

libquentier

0.5.0

Generated on Sun Dec 23 2018 16:01:34 for libquentier by Doxygen 1.9.8

Sun Dec 23 2018 16:01:34



<b>1 libquentier</b>	<b>1</b>
1.1 What's this	1
1.1.1 WARNING: libquentier is in alpha state right now, neither API nor ABI can be considered stable yet!	1
1.2 How to build/install	1
1.3 How to contribute	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>9</b>
4.1 File List	9
<b>5 Class Documentation</b>	<b>11</b>
5.1 quentier::Account Class Reference	11
5.1.1 Detailed Description	13
5.1.2 Member Function Documentation	13
5.1.2.1 displayName()	13
5.1.2.2 evernoteAccountType()	13
5.1.2.3 evernoteHost()	13
5.1.2.4 id()	13
5.1.2.5 isEmpty()	14
5.1.2.6 name()	14
5.1.2.7 print()	14
5.1.2.8 setDisplayName()	14
5.1.2.9 shardId()	14
5.1.2.10 type()	14
5.2 quentier::ApplicationSettings Class Reference	15
5.2.1 Detailed Description	16
5.2.2 Constructor & Destructor Documentation	16
5.2.2.1 ApplicationSettings() [1/3]	16
5.2.2.2 ApplicationSettings() [2/3]	17
5.2.2.3 ApplicationSettings() [3/3]	17
5.2.2.4 ~ApplicationSettings()	17
5.2.3 Member Function Documentation	17
5.2.3.1 beginGroup() [1/2]	17
5.2.3.2 beginGroup() [2/2]	18
5.2.3.3 beginReadArray() [1/2]	18
5.2.3.4 beginReadArray() [2/2]	18
5.2.3.5 beginWriteArray() [1/2]	19
5.2.3.6 beginWriteArray() [2/2]	19

5.2.3.7 contains() [1/2]	19
5.2.3.8 contains() [2/2]	20
5.2.3.9 print()	20
5.2.3.10 remove() [1/2]	20
5.2.3.11 remove() [2/2]	20
5.2.3.12 setValue() [1/2]	22
5.2.3.13 setValue() [2/2]	22
5.2.3.14 value() [1/2]	22
5.2.3.15 value() [2/2]	23
5.3 quantier::ApplicationSettingsInitializationException Class Reference	23
5.3.1 Detailed Description	25
5.3.2 Member Function Documentation	25
5.3.2.1 exceptionDisplayName()	25
5.4 quantier::ApplicationSettings::ArrayCloser Struct Reference	25
5.4.1 Detailed Description	26
5.5 quantier::AuthenticationManager Class Reference	26
5.5.1 Detailed Description	27
5.6 quantier::ResourceRecognitionIndexItem::BarcodeItem Struct Reference	27
5.7 quantier::DatabaseLockedException Class Reference	28
5.7.1 Member Function Documentation	29
5.7.1.1 exceptionDisplayName()	29
5.8 quantier::DatabaseLockFailedException Class Reference	29
5.8.1 Member Function Documentation	31
5.8.1.1 exceptionDisplayName()	31
5.9 quantier::DatabaseOpeningException Class Reference	31
5.9.1 Member Function Documentation	32
5.9.1.1 exceptionDisplayName()	32
5.10 quantier::DatabaseRequestException Class Reference	33
5.10.1 Detailed Description	34
5.10.2 Member Function Documentation	34
5.10.2.1 exceptionDisplayName()	34
5.11 quantier::DateTimePrint Class Reference	34
5.11.1 Detailed Description	35
5.11.2 Member Enumeration Documentation	35
5.11.2.1 Option	35
5.12 quantier::DecryptedTextManager Class Reference	35
5.13 quantier::DefaultLocalStorageCacheExpiryChecker Class Reference	35
5.13.1 Detailed Description	37
5.13.2 Member Function Documentation	37
5.13.2.1 checkLinkedNotebooks()	37
5.13.2.2 checkNotebooks()	38
5.13.2.3 checkNotes()	38

5.13.2.4 checkResources()	38
5.13.2.5 checkSavedSearches()	38
5.13.2.6 checkTags()	39
5.13.2.7 clone()	39
5.13.2.8 print()	39
5.14 quantier::EmptyDataElementException Class Reference	39
5.14.1 Member Function Documentation	41
5.14.1.1 exceptionDisplayName()	41
5.15 quantier::EncryptionManager Class Reference	41
5.15.1 Detailed Description	42
5.16 quantier::ENMLConverter Class Reference	42
5.16.1 Detailed Description	43
5.16.2 Member Function Documentation	43
5.16.2.1 cleanupExternalHtml()	43
5.16.2.2 exportNotesToEnex()	44
5.16.2.3 htmlToQTextDocument()	44
5.16.2.4 importEnex()	45
5.17 quantier::ErrorString Class Reference	45
5.17.1 Detailed Description	47
5.17.2 Member Function Documentation	47
5.17.2.1 print()	47
5.18 quantier::EventLoopWithExitStatus Class Reference	47
5.19 quantier::FileCopier Class Reference	49
5.20 quantier::FileIOProcessorAsync Class Reference	50
5.20.1 Detailed Description	51
5.20.2 Member Function Documentation	51
5.20.2.1 onReadFileRequest	51
5.20.2.2 onWriteFileRequest	52
5.20.2.3 readFileRequestProcessed	52
5.20.2.4 setIdleTimePeriod()	52
5.20.2.5 writeFileRequestProcessed	53
5.21 quantier::FileSystemWatcher Class Reference	53
5.22 quantier::ApplicationSettings::GroupCloser Struct Reference	54
5.22.1 Detailed Description	55
5.23 quantier::HTMLCleaner Class Reference	55
5.24 quantier::IAuthenticationManager Class Reference	55
5.25 quantier::IFavoritableDataElement Class Reference	56
5.25.1 Detailed Description	58
5.26 quantier::IKeychainService Class Reference	58
5.26.1 Detailed Description	59
5.26.2 Member Enumeration Documentation	59
5.26.2.1 ErrorCode	59

5.26.3 Member Function Documentation	60
5.26.3.1 deletePasswordJobFinished	60
5.26.3.2 readPasswordJobFinished	60
5.26.3.3 startDeletePasswordJob()	60
5.26.3.4 startReadPasswordJob()	61
5.26.3.5 startWritePasswordJob()	61
5.26.3.6 writePasswordJobFinished	61
5.27 quantier::ILocalStorageCacheExpiryChecker Class Reference	62
5.27.1 Detailed Description	64
5.27.2 Member Function Documentation	64
5.27.2.1 checkLinkedNotebooks()	64
5.27.2.2 checkNotebooks()	64
5.27.2.3 checkNotes()	64
5.27.2.4 checkResources()	64
5.27.2.5 checkSavedSearches()	65
5.27.2.6 checkTags()	65
5.27.2.7 clone()	65
5.27.2.8 print()	65
5.28 quantier::ILocalStorageDataElement Class Reference	66
5.29 quantier::ILocalStoragePatch Class Reference	66
5.29.1 Detailed Description	67
5.29.2 Member Function Documentation	67
5.29.2.1 apply()	67
5.29.2.2 backupLocalStorage()	68
5.29.2.3 backupProgress	68
5.29.2.4 fromVersion()	68
5.29.2.5 patchLongDescription()	69
5.29.2.6 patchShortDescription()	69
5.29.2.7 progress	69
5.29.2.8 removeLocalStorageBackup()	69
5.29.2.9 restoreBackupProgress	69
5.29.2.10 restoreLocalStorageFromBackup()	70
5.29.2.11 toVersion()	70
5.30 quantier::INoteEditorBackend Class Reference	70
5.31 quantier::INoteStore Class Reference	73
5.31.1 Detailed Description	75
5.31.2 Member Function Documentation	75
5.31.2.1 authenticateToSharedNotebook()	75
5.31.2.2 createNote()	75
5.31.2.3 createNotebook()	76
5.31.2.4 createSavedSearch()	76
5.31.2.5 createTag()	77

5.31.2.6 getLinkedNotebookSyncChunk()	77
5.31.2.7 getLinkedNotebookSyncState()	79
5.31.2.8 getNote()	80
5.31.2.9 getNoteAsync()	80
5.31.2.10 getResource()	81
5.31.2.11 getResourceAsync()	82
5.31.2.12 getSyncChunk()	82
5.31.2.13 getSyncState()	83
5.31.2.14 noteStoreUrl()	83
5.31.2.15 setAuthData()	83
5.31.2.16 setNoteStoreUrl()	84
5.31.2.17 stop()	84
5.31.2.18 updateNote()	84
5.31.2.19 updateNotebook()	84
5.31.2.20 updateSavedSearch()	85
5.31.2.21 updateTag()	85
5.32 quantier::INoteStoreDataElement Class Reference	86
5.33 quantier::IQuantierException Class Reference	87
5.33.1 Detailed Description	89
5.33.2 Member Function Documentation	89
5.33.2.1 print()	89
5.34 quantier::ISyncStateStorage::ISyncState Class Reference	90
5.34.1 Detailed Description	91
5.34.2 Member Function Documentation	91
5.34.2.1 print()	91
5.35 quantier::ISyncStateStorage Class Reference	91
5.35.1 Detailed Description	92
5.35.2 Member Function Documentation	92
5.35.2.1 notifySyncStateUpdated	92
5.36 quantier::IUserStore Class Reference	93
5.36.1 Detailed Description	93
5.36.2 Member Function Documentation	93
5.36.2.1 checkVersion()	93
5.36.2.2 getAccountLimits()	93
5.36.2.3 getUser()	94
5.36.2.4 setAuthData()	94
5.37 quantier::LimitedStack< T > Class Template Reference	95
5.37.1 Detailed Description	96
5.38 quantier::LinkedNotebook Class Reference	96
5.38.1 Member Function Documentation	98
5.38.1.1 checkParameters()	98
5.38.1.2 clear()	98

5.38.1.3 guid()	98
5.38.1.4 hasGuid()	98
5.38.1.5 hasUpdateSequenceNumber()	98
5.38.1.6 print()	99
5.38.1.7 setGuid()	99
5.38.1.8 setUpdateSequenceNumber()	99
5.38.1.9 updateSequenceNumber()	99
5.39 quotient::LocalStorageCacheManager Class Reference	99
5.39.1 Member Function Documentation	101
5.39.1.1 print()	101
5.40 quotient::LocalStorageCacheManagerException Class Reference	101
5.40.1 Member Function Documentation	103
5.40.1.1 exceptionDisplayName()	103
5.41 quotient::LocalStorageManager Class Reference	103
5.41.1 Member Enumeration Documentation	109
5.41.1.1 GetNoteOption	109
5.41.1.2 GetResourceOption	110
5.41.1.3 ListObjectsOption	110
5.41.1.4 StartupOption	110
5.41.1.5 UpdateNoteOption	111
5.41.2 Constructor & Destructor Documentation	111
5.41.2.1 LocalStorageManager()	111
5.41.3 Member Function Documentation	111
5.41.3.1 accountHighUsn()	111
5.41.3.2 addEnResource()	112
5.41.3.3 addLinkedNotebook()	112
5.41.3.4 addNote()	113
5.41.3.5 addNotebook()	113
5.41.3.6 addSavedSearch()	113
5.41.3.7 addTag()	114
5.41.3.8 addUser()	114
5.41.3.9 deleteUser()	115
5.41.3.10 enResourceCount()	115
5.41.3.11 expungeEnResource()	115
5.41.3.12 expungeLinkedNotebook()	116
5.41.3.13 expungeNote()	116
5.41.3.14 expungeNotebook()	116
5.41.3.15 expungeNotelessTagsFromLinkedNotebooks()	117
5.41.3.16 expungeSavedSearch()	117
5.41.3.17 expungeTag()	118
5.41.3.18 expungeUser()	118
5.41.3.19 findDefaultNotebook()	118

5.41.3.20 findDefaultOrLastUsedNotebook()	119
5.41.3.21 findEnResource()	119
5.41.3.22 findLastUsedNotebook()	120
5.41.3.23 findLinkedNotebook()	120
5.41.3.24 findNote()	120
5.41.3.25 findNotebook()	121
5.41.3.26 findNoteLocalUidsWithSearchQuery()	121
5.41.3.27 findNotesWithSearchQuery()	122
5.41.3.28 findSavedSearch()	122
5.41.3.29 findTag()	123
5.41.3.30 findUser()	123
5.41.3.31 highestSupportedLocalStorageVersion()	124
5.41.3.32 isLocalStorageVersionTooHigh()	124
5.41.3.33 linkedNotebookCount()	124
5.41.3.34 listAllLinkedNotebooks()	125
5.41.3.35 listAllNotebooks()	125
5.41.3.36 listAllSavedSearches()	126
5.41.3.37 listAllSharedNotebooks()	126
5.41.3.38 listAllTags()	127
5.41.3.39 listAllTagsPerNote()	127
5.41.3.40 listLinkedNotebooks()	128
5.41.3.41 listNotebooks()	128
5.41.3.42 listNotes()	129
5.41.3.43 listNotesByLocalUids()	130
5.41.3.44 listNotesPerNotebook()	130
5.41.3.45 listNotesPerNotebooksAndTags()	131
5.41.3.46 listNotesPerTag()	132
5.41.3.47 listSavedSearches()	132
5.41.3.48 listSharedNotebooksPerNotebookGuid()	133
5.41.3.49 listTags()	133
5.41.3.50 listTagsWithNoteLocalUids()	134
5.41.3.51 localStorageRequiresUpgrade()	135
5.41.3.52 localStorageVersion()	135
5.41.3.53 notebookCount()	136
5.41.3.54 noteCount()	136
5.41.3.55 noteCountPerNotebook()	136
5.41.3.56 noteCountPerNotebooksAndTags()	137
5.41.3.57 noteCountPerTag()	137
5.41.3.58 noteCountsPerAllTags()	137
5.41.3.59 requiredLocalStoragePatches()	138
5.41.3.60 savedSearchCount()	138
5.41.3.61 switchUser()	139

5.41.3.62 tagCount()	139
5.41.3.63 updateEnResource()	139
5.41.3.64 updateLinkedNotebook()	140
5.41.3.65 updateNote()	140
5.41.3.66 updateNotebook()	141
5.41.3.67 updateSavedSearch()	141
5.41.3.68 updateTag()	142
5.41.3.69 updateUser()	142
5.41.3.70 upgradeProgress	143
5.41.3.71 userCount()	143
5.42 quantier::LocalStorageManagerAsync Class Reference	143
5.43 quantier::LoggerInitializationException Class Reference	150
5.43.1 Member Function Documentation	152
5.43.1.1 exceptionDisplayName()	152
5.44 quantier::LRUCache< Key, Value, Allocator > Class Template Reference	152
5.45 quantier::Note Class Reference	153
5.45.1 Member Function Documentation	157
5.45.1.1 checkParameters()	157
5.45.1.2 clear()	157
5.45.1.3 guid()	157
5.45.1.4 hasGuid()	157
5.45.1.5 hasUpdateSequenceNumber()	157
5.45.1.6 print()	157
5.45.1.7 setGuid()	157
5.45.1.8 setUpdateSequenceNumber()	158
5.45.1.9 updateSequenceNumber()	158
5.46 quantier::Notebook Class Reference	158
5.46.1 Member Function Documentation	162
5.46.1.1 checkParameters()	162
5.46.1.2 clear()	162
5.46.1.3 guid()	163
5.46.1.4 hasGuid()	163
5.46.1.5 hasUpdateSequenceNumber()	163
5.46.1.6 print()	163
5.46.1.7 setGuid()	163
5.46.1.8 setUpdateSequenceNumber()	163
5.46.1.9 updateSequenceNumber()	163
5.47 quantier::ENMLConverter::NoteContentToHtmlExtraData Struct Reference	164
5.48 quantier::NoteEditor Class Reference	164
5.48.1 Detailed Description	167
5.48.2 Member Function Documentation	168
5.48.2.1 backend()	168

5.48.2.2 clear()	168
5.48.2.3 convertToNote	168
5.48.2.4 currentNoteLocalUid()	168
5.48.2.5 defaultFont()	168
5.48.2.6 defaultPalette()	168
5.48.2.7 idleTime()	169
5.48.2.8 inAppNoteLinkPasteRequested	169
5.48.2.9 initialize()	169
5.48.2.10 isEditorPageModified()	169
5.48.2.11 isModified()	170
5.48.2.12 isNoteLoaded()	170
5.48.2.13 saveNoteToLocalStorage	170
5.48.2.14 setAccount()	170
5.48.2.15 setBackend()	170
5.48.2.16 setCurrentNoteLocalUid()	170
5.48.2.17 setDefaultFont	171
5.48.2.18 setDefaultPalette	171
5.48.2.19 setFocus()	171
5.48.2.20 setInitialPageHtml()	171
5.48.2.21 setNoteDeletedPageHtml()	172
5.48.2.22 setNoteLoadingPageHtml()	172
5.48.2.23 setNoteNotFoundPageHtml()	172
5.48.2.24 setNoteTitle	172
5.48.2.25 setTagIds	172
5.48.2.26 setUndoStack()	173
5.48.2.27 undoStack()	173
5.49 quotient::NoteEditorInitializationException Class Reference	173
5.49.1 Member Function Documentation	175
5.49.1.1 exceptionDisplayName()	175
5.50 quotient::NoteEditorPluginInitializationException Class Reference	175
5.50.1 Member Function Documentation	176
5.50.1.1 exceptionDisplayName()	176
5.51 quotient::NoteSearchQuery Class Reference	177
5.51.1 Member Function Documentation	179
5.51.1.1 notebookModifier()	179
5.51.1.2 print()	179
5.51.1.3 queryString()	180
5.52 quotient::NullPtrException Class Reference	180
5.52.1 Member Function Documentation	181
5.52.1.1 exceptionDisplayName()	181
5.53 quotient::ResourceRecognitionIndexItem::ObjectItem Struct Reference	181
5.54 quotient::Printable Class Reference	182

5.54.1 Detailed Description	183
5.54.2 Member Function Documentation	183
5.54.2.1 print()	183
5.55 quantier::QuantierApplication Class Reference	183
5.56 quantier::QuantierUndoCommand Class Reference	184
5.56.1 Detailed Description	185
5.57 quantier::Resource Class Reference	185
5.57.1 Member Function Documentation	188
5.57.1.1 checkParameters()	188
5.57.1.2 clear()	188
5.57.1.3 guid()	188
5.57.1.4 hasGuid()	188
5.57.1.5 hasUpdateSequenceNumber()	189
5.57.1.6 print()	189
5.57.1.7 setGuid()	189
5.57.1.8 setUpdateSequenceNumber()	189
5.57.1.9 updateSequenceNumber()	189
5.58 quantier::ResourceRecognitionIndexItem Class Reference	189
5.58.1 Member Function Documentation	191
5.58.1.1 print()	191
5.59 quantier::ResourceRecognitionIndices Class Reference	192
5.59.1 Member Function Documentation	193
5.59.1.1 print()	193
5.60 quantier::SavedSearch Class Reference	193
5.60.1 Member Function Documentation	196
5.60.1.1 checkParameters()	196
5.60.1.2 clear()	196
5.60.1.3 guid()	196
5.60.1.4 hasGuid()	196
5.60.1.5 hasUpdateSequenceNumber()	196
5.60.1.6 print()	196
5.60.1.7 setGuid()	196
5.60.1.8 setUpdateSequenceNumber()	197
5.60.1.9 updateSequenceNumber()	197
5.61 quantier::ResourceRecognitionIndexItem::ShapeItem Struct Reference	197
5.62 quantier::SharedNote Class Reference	197
5.62.1 Member Function Documentation	200
5.62.1.1 print()	200
5.63 quantier::SharedNotebook Class Reference	200
5.63.1 Member Function Documentation	202
5.63.1.1 print()	202
5.64 quantier::ShortcutManager Class Reference	203

5.64.1 Member Function Documentation	204
5.64.1.1 defaultShortcut() [1/2]	204
5.64.1.2 defaultShortcut() [2/2]	204
5.64.1.3 shortcut() [1/2]	205
5.64.1.4 shortcut() [2/2]	205
5.64.1.5 userShortcut() [1/2]	205
5.64.1.6 userShortcut() [2/2]	205
5.65 quotientier::ENMLConverter::SkipHtmlElementRule Class Reference	206
5.65.1 Detailed Description	207
5.65.2 Member Function Documentation	207
5.65.2.1 print()	207
5.66 quotientier::SpellChecker Class Reference	208
5.67 quotientier::StringUtils::StringFilterPredicate Struct Reference	209
5.68 quotientier::StringUtils Class Reference	209
5.69 quotientier::SynchronizationManager Class Reference	209
5.69.1 Detailed Description	211
5.69.2 Constructor & Destructor Documentation	211
5.69.2.1 SynchronizationManager()	211
5.69.3 Member Function Documentation	212
5.69.3.1 active()	212
5.69.3.2 authenticate	212
5.69.3.3 authenticateCurrentAccount	212
5.69.3.4 authenticationFinished	212
5.69.3.5 authenticationRevoked	213
5.69.3.6 detectedConflictDuringLocalChangesSending	213
5.69.3.7 downloadNoteThumbnailsOption()	213
5.69.3.8 failed	213
5.69.3.9 finished	214
5.69.3.10 linkedNotebooksNotesDownloadProgress	214
5.69.3.11 linkedNotebooksResourcesDownloadProgress	214
5.69.3.12 linkedNotebooksSyncChunksDownloaded	215
5.69.3.13 linkedNotebookSyncChunksDownloadProgress	215
5.69.3.14 notesDownloadProgress	215
5.69.3.15 preparedDirtyObjectsForSending	215
5.69.3.16 preparedLinkedNotebooksDirtyObjectsForSending	216
5.69.3.17 rateLimitExceeded	216
5.69.3.18 remoteToLocalSyncDone	216
5.69.3.19 remoteToLocalSyncStopped	216
5.69.3.20 resourcesDownloadProgress	216
5.69.3.21 revokeAuthentication	217
5.69.3.22 sendLocalChangesStopped	217
5.69.3.23 setAccount	217

5.69.3.24 setAccountDone . . . . .	217
5.69.3.25 setDownloadInkNotelmages . . . . .	217
5.69.3.26 setDownloadInkNotelmagesDone . . . . .	218
5.69.3.27 setDownloadNoteThumbnails . . . . .	218
5.69.3.28 setDownloadNoteThumbnailsDone . . . . .	218
5.69.3.29 setInkNotelmagesStoragePath . . . . .	218
5.69.3.30 setInkNotelmagesStoragePathDone . . . . .	218
5.69.3.31 started . . . . .	219
5.69.3.32 stop . . . . .	219
5.69.3.33 stopped . . . . .	219
5.69.3.34 syncChunksDownloaded . . . . .	219
5.69.3.35 syncChunksDownloadProgress . . . . .	219
5.69.3.36 synchronize . . . . .	220
5.69.3.37 willRepeatRemoteToLocalSyncAfterSendingChanges . . . . .	220
5.70 quotienter::SysInfo Class Reference . . . . .	220
5.71 quotienter::Tag Class Reference . . . . .	220
5.71.1 Member Function Documentation . . . . .	222
5.71.1.1 checkParameters() . . . . .	222
5.71.1.2 clear() . . . . .	223
5.71.1.3 guid() . . . . .	223
5.71.1.4 hasGuid() . . . . .	223
5.71.1.5 hasUpdateSequenceNumber() . . . . .	223
5.71.1.6 print() . . . . .	223
5.71.1.7 setGuid() . . . . .	223
5.71.1.8 setUpdateSequenceNumber() . . . . .	223
5.71.1.9 updateSequenceNumber() . . . . .	224
5.72 quotienter::ResourceRecognitionIndexItem::TextItem Struct Reference . . . . .	224
5.73 quotienter::UidGenerator Class Reference . . . . .	224
5.74 quotienter::User Class Reference . . . . .	224
5.74.1 Member Function Documentation . . . . .	227
5.74.1.1 print() . . . . .	227
<b>6 File Documentation</b>	<b>229</b>
6.1 DecryptedTextManager.h . . . . .	229
6.2 ENMLConverter.h . . . . .	230
6.3 HTMLCleaner.h . . . . .	232
6.4 ApplicationSettingsInitializationException.h . . . . .	232
6.5 DatabaseLockedException.h . . . . .	233
6.6 DatabaseLockFailedException.h . . . . .	233
6.7 DatabaseOpeningException.h . . . . .	234
6.8 DatabaseRequestException.h . . . . .	234
6.9 EmptyDataElementException.h . . . . .	235

6.10 IQuentierException.h . . . . .	235
6.11 LocalStorageCacheManagerException.h . . . . .	236
6.12 LoggerInitializationException.h . . . . .	237
6.13 NoteEditorInitializationException.h . . . . .	237
6.14 NoteEditorPluginInitializationException.h . . . . .	238
6.15 NullPtrException.h . . . . .	238
6.16 DefaultLocalStorageCacheExpiryChecker.h . . . . .	239
6.17 ILocalStorageCacheExpiryChecker.h . . . . .	239
6.18 ILocalStoragePatch.h . . . . .	240
6.19 Lists.h . . . . .	241
6.20 LocalStorageCacheManager.h . . . . .	242
6.21 LocalStorageManager.h . . . . .	243
6.22 LocalStorageManagerAsync.h . . . . .	250
6.23 NoteSearchQuery.h . . . . .	259
6.24 QuentierLogger.h . . . . .	261
6.25 INoteEditorBackend.h . . . . .	262
6.26 NoteEditor.h . . . . .	265
6.27 SpellChecker.h . . . . .	268
6.28 AuthenticationManager.h . . . . .	269
6.29 synchronization/ForwardDeclarations.h . . . . .	270
6.30 utility/ForwardDeclarations.h . . . . .	270
6.31 IAuthenticationManager.h . . . . .	271
6.32 INoteStore.h . . . . .	272
6.33 ISyncStateStorage.h . . . . .	274
6.34 IUserStore.h . . . . .	274
6.35 SynchronizationManager.h . . . . .	275
6.36 Account.h . . . . .	277
6.37 ErrorString.h . . . . .	279
6.38 IFavoritableDataElement.h . . . . .	279
6.39 ILocalStorageDataElement.h . . . . .	280
6.40 INoteStoreDataElement.h . . . . .	281
6.41 LinkedNotebook.h . . . . .	283
6.42 Note.h . . . . .	284
6.43 Notebook.h . . . . .	287
6.44 RegisterMetatypes.h . . . . .	290
6.45 Resource.h . . . . .	290
6.46 ResourceRecognitionIndexItem.h . . . . .	292
6.47 ResourceRecognitionIndices.h . . . . .	294
6.48 SavedSearch.h . . . . .	295
6.49 SharedNote.h . . . . .	296
6.50 SharedNotebook.h . . . . .	298
6.51 Tag.h . . . . .	300

6.52 User.h . . . . .	301
6.53 ApplicationSettings.h . . . . .	303
6.54 Checks.h . . . . .	304
6.55 Compat.h . . . . .	305
6.56 DateTime.h . . . . .	305
6.57 EncryptionManager.h . . . . .	306
6.58 EventLoopWithExitStatus.h . . . . .	307
6.59 FileCopier.h . . . . .	308
6.60 FileIOProcessorAsync.h . . . . .	309
6.61 FileSystem.h . . . . .	310
6.62 FileSystemWatcher.h . . . . .	310
6.63 IKeychainService.h . . . . .	311
6.64 Initialize.h . . . . .	313
6.65 LimitedStack.h . . . . .	313
6.66 Linkage.h . . . . .	314
6.67 LRUCache.hpp . . . . .	315
6.68 MessageBox.h . . . . .	318
6.69 Printable.h . . . . .	318
6.70 QuentierApplication.h . . . . .	320
6.71 QuentierCheckPtr.h . . . . .	320
6.72 QuentierUndoCommand.h . . . . .	321
6.73 ShortcutManager.h . . . . .	321
6.74 Size.h . . . . .	323
6.75 StandardPaths.h . . . . .	324
6.76 StringUtils.h . . . . .	324
6.77 SuppressWarnings.h . . . . .	325
6.78 SysInfo.h . . . . .	326
6.79 System.h . . . . .	327
6.80 TagSortByParentChildRelations.h . . . . .	328
6.81 UidGenerator.h . . . . .	328
<b>Index</b>	<b>331</b>

# Chapter 1

## libquentier

Set of Qt/C++ APIs for feature rich desktop clients for Evernote service

### 1.1 What's this

This library presents a set of Qt/C++ APIs useful for applications working as feature rich desktop clients for Evernote service. The most important and useful components of the library are the following:

- Local storage - persistence of data downloaded from Evernote service in a local SQLite database
- Synchronization - the logics of exchanging new and/or modified data with Evernote service
- Note editor - the UI component capable for notes displaying and editing

The library is based on the lower level functionality provided by `QEverCloud` library. It also serves as the functional core of `Quentier` application.

**1.1.1 WARNING: libquentier is in alpha state right now, neither API nor ABI can be considered stable yet!**

### 1.2 How to build/install

Please see the building/installation guide.

### 1.3 How to contribute

Please see the contribution guide for detailed info.



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

quentier::ApplicationSettings::ArrayCloser . . . . .	25
quentier::ResourceRecognitionIndexItem::BarcodeItem . . . . .	27
quentier::DateTimePrint . . . . .	34
quentier::DecryptedTextManager . . . . .	35
quentier::ENMLConverter . . . . .	42
std::exception	
quentier::IQuentierException . . . . .	87
quentier::ApplicationSettingsInitializationException . . . . .	23
quentier::DatabaseLockFailedException . . . . .	29
quentier::DatabaseLockedException . . . . .	28
quentier::DatabaseOpeningException . . . . .	31
quentier::DatabaseRequestException . . . . .	33
quentier::EmptyDataElementException . . . . .	39
quentier::LocalStorageCacheManagerException . . . . .	101
quentier::LoggerInitializationException . . . . .	150
quentier::NoteEditorInitializationException . . . . .	173
quentier::NoteEditorPluginInitializationException . . . . .	175
quentier::NullPtrException . . . . .	180
quentier::ApplicationSettings::GroupCloser . . . . .	54
quentier::HTMLCleaner . . . . .	55
quentier::ILocalStorageDataElement . . . . .	66
quentier::INoteStoreDataElement . . . . .	86
quentier::IFavoritableDataElement . . . . .	56
quentier::Note . . . . .	153
quentier::Notebook . . . . .	158
quentier::SavedSearch . . . . .	193
quentier::Tag . . . . .	220
quentier::LinkedNotebook . . . . .	96
quentier::Resource . . . . .	185
quentier::INoteEditorBackend . . . . .	70
quentier::IUserStore . . . . .	93
quentier::LRUCache< Key, Value, Allocator > . . . . .	152
quentier::ENMLConverter::NoteContentToHtmlExtraData . . . . .	164
quentier::ResourceRecognitionIndexItem::ObjectItem . . . . .	181
quentier::Printable . . . . .	182

quentier::Account	11
quentier::ApplicationSettings	15
quentier::ENMLConverter::SkipHtmlElementRule	206
quentier::ErrorString	45
quentier::ILocalStorageCacheExpiryChecker	62
quentier::DefaultLocalStorageCacheExpiryChecker	35
quentier::INoteStoreDataElement	86
quentier::IQuentierException	87
quentier::ISyncStateStorage::ISyncState	90
quentier::LocalStorageCacheManager	99
quentier::NoteSearchQuery	177
quentier::ResourceRecognitionIndexItem	189
quentier::ResourceRecognitionIndices	192
quentier::SharedNote	197
quentier::SharedNotebook	200
quentier::User	224
QApplication	
quentier::QuentierApplication	183
QEventLoop	
quentier::EventLoopWithExitStatus	47
QObject	
quentier::EncryptionManager	41
quentier::FileCopier	49
quentier::FileIOProcessorAsync	50
quentier::FileSystemWatcher	53
quentier::IAuthenticationManager	55
quentier::AuthenticationManager	26
quentier::IKeychainService	58
quentier::ILocalStoragePatch	66
quentier::INoteStore	73
quentier::ISyncStateStorage	91
quentier::LocalStorageManager	103
quentier::LocalStorageManagerAsync	143
quentier::QuentierUndoCommand	184
quentier::ShortcutManager	203
quentier::SpellChecker	208
quentier::SynchronizationManager	209
QSettings	
quentier::ApplicationSettings	15
QStack	
quentier::LimitedStack< T >	95
QUndoCommand	
quentier::QuentierUndoCommand	184
QWidget	
quentier::NoteEditor	164
quentier::ResourceRecognitionIndexItem::ShapeItem	197
quentier::StringUtils::StringFilterPredicate	209
quentier::StringUtils	209
quentier::SysInfo	220
quentier::ResourceRecognitionIndexItem::TextItem	224
quentier::UidGenerator	224

# Chapter 3

## Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">quentier::Account</a>	Encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc . . . . .	11
<a href="#">quentier::ApplicationSettings</a>	Enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings . . . . .	15
<a href="#">quentier::ApplicationSettingsInitializationException</a>	The <a href="#">ApplicationSettingsInitializationException</a> can be thrown from methods of <a href="#">ApplicationSettings</a> class if it's unable to locate the file with persistent settings . . . . .	23
<a href="#">quentier::ApplicationSettings::ArrayCloser</a>		25
<a href="#">quentier::AuthenticationManager</a>	Libquentier's default implementation of <a href="#">IAuthenticationManager</a> interface; internally uses QEverCloud's OAuth widget . . . . .	26
<a href="#">quentier::ResourceRecognitionIndexItem::BarcodeItem</a>		27
<a href="#">quentier::DatabaseLockedException</a>		28
<a href="#">quentier::DatabaseLockFailedException</a>		29
<a href="#">quentier::DatabaseOpeningException</a>		31
<a href="#">quentier::DatabaseRequestException</a>	The <a href="#">DatabaseRequestException</a> is thrown when the local storage database encounters some internal error during the attempt to serve a request . . . . .	33
<a href="#">quentier::DateTimePrint</a>	Simply wraps the enum containing datetime printing options . . . . .	34
<a href="#">quentier::DecryptedTextManager</a>		35
<a href="#">quentier::DefaultLocalStorageCacheExpiryChecker</a>		35
<a href="#">quentier::EmptyDataElementException</a>		39
<a href="#">quentier::EncryptionManager</a>	Both synchronous methods to encrypt or decrypt given text with password, cipher and key length and their signal-slot based potentially asynchronous counterparts . . . . .	41
<a href="#">quentier::ENMLConverter</a>	Encapsulates a set of methods and helper data structures for performing the conversions between ENML and other note content formats, namely HTML . . . . .	42
<a href="#">quentier::ErrorString</a>	Encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description . . . . .	45
<a href="#">quentier::EventLoopWithExitStatus</a>		47

<a href="#">quentier::FileCopier</a>	49
<a href="#">quentier::FileIOProcessorAsync</a>	
Wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO	50
<a href="#">quentier::FileSystemWatcher</a>	53
<a href="#">quentier::ApplicationSettings::GroupCloser</a>	54
<a href="#">quentier::HTMLCleaner</a>	55
<a href="#">quentier::IAuthenticationManager</a>	55
<a href="#">quentier::IFavoritableDataElement</a>	56
<a href="#">quentier::IKeychainService</a>	
The <a href="#">IKeychainService</a> interface provides methods intended to start potentially asynchronous interaction with the keychain and signals intended to notify listeners about the completion of asynchronous interactions	58
<a href="#">quentier::ILocalStorageCacheExpiryChecker</a>	
Interface for cache expiry checker used by <a href="#">LocalStorageCacheManager</a> to see whether particular caches (of notes, notebooks, tags, linked notebooks and/or saved searches) need to be shrunk	62
<a href="#">quentier::ILocalStorageDataElement</a>	66
<a href="#">quentier::ILocalStoragePatch</a>	
Interface for patches of local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it	66
<a href="#">quentier::INoteEditorBackend</a>	70
<a href="#">quentier::INoteStore</a>	
<a href="#">INoteStore</a> is the interface which provides methods required for the implementation of NoteStore part of Evernote EDAM sync protocol	73
<a href="#">quentier::INoteStoreDataElement</a>	86
<a href="#">quentier::IQuentierException</a>	
Interface for exceptions specific to libquentier and applications based on it	87
<a href="#">quentier::ISyncStateStorage::ISyncState</a>	
The <a href="#">ISyncState</a> interface provides accessory methods to determine the sync state for the account	90
<a href="#">quentier::ISyncStateStorage</a>	
The <a href="#">ISyncStateStorage</a> interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states	91
<a href="#">quentier::IUserStore</a>	
<a href="#">IUserStore</a> is the interface which provides methods required for the implementation of UserStore part of Evernote EDAM sync protocol	93
<a href="#">quentier::LimitedStack&lt; T &gt;</a>	
The <a href="#">LimitedStack</a> template class implements a stack which may have a limitation for its size; when the size becomes too much according to the limit, the bottom element of the stack gets erased from it. Only limits greater than zero are considered	95
<a href="#">quentier::LinkedNotebook</a>	96
<a href="#">quentier::LocalStorageCacheManager</a>	99
<a href="#">quentier::LocalStorageCacheManagerException</a>	101
<a href="#">quentier::LocalStorageManager</a>	103
<a href="#">quentier::LocalStorageManagerAsync</a>	143
<a href="#">quentier::LoggerInitializationException</a>	150
<a href="#">quentier::LRUCache&lt; Key, Value, Allocator &gt;</a>	152
<a href="#">quentier::Note</a>	153
<a href="#">quentier::Notebook</a>	158
<a href="#">quentier::ENMLConverter::NoteContentToHtmlExtraData</a>	164
<a href="#">quentier::NoteEditor</a>	
Widget encapsulating all the functionality necessary for showing and editing notes	164
<a href="#">quentier::NoteEditorInitializationException</a>	173
<a href="#">quentier::NoteEditorPluginInitializationException</a>	175
<a href="#">quentier::NoteSearchQuery</a>	177
<a href="#">quentier::NullPtrException</a>	180
<a href="#">quentier::ResourceRecognitionIndexItem::ObjectItem</a>	181

<a href="#">quentier::Printable</a>	
Interface for Quentier's internal classes which should be able to write themselves into QText↔ Stream and/or convert to QString . . . . .	182
<a href="#">quentier::QuentierApplication</a> . . . . .	183
<a href="#">quentier::QuentierUndoCommand</a>	
Has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push QUndoCommand to QUndoStack, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that . . .	184
<a href="#">quentier::Resource</a> . . . . .	185
<a href="#">quentier::ResourceRecognitionIndexItem</a> . . . . .	189
<a href="#">quentier::ResourceRecognitionIndices</a> . . . . .	192
<a href="#">quentier::SavedSearch</a> . . . . .	193
<a href="#">quentier::ResourceRecognitionIndexItem::ShapelItem</a> . . . . .	197
<a href="#">quentier::SharedNote</a> . . . . .	197
<a href="#">quentier::SharedNotebook</a> . . . . .	200
<a href="#">quentier::ShortcutManager</a> . . . . .	203
<a href="#">quentier::ENMLConverter::SkipHtmlElementRule</a>	
Describes the set of rules for HTML -> ENML conversion about the HTML elements that should not be actually converted to ENML due to their nature of being "helper" elements for the display or functioning of something within the note editor's page. The HTML -> ENML conversion would ignore tags and attributes forbidden by ENML even without these rules conditionally preserving or skipping the contents and nested elements of skipped elements . . . . .	206
<a href="#">quentier::SpellChecker</a> . . . . .	208
<a href="#">quentier::StringUtils::StringFilterPredicate</a> . . . . .	209
<a href="#">quentier::StringUtils</a> . . . . .	209
<a href="#">quentier::SynchronizationManager</a>	
Encapsulates methods and signals & slots required to perform the full or partial synchronization of data with remote Evernote servers. The class also deals with authentication with Evernote service through OAuth . . . . .	209
<a href="#">quentier::SysInfo</a> . . . . .	220
<a href="#">quentier::Tag</a> . . . . .	220
<a href="#">quentier::ResourceRecognitionIndexItem::TextItem</a> . . . . .	224
<a href="#">quentier::UidGenerator</a> . . . . .	224
<a href="#">quentier::User</a> . . . . .	224



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">DecryptedTextManager.h</a>	229
<a href="#">ENMLConverter.h</a>	230
<a href="#">HTMLCleaner.h</a>	232
<a href="#">ApplicationSettingsInitializationException.h</a>	232
<a href="#">DatabaseLockedException.h</a>	233
<a href="#">DatabaseLockFailedException.h</a>	233
<a href="#">DatabaseOpeningException.h</a>	234
<a href="#">DatabaseRequestException.h</a>	234
<a href="#">EmptyDataElementException.h</a>	235
<a href="#">IQuentierException.h</a>	235
<a href="#">LocalStorageCacheManagerException.h</a>	236
<a href="#">LoggerInitializationException.h</a>	237
<a href="#">NoteEditorInitializationException.h</a>	237
<a href="#">NoteEditorPluginInitializationException.h</a>	238
<a href="#">NullPtrException.h</a>	238
<a href="#">DefaultLocalStorageCacheExpiryChecker.h</a>	239
<a href="#">ILocalStorageCacheExpiryChecker.h</a>	239
<a href="#">ILocalStoragePatch.h</a>	240
<a href="#">Lists.h</a>	241
<a href="#">LocalStorageCacheManager.h</a>	242
<a href="#">LocalStorageManager.h</a>	243
<a href="#">LocalStorageManagerAsync.h</a>	250
<a href="#">NoteSearchQuery.h</a>	259
<a href="#">QuentierLogger.h</a>	261
<a href="#">INoteEditorBackend.h</a>	262
<a href="#">NoteEditor.h</a>	265
<a href="#">SpellChecker.h</a>	268
<a href="#">AuthenticationManager.h</a>	269
<a href="#">synchronization/ForwardDeclarations.h</a>	270
<a href="#">utility/ForwardDeclarations.h</a>	270
<a href="#">IAuthenticationManager.h</a>	271
<a href="#">INoteStore.h</a>	272
<a href="#">ISyncStateStorage.h</a>	274
<a href="#">IUserStore.h</a>	274
<a href="#">SynchronizationManager.h</a>	275

Account.h	277
ErrorString.h	279
IFavoritableDataElement.h	279
ILocalStorageDataElement.h	280
INoteStoreDataElement.h	281
LinkedNotebook.h	283
Note.h	284
Notebook.h	287
RegisterMetatypes.h	290
Resource.h	290
ResourceRecognitionIndexItem.h	292
ResourceRecognitionIndices.h	294
SavedSearch.h	295
SharedNote.h	296
SharedNotebook.h	298
Tag.h	300
User.h	301
ApplicationSettings.h	303
Checks.h	304
Compat.h	305
DateTime.h	305
EncryptionManager.h	306
EventLoopWithExitStatus.h	307
FileCopier.h	308
FileIOProcessorAsync.h	309
FileSystem.h	310
FileSystemWatcher.h	310
IKeychainService.h	311
Initialize.h	313
LimitedStack.h	313
Linkage.h	314
LRUCache.hpp	315
MessageBox.h	318
Printable.h	318
QuentierApplication.h	320
QuentierCheckPtr.h	320
QuentierUndoCommand.h	321
ShortcutManager.h	321
Size.h	323
StandardPaths.h	324
StringUtils.h	324
SuppressWarnings.h	325
SysInfo.h	326
System.h	327
TagSortByParentChildRelations.h	328
UidGenerator.h	328

## Chapter 5

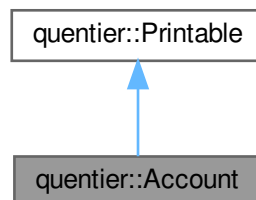
# Class Documentation

### 5.1 `quentier::Account` Class Reference

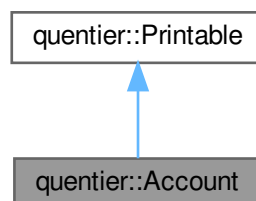
The `Account` class encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc.

```
#include <Account.h>
```

Inheritance diagram for `quentier::Account`:



Collaboration diagram for `quentier::Account`:



## Public Types

- enum class **Type** { **Local** = 0 , **Evernote** }
- enum class **EvernoteAccountType** { **Free** = 0 , **Plus** , **Premium** , **Business** }

## Public Member Functions

- **Account** (QString [name](#), const Type [type](#), const qevercloud::UserID [userId](#)=-1, const EvernoteAccountType [evernoteAccountType](#)=EvernoteAccountType::Free, QString [evernoteHost](#)={}, QString [shardId](#)={})
- **Account** (const [Account](#) &other)
- [Account](#) & **operator=** (const [Account](#) &other)
- bool **operator==** (const [Account](#) &other) const
- bool **operator!=** (const [Account](#) &other) const
- bool **isEmpty** () const
- QString [name](#) () const
- void **setName** (QString [name](#))  
*setName sets the username to the account*
- QString [displayName](#) () const
- void **setDisplayName** (QString [displayName](#))
- Type [type](#) () const
- qevercloud::UserID [id](#) () const
- EvernoteAccountType [evernoteAccountType](#) () const
- QString [evernoteHost](#) () const
- QString [shardId](#) () const
- void **setEvernoteAccountType** (const EvernoteAccountType [evernoteAccountType](#))
- void **setEvernoteHost** (QString [evernoteHost](#))
- void **setShardId** (QString [shardId](#))
- qint32 **mailLimitDaily** () const
- qint64 **noteSizeMax** () const
- qint64 **resourceSizeMax** () const
- qint32 **linkedNotebookMax** () const
- qint32 **noteCountMax** () const
- qint32 **notebookCountMax** () const
- qint32 **tagCountMax** () const
- qint32 **noteTagCountMax** () const
- qint32 **savedSearchCountMax** () const
- qint32 **noteResourceCountMax** () const
- void **setEvernoteAccountLimits** (const qevercloud::AccountLimits &limits)
- virtual QTextStream & [print](#) (QTextStream &strm) const override

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

## Friends

- QUENTIER\_EXPORT QTextStream & **operator**<< (QTextStream &strm, const Type [type](#))
- QUENTIER\_EXPORT QDebug & **operator**<< (QDebug &dbg, const Type [type](#))
- QUENTIER\_EXPORT QTextStream & **operator**<< (QTextStream &strm, const EvernoteAccountType [type](#))
- QUENTIER\_EXPORT QDebug & **operator**<< (QDebug &dbg, const EvernoteAccountType [type](#))

## Additional Inherited Members

## Protected Member Functions inherited from `quentier::Printable`

- `Printable` (const `Printable` &other)
- `Printable` & `operator=` (const `Printable` &other)

### 5.1.1 Detailed Description

The `Account` class encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc.

### 5.1.2 Member Function Documentation

#### 5.1.2.1 `displayName()`

```
QString quentier::Account::displayName ( ) const
```

##### Returns

`Printable` user's name that is not used to uniquely identify the account, so this name may repeat across different local and Evernote accounts

#### 5.1.2.2 `evernoteAccountType()`

```
EvernoteAccountType quentier::Account::evernoteAccountType ( ) const
```

##### Returns

The type of the Evernote account; if applied to free account, returns "Free"

#### 5.1.2.3 `evernoteHost()`

```
QString quentier::Account::evernoteHost ( ) const
```

##### Returns

The Evernote server host with which the account is associated

#### 5.1.2.4 `id()`

```
qevercloud::UserID quentier::Account::id ( ) const
```

##### Returns

`User` id for Evernote accounts, -1 for local accounts (as the concept of user id is not defined for local accounts)

#### 5.1.2.5 isEmpty()

```
bool quantier::Account::isEmpty ( ) const
```

##### Returns

True if either the account is local but the name is empty or if the account is Evernote but user id is negative; in all other cases return false

#### 5.1.2.6 name()

```
QString quantier::Account::name ( ) const
```

##### Returns

Username for either local or Evernote account

#### 5.1.2.7 print()

```
virtual QTextStream & quantier::Account::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quantier::Printable](#).

#### 5.1.2.8 setDisplayName()

```
void quantier::Account::setDisplayName (
    QString displayName )
```

Set the printable name of the account

#### 5.1.2.9 shardId()

```
QString quantier::Account::shardId ( ) const
```

##### Returns

Shard id for Evernote accounts, empty string for local accounts (as the concept of shard id is not defined for local accounts)

#### 5.1.2.10 type()

```
Type quantier::Account::type ( ) const
```

##### Returns

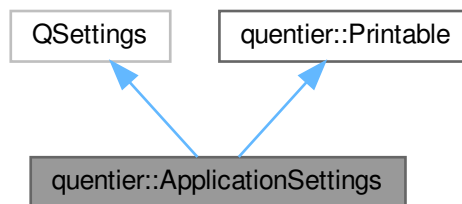
The type of the account: either local or Evernote

## 5.2 quantier::ApplicationSettings Class Reference

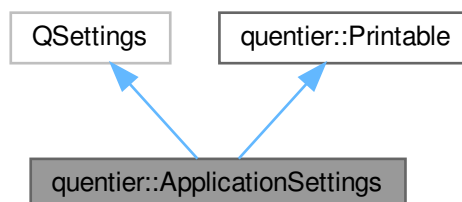
The [ApplicationSettings](#) class enhances the functionality of `QSettings`, in particular it simplifies the way of working with either application-wide or account-specific settings.

```
#include <ApplicationSettings.h>
```

Inheritance diagram for `quantier::ApplicationSettings`:



Collaboration diagram for `quantier::ApplicationSettings`:



### Classes

- struct [ArrayCloser](#)
- struct [GroupCloser](#)

### Public Member Functions

- [ApplicationSettings](#) (const `QString` &settingsName={})
- [ApplicationSettings](#) (const [Account](#) &account, const `QString` &settingsName={})
- [ApplicationSettings](#) (const [Account](#) &account, const `char` \*settingsName, const `int` settingsNameSize=-1)
- virtual `~ApplicationSettings` () override
- void [beginGroup](#) (const `QString` &prefix)
- void [beginGroup](#) (const `char` \*prefix, const `int` size=-1)

- int [beginReadArray](#) (const QString &prefix)
- int [beginReadArray](#) (const char \*prefix, const int size=-1)
- void [beginWriteArray](#) (const QString &prefix, const int arraySize=-1)
- void [beginWriteArray](#) (const char \*prefix, const int arraySize=-1, const int prefixSize=-1)
- bool [contains](#) (const QString &key) const
- bool [contains](#) (const char \*key, const int size=-1) const
- void [remove](#) (const QString &key)
- void [remove](#) (const char \*key, const int size=-1)
- void [setValue](#) (const QString &key, const QVariant &value)
- void [setValue](#) (const char \*key, const QVariant &value, const int keySize=-1)
- QVariant [value](#) (const QString &key, const QVariant &defaultValue={}) const
- QVariant [value](#) (const char \*key, const QVariant &defaultValue={}, const int keySize=-1) const
- virtual QTextStream & [print](#) (QTextStream &strm) const override

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString [toString](#) () const

## Additional Inherited Members

## Protected Member Functions inherited from [quentier::Printable](#)

- [Printable](#) (const [Printable](#) &other)
- [Printable](#) & [operator=](#) (const [Printable](#) &other)

### 5.2.1 Detailed Description

The [ApplicationSettings](#) class enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings.

### 5.2.2 Constructor & Destructor Documentation

#### 5.2.2.1 ApplicationSettings() [1/3]

```
quentier::ApplicationSettings::ApplicationSettings (
    const QString & settingsName = {} )
```

Constructor for application settings not being account-specific

#### Parameters

<i>settingsName</i>	If not empty, the created application settings would manage the settings stored in a file with a specific name within the common settings storage; otherwise they would be stored in the default settings file for the account
---------------------	--

### 5.2.2.2 ApplicationSettings() [2/3]

```
quentier::ApplicationSettings::ApplicationSettings (
    const Account & account,
    const QString & settingsName = {} )
```

Constructor for application settings specific to the account

#### Parameters

<i>account</i>	The account for which the settings are to be stored or read
<i>settingsName</i>	If not empty, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account

### 5.2.2.3 ApplicationSettings() [3/3]

```
quentier::ApplicationSettings::ApplicationSettings (
    const Account & account,
    const char * settingsName,
    const int settingsNameSize = -1 )
```

Constructor for application settings specific to the account

#### Parameters

<i>account</i>	The account for which the settings are to be stored or read
<i>settingsName</i>	If not nullptr, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>settingsNameSize</i>	Size of the settingsName string. If negative (the default), the settingsName size is taken to be strlen(settingsName)

### 5.2.2.4 ~ApplicationSettings()

```
virtual quentier::ApplicationSettings::~~ApplicationSettings ( ) [override], [virtual]
```

Destructor

## 5.2.3 Member Function Documentation

### 5.2.3.1 beginGroup() [1/2]

```
void quentier::ApplicationSettings::beginGroup (
    const char * prefix,
    const int size = -1 )
```

Appends prefix to the current group. Overload of beginGroup accepting const char \* and optionally the size of the string

## Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix).

**5.2.3.2 beginGroup()** [2/2]

```
void quantier::ApplicationSettings::beginGroup (
    const QString & prefix )
```

Appends prefix to the current group. The call is redirected to QSettings::beginGroup. It is required in this class only to workaround hiding QSettings method due to overloads

## Parameters

<i>prefix</i>	String containing the prefix name
---------------	-----------------------------------

**5.2.3.3 beginReadArray()** [1/2]

```
int quantier::ApplicationSettings::beginReadArray (
    const char * prefix,
    const int size = -1 )
```

Adds prefix to the current group and starts reading from an array. Overload of beginReadArray accepting const char \* and optionally the size of the string

## Parameters

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix)

**5.2.3.4 beginReadArray()** [2/2]

```
int quantier::ApplicationSettings::beginReadArray (
    const QString & prefix )
```

Adds prefix to the current group and starts reading from an array. The call is redirected to QSettings::beginReadArray. It is required in this class only to workaround hiding QSettings method due to overloads

## Parameters

<i>prefix</i>	String containing the prefix name
---------------	-----------------------------------

**Returns**

The size of the array

**5.2.3.5 beginWriteArray() [1/2]**

```
void quentier::ApplicationSettings::beginWriteArray (
    const char * prefix,
    const int arraySize = -1,
    const int prefixSize = -1 )
```

Adds prefix to the current group and starts writing an array of size *arraySize*. Overload of *beginWriteArray* accepting *const char \** and optionally the size of the string

**Parameters**

<i>prefix</i>	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>arraySize</i>	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.
<i>prefixSize</i>	Size of the prefix string. If negative (the default), the prefix size is taken to be strlen(prefix)

**5.2.3.6 beginWriteArray() [2/2]**

```
void quentier::ApplicationSettings::beginWriteArray (
    const QString & prefix,
    const int arraySize = -1 )
```

Adds prefix to the current group and starts writing an array of size *arraySize*. The call is redirected to QSettings::beginWriteArray. It is required in this class only to workaround hiding QSettings method due to overloads

**Parameters**

<i>prefix</i>	String containing the prefix name
<i>arraySize</i>	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.

**5.2.3.7 contains() [1/2]**

```
bool quentier::ApplicationSettings::contains (
    const char * key,
    const int size = -1 ) const
```

Overload of *contains* accepting *const char \** and optionally the size of the string

**Parameters**

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>size</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key)

**Returns**

True if there exists a setting called `key`; false otherwise

**5.2.3.8 contains() [2/2]**

```
bool quantier::ApplicationSettings::contains (
    const QString & key ) const
```

The call is redirected to `QSettings::contains`. It is required in this class only to workaround hiding `QSettings` method due to overloads

**Parameters**

<i>key</i>	The key being checked for presence
------------	------------------------------------

**Returns**

True if there exists a setting called `key`; false otherwise

**5.2.3.9 print()**

```
virtual QTextStream & quantier::ApplicationSettings::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quantier::Printable](#).

**5.2.3.10 remove() [1/2]**

```
void quantier::ApplicationSettings::remove (
    const char * key,
    const int size = -1 )
```

Removes the setting `key` and any sub-settings of `key`. Overload of `remove` accepting `const char *` and optionally the size of the string

**Parameters**

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to <code>QString</code> via <code>QString::fromUtf8</code>
<i>size</i>	Size of the key string. If negative (the default), the key size is taken to be <code>strlen(key)</code> .

**5.2.3.11 remove() [2/2]**

```
void quantier::ApplicationSettings::remove (
    const QString & key )
```

Removes the setting key and any sub-settings of key. The call is redirected to `QSettings::remove`. It is required in this class only to workaround hiding `QSettings` method due to overloads

## Parameters

<i>key</i>	String containing the setting name
------------	------------------------------------

**5.2.3.12 setValue()** [1/2]

```
void quantier::ApplicationSettings::setValue (
    const char * key,
    const QVariant & value,
    const int keySize = -1 )
```

Sets the value of setting. Overload of setValue accepting const char \* and optionally the size of the string

## Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>value</i>	Value for setting key
<i>keySize</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key).

**5.2.3.13 setValue()** [2/2]

```
void quantier::ApplicationSettings::setValue (
    const QString & key,
    const QVariant & value )
```

Sets the value of setting. The call is redirected to QSettings::setValue. It is required in this class only to workaround hiding QSettings method due to overloads

## Parameters

<i>key</i>	String containing the setting name
<i>value</i>	Value for setting key

**5.2.3.14 value()** [1/2]

```
QVariant quantier::ApplicationSettings::value (
    const char * key,
    const QVariant & defaultValue = {},
    const int keySize = -1 ) const
```

Fetches the value of setting. Overload of value accepting const char \* and optionally the size of the string

## Parameters

<i>key</i>	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
<i>defaultValue</i>	Default value returned if the setting doesn't exist
<i>keySize</i>	Size of the key string. If negative (the default), the key size is taken to be strlen(key)

**Returns**

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

**5.2.3.15 value() [2/2]**

```
QVariant quantier::ApplicationSettings::value (
    const QString & key,
    const QVariant & defaultValue = {} ) const
```

Fetches the value of setting. The call is redirected to QSettings::value. It is required in this class only to workaround hiding QSettings method due to overloads

**Parameters**

<i>key</i>	String containing the setting name
<i>defaultValue</i>	Default value returned if the setting doesn't exist

**Returns**

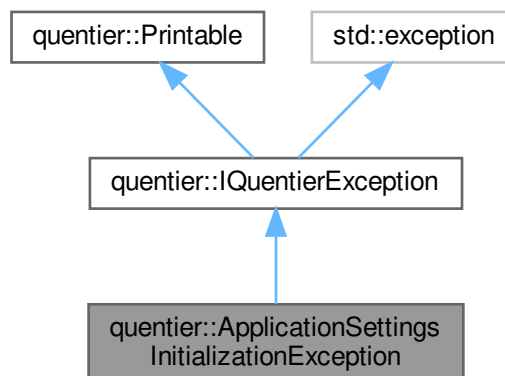
The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

**5.3 quantier::ApplicationSettingsInitializationException Class Reference**

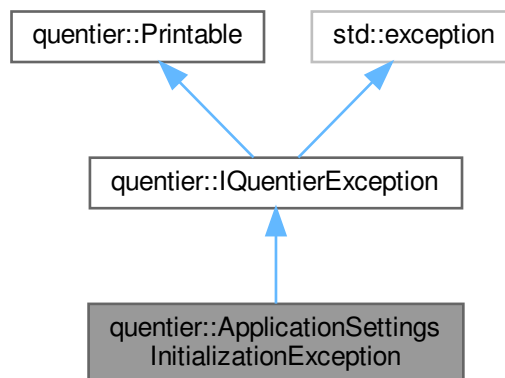
The [ApplicationSettingsInitializationException](#) can be thrown from methods of [ApplicationSettings](#) class if it's unable to locate the file with persistent settings.

```
#include <ApplicationSettingsInitializationException.h>
```

Inheritance diagram for quantier::ApplicationSettingsInitializationException:



Collaboration diagram for `quentier::ApplicationSettingsInitializationException`:



#### Public Member Functions

- **ApplicationSettingsInitializationException** (const [ErrorString](#) &message)

#### Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

#### Public Member Functions inherited from [quentier::Printable](#)

- virtual const **QString toString** () const

#### Protected Member Functions

- virtual const **QString exceptionDisplayName** () const override

#### Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [IQuentierException](#) &other)
- **IQuentierException & operator=** (const [IQuentierException](#) &other)

#### Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable & operator=** (const [Printable](#) &other)

### 5.3.1 Detailed Description

The [ApplicationSettingsInitializationException](#) can be thrown from methods of [ApplicationSettings](#) class if it's unable to locate the file with persistent settings.

### 5.3.2 Member Function Documentation

#### 5.3.2.1 `exceptionDisplayName()`

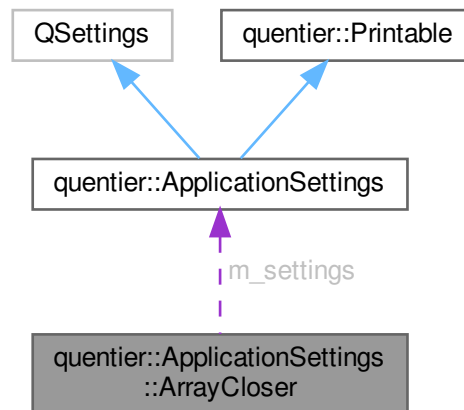
```
virtual const QString quentier::ApplicationSettingsInitializationException::exceptionDisplay←
Name ( ) const [override], [protected], [virtual]
```

Implements [quentier::IQuentierException](#).

## 5.4 `quentier::ApplicationSettings::ArrayCloser` Struct Reference

```
#include <ApplicationSettings.h>
```

Collaboration diagram for `quentier::ApplicationSettings::ArrayCloser`:



### Public Member Functions

- **ArrayCloser** ([ApplicationSettings](#) &settings)

### Public Attributes

- [ApplicationSettings](#) & **m\_settings**

### 5.4.1 Detailed Description

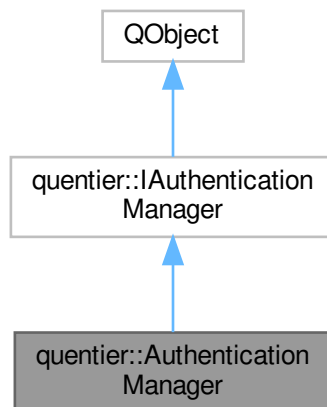
Helper struct for RAII style of ensuring the array once began would be closed even if exception is thrown after beginning the array

## 5.5 `quentier::AuthenticationManager` Class Reference

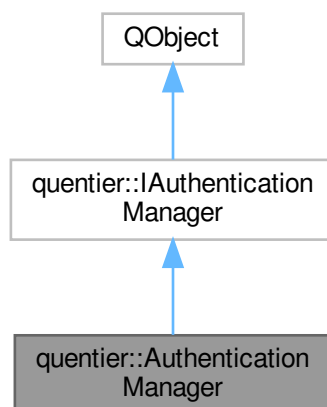
The `AuthenticationManager` class is libquentier's default implementation of `IAuthenticationManager` interface; internally uses QEverCloud's OAuth widget.

```
#include <AuthenticationManager.h>
```

Inheritance diagram for `quentier::AuthenticationManager`:



Collaboration diagram for `quentier::AuthenticationManager`:



### Public Slots

- virtual void **onAuthenticationRequest** () override

### Public Slots inherited from `quentier::IAuthenticationManager`

- virtual void **onAuthenticationRequest** ()=0

### Public Member Functions

- **AuthenticationManager** (const QString &consumerKey, const QString &consumerSecret, const QString &host, QObject \*parent=nullptr)

### Additional Inherited Members

### Signals inherited from `quentier::IAuthenticationManager`

- void **sendAuthenticationResult** (bool success, qevercloud::UserID userId, QString authToken, qevercloud::Timestamp authTokenExpirationTime, QString shardId, QString noteStoreUrl, QString webApiUrlPrefix, QList< QNetworkCookie > userStoreCookies, [ErrorString](#) errorDescription)

### Protected Member Functions inherited from `quentier::IAuthenticationManager`

- **IAuthenticationManager** (QObject \*parent=nullptr)

## 5.5.1 Detailed Description

The [AuthenticationManager](#) class is libquentier's default implementation of [IAuthenticationManager](#) interface; internally uses QEverCloud's OAuth widget.

## 5.6 `quentier::ResourceRecognitionIndexItem::BarcodeItem` Struct Reference

### Public Member Functions

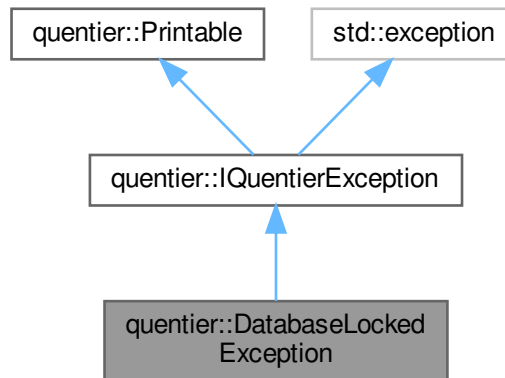
- bool **operator==** (const [BarcodeItem](#) &other) const

### Public Attributes

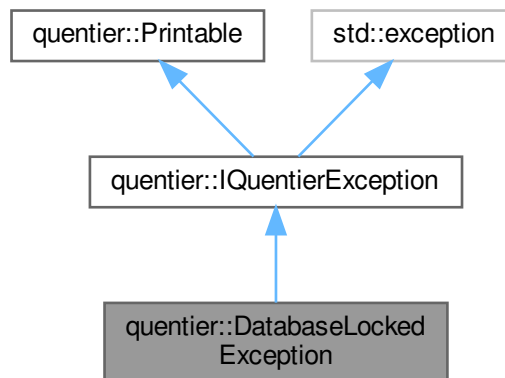
- QString **m\_barcode**
- int **m\_weight** = -1

## 5.7 `quentier::DatabaseLockedException` Class Reference

Inheritance diagram for `quentier::DatabaseLockedException`:



Collaboration diagram for `quentier::DatabaseLockedException`:



### Public Member Functions

- **DatabaseLockedException** (const [ErrorString](#) &message)

### Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

**Public Member Functions inherited from [quentier::Printable](#)**

- virtual const QString **toString** () const

**Protected Member Functions**

- virtual const QString [exceptionDisplayName](#) () const override

**Protected Member Functions inherited from [quentier::IQuentierException](#)**

- **IQuentierException** (const [IQuentierException](#) &other)
- [IQuentierException](#) & **operator=** (const [IQuentierException](#) &other)

**Protected Member Functions inherited from [quentier::Printable](#)**

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

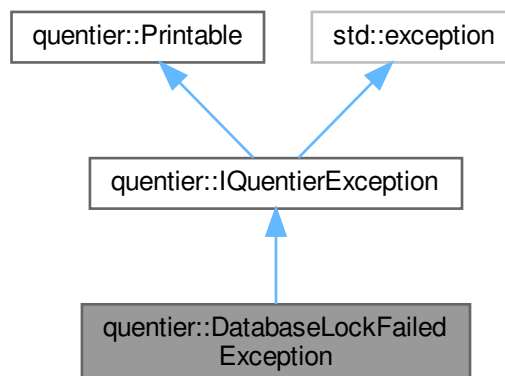
**5.7.1 Member Function Documentation****5.7.1.1 exceptionDisplayName()**

```
virtual const QString quentier::DatabaseLockedException::exceptionDisplayName ( ) const [override],
[protected], [virtual]
```

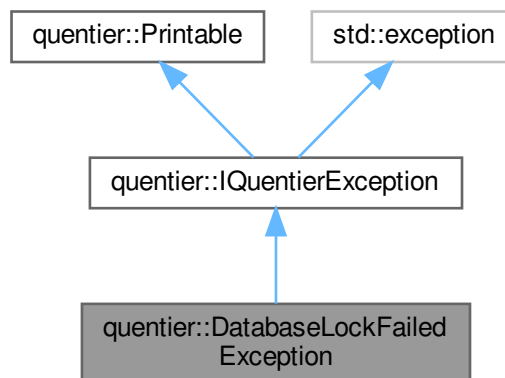
Implements [quentier::IQuentierException](#).

**5.8 quentier::DatabaseLockFailedException Class Reference**

Inheritance diagram for quentier::DatabaseLockFailedException:



Collaboration diagram for `quentier::DatabaseLockFailedException`:



#### Public Member Functions

- **DatabaseLockFailedException** (const [ErrorString](#) &message)

#### Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

#### Public Member Functions inherited from [quentier::Printable](#)

- virtual const **QString toString** () const

#### Protected Member Functions

- virtual const **QString exceptionDisplayName** () const override

#### Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [IQuentierException](#) &other)
- **IQuentierException & operator=** (const [IQuentierException](#) &other)

#### Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable & operator=** (const [Printable](#) &other)

## 5.8.1 Member Function Documentation

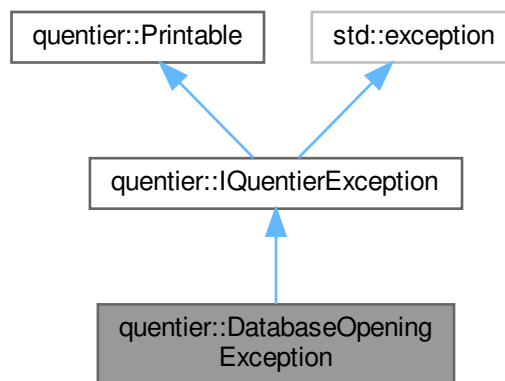
### 5.8.1.1 exceptionDisplayName()

```
virtual const QString quantier::DatabaseLockFailedException::exceptionDisplayName ( ) const  
[override], [protected], [virtual]
```

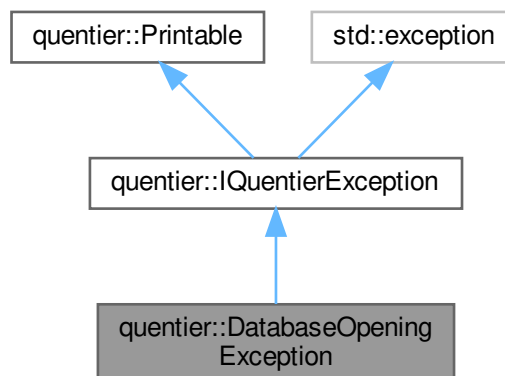
Implements [quantier::IQuantierException](#).

## 5.9 quantier::DatabaseOpeningException Class Reference

Inheritance diagram for quantier::DatabaseOpeningException:



Collaboration diagram for quantier::DatabaseOpeningException:



## Public Member Functions

- **DatabaseOpeningException** (const [ErrorString](#) &message)

## Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const **QString toString** () const

## Protected Member Functions

- virtual const **QString exceptionDisplayName** () const override

## Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [IQuentierException](#) &other)
- **IQuentierException & operator=** (const [IQuentierException](#) &other)

## Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable & operator=** (const [Printable](#) &other)

## 5.9.1 Member Function Documentation

### 5.9.1.1 exceptionDisplayName()

```
virtual const QString quentier::DatabaseOpeningException::exceptionDisplayName ( ) const [override],  
[protected], [virtual]
```

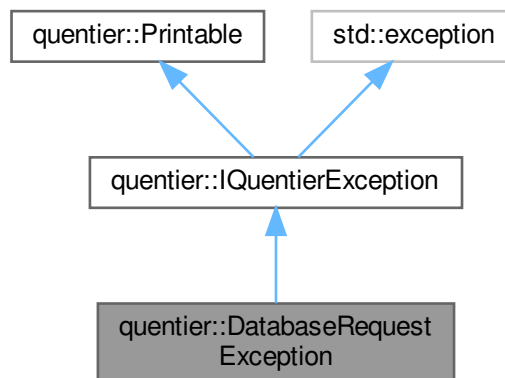
Implements [quentier::IQuentierException](#).

## 5.10 quantier::DatabaseRequestException Class Reference

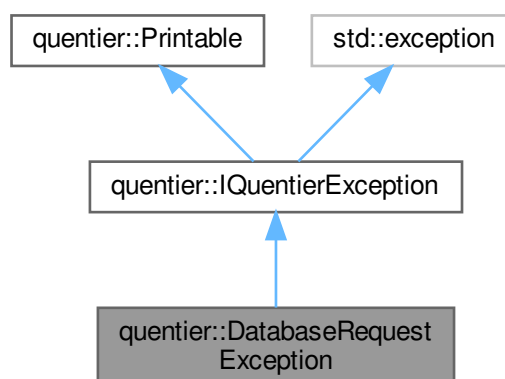
The [DatabaseRequestException](#) is thrown when the local storage database encounters some internal error during the attempt to serve a request.

```
#include <DatabaseRequestException.h>
```

Inheritance diagram for quantier::DatabaseRequestException:



Collaboration diagram for quantier::DatabaseRequestException:



### Public Member Functions

- **DatabaseRequestException** (const [ErrorString](#) &message)

## Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- QString **localizedErrorMessage** () const
- QString **nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual QTextStream & **print** (QTextStream &strm) const override

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

## Protected Member Functions

- virtual const QString **exceptionDisplayName** () const override

## Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [IQuentierException](#) &other)
- [IQuentierException](#) & **operator=** (const [IQuentierException](#) &other)

## Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

### 5.10.1 Detailed Description

The [DatabaseRequestException](#) is thrown when the local storage database encounters some internal error during the attempt to serve a request.

### 5.10.2 Member Function Documentation

#### 5.10.2.1 **exceptionDisplayName()**

```
virtual const QString quentier::DatabaseRequestException::exceptionDisplayName ( ) const [override],
[protected], [virtual]
```

Implements [quentier::IQuentierException](#).

## 5.11 [quentier::DateTimePrint](#) Class Reference

The [DateTimePrint](#) class simply wraps the enum containing datetime printing options.

```
#include <DateTime.h>
```

## Public Types

- enum `Option` { `IncludeNumericTimestamp` = 1 << 1 , `IncludeMilliseconds` = 1 << 2 , `IncludeTimezone` = 1 << 3 }

### 5.11.1 Detailed Description

The `DateTimePrint` class simply wraps the enum containing datetime printing options.

### 5.11.2 Member Enumeration Documentation

#### 5.11.2.1 Option

```
enum quentier::DateTimePrint::Option
```

Available printing options for datetime

#### Enumerator

<code>IncludeNumericTimestamp</code>	Include the numeric representation of the timestamp into the printed string
<code>IncludeMilliseconds</code>	Include milliseconds into the printed string
<code>IncludeTimezone</code>	Include timezone into the printed string WARNING: currently this option has no effect on Windows platform, the timezone is not included anyway.

## 5.12 `quentier::DecryptedTextManager` Class Reference

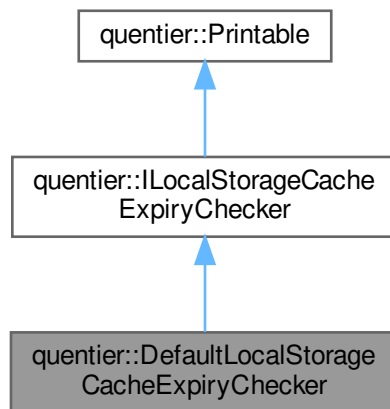
### Public Member Functions

- void **addEntry** (const QString &hash, const QString &decryptedText, const bool rememberForSession, const QString &passphrase, const QString &cipher, const size\_t keyLength)
- void **removeEntry** (const QString &hash)
- void **clearNonRememberedForSessionEntries** ()
- bool **findDecryptedTextByEncryptedText** (const QString &encryptedText, QString &decryptedText, bool &rememberForSession) const
- bool **modifyDecryptedText** (const QString &originalEncryptedText, const QString &newDecryptedText, QString &newEncryptedText)

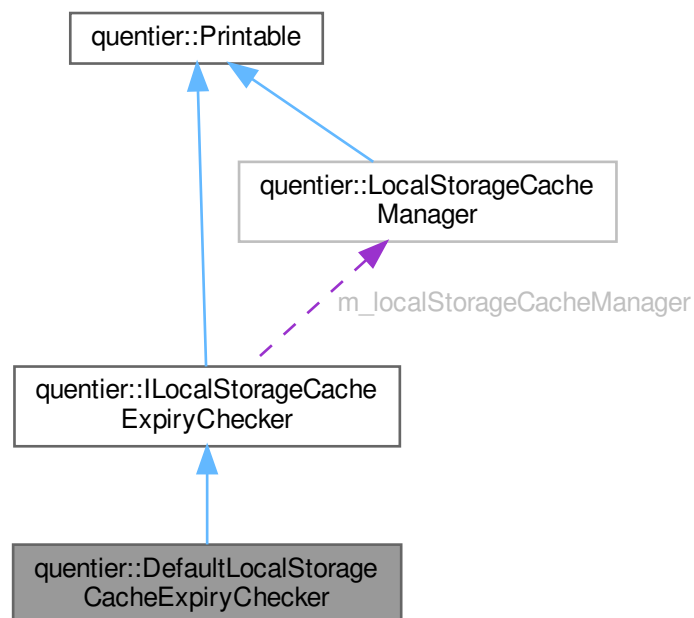
## 5.13 `quentier::DefaultLocalStorageCacheExpiryChecker` Class Reference

```
#include <DefaultLocalStorageCacheExpiryChecker.h>
```

Inheritance diagram for `quentier::DefaultLocalStorageCacheExpiryChecker`:



Collaboration diagram for `quentier::DefaultLocalStorageCacheExpiryChecker`:



## Public Member Functions

- **DefaultLocalStorageCacheExpiryChecker** (const [LocalStorageCacheManager](#) &cacheManager)

- virtual [DefaultLocalStorageCacheExpiryChecker](#) \* `clone` () const override
- virtual bool `checkNotes` () const override
- virtual bool `checkResources` () const override
- virtual bool `checkNotebooks` () const override
- virtual bool `checkTags` () const override
- virtual bool `checkLinkedNotebooks` () const override
- virtual bool `checkSavedSearches` () const override
- virtual QTextStream & `print` (QTextStream &strm) const override

*Print the internal information about the current [DefaultLocalStorageCacheExpiryChecker](#) instance to the text stream.*

### Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString `toString` () const

### Additional Inherited Members

### Protected Member Functions inherited from [quentier::ILocalStorageCacheExpiryChecker](#)

- [ILocalStorageCacheExpiryChecker](#) (const [LocalStorageCacheManager](#) &cacheManager)

### Protected Member Functions inherited from [quentier::Printable](#)

- [Printable](#) (const [Printable](#) &other)
- [Printable](#) & `operator=` (const [Printable](#) &other)

### Protected Attributes inherited from [quentier::ILocalStorageCacheExpiryChecker](#)

- const [LocalStorageCacheManager](#) & `m_localStorageCacheManager`

## 5.13.1 Detailed Description

brief The [DefaultLocalStorageCacheExpiryChecker](#) class is the implementation of [ILocalStorageCacheExpiryChecker](#) interface used by [LocalStorageCacheManager](#) by default, if no another implementation of [ILocalStorageCacheExpiryChecker](#) is set to be used by [LocalStorageCacheManager](#)

## 5.13.2 Member Function Documentation

### 5.13.2.1 `checkLinkedNotebooks()`

```
virtual bool quentier::DefaultLocalStorageCacheExpiryChecker::checkLinkedNotebooks ( ) const
[override], [virtual]
```

#### Returns

False if the current number of cached linked notebooks is higher than a reasonable limit, true otherwise

Implements [quentier::ILocalStorageCacheExpiryChecker](#).

### 5.13.2.2 checkNotebooks()

```
virtual bool quentier::DefaultLocalStorageCacheExpiryChecker::checkNotebooks ( ) const [override],  
[virtual]
```

#### Returns

False if the current number of cached notebooks is higher than a reasonable limit, true otherwise

Implements [quentier::ILocalStorageCacheExpiryChecker](#).

### 5.13.2.3 checkNotes()

```
virtual bool quentier::DefaultLocalStorageCacheExpiryChecker::checkNotes ( ) const [override],  
[virtual]
```

#### Returns

False if the current number of cached notes is higher than a reasonable limit, true otherwise

Implements [quentier::ILocalStorageCacheExpiryChecker](#).

### 5.13.2.4 checkResources()

```
virtual bool quentier::DefaultLocalStorageCacheExpiryChecker::checkResources ( ) const [override],  
[virtual]
```

#### Returns

False if the current number of cached resource is higher than a reasonable limit, true otherwise

Implements [quentier::ILocalStorageCacheExpiryChecker](#).

### 5.13.2.5 checkSavedSearches()

```
virtual bool quentier::DefaultLocalStorageCacheExpiryChecker::checkSavedSearches ( ) const  
[override], [virtual]
```

#### Returns

False if the current number of cached saved searches is higher than a reasonable limit, true otherwise

Implements [quentier::ILocalStorageCacheExpiryChecker](#).

### 5.13.2.6 checkTags()

```
virtual bool quantier::DefaultLocalStorageCacheExpiryChecker::checkTags ( ) const [override],  
[virtual]
```

#### Returns

False if the current number of cached tags is higher than a reasonable limit, true otherwise

Implements [quantier::ILocalStorageCacheExpiryChecker](#).

### 5.13.2.7 clone()

```
virtual DefaultLocalStorageCacheExpiryChecker * quantier::DefaultLocalStorageCacheExpiry↔  
Checker::clone ( ) const [override], [virtual]
```

#### Returns

A pointer to the newly allocated copy of the current [DefaultLocalStorageCacheExpiryChecker](#)

Implements [quantier::ILocalStorageCacheExpiryChecker](#).

### 5.13.2.8 print()

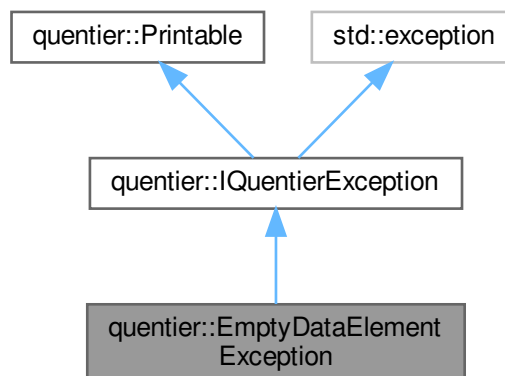
```
virtual QTextStream & quantier::DefaultLocalStorageCacheExpiryChecker::print (   
    QTextStream & strm ) const [override], [virtual]
```

Print the internal information about the current [DefaultLocalStorageCacheExpiryChecker](#) instance to the text stream.

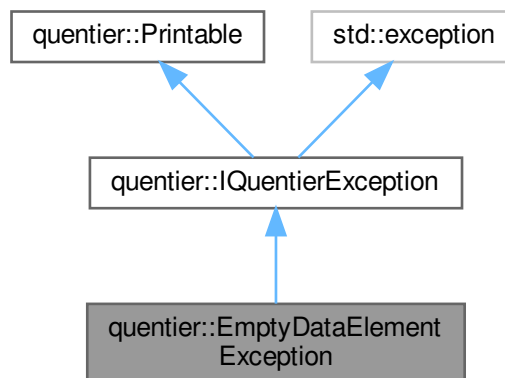
Implements [quantier::ILocalStorageCacheExpiryChecker](#).

## 5.14 quantier::EmptyDataElementException Class Reference

Inheritance diagram for `quantier::EmptyDataElementException`:



Collaboration diagram for `quentier::EmptyDataElementException`:



#### Public Member Functions

- **EmptyDataElementException** (const [ErrorString](#) &message)

#### Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

#### Public Member Functions inherited from [quentier::Printable](#)

- virtual const **QString toString** () const

#### Protected Member Functions

- virtual const **QString exceptionDisplayName** () const override

#### Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [IQuentierException](#) &other)
- **IQuentierException & operator=** (const [IQuentierException](#) &other)

#### Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable & operator=** (const [Printable](#) &other)

### 5.14.1 Member Function Documentation

#### 5.14.1.1 exceptionDisplayName()

```
virtual const QString quantier::EmptyDataElementException::exceptionDisplayName ( ) const
[override], [protected], [virtual]
```

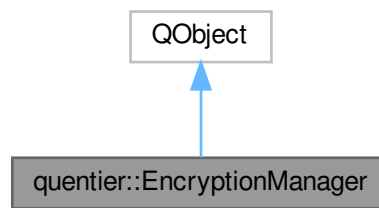
Implements [quantier::IQuantierException](#).

## 5.15 quantier::EncryptionManager Class Reference

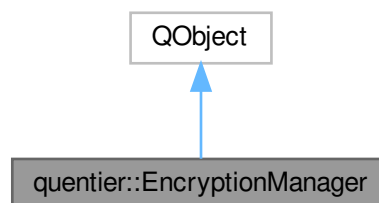
The [EncryptionManager](#) class provides both synchronous methods to encrypt or decrypt given text with password, cipher and key length and their signal-slot based potentially asynchronous counterparts.

```
#include <EncryptionManager.h>
```

Inheritance diagram for quantier::EncryptionManager:



Collaboration diagram for quantier::EncryptionManager:



### Public Slots

- void **onDecryptTextRequest** (QString encryptedText, QString passphrase, QString cipher, size\_t keyLength, QUuid requestId)
- void **onEncryptTextRequest** (QString textToEncrypt, QString passphrase, QString cipher, size\_t keyLength, QUuid requestId)

## Signals

- void **decryptedText** (QString text, bool success, [ErrorString](#) errorDescription, QUuid requestId)
- void **encryptedText** (QString encryptedText, bool success, [ErrorString](#) errorDescription, QUuid requestId)

## Public Member Functions

- **EncryptionManager** (QObject \*parent=nullptr)
- bool **decrypt** (const QString &encryptedText, const QString &passphrase, const QString &cipher, const size\_t keyLength, QString &decryptedText, [ErrorString](#) &errorDescription)
- bool **encrypt** (const QString &textToEncrypt, const QString &passphrase, QString &cipher, size\_t &keyLength, QString &encryptedText, [ErrorString](#) &errorDescription)

### 5.15.1 Detailed Description

The [EncryptionManager](#) class provides both synchronous methods to encrypt or decrypt given text with password, cipher and key length and their signal-slot based potentially asynchronous counterparts.

## 5.16 quotientier::ENMLConverter Class Reference

The [ENMLConverter](#) class encapsulates a set of methods and helper data structures for performing the conversions between ENML and other note content formats, namely HTML.

```
#include <ENMLConverter.h>
```

## Classes

- struct [NoteContentToHtmlExtraData](#)
- class [SkipHtmlElementRule](#)

*The [SkipHtmlElementRule](#) class describes the set of rules for HTML -> ENML conversion about the HTML elements that should not be actually converted to ENML due to their nature of being "helper" elements for the display or functioning of something within the note editor's page. The HTML -> ENML conversion would ignore tags and attributes forbidden by ENML even without these rules conditionally preserving or skipping the contents and nested elements of skipped elements.*

## Public Types

- enum class [EnexExportTags](#) { **Yes** = 0 , **No** }

*The [EnexExportTags](#) enum allows to specify whether export of note(s) to ENEX should include the names of note's tags.*

## Public Member Functions

- bool **htmlToNoteContent** (const QString &html, QString &noteContent, [DecryptedTextManager](#) &decryptedTextManager, [ErrorString](#) &errorDescription, const QVector< [SkipHtmlElementRule](#) > &skipRules=()) const
- bool **cleanupExternalHtml** (const QString &inputHtml, QString &cleanedUpHtml, [ErrorString](#) &errorDescription) const  
*cleanupExternalHtml method cleans up a piece of HTML coming from some external source: the cleanup includes the removal (or replacement with equivalents/alternatives) of any tags and attributes not supported by the ENML representation of note page's HTML*
- bool **htmlToQTextDocument** (const QString &html, QTextDocument &doc, [ErrorString](#) &errorDescription, const QVector< [SkipHtmlElementRule](#) > &skipRules=()) const
- bool **noteContentToHtml** (const QString &noteContent, QString &html, [ErrorString](#) &errorDescription, [DecryptedTextManager](#) &decryptedTextManager, [NoteContentToHtmlExtraData](#) &extraData) const
- bool **validateEnml** (const QString &enml, [ErrorString](#) &errorDescription) const
- bool **validateAndFixupEnml** (QString &enml, [ErrorString](#) &errorDescription) const
- bool **exportNotesToEnex** (const QVector< [Note](#) > &notes, const QHash< QString, QString > &tagNamesByTagLocalUids, const [EnexExportTags](#) exportTagsOption, QString &enex, [ErrorString](#) &errorDescription, const QString &version=()) const  
*exportNotesToEnex exports either a single note or a set of notes into ENEX format*
- bool **importEnex** (const QString &enex, QVector< [Note](#) > &notes, QHash< QString, QStringList > &tagNamesByNoteLocalUid, [ErrorString](#) &errorDescription) const  
*importEnex reads the content of input ENEX file and converts it into a set of notes and tag names.*

## Static Public Member Functions

- static bool **noteContentToPlainText** (const QString &noteContent, QString &plainText, [ErrorString](#) &errorMessage)
- static bool **noteContentToListOfWords** (const QString &noteContent, QStringList &listOfWords, [ErrorString](#) &errorMessage, QString \*plainText=nullptr)
- static QStringList **plainTextToListOfWords** (const QString &plainText)
- static QString **toDoCheckboxHtml** (const bool checked, const quint64 idNumber)
- static QString **encryptedTextHtml** (const QString &encryptedText, const QString &hint, const QString &cipher, const size\_t keyLength, const quint64 enCryptIndex)
- static QString **decryptedTextHtml** (const QString &decryptedText, const QString &encryptedText, const QString &hint, const QString &cipher, const size\_t keyLength, const quint64 enDecryptedIndex)
- static QString **resourceHtml** (const [Resource](#) &resource, [ErrorString](#) &errorDescription)
- static void **escapeString** (QString &string, const bool simplify=true)

### 5.16.1 Detailed Description

The [ENMLConverter](#) class encapsulates a set of methods and helper data structures for performing the conversions between ENML and other note content formats, namely HTML.

### 5.16.2 Member Function Documentation

#### 5.16.2.1 cleanupExternalHtml()

```
bool quantier::ENMLConverter::cleanupExternalHtml (
    const QString & inputHtml,
    QString & cleanedUpHtml,
    ErrorString & errorDescription ) const
```

cleanupExternalHtml method cleans up a piece of HTML coming from some external source: the cleanup includes the removal (or replacement with equivalents/alternatives) of any tags and attributes not supported by the ENML representation of note page's HTML

## Parameters

<i>inputHtml</i>	- the input HTML to be cleaned up
<i>cleanedUpHtml</i>	- the result of the method's work
<i>errorDescription</i>	- the textual description of the error if conversion of input HTML into QTextDocument has failed

## Returns

true in case of successful conversion, false otherwise

## 5.16.2.2 exportNotesToEnex()

```
bool quantier::ENMLConverter::exportNotesToEnex (
    const QVector< Note > & notes,
    const QHash< QString, QString > & tagNamesByTagLocalUids,
    const EnexExportTags exportTagsOption,
    QString & enex,
    ErrorString & errorDescription,
    const QString & version = {} ) const
```

exportNotesToEnex exports either a single note or a set of notes into ENEX format

## Parameters

<i>notes</i>	The notes to be exported into the enex format. The connection of particular notes to tags is expected to follow from note's tag local uids. In other words, if some note has no tag local uids, its corresponding fragment of ENEX won't contain tag names associated with the note
<i>tagNamesByTagLocalUids</i>	Tag names for all tag local uids across all passed in notes. The lack of any tag name for any tag local uid is considered an error and the overall export attempt fails
<i>exportTagsOption</i>	Whether the export to ENEX should include the names of notes' tags
<i>enex</i>	The output of the method
<i>errorDescription</i>	The textual description of the error, if any
<i>version</i>	Optional "version" tag for the ENEX. If not set, the corresponding ENEX tag is set to empty value

## Returns

True if the export completed successfully, false otherwise

## 5.16.2.3 htmlToQTextDocument()

```
bool quantier::ENMLConverter::htmlToQTextDocument (
    const QString & html,
    QTextDocument & doc,
    ErrorString & errorDescription,
    const QVector< SkipHtmlElementRule > & skipRules = {} ) const
```

Converts the passed in HTML into its simplified form acceptable by QTextDocument (see <http://doc.qt.io/qt-5/richtext-html-subset.html> for the list of elements supported by QTextDocument)

## Parameters

<i>html</i>	- the input HTML which needs to be converted to QTextDocument
<i>doc</i>	- QTextDocument filled with the result of the method's work
<i>errorDescription</i>	- the textual description of the error if conversion of input HTML into QTextDocument has failed
<i>skipRules</i>	- rules for skipping the particular elements

## Returns

true in case of successful conversion, false otherwise

## 5.16.2.4 importEnex()

```
bool quantier::ENMLConverter::importEnex (
    const QString & enex,
    QVector< Note > & notes,
    QHash< QString, QStringList > & tagNamesByNoteLocalUid,
    ErrorString & errorDescription ) const
```

importEnex reads the content of input ENEX file and converts it into a set of notes and tag names.

## Parameters

<i>enex</i>	The input ENEX file contents
<i>notes</i>	Notes read from the ENEX
<i>tagNamesByNoteLocalUid</i>	Tag names per each read note; it is the responsibility of the method caller to find the actual tags corresponding to these names and set the tag local uids and/or guids to the note
<i>errorDescription</i>	The textual description of the error if the ENEX file could not be read and converted into a set of notes and tag names for them

## Returns

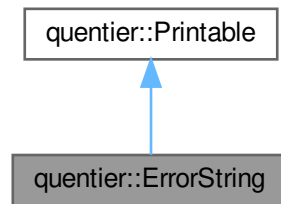
True if the ENEX file was read and converted into a set of notes and tag names successfully, false otherwise

## 5.17 quantier::ErrorString Class Reference

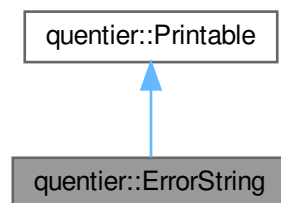
The [ErrorString](#) class encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description.

```
#include <ErrorString.h>
```

Inheritance diagram for `quentier::ErrorString`:



Collaboration diagram for `quentier::ErrorString`:



## Public Member Functions

- **ErrorString** (const char \*error=nullptr)
- **ErrorString** (const QString &error)
- **ErrorString** (const [ErrorString](#) &other)
- [ErrorString](#) & **operator=** (const [ErrorString](#) &other)
- const QString & **base** () const
- QString & **base** ()
- const QStringList & **additionalBases** () const
- QStringList & **additionalBases** ()
- const QString & **details** () const
- QString & **details** ()
- void **setBase** (const QString &error)
- void **setBase** (const char \*error)
- void **appendBase** (const QString &error)
- void **appendBase** (const QStringList &errors)
- void **appendBase** (const char \*error)
- void **setDetails** (const QString &error)
- void **setDetails** (const char \*error)
- bool **isEmpty** () const
- void **clear** ()
- QString **localizedString** () const
- QString **nonLocalizedString** () const
- virtual QTextStream & [print](#) (QTextStream &strm) const override

## Public Member Functions inherited from `quentier::Printable`

- virtual const QString **toString** () const

## Additional Inherited Members

## Protected Member Functions inherited from `quentier::Printable`

- **Printable** (const `Printable` &other)
- `Printable` & **operator=** (const `Printable` &other)

### 5.17.1 Detailed Description

The `ErrorString` class encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description.

1. `base()` methods return const and non-const links to the primary translatable string
2. `details()` methods return const and non-const links to non-translatable string (coming from some third party library etc)
3. `additionalBases()` methods return const and non-const links to additional translatable strings; one translatable string is not always enough because the error message might be composed from different parts

### 5.17.2 Member Function Documentation

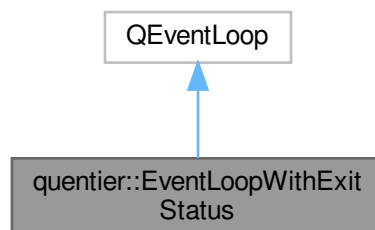
#### 5.17.2.1 `print()`

```
virtual QTextStream & quentier::ErrorString::print (
    QTextStream & strm ) const [override], [virtual]
```

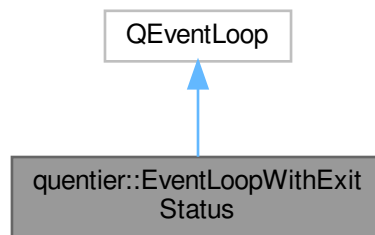
Implements `quentier::Printable`.

## 5.18 `quentier::EventLoopWithExitStatus` Class Reference

Inheritance diagram for `quentier::EventLoopWithExitStatus`:



Collaboration diagram for `quentier::EventLoopWithExitStatus`:



### Public Types

- enum class **ExitStatus** { **Success** , **Failure** , **Timeout** }

### Public Slots

- void **exitAsSuccess** ()
- void **exitAsFailure** ()
- void **exitAsFailureWithError** (QString errorDescription)
- void **exitAsFailureWithErrorString** ([ErrorString](#) errorDescription)
- void **exitAsTimeout** ()

### Public Member Functions

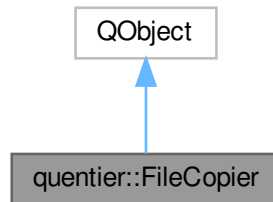
- **EventLoopWithExitStatus** (QObject \*parent=nullptr)
- ExitStatus **exitStatus** () const
- const [ErrorString](#) & **errorDescription** () const

### Friends

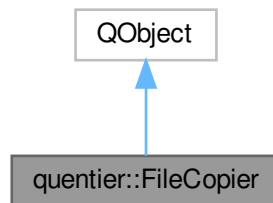
- QDebug & **operator**<< (QDebug &dbg, const ExitStatus status)
- QTextStream & **operator**<< (QTextStream &strm, const ExitStatus status)

## 5.19 quantier::FileCopier Class Reference

Inheritance diagram for quantier::FileCopier:



Collaboration diagram for quantier::FileCopier:



### Public Types

- enum class **State** { **Idle** = 0 , **Copying** , **Cancelling** }

### Public Slots

- void **copyFile** (QString sourcePath, QString destPath)
- void **cancel** ()

### Signals

- void **progressUpdate** (double progress)
- void **finished** (QString sourcePath, QString destPath)
- void **canceled** (QString sourcePath, QString destPath)
- void **notifyError** ([ErrorString](#) error)

### Public Member Functions

- **FileCopier** (QObject \*parent=nullptr)
- State **state** () const
- QString **sourceFilePath** () const
- QString **destinationFilePath** () const
- double **currentProgress** () const

### Friends

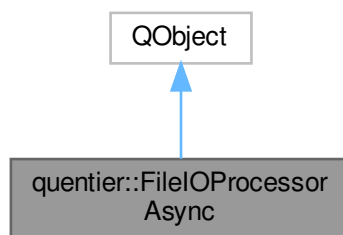
- QDebug & **operator**<< (QDebug &dbg, const State state)
- QTextStream & **operator**<< (QTextStream &strm, const State state)

## 5.20 quantier::FileIOProcessorAsync Class Reference

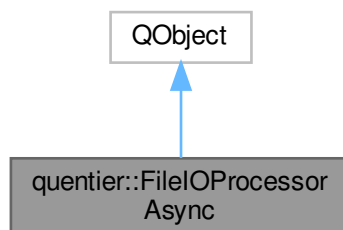
The [FileIOProcessorAsync](#) class is a wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO.

```
#include <FileIOProcessorAsync.h>
```

Inheritance diagram for quantier::FileIOProcessorAsync:



Collaboration diagram for quantier::FileIOProcessorAsync:



## Public Slots

- void [onWriteFileRequest](#) (QString absoluteFilePath, QByteArray data, QUuid requestId, bool append)  
*onWriteFileRequest slot processes file write requests with given request ids*
- void [onReadFileRequest](#) (QString absoluteFilePath, QUuid requestId)  
*onReadFileRequest slot processes file read requests with given request ids*

## Signals

- void **readyForIO** ()  
*readyForIO signal is emitted when the queue for file IO is empty for some time (30 seconds by default, can also be configured via [setIdleTimePeriod](#) method) after the last IO event to signal listeners that they can perform some IO via the [FileIOProcessorAsync](#)*
- void [writeFileRequestProcessed](#) (bool success, [ErrorString](#) errorDescription, QUuid requestId)  
*writeFileRequestProcessed signal is emitted when the file write request with given id is finished*
- void [readFileRequestProcessed](#) (bool success, [ErrorString](#) errorDescription, QByteArray data, QUuid requestId)  
*readFileRequestProcessed signal is emitted when the file read request with given id is finished*

## Public Member Functions

- **FileIOProcessorAsync** (QObject \*parent=nullptr)
- void [setIdleTimePeriod](#) (qint32 seconds)  
*setIdleTimePeriod sets time period defining the idle state of [FileIOProcessorAsync](#): once the time measured since the last IO operation is over the specified number of seconds, the class emits readyForIO signal to any interested listeners of this event. If this method is not called ever, the default idle time period would be 30 seconds.*

### 5.20.1 Detailed Description

The [FileIOProcessorAsync](#) class is a wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO.

### 5.20.2 Member Function Documentation

#### 5.20.2.1 onReadFileRequest

```
void quantier::FileIOProcessorAsync::onReadFileRequest (
    QString absoluteFilePath,
    QUuid requestId ) [slot]
```

onReadFileRequest slot processes file read requests with given request ids

#### Parameters

<i>absoluteFilePath</i>	Absolute file path to be read
<i>requestId</i>	Unique identifier of the file read request

### 5.20.2.2 onWriteFileRequest

```
void quentier::FileIOProcessorAsync::onWriteFileRequest (
    QString absoluteFilePath,
    QByteArray data,
    QUuid requestId,
    bool append ) [slot]
```

onWriteFileRequest slot processes file write requests with given request ids

#### Parameters

<i>absoluteFilePath</i>	Absolute file path to be written
<i>data</i>	Data to be written to the file
<i>requestId</i>	Unique identifier of the file write request
<i>append</i>	If true, the data would be appended to file, otherwise the entire file would be erased before with the data is written

### 5.20.2.3 readFileRequestProcessed

```
void quentier::FileIOProcessorAsync::readFileRequestProcessed (
    bool success,
    QString errorDescription,
    QByteArray data,
    QUuid requestId ) [signal]
```

readFileRequestProcessed signal is emitted when the file read request with given id is finished

#### Parameters

<i>success</i>	True if read operation was successful, false otherwise
<i>errorDescription</i>	Textual description of the error
<i>data</i>	Data read from file
<i>requestId</i>	Unique identifier of the file read request

### 5.20.2.4 setIdleTimePeriod()

```
void quentier::FileIOProcessorAsync::setIdleTimePeriod (
    quint32 seconds )
```

setIdleTimePeriod sets time period defining the idle state of [FileIOProcessorAsync](#): once the time measured since the last IO operation is over the specified number of seconds, the class emits readyForIO signal to any interested listeners of this event. If this method is not called ever, the default idle time period would be 30 seconds.

#### Parameters

<i>seconds</i>	Number of seconds for idle time period
----------------	--

### 5.20.2.5 writeFileRequestProcessed

```
void quantier::FileIOProcessorAsync::writeFileRequestProcessed (
    bool success,
    ErrorString errorDescription,
    QUuid requestId ) [signal]
```

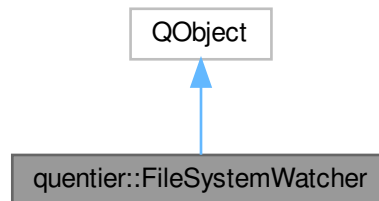
writeFileRequestProcessed signal is emitted when the file write request with given id is finished

#### Parameters

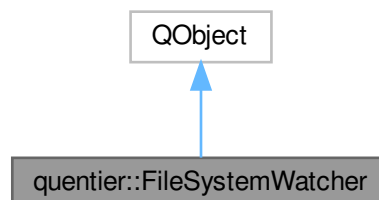
<i>success</i>	True if write operation was successful, false otherwise
<i>errorDescription</i>	Textual description of the error
<i>requestId</i>	Unique identifier of the file write request

## 5.21 quantier::FileSystemWatcher Class Reference

Inheritance diagram for quantier::FileSystemWatcher:



Collaboration diagram for quantier::FileSystemWatcher:



## Signals

- void **directoryChanged** (const QString &path)
- void **directoryRemoved** (const QString &path)
- void **fileChanged** (const QString &path)
- void **fileRemoved** (const QString &path)

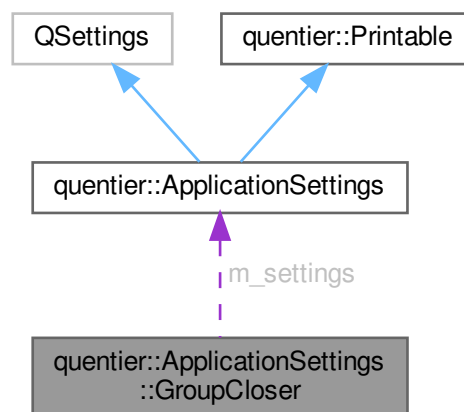
## Public Member Functions

- **FileSystemWatcher** (const int removalTimeoutMSec=FILE\_SYSTEM\_WATCHER\_DEFAULT\_REMOVAL\_TIMEOUT\_MSEC, QObject \*parent=nullptr)
- **FileSystemWatcher** (const QStringList &paths, const int removalTimeoutMSec=FILE\_SYSTEM\_WATCHER\_DEFAULT\_REMOVAL\_TIMEOUT\_MSEC, QObject \*parent=nullptr)
- void **addPath** (const QString &path)
- void **addPaths** (const QStringList &paths)
- QStringList **directories** () const
- QStringList **files** () const
- void **removePath** (const QString &path)
- void **removePaths** (const QStringList &paths)

## 5.22 quantier::ApplicationSettings::GroupCloser Struct Reference

```
#include <ApplicationSettings.h>
```

Collaboration diagram for quantier::ApplicationSettings::GroupCloser:



## Public Member Functions

- **GroupCloser** ([ApplicationSettings](#) &settings)

### Public Attributes

- [ApplicationSettings](#) & `m_settings`

#### 5.22.1 Detailed Description

Helper struct for RAIL style of ensuring the group once opened would be closed even if exception is thrown after beginning the group

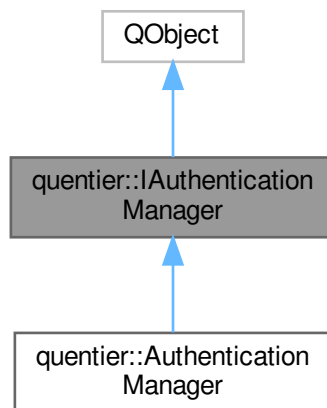
## 5.23 `quentier::HTMLCleaner` Class Reference

### Public Member Functions

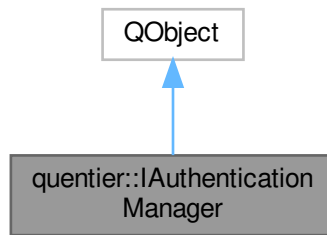
- `bool htmlToXml` (const QString &html, QString &output, QString &errorDescription)
- `bool htmlToXhtml` (const QString &html, QString &output, QString &errorDescription)
- `bool cleanupHtml` (QString &html, QString &errorDescription)

## 5.24 `quentier::IAuthenticationManager` Class Reference

Inheritance diagram for `quentier::IAuthenticationManager`:



Collaboration diagram for `quentier::IAuthenticationManager`:



### Public Slots

- virtual void **onAuthenticationRequest** ()=0

### Signals

- void **sendAuthenticationResult** (bool success, `qevercloud::UserID` userId, `QString` authToken, `qevercloud::Timestamp` authTokenExpirationTime, `QString` shardId, `QString` noteStoreUrl, `QString` webApiUrlPrefix, `QList< QNetworkCookie >` userStoreCookies, `ErrorString` errorDescription)

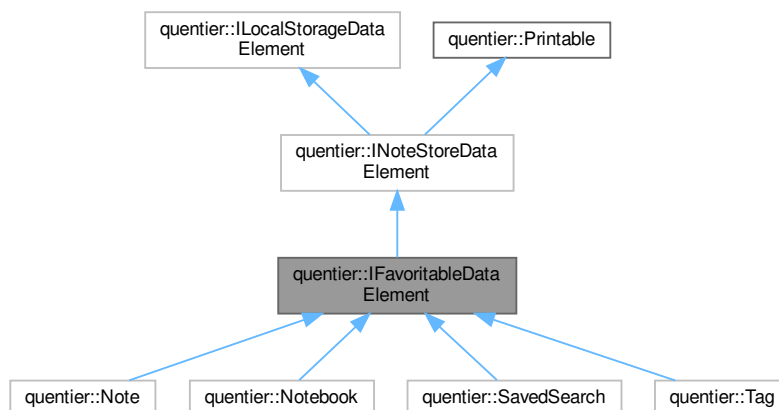
### Protected Member Functions

- **IAuthenticationManager** (`QObject *parent=nullptr`)

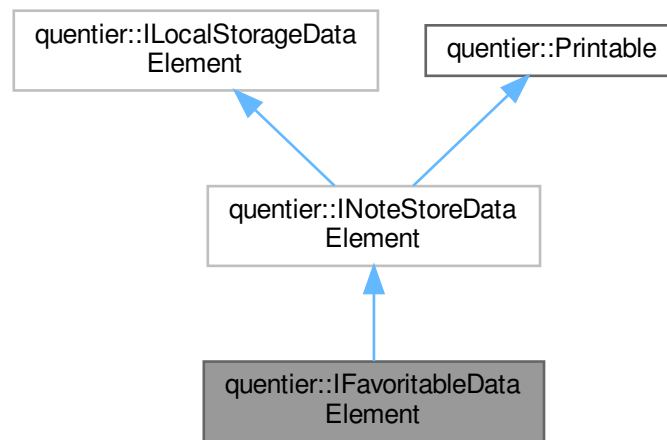
## 5.25 quentier::IFavoritableDataElement Class Reference

```
#include <IFavoritableDataElement.h>
```

Inheritance diagram for `quentier::IFavoritableDataElement`:



Collaboration diagram for quantier::IFavoritableDataElement:



#### Public Member Functions

- virtual bool **isFavorited** () const =0
- virtual void **setFavorited** (const bool favored)=0

#### Public Member Functions inherited from [quantier::INoteStoreDataElement](#)

- virtual void **clear** ()=0
- virtual bool **hasGuid** () const =0
- virtual const QString & **guid** () const =0
- virtual void **setGuid** (const QString &guid)=0
- virtual bool **hasUpdateSequenceNumber** () const =0
- virtual qint32 **updateSequenceNumber** () const =0
- virtual void **setUpdateSequenceNumber** (const qint32 usn)=0
- virtual bool **checkParameters** ([ErrorString](#) &errorDescription) const =0
- virtual bool **isDirty** () const =0
- virtual void **setDirty** (const bool dirty)=0
- virtual bool **isLocal** () const =0
- virtual void **setLocal** (const bool local)=0

#### Public Member Functions inherited from [quantier::ILocalStorageDataElement](#)

- virtual const QString **localUid** () const =0
- virtual void **setLocalUid** (const QString &guid)=0
- virtual void **unsetLocalUid** ()=0

#### Public Member Functions inherited from [quantier::Printable](#)

- virtual QTextStream & **print** (QTextStream &strm) const =0
- virtual const QString **toString** () const

## Additional Inherited Members

## Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

### 5.25.1 Detailed Description

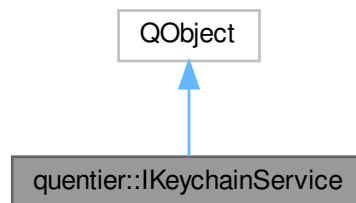
Class adding one more attribute to each note store data element subclassing it: "favorited" flag. Favorited data elements are expected to be somehow arranged for quick access in the client application's UI.

## 5.26 [quentier::IKeychainService](#) Class Reference

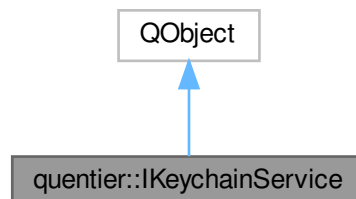
The [IKeychainService](#) interface provides methods intended to start potentially asynchronous interaction with the keychain and signals intended to notify listeners about the completion of asynchronous interactions.

```
#include <IKeychainService.h>
```

Inheritance diagram for [quentier::IKeychainService](#):



Collaboration diagram for [quentier::IKeychainService](#):



## Public Types

- enum class [ErrorCode](#) {  
[NoError](#) , [EntryNotFound](#) , [CouldNotDeleteEntry](#) , [AccessDeniedByUser](#) ,  
[AccessDenied](#) , [NoBackendAvailable](#) , [NotImplemented](#) , [OtherError](#) }

## Signals

- void [writePasswordJobFinished](#) (QUuid requestId, [ErrorCode](#) errorCode, [ErrorString](#) errorDescription)
- void [readPasswordJobFinished](#) (QUuid requestId, [ErrorCode](#) errorCode, [ErrorString](#) errorDescription, QString password)
- void [deletePasswordJobFinished](#) (QUuid requestId, [ErrorCode](#) errorCode, [ErrorString](#) errorDescription)

## Public Member Functions

- virtual QUuid [startWritePasswordJob](#) (const QString &service, const QString &key, const QString &password)=0
- virtual QUuid [startReadPasswordJob](#) (const QString &service, const QString &key)=0
- virtual QUuid [startDeletePasswordJob](#) (const QString &service, const QString &key)=0

## Protected Member Functions

- [IKeychainService](#) (QObject \*parent=nullptr)

## Friends

- QTextStream & **operator**<< (QTextStream &strm, const [ErrorCode](#) errorCode)
- QDebug & **operator**<< (QDebug &dbg, const [ErrorCode](#) errorCode)

### 5.26.1 Detailed Description

The [IKeychainService](#) interface provides methods intended to start potentially asynchronous interaction with the keychain and signals intended to notify listeners about the completion of asynchronous interactions.

### 5.26.2 Member Enumeration Documentation

#### 5.26.2.1 ErrorCode

```
enum class quantier::IKeychainService::ErrorCode [strong]
```

Error codes for results of operations with the keychain service

#### Enumerator

<a href="#">NoError</a>	No error occurred, operation was successful
<a href="#">EntryNotFound</a>	For the given key no data was found
<a href="#">CouldNotDeleteEntry</a>	Could not delete existing secret data
<a href="#">AccessDeniedByUser</a>	<a href="#">User</a> denied access to keychain
<a href="#">AccessDenied</a>	Access denied for some reason
<a href="#">NoBackendAvailable</a>	No platform-specific keychain service available
<a href="#">NotImplemented</a>	Not implemented on platform
<a href="#">OtherError</a>	Something else went wrong, the error description specifies what

### 5.26.3 Member Function Documentation

#### 5.26.3.1 deletePasswordJobFinished

```
void quantier::IKeychainService::deletePasswordJobFinished (
    QUuid requestId,
    ErrorCode errorCode,
    ErrorString errorDescription ) [signal]
```

deletePasswordJobFinished signal should be emitted in response to the call of startDeletePasswordJob method

##### Parameters

<i>requestId</i>	Request id returned from startDeletePasswordJob method
<i>errorCode</i>	Error code determining whether the operation was successful or some error has occurred
<i>errorDescription</i>	Textual description of error in case of unsuccessful execution

#### 5.26.3.2 readPasswordJobFinished

```
void quantier::IKeychainService::readPasswordJobFinished (
    QUuid requestId,
    ErrorCode errorCode,
    ErrorString errorDescription,
    QString password ) [signal]
```

readPasswordJobFinished signal should be emitted in response to the call of startReadPasswordJob method

##### Parameters

<i>requestId</i>	Request id returned from startReadPasswordJob method
<i>errorCode</i>	Error code determining whether the operation was successful or some error has occurred
<i>errorDescription</i>	Textual description of error in case of unsuccessful execution
<i>password</i>	Password read from the keychain

#### 5.26.3.3 startDeletePasswordJob()

```
virtual QUuid quantier::IKeychainService::startDeletePasswordJob (
    const QString & service,
    const QString & key ) [pure virtual]
```

startDeletePasswordJob slot should start the potentially asynchronous process of deleting the password from the keychain. When ready, this slot is expected to emit deletePasswordJobFinished signal.

##### Parameters

<i>service</i>	Name of service within the keychain
<i>key</i>	Key under which the password is stored

**Returns**

Unique identifier assigned to this delete password request

**5.26.3.4 startReadPasswordJob()**

```
virtual QUuid quentier::IKeychainService::startReadPasswordJob (
    const QString & service,
    const QString & key ) [pure virtual]
```

startReadPasswordJob slot should start the potentially asynchronous process of reading the password from the keychain. When ready, this slot is expected to emit readPasswordJobFinished signal.

**Parameters**

<i>service</i>	Name of service within the keychain
<i>key</i>	Key under which the password is stored

**Returns**

Unique identifier assigned to this read password request

**5.26.3.5 startWritePasswordJob()**

```
virtual QUuid quentier::IKeychainService::startWritePasswordJob (
    const QString & service,
    const QString & key,
    const QString & password ) [pure virtual]
```

startWritePasswordJob slot should start the potentially asynchronous process of storing the password in the keychain. When ready, this slot is expected to emit writePasswordJobFinished signal.

**Parameters**

<i>service</i>	Name of service within the keychain
<i>key</i>	Key to store the password under
<i>password</i>	Password to store in the keychain

**Returns**

Unique identifier assigned to this write password request

**5.26.3.6 writePasswordJobFinished**

```
void quentier::IKeychainService::writePasswordJobFinished (
    QUuid requestId,
    ErrorCode errorCode,
    ErrorString errorDescription ) [signal]
```

writePasswordJobFinished signal should be emitted in response to the call of startWritePasswordJob method

## Parameters

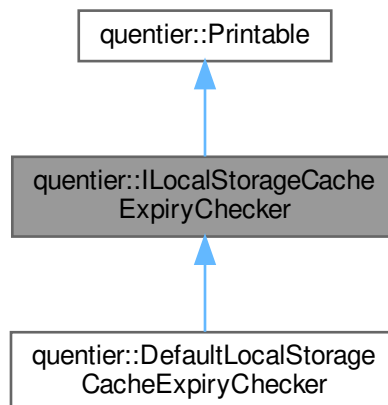
<i>requestId</i>	Request id returned from startWritePasswordJob method
<i>errorCode</i>	Error code determining whether the operation was successful or some error has occurred
<i>errorDescription</i>	Textual description of error in case of unsuccessful execution

## 5.27 quantier::ILocalStorageCacheExpiryChecker Class Reference

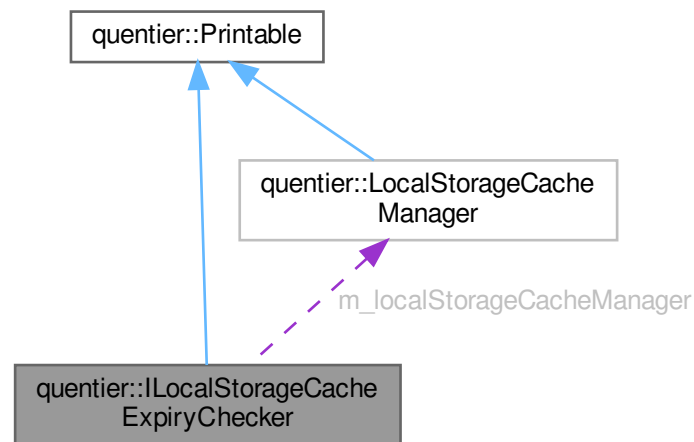
The [ILocalStorageCacheExpiryChecker](#) class represents the interface for cache expiry checker used by [LocalStorageCacheManager](#) to see whether particular caches (of notes, notebooks, tags, linked notebooks and/or saved searches) need to be shrunk.

```
#include <ILocalStorageCacheExpiryChecker.h>
```

Inheritance diagram for quantier::ILocalStorageCacheExpiryChecker:



Collaboration diagram for quantier::ILocalStorageCacheExpiryChecker:



### Public Member Functions

- virtual [ILocalStorageCacheExpiryChecker](#) \* [clone](#) () const =0
- virtual bool [checkNotes](#) () const =0
- virtual bool [checkResources](#) () const =0
- virtual bool [checkNotebooks](#) () const =0
- virtual bool [checkTags](#) () const =0
- virtual bool [checkLinkedNotebooks](#) () const =0
- virtual bool [checkSavedSearches](#) () const =0
- virtual QTextStream & [print](#) (QTextStream &strm) const =0

*Print the internal information about [ILocalStorageCacheExpiryChecker](#) implementation instance to the text stream.*

### Public Member Functions inherited from [quantier::Printable](#)

- virtual const QString [toString](#) () const

### Protected Member Functions

- [ILocalStorageCacheExpiryChecker](#) (const [LocalStorageCacheManager](#) &cacheManager)

### Protected Member Functions inherited from [quantier::Printable](#)

- [Printable](#) (const [Printable](#) &other)
- [Printable](#) & [operator=](#) (const [Printable](#) &other)

### Protected Attributes

- const [LocalStorageCacheManager](#) & [m\\_localStorageCacheManager](#)

### 5.27.1 Detailed Description

The [ILocalStorageCacheExpiryChecker](#) class represents the interface for cache expiry checker used by [LocalStorageCacheManager](#) to see whether particular caches (of notes, notebooks, tags, linked notebooks and/or saved searches) need to be shrunk.

### 5.27.2 Member Function Documentation

#### 5.27.2.1 `checkLinkedNotebooks()`

```
virtual bool quantier::ILocalStorageCacheExpiryChecker::checkLinkedNotebooks ( ) const [pure virtual]
```

##### Returns

False if the cache of linked notebooks needs to be shrunk (due to its size or whatever other reason), true otherwise

Implemented in [quantier::DefaultLocalStorageCacheExpiryChecker](#).

#### 5.27.2.2 `checkNotebooks()`

```
virtual bool quantier::ILocalStorageCacheExpiryChecker::checkNotebooks ( ) const [pure virtual]
```

##### Returns

False if the cache of notebooks needs to be shrunk (due to its size or whatever other reason), true otherwise

Implemented in [quantier::DefaultLocalStorageCacheExpiryChecker](#).

#### 5.27.2.3 `checkNotes()`

```
virtual bool quantier::ILocalStorageCacheExpiryChecker::checkNotes ( ) const [pure virtual]
```

##### Returns

False if the cache of notes needs to be shrunk (due to its size or whatever other reason), true otherwise

Implemented in [quantier::DefaultLocalStorageCacheExpiryChecker](#).

#### 5.27.2.4 `checkResources()`

```
virtual bool quantier::ILocalStorageCacheExpiryChecker::checkResources ( ) const [pure virtual]
```

##### Returns

False if the cache of resources needs to be shrunk (due to its size or whatever other reason), true otherwise

Implemented in [quantier::DefaultLocalStorageCacheExpiryChecker](#).

#### 5.27.2.5 `checkSavedSearches()`

```
virtual bool quentier::ILocalStorageCacheExpiryChecker::checkSavedSearches ( ) const [pure virtual]
```

##### Returns

False if the cache of saved searches needs to be shrunk (due to its size or whatever other reason), true otherwise

Implemented in [quentier::DefaultLocalStorageCacheExpiryChecker](#).

#### 5.27.2.6 `checkTags()`

```
virtual bool quentier::ILocalStorageCacheExpiryChecker::checkTags ( ) const [pure virtual]
```

##### Returns

False if the cache of tags needs to be shrunk (due to its size or whatever other reason), true otherwise

Implemented in [quentier::DefaultLocalStorageCacheExpiryChecker](#).

#### 5.27.2.7 `clone()`

```
virtual ILocalStorageCacheExpiryChecker * quentier::ILocalStorageCacheExpiryChecker::clone ( ) const [pure virtual]
```

##### Returns

A pointer to the newly allocated copy of a particular [ILocalStorageCacheExpiryChecker](#) implementation

Implemented in [quentier::DefaultLocalStorageCacheExpiryChecker](#).

#### 5.27.2.8 `print()`

```
virtual QTextStream & quentier::ILocalStorageCacheExpiryChecker::print ( QTextStream & strm ) const [pure virtual]
```

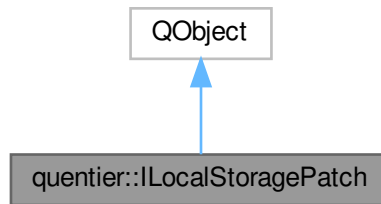
Print the internal information about [ILocalStorageCacheExpiryChecker](#) implementation instance to the text stream.

Implements [quentier::Printable](#).

Implemented in [quentier::DefaultLocalStorageCacheExpiryChecker](#).



Collaboration diagram for quantier::ILocalStoragePatch:



### Signals

- void [progress](#) (double progress)
- void [backupProgress](#) (double progress)
- void [restoreBackupProgress](#) (double progress)

### Public Member Functions

- virtual int [fromVersion](#) () const =0
- virtual int [toVersion](#) () const =0
- virtual QString [patchShortDescription](#) () const =0
- virtual QString [patchLongDescription](#) () const =0
- virtual bool [backupLocalStorage](#) (ErrorString &errorDescription)=0
- virtual bool [restoreLocalStorageFromBackup](#) (ErrorString &errorDescription)=0
- virtual bool [removeLocalStorageBackup](#) (ErrorString &errorDescription)=0
- virtual bool [apply](#) (ErrorString &errorDescription)=0

### Protected Member Functions

- **ILocalStoragePatch** (QObject \*parent=nullptr)

### Friends

- class **LocalStorageDatabaseUpgrader**

## 5.29.1 Detailed Description

The [ILocalStoragePatch](#) class represents the interface for patches of local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquantier can be used to work with it.

## 5.29.2 Member Function Documentation

### 5.29.2.1 [apply\(\)](#)

```
virtual bool quantier::ILocalStoragePatch::apply (
    ErrorString & errorDescription ) [pure virtual]
```

Apply the patch to local storage

**Parameters**

<i>errorDescription</i>	The textual description of the error in case of patch application failure
-------------------------	---

**Returns**

True in case of successful patch application, false otherwise

**5.29.2.2 backupLocalStorage()**

```
virtual bool quantier::ILocalStoragePatch::backupLocalStorage (
    ErrorString & errorDescription ) [pure virtual]
```

Backup either the entire local storage or its parts affected by the particular patch, should be called before applying the patch (but can be skipped if not desired).

**Parameters**

<i>errorDescription</i>	The textual description of the error in case of backup preparation failure
-------------------------	--

**Returns**

True if the local storage was backed up for the patch successfully, false otherwise

**5.29.2.3 backupProgress**

```
void quantier::ILocalStoragePatch::backupProgress (
    double progress ) [signal]
```

Local storage backup preparation progress

**Parameters**

<i>progress</i>	Backup preparation progress value, from 0 to 1
-----------------	--

**5.29.2.4 fromVersion()**

```
virtual int quantier::ILocalStoragePatch::fromVersion ( ) const [pure virtual]
```

**Returns**

Version of local storage to which the patch needs to be applied

**5.29.2.5 patchLongDescription()**

```
virtual QString quentier::ILocalStoragePatch::patchLongDescription ( ) const [pure virtual]
```

**Returns**

Long i.e. detailed description of the patch

**5.29.2.6 patchShortDescription()**

```
virtual QString quentier::ILocalStoragePatch::patchShortDescription ( ) const [pure virtual]
```

**Returns**

Short description of the patch

**5.29.2.7 progress**

```
void quentier::ILocalStoragePatch::progress (
    double progress ) [signal]
```

Patch application progress signal

**Parameters**

<i>progress</i>	Patch application progress value, from 0 to 1
-----------------	---

**5.29.2.8 removeLocalStorageBackup()**

```
virtual bool quentier::ILocalStoragePatch::removeLocalStorageBackup (
    ErrorString & errorDescription ) [pure virtual]
```

Remove the previously made backup of local storage, presumably after successful application of the patch so the backup is no longer needed. It won't work if no backup was made before applying a patch, obviously.

**Parameters**

<i>errorDescription</i>	The textual description of the error in case of failure to remove the local storage backup
-------------------------	--

**Returns**

True if local storage backup was successfully removed, false otherwise

**5.29.2.9 restoreBackupProgress**

```
void quentier::ILocalStoragePatch::restoreBackupProgress (
    double progress ) [signal]
```

Local storage restoration from backup progress

#### Parameters

<i>progress</i>	Local storage restoration from backup progress value, from 0 to 1
-----------------	---

#### 5.29.2.10 restoreLocalStorageFromBackup()

```
virtual bool quantier::ILocalStoragePatch::restoreLocalStorageFromBackup (
    ErrorString & errorDescription ) [pure virtual]
```

Restore local storage from previously made backup, presumably after the failed attempt to apply a patch. Won't work if no backup was made before applying a patch, obviously.

#### Parameters

<i>errorDescription</i>	The textual description of the error in case of failure to restore the local storage from backup
-------------------------	--

#### Returns

True if local storage was successfully restored from backup, false otherwise

#### 5.29.2.11 toVersion()

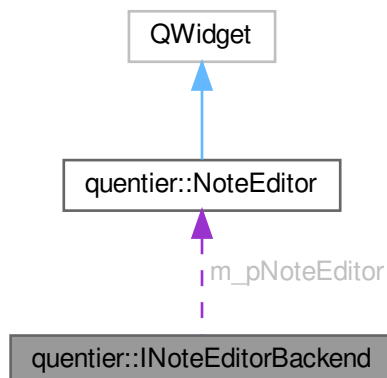
```
virtual int quantier::ILocalStoragePatch::toVersion ( ) const [pure virtual]
```

#### Returns

Version of local storage to which the patch would upgrade the local storage

## 5.30 quantier::INoteEditorBackend Class Reference

Collaboration diagram for quantier::INoteEditorBackend:



## Public Types

- enum class **Rotation** { **Clockwise** = 0 , **Counterclockwise** }

## Public Member Functions

- virtual void **initialize** ([LocalStorageManagerAsync](#) &localStorageManager, [SpellChecker](#) &spellChecker, const [Account](#) &account, QThread \*pBackgroundJobsThread)=0
- virtual QObject \* **object** ()=0
- virtual QWidget \* **widget** ()=0
- virtual void **setAccount** (const [Account](#) &account)=0
- virtual void **setUndoStack** (QUndoStack \*pUndoStack)=0
- virtual void **setInitialPageHtml** (const QString &html)=0
- virtual void **setNoteNotFoundPageHtml** (const QString &html)=0
- virtual void **setNoteDeletedPageHtml** (const QString &html)=0
- virtual void **setNoteLoadingPageHtml** (const QString &html)=0
- virtual bool **isNoteLoaded** () const =0
- virtual qint64 **idleTime** () const =0
- virtual void **convertToNote** ()=0
- virtual void **saveNoteToLocalStorage** ()=0
- virtual void **setNoteTitle** (const QString &noteTitle)=0
- virtual void **setTagIds** (const QStringList &tagLocalUids, const QStringList &tagGuids)=0
- virtual void **undo** ()=0
- virtual void **redo** ()=0
- virtual void **cut** ()=0
- virtual void **copy** ()=0
- virtual void **paste** ()=0
- virtual void **pasteUnformatted** ()=0
- virtual void **selectAll** ()=0
- virtual void **formatSelectionAsSourceCode** ()=0
- virtual void **fontMenu** ()=0
- virtual void **textBold** ()=0
- virtual void **textItalic** ()=0
- virtual void **textUnderline** ()=0
- virtual void **textStrikethrough** ()=0
- virtual void **textHighlight** ()=0
- virtual void **alignLeft** ()=0
- virtual void **alignCenter** ()=0
- virtual void **alignRight** ()=0
- virtual void **alignFull** ()=0
- virtual QString **selectedText** () const =0
- virtual bool **hasSelection** () const =0
- virtual void **findNext** (const QString &text, const bool matchCase) const =0
- virtual void **findPrevious** (const QString &text, const bool matchCase) const =0
- virtual void **replace** (const QString &textToReplace, const QString &replacementText, const bool match↵Case)=0
- virtual void **replaceAll** (const QString &textToReplace, const QString &replacementText, const bool match↵Case)=0
- virtual void **insertToDoCheckbox** ()=0
- virtual void **insertInAppNoteLink** (const QString &userId, const QString &shardId, const QString &noteGuid, const QString &linkText)=0
- virtual void **setSpellcheck** (const bool enabled)=0
- virtual bool **spellCheckEnabled** () const =0
- virtual void **setFont** (const QFont &font)=0

- virtual void **setFontHeight** (const int height)=0
- virtual void **setFontColor** (const QColor &color)=0
- virtual void **setBackgroundColor** (const QColor &color)=0
- virtual QPalette **defaultPalette** () const =0
- virtual void **setDefaultPalette** (const QPalette &pal)=0
- virtual const QFont \* **defaultFont** () const =0
- virtual void **setDefaultFont** (const QFont &font)=0
- virtual void **insertHorizontalLine** ()=0
- virtual void **increaseFontSize** ()=0
- virtual void **decreaseFontSize** ()=0
- virtual void **increaseIndentation** ()=0
- virtual void **decreaseIndentation** ()=0
- virtual void **insertBulletedList** ()=0
- virtual void **insertNumberedList** ()=0
- virtual void **insertTableDialog** ()=0
- virtual void **insertFixedWidthTable** (const int rows, const int columns, const int widthInPixels)=0
- virtual void **insertRelativeWidthTable** (const int rows, const int columns, const double relativeWidth)=0
- virtual void **insertTableRow** ()=0
- virtual void **insertTableColumn** ()=0
- virtual void **removeTableRow** ()=0
- virtual void **removeTableColumn** ()=0
- virtual void **addAttachmentDialog** ()=0
- virtual void **saveAttachmentDialog** (const QByteArray &resourceHash)=0
- virtual void **saveAttachmentUnderCursor** ()=0
- virtual void **openAttachment** (const QByteArray &resourceHash)=0
- virtual void **openAttachmentUnderCursor** ()=0
- virtual void **copyAttachment** (const QByteArray &resourceHash)=0
- virtual void **copyAttachmentUnderCursor** ()=0
- virtual void **removeAttachment** (const QByteArray &resourceHash)=0
- virtual void **removeAttachmentUnderCursor** ()=0
- virtual void **renameAttachment** (const QByteArray &resourceHash)=0
- virtual void **renameAttachmentUnderCursor** ()=0
- virtual void **rotateImageAttachment** (const QByteArray &resourceHash, const Rotation rotationDirection)=0
- virtual void **rotateImageAttachmentUnderCursor** (const Rotation rotationDirection)=0
- virtual void **encryptSelectedText** ()=0
- virtual void **decryptEncryptedTextUnderCursor** ()=0
- virtual void **decryptEncryptedText** (QString encryptedText, QString cipher, QString keyLength, QString hint, QString enCryptIndex)=0
- virtual void **hideDecryptedTextUnderCursor** ()=0
- virtual void **hideDecryptedText** (QString encryptedText, QString decryptedText, QString cipher, QString keyLength, QString hint, QString enDecryptedIndex)=0
- virtual void **editHyperlinkDialog** ()=0
- virtual void **copyHyperlink** ()=0
- virtual void **removeHyperlink** ()=0
- virtual void **onNoteLoadCancelled** ()=0
- virtual bool **print** (QPrinter &printer, [ErrorString](#) &errorDescription)=0
- virtual bool **exportToPdf** (const QString &absoluteFilePath, [ErrorString](#) &errorDescription)=0
- virtual bool **exportToEnex** (const QStringList &tagNames, QString &enex, [ErrorString](#) &errorDescription)=0
- virtual QString **currentNoteLocalUid** () const =0
- virtual void **setCurrentNoteLocalUid** (const QString &noteLocalUid)=0
- virtual void **clear** ()=0
- virtual bool **isModified** () const =0
- virtual bool **isEditorPageModified** () const =0
- virtual void **setFocusToEditor** ()=0

**Protected Member Functions**

- **INoteEditorBackend** ([NoteEditor](#) \*parent)

**Protected Attributes**

- [NoteEditor](#) \* **m\_pNoteEditor**

**Friends**

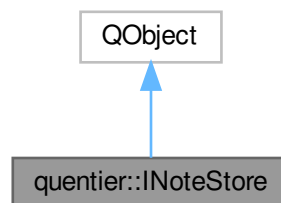
- QUENTIER\_EXPORT QTextStream & **operator**<< (QTextStream &strm, const Rotation rotation)
- QUENTIER\_EXPORT QDebug & **operator**<< (QDebug &dbg, const Rotation rotation)

## 5.31 quentier::INoteStore Class Reference

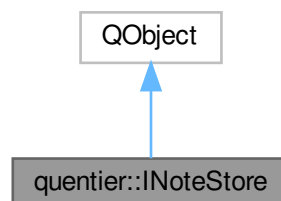
[INoteStore](#) is the interface which provides methods required for the implementation of NoteStore part of Evernote EDAM sync protocol.

```
#include <INoteStore.h>
```

Inheritance diagram for quentier::INoteStore:



Collaboration diagram for quentier::INoteStore:



## Signals

- void **getNoteAsyncFinished** (qint32 errorCode, qevercloud::Note note, qint32 rateLimitSeconds, [ErrorString](#) errorDescription)
- void **getResourceAsyncFinished** (qint32 errorCode, qevercloud::Resource resource, qint32 rateLimitSeconds, [ErrorString](#) errorDescription)

## Public Member Functions

- virtual [INoteStore](#) \* **create** () const =0
- virtual QString **noteStoreUrl** () const =0
- virtual void **setNoteStoreUrl** (QString [noteStoreUrl](#))=0
- virtual void **setAuthData** (QString authenticationToken, QList< QNetworkCookie > cookies)=0
- virtual void **stop** ()=0
- virtual qint32 **createNotebook** ([Notebook](#) &notebook, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds, QString linkedNotebookAuthToken={})=0
- virtual qint32 **updateNotebook** ([Notebook](#) &notebook, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds, QString linkedNotebookAuthToken={})=0
- virtual qint32 **createNote** ([Note](#) &note, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds, QString linkedNotebookAuthToken={})=0
- virtual qint32 **updateNote** ([Note](#) &note, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds, QString linkedNotebookAuthToken={})=0
- virtual qint32 **createTag** ([Tag](#) &tag, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds, QString linkedNotebookAuthToken={})=0
- virtual qint32 **updateTag** ([Tag](#) &tag, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds, QString linkedNotebookAuthToken={})=0
- virtual qint32 **createSavedSearch** ([SavedSearch](#) &savedSearch, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual qint32 **updateSavedSearch** ([SavedSearch](#) &savedSearch, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual qint32 **getSyncState** (qevercloud::SyncState &syncState, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual qint32 **getSyncChunk** (const qint32 afterUSN, const qint32 maxEntries, const qevercloud::SyncChunkFilter &filter, qevercloud::SyncChunk &syncChunk, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual qint32 **getLinkedNotebookSyncState** (const qevercloud::LinkedNotebook &linkedNotebook, const QString &authToken, qevercloud::SyncState &syncState, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual qint32 **getLinkedNotebookSyncChunk** (const qevercloud::LinkedNotebook &linkedNotebook, const qint32 afterUSN, const qint32 maxEntries, const QString &linkedNotebookAuthToken, const bool fullSyncOnly, qevercloud::SyncChunk &syncChunk, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual qint32 **getNote** (const bool withContent, const bool withResourcesData, const bool withResourcesRecognition, const bool withResourceAlternateData, [Note](#) &note, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual bool **getNoteAsync** (const bool withContent, const bool withResourceData, const bool withResourcesRecognition, const bool withResourceAlternateData, const bool withSharedNotes, const bool withNoteAppDataValues, const bool withResourceAppDataValues, const bool withNoteLimits, const QString &noteGuid, const QString &authToken, [ErrorString](#) &errorDescription)=0
- virtual qint32 **getResource** (const bool withDataBody, const bool withRecognitionDataBody, const bool withAlternateDataBody, const bool withAttributes, const QString &authToken, [Resource](#) &resource, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual bool **getResourceAsync** (const bool withDataBody, const bool withRecognitionDataBody, const bool withAlternateDataBody, const bool withAttributes, const QString &resourceGuid, const QString &authToken, [ErrorString](#) &errorDescription)=0
- virtual qint32 **authenticateToSharedNotebook** (const QString &shareKey, qevercloud::AuthenticationResult &authResult, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0

## Protected Member Functions

- **INoteStore** (QObject \*parent=nullptr)

### 5.31.1 Detailed Description

**INoteStore** is the interface which provides methods required for the implementation of NoteStore part of Evernote EDAM sync protocol.

### 5.31.2 Member Function Documentation

#### 5.31.2.1 authenticateToSharedNotebook()

```
virtual qint32 quentier::INoteStore::authenticateToSharedNotebook (
    const QString & shareKey,
    qevercloud::AuthenticationResult & authResult,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

Authenticate to shared notebook

##### Parameters

<i>shareKey</i>	The shared notebook global identifier
<i>authResult</i>	Output parameter, the result of authentication
<i>errorDescription</i>	The textual description of the error if authentication to shared notebook could not be performed
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED

##### Returns

Error code, 0 in case of successful authentication to shared notebook, other values corresponding to qevercloud::EDAMErrorCode enumeration instead

#### 5.31.2.2 createNote()

```
virtual qint32 quentier::INoteStore::createNote (
    Note & note,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds,
    QString linkedNotebookAuthToken = {} ) [pure virtual]
```

Create note

##### Parameters

<i>note</i>	<b>Note</b> to be created
-------------	---------------------------

## Parameters

<i>errorDescription</i>	The textual description of the error in case note could not be created
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED
<i>linkedNotebookAuthToken</i>	If a note is created within another user's account, the corresponding auth token should be set, otherwise the note would be created in user's own account

## Returns

Error code, 0 in case of successful note creation, other values corresponding to qevercloud::EDAMErrorCode enumeration instead

## 5.31.2.3 createNotebook()

```
virtual qint32 quentier::INoteStore::createNotebook (
    Notebook & notebook,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds,
    QString linkedNotebookAuthToken = {} ) [pure virtual]
```

## Create notebook

## Parameters

<i>notebook</i>	<a href="#">Notebook</a> to be created, must have name set and either "active" or "default notebook" fields may be set
<i>errorDescription</i>	The textual description of the error in case notebook could not be created
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED
<i>linkedNotebookAuthToken</i>	If a notebook is created within another user's account, the corresponding auth token should be set, otherwise the notebook would be created in user's own account

## Returns

Error code, 0 in case of successful notebook creation, other values corresponding to qevercloud::EDAMErrorCode enumeration instead

## 5.31.2.4 createSavedSearch()

```
virtual qint32 quentier::INoteStore::createSavedSearch (
    SavedSearch & savedSearch,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

## Create saved search

## Parameters

<i>savedSearch</i>	Saved search to be created, must have name and query set, can also have search scope set
<i>errorDescription</i>	The textual description of the error in case saved search could not be created
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED

## Returns

Error code, 0 in case of successful saved search creation, other values corresponding to qevercloud::EDAMErrorCode enumeration instead

## 5.31.2.5 createTag()

```
virtual qint32 quentier::INoteStore::createTag (
    Tag & tag,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds,
    QString linkedNotebookAuthToken = {} ) [pure virtual]
```

## Create tag

## Parameters

<i>note</i>	<a href="#">Tag</a> to be created, must have name set, can also have parent guid set
<i>errorDescription</i>	The textual description of the error in case tag could not be created
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED
<i>linkedNotebookAuthToken</i>	If a tag is created within another user's account, the corresponding auth token should be set, otherwise the tag would be created in user's own account

## Returns

Error code, 0 in case of successful tag creation, other values corresponding to qevercloud::EDAMErrorCode enumeration instead

## 5.31.2.6 getLinkedNotebookSyncChunk()

```
virtual qint32 quentier::INoteStore::getLinkedNotebookSyncChunk (
    const qevercloud::LinkedNotebook & linkedNotebook,
    const qint32 afterUSN,
    const qint32 maxEntries,
    const QString & linkedNotebookAuthToken,
    const bool fullSyncOnly,
    qevercloud::SyncChunk & syncChunk,
```

```
ErrorString & errorDescription,  
qint32 & rateLimitSeconds ) [pure virtual]
```

Get linked notebook sync chunk

## Parameters

<i>linkedNotebook</i>	The linked notebook for which the sync chunk is being retrieved, must contain identifying information and permissions to access the notebook in question
<i>afterUSN</i>	The USN after which the sync chunks are being requested
<i>maxEntries</i>	Max number of items within the sync chunk to be returned
<i>linkedNotebookAuthToken</i>	The authentication token to use for the data from the linked notebook
<i>fullSyncOnly</i>	If true then client only wants initial data for a full sync. In this case Evernote service will not return any expunged objects and will not return any resources since these are also provided in their corresponding notes
<i>syncChunk</i>	Output parameter, the sync chunk
<i>errorDescription</i>	The textual description of the error in case linked notebook sync chunk could not be retrieved
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED

## Returns

Error code, 0 in case of successful linked notebook sync chunk retrieval, other values corresponding to qevercloud::EDAMErrorCode enumeration instead

## 5.31.2.7 getLinkedNotebookSyncState()

```
virtual qint32 quantier::INoteStore::getLinkedNotebookSyncState (
    const qevercloud::LinkedNotebook & linkedNotebook,
    const QString & authToken,
    qevercloud::SyncState & syncState,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

Get linked notebook sync state

## Parameters

<i>linkedNotebook</i>	The linked notebook for which the sync state is being retrieved, must contain identifying information and permissions to access the notebook in question
<i>authToken</i>	The authentication token to use for the data from the linked notebook
<i>syncState</i>	Output parameter, the sync state
<i>errorDescription</i>	The textual description of the error in case linked notebook sync state could not be retrieved
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED

## Returns

Error code, 0 in case of successful linked notebook sync state retrieval, other values corresponding to qevercloud::EDAMErrorCode enumeration instead

### 5.31.2.8 `getNote()`

```
virtual qint32 quentier::INoteStore::getNote (
    const bool withContent,
    const bool withResourcesData,
    const bool withResourcesRecognition,
    const bool withResourceAlternateData,
    Note & note,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

Get note synchronously

#### Parameters

<i>withContent</i>	If true, the returned note would include the content
<i>withResourcesData</i>	If true, any resources the note might have would include their full data
<i>withResourcesRecognition</i>	If true, any resources the note might have and which have Evernote supplied recognition would include their full recognition data
<i>withResourceAlternateData</i>	If true, any resources the note might have would include their full alternate data
<i>note</i>	Input and output parameter, the retrieved note, must have guid set
<i>errorDescription</i>	The textual description of the error in case the note could not be retrieved
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches <code>qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED</code>

#### Returns

Error code, 0 in case of successful note retrieval, other values corresponding to `qevercloud::EDAMErrorCode` enumeration instead

### 5.31.2.9 `getNoteAsync()`

```
virtual bool quentier::INoteStore::getNoteAsync (
    const bool withContent,
    const bool withResourceData,
    const bool withResourcesRecognition,
    const bool withResourceAlternateData,
    const bool withSharedNotes,
    const bool withNoteAppDataValues,
    const bool withResourceAppDataValues,
    const bool withNoteLimits,
    const QString & noteGuid,
    const QString & authToken,
    ErrorString & errorDescription ) [pure virtual]
```

Get note asynchronously

If the method returned true, the actual result of the method invocation would be returned via the emission of `getNoteAsyncFinished` signal.

## Parameters

<i>withContent</i>	If true, the returned note would include the content
<i>withResourcesData</i>	If true, any resources the note might have would include their full data
<i>withResourcesRecognition</i>	If true, any resources the note might have and which have Evernote supplied recognition would include their full recognition data
<i>withResourceAlternateData</i>	If true, any resources the note might have would include their full alternate data
<i>withSharedNotes</i>	If true, any shared notes contained within the note would be provided along with the asynchronously fetched result
<i>withNoteAppDataValues</i>	If true, the asynchronously fetched note would contain the app data values
<i>withResourceAppDataValues</i>	If true, the resources of asynchronously fetched note would contain the app data values
<i>withNoteLimits</i>	If true, the asynchronously fetched note would contain note limits
<i>noteGuid</i>	The guid of the note to be retrieved
<i>authToken</i>	Authentication token to use for note retrieval
<i>errorDescription</i>	The textual description of the error if the launch of async note retrieval has failed

## Returns

True if the launch of async note retrieval was successful, false otherwise

## 5.31.2.10 getResource()

```
virtual qint32 quantier::INoteStore::getResource (
    const bool withDataBody,
    const bool withRecognitionDataBody,
    const bool withAlternateDataBody,
    const bool withAttributes,
    const QString & authToken,
    Resource & resource,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

Get resource synchronously

## Parameters

<i>withDataBody</i>	If true, the returned resource would include its data body
<i>withRecognitionDataBody</i>	If true, the returned resource would include its recognition data body
<i>withAlternateDataBody</i>	If true, the returned resource would include its alternate data body
<i>withAttributes</i>	If true, the returned resource would include its attributes
<i>authToken</i>	Authentication token to use for resource retrieval
<i>resource</i>	Input and output parameter, the retrieved resource, must have guid set
<i>errorDescription</i>	The textual description of the error in case the resource could not be retrieved
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches <code>qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED</code>

**Returns**

Error code, 0 in case of successful resource retrieval, other values corresponding to `qevercloud::EDAMError`↔  
Code enumeration instead

**5.31.2.11 getResourceAsync()**

```
virtual bool quentier::INoteStore::getResourceAsync (
    const bool withDataBody,
    const bool withRecognitionDataBody,
    const bool withAlternateDataBody,
    const bool withAttributes,
    const QString & resourceGuid,
    const QString & authToken,
    ErrorString & errorDescription ) [pure virtual]
```

Get resource asynchronously

If the method returned true, the actual result of the method invocation would be returned via the emission of `get`↔  
`ResourceAsyncFinished` signal.

**Parameters**

<i>withDataBody</i>	If true, the returned resource would include its data body
<i>withRecognitionDataBody</i>	If true, the returned resource would include its recognition data body
<i>withAlternateDataBody</i>	If true, the returned resource would include its alternate data body
<i>withAttributes</i>	If true, the returned resource would include its attributes
<i>resourceGuid</i>	The guid of the resource to be retrieved
<i>authToken</i>	Authentication token to use for resource retrieval
<i>errorDescription</i>	The textual description of the error if the launch of async resource retrieval has failed

**Returns**

True if the launch of async resource retrieval was successful, false otherwise

**5.31.2.12 getSyncChunk()**

```
virtual qint32 quentier::INoteStore::getSyncChunk (
    const qint32 afterUSN,
    const qint32 maxEntries,
    const qevercloud::SyncChunkFilter & filter,
    qevercloud::SyncChunk & syncChunk,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

Get sync chunk

**Parameters**

<i>afterUSN</i>	The USN after which the sync chunks are being requested
<i>maxEntries</i>	Max number of items within the sync chunk to be returned

## Parameters

<i>filter</i>	Filter for items to be returned within the sync chunks
<i>syncChunk</i>	Output parameter, the sync chunk
<i>errorDescription</i>	The textual description of the error in case sync chunk could not be retrieved
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED

## Returns

Error code, 0 in case of successful sync chunk retrieval, other values corresponding to qevercloud::↵ EDAMErrorCode enumeration instead

## 5.31.2.13 getSyncState()

```
virtual qint32 quantier::INoteStore::getSyncState (
    qevercloud::SyncState & syncState,
    QString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

Get sync state

## Parameters

<i>syncState</i>	Output parameter, the sync state
<i>errorDescription</i>	The textual description of the error in case sync state could not be retrieved
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED

## Returns

Error code, 0 in case of successful sync state retrieval, other values corresponding to qevercloud::↵ EDAMErrorCode enumeration instead

## 5.31.2.14 noteStoreUrl()

```
virtual QString quantier::INoteStore::noteStoreUrl ( ) const [pure virtual]
```

Provide note store URL used by this [INoteStore](#) instance

## 5.31.2.15 setAuthData()

```
virtual void quantier::INoteStore::setAuthData (
    QString authenticationToken,
    QList< QNetworkCookie > cookies ) [pure virtual]
```

Set authentication data to be used by this [INoteStore](#) instance

### 5.31.2.16 setNoteStoreUrl()

```
virtual void quentier::INoteStore::setNoteStoreUrl (
    QString noteStoreUrl ) [pure virtual]
```

Set note store URL to be used by this [INoteStore](#) instance

### 5.31.2.17 stop()

```
virtual void quentier::INoteStore::stop ( ) [pure virtual]
```

Stop asynchronous queries for notes or resources which might be running at the moment

### 5.31.2.18 updateNote()

```
virtual qint32 quentier::INoteStore::updateNote (
    Note & note,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds,
    QString linkedNotebookAuthToken = {} ) [pure virtual]
```

Update note

#### Parameters

<i>note</i>	<a href="#">Note</a> to be updated, must have guid set
<i>errorDescription</i>	The textual description of the error in case note could not be updated
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches <code>qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED</code>
<i>linkedNotebookAuthToken</i>	If a note is updated within another user's account, the corresponding auth token should be set, otherwise the note would be updated within user's own account

#### Returns

Error code, 0 in case of successful note update, other values corresponding to `qevercloud::EDAMErrorCode` enumeration instead

### 5.31.2.19 updateNotebook()

```
virtual qint32 quentier::INoteStore::updateNotebook (
    Notebook & notebook,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds,
    QString linkedNotebookAuthToken = {} ) [pure virtual]
```

Update notebook

## Parameters

<i>notebook</i>	<a href="#">Notebook</a> to be updated, must have guid set
<i>errorDescription</i>	The textual description of the error in case notebook could not be updated
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches <code>qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED</code>
<i>linkedNotebookAuthToken</i>	If a notebook is updated within another user's account, the corresponding auth token should be set, otherwise the notebook would be updated within user's own account

## Returns

Error code, 0 in case of successful notebook update, other values corresponding to `qevercloud::EDAMErrorCode` enumeration instead

## 5.31.2.20 updateSavedSearch()

```
virtual qint32 quantier::INoteStore::updateSavedSearch (
    SavedSearch & savedSearch,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

Update saved search

## Parameters

<i>savedSearch</i>	Saved search to be updated, must have guid set
<i>errorDescription</i>	The textual description of the error in case saved search could not be updated
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches <code>qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED</code>

## Returns

Error code, 0 in case of successful saved search update, other values corresponding to `qevercloud::EDAMErrorCode` enumeration instead

## 5.31.2.21 updateTag()

```
virtual qint32 quantier::INoteStore::updateTag (
    Tag & tag,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds,
    QString linkedNotebookAuthToken = {} ) [pure virtual]
```

Update tag

## Parameters

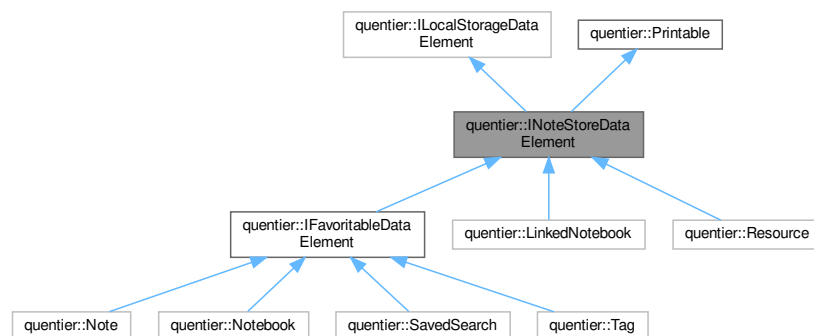
<i>tag</i>	<a href="#">Tag</a> to be updated, must have guid set
<i>errorDescription</i>	The textual description of the error in case tag could not be updated
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches <code>qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED</code>
<i>linkedNotebookAuthToken</i>	If a tag is updated within another user's account, the corresponding auth token should be set, otherwise the tag would be updated within user's own account

## Returns

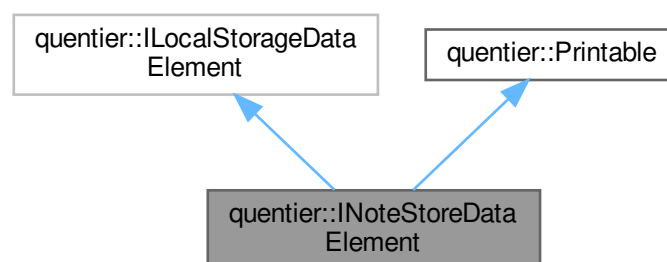
Error code, 0 in case of successful tag update, other values corresponding to `qevercloud::EDAMErrorCode` enumeration instead

## 5.32 quantier::INoteStoreDataElement Class Reference

Inheritance diagram for `quantier::INoteStoreDataElement`:



Collaboration diagram for `quantier::INoteStoreDataElement`:



**Public Member Functions**

- virtual void **clear** ()=0
- virtual bool **hasGuid** () const =0
- virtual const QString & **guid** () const =0
- virtual void **setGuid** (const QString &guid)=0
- virtual bool **hasUpdateSequenceNumber** () const =0
- virtual qint32 **updateSequenceNumber** () const =0
- virtual void **setUpdateSequenceNumber** (const qint32 usn)=0
- virtual bool **checkParameters** (ErrorString &errorDescription) const =0
- virtual bool **isDirty** () const =0
- virtual void **setDirty** (const bool dirty)=0
- virtual bool **isLocal** () const =0
- virtual void **setLocal** (const bool local)=0

**Public Member Functions inherited from [quantier::ILocalStorageDataElement](#)**

- virtual const QString **localUid** () const =0
- virtual void **setLocalUid** (const QString &guid)=0
- virtual void **unsetLocalUid** ()=0

**Public Member Functions inherited from [quantier::Printable](#)**

- virtual QTextStream & **print** (QTextStream &strm) const =0
- virtual const QString **toString** () const

**Additional Inherited Members****Protected Member Functions inherited from [quantier::Printable](#)**

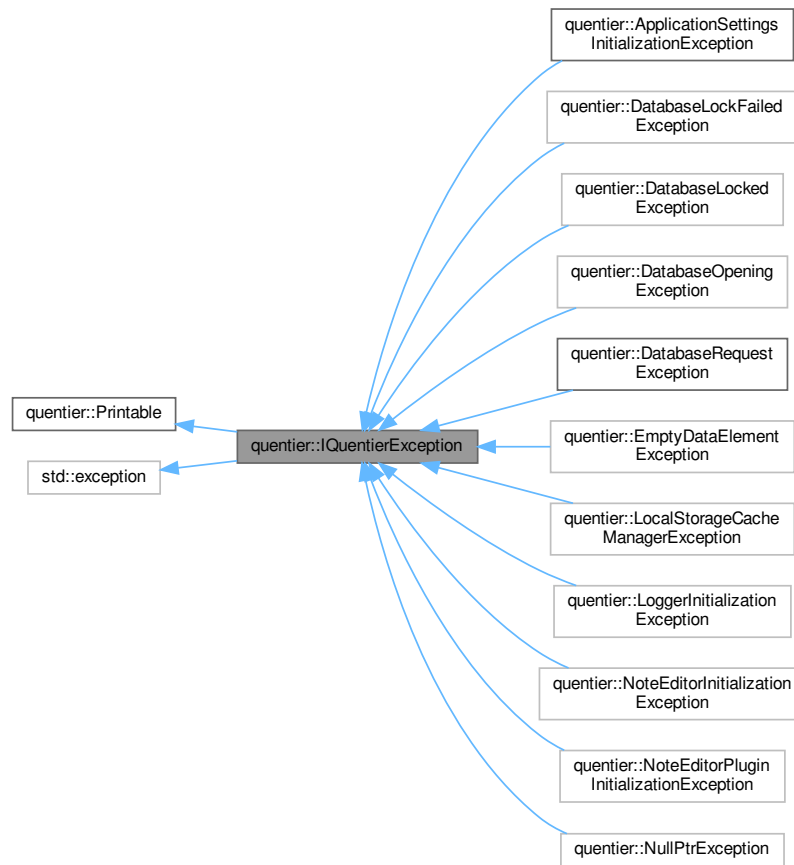
- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

## 5.33 quantier::IQuantierException Class Reference

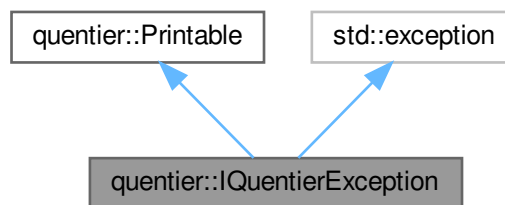
The [IQuantierException](#) class represents the interface for exceptions specific to libquantier and applications based on it.

```
#include <IQuantierException.h>
```

Inheritance diagram for `quentier::IQuentierException`:



Collaboration diagram for `quentier::IQuentierException`:



## Public Member Functions

- `IQuentierException` (const [ErrorString](#) &message)
- `QString localizedErrorMessage () const`

- `QString nonLocalizedErrorMessage () const`
- `virtual const char * what () const noexcept override`
- `virtual QTextStream & print (QTextStream &strm) const override`

### Public Member Functions inherited from `quentier::Printable`

- `virtual const QString toString () const`

### Protected Member Functions

- `IQuentierException (const IQuentierException &other)`
- `IQuentierException & operator= (const IQuentierException &other)`
- `virtual const QString exceptionDisplayName () const =0`

### Protected Member Functions inherited from `quentier::Printable`

- `Printable (const Printable &other)`
- `Printable & operator= (const Printable &other)`

## 5.33.1 Detailed Description

The `IQuentierException` class represents the interface for exceptions specific to libquentier and applications based on it.

In addition to standard exception features inherited from `std::exception`, `IQuentierException` based exceptions can provide both localized and non-localized error messages.

## 5.33.2 Member Function Documentation

### 5.33.2.1 `print()`

```
virtual QTextStream & quentier::IQuentierException::print (  
    QTextStream & strm ) const [override], [virtual]
```

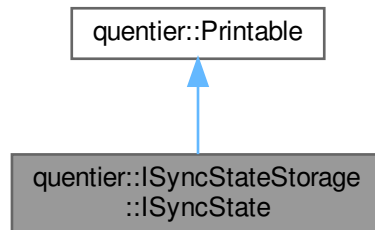
Implements `quentier::Printable`.

## 5.34 quantier::ISyncStateStorage::ISyncState Class Reference

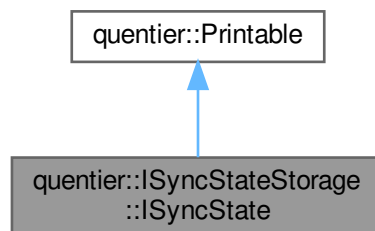
The [ISyncState](#) interface provides accessory methods to determine the sync state for the account.

```
#include <ISyncStateStorage.h>
```

Inheritance diagram for quantier::ISyncStateStorage::ISyncState:



Collaboration diagram for quantier::ISyncStateStorage::ISyncState:



### Public Member Functions

- virtual qint32 **userDataUpdateCount** () const =0
- virtual qevercloud::Timestamp **userDataLastSyncTime** () const =0
- virtual QHash< QString, qint32 > **linkedNotebookUpdateCounts** () const =0
- virtual QHash< QString, qevercloud::Timestamp > **linkedNotebookLastSyncTimes** () const =0
- virtual QTextStream & [print](#) (QTextStream &strm) const override

### Public Member Functions inherited from [quantier::Printable](#)

- virtual const QString **toString** () const

### Additional Inherited Members

### Protected Member Functions inherited from `quentier::Printable`

- **Printable** (const `Printable` &other)
- `Printable` & **operator=** (const `Printable` &other)

#### 5.34.1 Detailed Description

The `ISyncState` interface provides accessory methods to determine the sync state for the account.

#### 5.34.2 Member Function Documentation

##### 5.34.2.1 `print()`

```
virtual QTextStream & quentier::ISyncStateStorage::ISyncState::print (  
    QTextStream & strm ) const [override], [virtual]
```

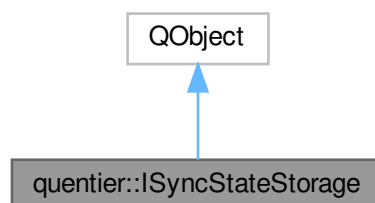
Implements `quentier::Printable`.

## 5.35 `quentier::ISyncStateStorage` Class Reference

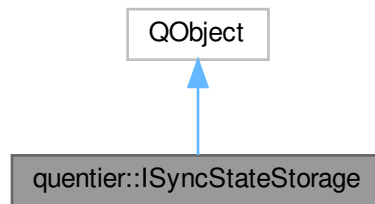
The `ISyncStateStorage` interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states.

```
#include <ISyncStateStorage.h>
```

Inheritance diagram for `quentier::ISyncStateStorage`:



Collaboration diagram for `quentier::ISyncStateStorage`:



## Classes

- class [ISyncState](#)

The [ISyncState](#) interface provides accessory methods to determine the sync state for the account.

## Public Types

- using **ISyncStatePtr** = `std::shared_ptr< ISyncState >`

## Signals

- void [notifySyncStateUpdated](#) ([Account](#) account, `ISyncStatePtr syncState`)

## Public Member Functions

- **ISyncStateStorage** (`QObject *parent=nullptr`)
- virtual `ISyncStatePtr` **getSyncState** (`const Account &account`)=0
- virtual void **setSyncState** (`const Account &account`, `ISyncStatePtr syncState`)=0

### 5.35.1 Detailed Description

The [ISyncStateStorage](#) interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states.

### 5.35.2 Member Function Documentation

#### 5.35.2.1 notifySyncStateUpdated

```
void quentier::ISyncStateStorage::notifySyncStateUpdated (
    Account account,
    ISyncStatePtr syncState ) [signal]
```

Classes implementing [ISyncStateStorage](#) interface are expected to emit `notifySyncStateUpdated` signal each time when sync state for the corresponding account is updated

## 5.36 quantier::IUserStore Class Reference

[IUserStore](#) is the interface which provides methods required for the implementation of UserStore part of Evernote EDAM sync protocol.

```
#include <IUserStore.h>
```

### Public Member Functions

- virtual void [setAuthData](#) (QString authenticationToken, QList< QNetworkCookie > cookies)=0
- virtual bool [checkVersion](#) (const QString &clientName, qint16 edamVersionMajor, qint16 edamVersionMinor, [ErrorString](#) &errorDescription)=0
- virtual qint32 [getUser](#) ([User](#) &user, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0
- virtual qint32 [getAccountLimits](#) (const qevercloud::ServiceLevel serviceLevel, qevercloud::AccountLimits &limits, [ErrorString](#) &errorDescription, qint32 &rateLimitSeconds)=0

### 5.36.1 Detailed Description

[IUserStore](#) is the interface which provides methods required for the implementation of UserStore part of Evernote EDAM sync protocol.

### 5.36.2 Member Function Documentation

#### 5.36.2.1 checkVersion()

```
virtual bool quantier::IUserStore::checkVersion (
    const QString & clientName,
    qint16 edamVersionMajor,
    qint16 edamVersionMinor,
    ErrorString & errorDescription ) [pure virtual]
```

Check the version of EDAM protocol

#### Parameters

<i>clientName</i>	Application name + application version + platform name string
<i>edamVersionMajor</i>	The major version of EDAM protocol the application wants to use to connect to Evernote
<i>edamVersionMinor</i>	The minor version of EDAM protocol the application wants to use to connect to Evernote
<i>errorDescription</i>	The textual description of the error if the supplied protocol version cannot be used to connect to Evernote

#### Returns

True if protocol check was successful i.e. the service can talk to the client using the supplied protocol version, false otherwise

#### 5.36.2.2 getAccountLimits()

```
virtual qint32 quantier::IUserStore::getAccountLimits (
```

```
const qevercloud::ServiceLevel serviceLevel,
qevercloud::AccountLimits & limits,
ErrorString & errorDescription,
qint32 & rateLimitSeconds ) [pure virtual]
```

Retrieve account limits corresponding to certain provided service level

#### Parameters

<i>serviceLevel</i>	The level of Evernote service for which account limits are requested
<i>limits</i>	Output account limits
<i>errorDescription</i>	The textual description of the error if account limits could not be retrieved
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED

#### Returns

Error code, 0 in case of successful retrieval of account limits for the given service level, other values corresponding to qevercloud::EDAMErrorCode::type enumeration instead

### 5.36.2.3 getUser()

```
virtual qint32 qentier::IUserStore::getUser (
    User & user,
    ErrorString & errorDescription,
    qint32 & rateLimitSeconds ) [pure virtual]
```

Retrieve full information about user (account)

#### Parameters

<i>user</i>	Input and output parameter; on input needs to have user id set
<i>errorDescription</i>	The textual description of the error if full user information could not be retrieved
<i>rateLimitSeconds</i>	Output parameter, the number of seconds the client needs to wait before attempting to call this method or any other method calling Evernote API again; only meaningful if returned value matches qevercloud::EDAMErrorCode::RATE_LIMIT_REACHED

#### Returns

Error code, 0 in case of successful retrieval of full user information, other values corresponding to qevercloud::EDAMErrorCode enumeration instead

### 5.36.2.4 setAuthData()

```
virtual void qentier::IUserStore::setAuthData (
    QString authenticationToken,
    QList< QNetworkCookie > cookies ) [pure virtual]
```

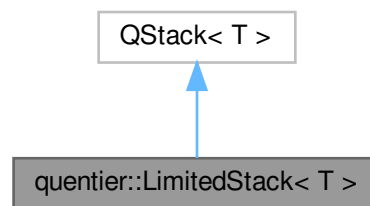
Set authentication data to be used by this [IUserStore](#) instance

## 5.37 `quentier::LimitedStack< T >` Class Template Reference

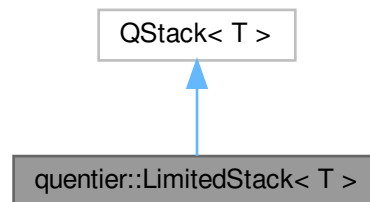
The `LimitedStack` template class implements a stack which may have a limitation for its size; when the size becomes too much according to the limit, the bottom element of the stack gets erased from it. Only limits greater than zero are considered.

```
#include <LimitedStack.h>
```

Inheritance diagram for `quentier::LimitedStack< T >`:



Collaboration diagram for `quentier::LimitedStack< T >`:



### Public Member Functions

- **LimitedStack** (`void(*deleter)(T &)=nullptr`)
- **LimitedStack** (`const LimitedStack< T > &other`)
- **LimitedStack** (`LimitedStack< T > &&other`)
- `LimitedStack< T > & operator=` (`const LimitedStack< T > &other`)
- `LimitedStack< T > & operator=` (`LimitedStack< T > &&other`)
- `void swap` (`const LimitedStack< T > &other`)
- `int limit` () `const`
- `void setLimit` (`const int limit`)
- `void push` (`const T &t`)

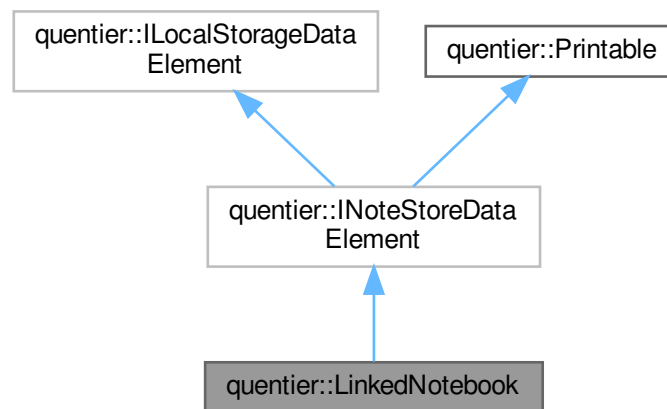
### 5.37.1 Detailed Description

```
template<class T>
class quantier::LimitedStack< T >
```

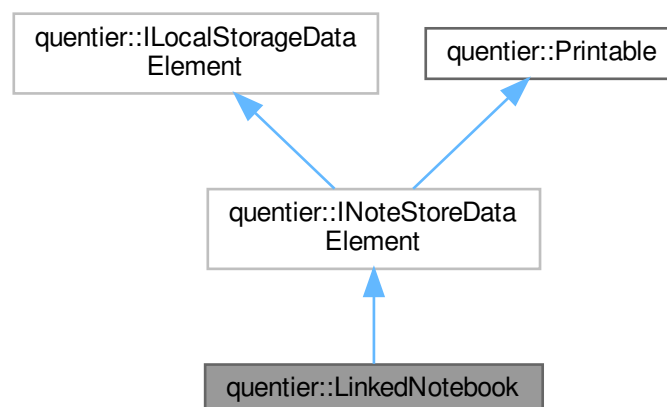
The [LimitedStack](#) template class implements a stack which may have a limitation for its size; when the size becomes too much according to the limit, the bottom element of the stack gets erased from it. Only limits greater than zero are considered.

## 5.38 quantier::LinkedNotebook Class Reference

Inheritance diagram for quantier::LinkedNotebook:



Collaboration diagram for quantier::LinkedNotebook:



## Public Member Functions

- **LinkedNotebook** (const [LinkedNotebook](#) &other)
- **LinkedNotebook** ([LinkedNotebook](#) &&other)
- [LinkedNotebook](#) & **operator=** (const [LinkedNotebook](#) &other)
- [LinkedNotebook](#) & **operator=** ([LinkedNotebook](#) &&other)
- **LinkedNotebook** (const qevercloud::LinkedNotebook &linkedNotebook)
- **LinkedNotebook** (qevercloud::LinkedNotebook &&linkedNotebook)
- const qevercloud::LinkedNotebook & **qevercloudLinkedNotebook** () const
- qevercloud::LinkedNotebook & **qevercloudLinkedNotebook** ()
- bool **operator==** (const [LinkedNotebook](#) &other) const
- bool **operator!=** (const [LinkedNotebook](#) &other) const
- virtual void **clear** () override
- virtual bool **hasGuid** () const override
- virtual const QString & **guid** () const override
- virtual void **setGuid** (const QString &guid) override
- virtual bool **hasUpdateSequenceNumber** () const override
- virtual qint32 **updateSequenceNumber** () const override
- virtual void **setUpdateSequenceNumber** (const qint32 usn) override
- virtual bool **checkParameters** ([ErrorString](#) &errorDescription) const override
- bool **hasShareName** () const
- const QString & **shareName** () const
- void **setShareName** (const QString &shareName)
- bool **hasUsername** () const
- const QString & **username** () const
- void **setUsername** (const QString &username)
- bool **hasShardId** () const
- const QString & **shardId** () const
- void **setShardId** (const QString &shardId)
- bool **hasSharedNotebookGlobalId** () const
- const QString & **sharedNotebookGlobalId** () const
- void **setSharedNotebookGlobalId** (const QString &sharedNotebookGlobalId)
- bool **hasUri** () const
- const QString & **uri** () const
- void **setUri** (const QString &uri)
- bool **hasNoteStoreUrl** () const
- const QString & **noteStoreUrl** () const
- void **setNoteStoreUrl** (const QString &noteStoreUrl)
- bool **hasWebApiUrlPrefix** () const
- const QString & **webApiUrlPrefix** () const
- void **setWebApiUrlPrefix** (const QString &webApiUrlPrefix)
- bool **hasStack** () const
- const QString & **stack** () const
- void **setStack** (const QString &stack)
- bool **hasBusinessId** () const
- qint32 **businessId** () const
- void **setBusinessId** (const qint32 businessId)
- virtual QTextStream & **print** (QTextStream &strm) const override

Public Member Functions inherited from [quantier::INoteStoreDataElement](#)

- virtual bool **isDirty** () const =0
- virtual void **setDirty** (const bool dirty)=0

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

## Additional Inherited Members

## Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

## 5.38.1 Member Function Documentation

### 5.38.1.1 checkParameters()

```
virtual bool quentier::LinkedNotebook::checkParameters (  
    ErrorString & errorDescription ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

### 5.38.1.2 clear()

```
virtual void quentier::LinkedNotebook::clear ( ) [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

### 5.38.1.3 guid()

```
virtual const QString & quentier::LinkedNotebook::guid ( ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

### 5.38.1.4 hasGuid()

```
virtual bool quentier::LinkedNotebook::hasGuid ( ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

### 5.38.1.5 hasUpdateSequenceNumber()

```
virtual bool quentier::LinkedNotebook::hasUpdateSequenceNumber ( ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

#### 5.38.1.6 `print()`

```
virtual QTextStream & quentier::LinkedNotebook::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quentier::Printable](#).

#### 5.38.1.7 `setGuid()`

```
virtual void quentier::LinkedNotebook::setGuid (
    const QString & guid ) [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

#### 5.38.1.8 `setUpdateSequenceNumber()`

```
virtual void quentier::LinkedNotebook::setUpdateSequenceNumber (
    const quint32 usn ) [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

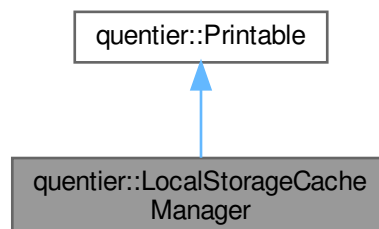
#### 5.38.1.9 `updateSequenceNumber()`

```
virtual quint32 quentier::LinkedNotebook::updateSequenceNumber ( ) const [override], [virtual]
```

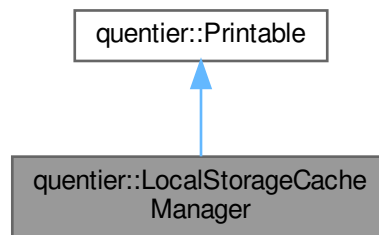
Implements [quentier::INoteStoreDataElement](#).

## 5.39 `quentier::LocalStorageCacheManager` Class Reference

Inheritance diagram for `quentier::LocalStorageCacheManager`:



Collaboration diagram for `quentier::LocalStorageCacheManager`:



## Public Types

- enum **WhichUid** { **LocalUid** , **Guid** }

## Public Member Functions

- void **clear** ()
- bool **empty** () const
- size\_t **numCachedNotes** () const
- void **cacheNote** (const [Note](#) &note)
- void **expungeNote** (const [Note](#) &note)
- const [Note](#) \* **findNote** (const QString &uid, const WhichUid whichUid) const
- void **clearAllNotes** ()
- size\_t **numCachedResources** () const
- void **cacheResource** (const [Resource](#) &resource)
- void **expungeResource** (const [Resource](#) &resource)
- const [Resource](#) \* **findResource** (const QString &id, const WhichUid whichUid) const
- void **clearAllResources** ()
- size\_t **numCachedNotebooks** () const
- void **cacheNotebook** (const [Notebook](#) &notebook)
- void **expungeNotebook** (const [Notebook](#) &notebook)
- const [Notebook](#) \* **findNotebook** (const QString &uid, const WhichUid whichUid) const
- const [Notebook](#) \* **findNotebookByName** (const QString &name) const
- void **clearAllNotebooks** ()
- size\_t **numCachedTags** () const
- void **cacheTag** (const [Tag](#) &tag)
- void **expungeTag** (const [Tag](#) &tag)
- const [Tag](#) \* **findTag** (const QString &uid, const WhichUid whichUid) const
- const [Tag](#) \* **findTagByName** (const QString &name) const
- void **clearAllTags** ()
- size\_t **numCachedLinkedNotebooks** () const
- void **cacheLinkedNotebook** (const [LinkedNotebook](#) &linkedNotebook)
- void **expungeLinkedNotebook** (const [LinkedNotebook](#) &linkedNotebook)
- const [LinkedNotebook](#) \* **findLinkedNotebook** (const QString &guid) const
- void **clearAllLinkedNotebooks** ()
- size\_t **numCachedSavedSearches** () const

- void **cacheSavedSearch** (const [SavedSearch](#) &savedSearch)
- void **expungeSavedSearch** (const [SavedSearch](#) &savedSearch)
- const [SavedSearch](#) \* **findSavedSearch** (const QString &uid, const WhichUid whichUid) const
- const [SavedSearch](#) \* **findSavedSearchByName** (const QString &name) const
- void **clearAllSavedSearches** ()
- void **installCacheExpiryFunction** (const [ILocalStorageCacheExpiryChecker](#) &checker)
- virtual QTextStream & **print** (QTextStream &strm) const override

### Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

### Additional Inherited Members

### Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

## 5.39.1 Member Function Documentation

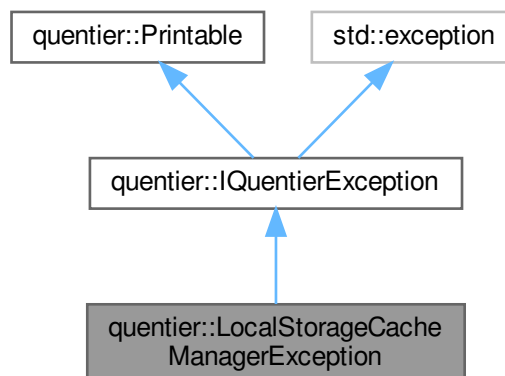
### 5.39.1.1 `print()`

```
virtual QTextStream & quentier::LocalStorageCacheManager::print (
    QTextStream & strm ) const [override], [virtual]
```

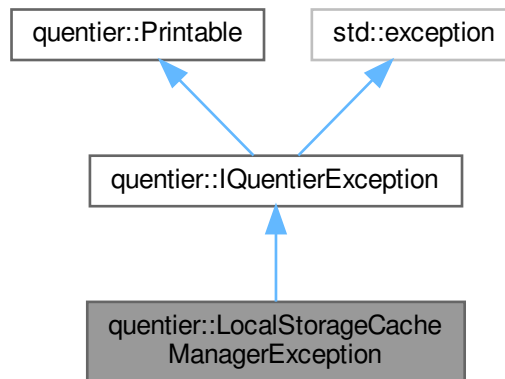
Implements [quentier::Printable](#).

## 5.40 `quentier::LocalStorageCacheManagerException` Class Reference

Inheritance diagram for `quentier::LocalStorageCacheManagerException`:



Collaboration diagram for `quentier::LocalStorageCacheManagerException`:



#### Public Member Functions

- **LocalStorageCacheManagerException** (const [ErrorString](#) &message)

#### Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

#### Public Member Functions inherited from [quentier::Printable](#)

- virtual const **QString toString** () const

#### Protected Member Functions

- virtual const **QString exceptionDisplayName** () const override

#### Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [IQuentierException](#) &other)
- **IQuentierException & operator=** (const [IQuentierException](#) &other)

#### Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable & operator=** (const [Printable](#) &other)

## 5.40.1 Member Function Documentation

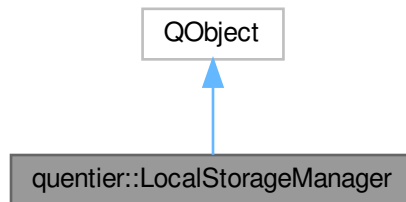
### 5.40.1.1 exceptionDisplayName()

```
virtual const QString quantier::LocalStorageCacheManagerException::exceptionDisplayName ( )  
const [override], [protected], [virtual]
```

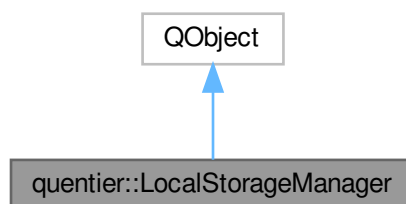
Implements [quantier::IQuantierException](#).

## 5.41 quantier::LocalStorageManager Class Reference

Inheritance diagram for quantier::LocalStorageManager:



Collaboration diagram for quantier::LocalStorageManager:



## Public Types

- enum class [StartupOption](#) { [ClearDatabase](#) = 1 , [OverrideLock](#) = 2 }

*The StartupOption enum is a QFlags enum which allows to specify some options to be applied to the local storage database on startup or on call to switchUser method.*

- enum class [ListObjectsOption](#) {  
**ListAll** = 0 , **ListDirty** = 1 , **ListNonDirty** = 2 , **ListElementsWithoutGuid** = 4 ,  
**ListElementsWithGuid** = 8 , **ListLocal** = 16 , **ListNonLocal** = 32 , **ListFavoritedElements** = 64 ,  
**ListNonFavoritedElements** = 128 }  
*The ListObjectsOption enum is a QFlags enum which allows to specify the desired local storage elements in calls to methods listing them from the database.*
- enum class [OrderDirection](#) { **Ascending** = 0 , **Descending** }  
*The OrderDirection enum specifies the direction of ordering of the results for methods listing the objects from the local storage database.*
- enum class [ListNotebooksOrder](#) {  
**ByUpdateSequenceNumber** = 0 , **ByNotebookName** , **ByCreationTimestamp** , **ByModificationTimestamp** ,  
**NoOrder** }  
*The ListNotebooksOrder allows to specify the results ordering for methods listing notebooks from the local storage database.*
- enum class [ListLinkedNotebooksOrder](#) { **ByUpdateSequenceNumber** = 0 , **ByShareName** , **ByUsername** , **NoOrder** }  
*The ListLinkedNotebooksOrder enum allows to specify the results ordering for methods listing linked notebooks from local storage.*
- enum class [NoteCountOption](#) { **IncludeNonDeletedNotes** = 1 , **IncludeDeletedNotes** = 2 }  
*The NoteCountOption enum is a QFlags enum which allows to specify some options for methods returning note counts from local storage.*
- enum class [UpdateNoteOption](#) { [UpdateResourceMetadata](#) = 1 , [UpdateResourceBinaryData](#) = 2 , [UpdateTags](#) = 4 }  
*The UpdateNoteOption enum is a QFlags enum which allows to specify which note fields should be updated when updateNote method is called.*
- enum class [GetNoteOption](#) { [WithResourceMetadata](#) = 1 , [WithResourceBinaryData](#) = 2 }  
*The GetNoteOption enum is a QFlags enum which allows to specify which note fields should be included when findNote or one of listNote\* methods is called.*
- enum class [ListNotesOrder](#) {  
**ByUpdateSequenceNumber** = 0 , **ByTitle** , **ByCreationTimestamp** , **ByModificationTimestamp** ,  
**ByDeletionTimestamp** , **ByAuthor** , **BySource** , **BySourceApplication** ,  
**ByReminderTime** , **ByPlaceName** , **NoOrder** }  
*The ListNotesOrder enum allows to specify the results ordering for methods listing notes from the local storage database.*
- enum class [ListTagsOrder](#) { **ByUpdateSequenceNumber** , **ByName** , **NoOrder** }  
*The ListTagsOrder enum allows to specify the results ordering for methods listing tags from the local storage database.*
- enum class [GetResourceOption](#) { [WithBinaryData](#) = 1 }  
*The GetResourceOption enum is a QFlags enum which allows to specify which resource fields should be included when findEnResource method is called.*
- enum class [ListSavedSearchesOrder](#) { **ByUpdateSequenceNumber** = 0 , **ByName** , **ByFormat** , **NoOrder** }  
*The ListSavedSearchesOrder enum allows to specify the results ordering for methods listing saved searches from local storage.*

## Signals

- void [upgradeProgress](#) (double progress)  
[LocalStorageManager](#) is capable of performing automatic database upgrades if/when it is necessary.

## Public Member Functions

- [LocalStorageManager](#) (const [Account](#) &account, const StartupOptions options={}, QObject \*parent=nullptr)  
*LocalStorageManager* - constructor. Takes in the account for which the [LocalStorageManager](#) instance is created plus some other parameters determining the startup behaviour.
- void [switchUser](#) (const [Account](#) &account, const StartupOptions options={})  
*switchUser* - switches to another local storage database file associated with the passed in account
- bool [isLocalStorageVersionTooHigh](#) ([ErrorString](#) &errorDescription)
- bool [localStorageRequiresUpgrade](#) ([ErrorString](#) &errorDescription)
- QVector< std::shared\_ptr< [ILocalStoragePatch](#) > > [requiredLocalStoragePatches](#) ()
- qint32 [localStorageVersion](#) ([ErrorString](#) &errorDescription)
- qint32 [highestSupportedLocalStorageVersion](#) () const
- int [userCount](#) ([ErrorString](#) &errorDescription) const  
*userCount* returns the number of non-deleted users currently stored in the local storage database
- bool [addUser](#) (const [User](#) &user, [ErrorString](#) &errorDescription)  
*addUser* adds the passed in [User](#) object to the local storage database
- bool [updateUser](#) (const [User](#) &user, [ErrorString](#) &errorDescription)  
*updateUser* updates the passed in [User](#) object in the local storage database
- bool [findUser](#) ([User](#) &user, [ErrorString](#) &errorDescription) const  
*findUser* attempts to find and fill the fields of the passed in [User](#) object which must have "id" field set as this value is used as the identifier of [User](#) objects in the local storage database
- bool [deleteUser](#) (const [User](#) &user, [ErrorString](#) &errorDescription)  
*deleteUser* marks the user as deleted in local storage
- bool [expungeUser](#) (const [User](#) &user, [ErrorString](#) &errorDescription)  
*expungeUser* permanently deletes the user from the local storage database
- int [notebookCount](#) ([ErrorString](#) &errorDescription) const  
*notebookCount* returns the number of notebooks currently stored in the local storage database
- bool [addNotebook](#) ([Notebook](#) &notebook, [ErrorString](#) &errorDescription)  
*addNotebook* adds the passed in [Notebook](#) to the local storage database
- bool [updateNotebook](#) ([Notebook](#) &notebook, [ErrorString](#) &errorDescription)  
*updateNotebook* updates the passed in [Notebook](#) in the local storage database
- bool [findNotebook](#) ([Notebook](#) &notebook, [ErrorString](#) &errorDescription) const  
*findNotebook* attempts to find and set all found fields of the passed in [Notebook](#) object
- bool [findDefaultNotebook](#) ([Notebook](#) &notebook, [ErrorString](#) &errorDescription) const  
*findDefaultNotebook* attempts to find the default notebook in the local storage database.
- bool [findLastUsedNotebook](#) ([Notebook](#) &notebook, [ErrorString](#) &errorDescription) const  
*findLastUsedNotebook* attempts to find the last used notebook in the local storage database.
- bool [findDefaultOrLastUsedNotebook](#) ([Notebook](#) &notebook, [ErrorString](#) &errorDescription) const  
*findDefaultOrLastUsedNotebook* attempts to find either the default or the last used notebook in the local storage database.
- QList< [Notebook](#) > [listAllNotebooks](#) ([ErrorString](#) &errorDescription, const size\_t limit=0, const size\_t offset=0, const [ListNotebooksOrder](#) order=ListNotebooksOrder::NoOrder, const [OrderDirection](#) orderDirection=OrderDirection::Ascending, const QString &linkedNotebookGuid=QString()) const  
*listAllNotebooks* attempts to list all notebooks within the current account from the local storage database.
- QList< [Notebook](#) > [listNotebooks](#) (const ListObjectsOptions flag, [ErrorString](#) &errorDescription, const size\_t limit=0, const size\_t offset=0, const [ListNotebooksOrder](#) order=ListNotebooksOrder::NoOrder, const [OrderDirection](#) orderDirection=OrderDirection::Ascending, const QString &linkedNotebookGuid=QString()) const  
*listNotebooks* attempts to list notebooks within the account according to the specified input flag
- QList< [SharedNotebook](#) > [listAllSharedNotebooks](#) ([ErrorString](#) &errorDescription) const  
*listAllSharedNotebooks* attempts to list all shared notebooks within the account.
- QList< [SharedNotebook](#) > [listSharedNotebooksPerNotebookGuid](#) (const QString &notebookGuid, [ErrorString](#) &errorDescription) const

*listSharedNotebooksPerNotebookGuid* - attempts to list all shared notebooks per given notebook's remote guid (not local uid, it's important).

- bool [expungeNotebook](#) (Notebook &notebook, [ErrorString](#) &errorDescription)  
*expungeNotebook* permanently deletes the specified notebook from the local storage database.
- int [linkedNotebookCount](#) ([ErrorString](#) &errorDescription) const  
*linkedNotebookCount* returns the number of linked notebooks stored in the local storage database.
- bool [addLinkedNotebook](#) (const [LinkedNotebook](#) &linkedNotebook, [ErrorString](#) &errorDescription)  
*addLinkedNotebook* adds passed in [LinkedNotebook](#) to the local storage database; [LinkedNotebook](#) must have "remote" Evernote service's guid set. It is not possible to add a linked notebook in offline mode so it doesn't make sense for [LinkedNotebook](#) objects to not have guid.
- bool [updateLinkedNotebook](#) (const [LinkedNotebook](#) &linkedNotebook, [ErrorString](#) &errorDescription)  
*updateLinkedNotebook* updates passed in [LinkedNotebook](#) in the local storage database; [LinkedNotebook](#) must have "remote" Evernote service's guid set.
- bool [findLinkedNotebook](#) ([LinkedNotebook](#) &linkedNotebook, [ErrorString](#) &errorDescription) const  
*findLinkedNotebook* attempts to find and set all found fields for passed in by reference [LinkedNotebook](#) object. For [LinkedNotebook](#) local uid doesn't mean anything because it can only be considered valid if it has "remote" Evernote service's guid set. So this passed in [LinkedNotebook](#) object must have guid set to identify the linked notebook in the local storage database.
- QList< [LinkedNotebook](#) > [listAllLinkedNotebooks](#) ([ErrorString](#) &errorDescription, const size\_t limit=0, const size\_t offset=0, const [ListLinkedNotebooksOrder](#) order=ListLinkedNotebooksOrder::NoOrder, const [OrderDirection](#) orderDirection=OrderDirection::Ascending) const  
*listAllLinkedNotebooks* - attempts to list all linked notebooks within the account.
- QList< [LinkedNotebook](#) > [listLinkedNotebooks](#) (const ListObjectsOptions flag, [ErrorString](#) &errorDescription, const size\_t limit=0, const size\_t offset=0, const [ListLinkedNotebooksOrder](#) order=ListLinkedNotebooksOrder::NoOrder, const [OrderDirection](#) orderDirection=OrderDirection::Ascending) const  
*listLinkedNotebooks* attempts to list linked notebooks within the account according to the specified input flag.
- bool [expungeLinkedNotebook](#) (const [LinkedNotebook](#) &linkedNotebook, [ErrorString](#) &errorDescription)  
*expungeLinkedNotebook* permanently deletes specified linked notebook from the local storage database.
- int [noteCount](#) ([ErrorString](#) &errorDescription, const NoteCountOptions options=NoteCountOption::IncludeNonDeletedNotes) const  
*noteCount* returns the number of notes currently stored in the local storage database.
- int [noteCountPerNotebook](#) (const [Notebook](#) &notebook, [ErrorString](#) &errorDescription, const NoteCountOptions options=NoteCountOption::IncludeNonDeletedNotes) const  
*noteCountPerNotebook* returns the number of notes currently stored in the local storage database per given notebook.
- int [noteCountPerTag](#) (const [Tag](#) &tag, [ErrorString](#) &errorDescription, const NoteCountOptions options=NoteCountOption::IncludeNonDeletedNotes) const  
*noteCountPerTag* returns the number of notes currently stored in local storage database labeled with given tag.
- bool [noteCountsPerAllTags](#) (QHash< QString, int > &noteCountsPerTagLocalUid, [ErrorString](#) &errorDescription, const NoteCountOptions options=NoteCountOption::IncludeNonDeletedNotes) const  
*noteCountsPerAllTags* returns the number of notes currently stored in local storage database labeled with each tag stored in the local storage database.
- int [noteCountPerNotebooksAndTags](#) (const QStringList &notebookLocalUids, const QStringList &tagLocalUids, [ErrorString](#) &errorDescription, const NoteCountOptions options=NoteCountOption::IncludeNonDeletedNotes) const  
*noteCountPerNotebooksAndTags* returns the number of notes currently stored in local storage database belonging to one of notebooks corresponding to given notebook local uids and labeled by at least one of tags corresponding to given tag local uids
- bool [addNote](#) ([Note](#) &note, [ErrorString](#) &errorDescription)  
*addNote* adds passed in [Note](#) to the local storage database.
- bool [updateNote](#) ([Note](#) &note, const UpdateNoteOptions options, [ErrorString](#) &errorDescription)  
*updateNote* updates passed in [Note](#) in the local storage database.
- bool [findNote](#) ([Note](#) &note, const GetNoteOptions options, [ErrorString](#) &errorDescription) const  
*findNote* - attempts to find note in the local storage database

- `QList< Note > listNotesPerNotebook` (const `Notebook` &notebook, const `GetNoteOptions` options, `ErrorString` &errorDescription, const `ListObjectsOptions` &flag=`ListObjectsOption::ListAll`, const `size_t` limit=0, const `size_t` offset=0, const `ListNotesOrder` &order=`ListNotesOrder::NoOrder`, const `OrderDirection` &orderDirection=`OrderDirection::Ascending`) const  
*listNotesPerNotebook attempts to list notes per given notebook*
- `QList< Note > listNotesPerTag` (const `Tag` &tag, const `GetNoteOptions` options, `ErrorString` &errorDescription, const `ListObjectsOptions` &flag=`ListObjectsOption::ListAll`, const `size_t` limit=0, const `size_t` offset=0, const `ListNotesOrder` &order=`ListNotesOrder::NoOrder`, const `OrderDirection` &orderDirection=`OrderDirection::Ascending`) const  
*listNotesPerTag attempts to list notes labeled with a given tag*
- `QList< Note > listNotesPerNotebooksAndTags` (const `QStringList` &notebookLocalUids, const `QStringList` &tagLocalUids, const `GetNoteOptions` options, `ErrorString` &errorDescription, const `ListObjectsOptions` &flag=`ListObjectsOption::ListAll`, const `size_t` limit=0, const `size_t` offset=0, const `ListNotesOrder` &order=`ListNotesOrder::NoOrder`, const `OrderDirection` &orderDirection=`OrderDirection::Ascending`) const  
*listNotesPerNotebooksAndTags attempts to list notes which are present within one of specified notebooks and are labeled with at least one of specified tags*
- `QList< Note > listNotesByLocalUids` (const `QStringList` &noteLocalUids, const `GetNoteOptions` options, `ErrorString` &errorDescription, const `ListObjectsOptions` &flag=`ListObjectsOption::ListAll`, const `size_t` limit=0, const `size_t` offset=0, const `ListNotesOrder` &order=`ListNotesOrder::NoOrder`, const `OrderDirection` &orderDirection=`OrderDirection::Ascending`) const  
*listNotesByLocalUids attempts to list notes given their local uids*
- `QList< Note > listNotes` (const `ListObjectsOptions` flag, const `GetNoteOptions` options, `ErrorString` &errorDescription, const `size_t` limit=0, const `size_t` offset=0, const `ListNotesOrder` order=`ListNotesOrder::NoOrder`, const `OrderDirection` orderDirection=`OrderDirection::Ascending`, const `QString` &linkedNotebookGuid=`QString()`) const  
*listNotes attempts to list notes within the account according to the specified input flag.*
- `QStringList findNoteLocalUidsWithSearchQuery` (const `NoteSearchQuery` &noteSearchQuery, `ErrorString` &errorDescription) const  
*findNoteLocalUidsWithSearchQuery attempts to find note local uids of notes corresponding to the passed in NoteSearchQuery object.*
- `NoteList findNotesWithSearchQuery` (const `NoteSearchQuery` &noteSearchQuery, const `GetNoteOptions` options, `ErrorString` &errorDescription) const  
*findNotesWithSearchQuery attempts to find notes corresponding to the passed in NoteSearchQuery object.*
- `bool expungeNote` (`Note` &note, `ErrorString` &errorDescription)  
*expungeNote permanently deletes note from local storage.*
- `int tagCount` (`ErrorString` &errorDescription) const  
*tagCount returns the number of non-deleted tags currently stored in the local storage database.*
- `bool addTag` (`Tag` &tag, `ErrorString` &errorDescription)  
*addTag adds passed in Tag to the local storage database. If tag has "remote" Evernote service's guid set, it is identified in the database by this guid. Otherwise it is identified by local uid.*
- `bool updateTag` (`Tag` &tag, `ErrorString` &errorDescription)  
*updateTag updates passed in Tag in the local storage database.*
- `bool findTag` (`Tag` &tag, `ErrorString` &errorDescription) const  
*findTag attempts to find and fill the fields of passed in tag object.*
- `QList< Tag > listAllTagsPerNote` (const `Note` &note, `ErrorString` &errorDescription, const `ListObjectsOptions` &flag=`ListObjectsOption::ListAll`, const `size_t` limit=0, const `size_t` offset=0, const `ListTagsOrder` &order=`ListTagsOrder::NoOrder`, const `OrderDirection` &orderDirection=`OrderDirection::Ascending`) const  
*listAllTagsPerNote lists all tags per given note*
- `QList< Tag > listAllTags` (`ErrorString` &errorDescription, const `size_t` limit=0, const `size_t` offset=0, const `ListTagsOrder` order=`ListTagsOrder::NoOrder`, const `OrderDirection` orderDirection=`OrderDirection::Ascending`, const `QString` &linkedNotebookGuid=`QString()`) const  
*listAllTags lists all tags within the current user's account.*
- `QList< Tag > listTags` (const `ListObjectsOptions` flag, `ErrorString` &errorDescription, const `size_t` limit=0, const `size_t` offset=0, const `ListTagsOrder` &order=`ListTagsOrder::NoOrder`, const `OrderDirection` orderDirection=`OrderDirection::Ascending`, const `QString` &linkedNotebookGuid=`QString()`) const

- listTags attempts to list tags within the account according to the specified input flag.*
- `QList< std::pair< Tag, QStringList > > listTagsWithNoteLocalUids` (const ListObjectsOptions flag, `QString` &errorDescription, const size\_t limit=0, const size\_t offset=0, const `ListTagsOrder` &order=ListTagsOrder::NoOrder, const `OrderDirection` orderDirection=OrderDirection::Ascending, const `QString` &linkedNotebookGuid=QString()) const
  - listTagsWithNoteLocalUids attempts to list tags and their corresponding local uids within the account according to the specified input flag*
- `bool expungeTag` (`Tag` &tag, `QStringList` &expungedChildTagLocalUids, `QString` &errorDescription)
  - expungeTag permanently deletes tag from the local storage database.*
- `bool expungeNotelessTagsFromLinkedNotebooks` (`QString` &errorDescription)
  - expungeNotelessTagsFromLinkedNotebooks permanently deletes from the local storage database those tags which belong to some linked notebook and are not linked with any notes.*
- `int enResourceCount` (`QString` &errorDescription) const
  - enResourceCount (the name is not `Resource` to prevent problems with macro defined on some versions of Windows) returns the number of resources currently stored in the local storage database.*
- `bool addEnResource` (`Resource` &resource, `QString` &errorDescription)
  - addEnResource adds passed in resource to the local storage database.*
- `bool updateEnResource` (`Resource` &resource, `QString` &errorDescription)
  - updateEnResource updates passed in resource in the local storage database.*
- `bool findEnResource` (`Resource` &resource, const `GetResourceOptions` options, `QString` &errorDescription) const
  - findEnResource method attempts to find resource in the local storage database*
- `bool expungeEnResource` (`Resource` &resource, `QString` &errorDescription)
  - expungeResource permanently deletes resource from the local storage database.*
- `int savedSearchCount` (`QString` &errorDescription) const
  - savedSearchCount returns the number of saved searches currently stored in local storage database.*
- `bool addSavedSearch` (`SavedSearch` &search, `QString` &errorDescription)
  - addSavedSearch adds passed in `SavedSearch` to the local storage database; if search has "remote" Evernote service's guid set, it is identified in the database by this guid. Otherwise it is identified by local uid.*
- `bool updateSavedSearch` (`SavedSearch` &search, `QString` &errorDescription)
  - updateSavedSearch updates passed in `SavedSearch` in the local storage database.*
- `bool findSavedSearch` (`SavedSearch` &search, `QString` &errorDescription) const
  - findSavedSearch attempts to find and fill the fields of passed in saved search object.*
- `QList< SavedSearch > listAllSavedSearches` (`QString` &errorDescription, const size\_t limit=0, const size\_t offset=0, const `ListSavedSearchesOrder` order=ListSavedSearchesOrder::NoOrder, const `OrderDirection` orderDirection=OrderDirection::Ascending) const
  - listAllSavedSearches lists all saved searches within the account.*
- `QList< SavedSearch > listSavedSearches` (const ListObjectsOptions flag, `QString` &errorDescription, const size\_t limit=0, const size\_t offset=0, const `ListSavedSearchesOrder` order=ListSavedSearchesOrder::NoOrder, const `OrderDirection` orderDirection=OrderDirection::Ascending) const
  - listSavedSearches attempts to list saved searches within the account according to the specified input flag.*
- `bool expungeSavedSearch` (`SavedSearch` &search, `QString` &errorDescription)
  - expungeSavedSearch permanently deletes saved search from the local storage database.*
- `qint32 accountHighUsn` (const `QString` &linkedNotebookGuid, `QString` &errorDescription)
  - accountHighUsn returns the highest update sequence number within the data elements stored in the local storage database, either for user's own account or for some linked notebook.*

## Friends

- `QUENTIER_EXPORT QTextStream &operator<<` (`QTextStream` &strm, const `StartupOption` option)
- `QUENTIER_EXPORT QDebug &operator<<` (`QDebug` &dbg, const `StartupOption` option)
- `QUENTIER_EXPORT QTextStream &operator<<` (`QTextStream` &strm, const `StartupOptions` options)

- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const StartupOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListObjectsOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListObjectsOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListObjectsOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListObjectsOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const OrderDirection order↔ Direction)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const OrderDirection orderDirection)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListNotebooksOrder order)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListNotebooksOrder order)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListLinkedNotebooksOrder order)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListLinkedNotebooksOrder order)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const NoteCountOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const NoteCountOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const NoteCountOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const NoteCountOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const UpdateNoteOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const UpdateNoteOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const UpdateNoteOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const UpdateNoteOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const GetNoteOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const GetNoteOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const GetNoteOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const GetNoteOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListNotesOrder order)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListNotesOrder order)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListTagsOrder order)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListTagsOrder order)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const GetResourceOption option)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const GetResourceOption option)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const GetResourceOptions options)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const GetResourceOptions options)`
- `QUENTIER_EXPORT QTextStream & operator<< (QTextStream &strm, const ListSavedSearchesOrder order)`
- `QUENTIER_EXPORT QDebug & operator<< (QDebug &dbg, const ListSavedSearchesOrder order)`

## 5.41.1 Member Enumeration Documentation

### 5.41.1.1 `GetNoteOption`

```
enum class quentier::LocalStorageManager::GetNoteOption [strong]
```

The `GetNoteOption` enum is a `QFlags` enum which allows to specify which note fields should be included when `findNote` or one of `listNote*` methods is called.

Most note data is included unconditionally - note title, content, attributes (if any) etc. However, some specific data can be opted to not be included into the returned note data - notably, metadata of resources and binary data of resources. If these are omitted, `findNote` or any of `listNote*` methods might work faster than otherwise

## Enumerator

WithResourceMetadata	WithResourceMetadata value specifies that fields aside <code>dataBody</code> , <code>dataSize</code> , <code>dataHash</code> , <code>alternateDataBody</code> , <code>alternateDataSize</code> , <code>alternateDataHash</code> for each note's resource should be included
WithResourceBinaryData	WithResourceBinaryData value specifies that <code>dataBody</code> , its size and hash and <code>alternateDataBody</code> , its size and hash should be included into each of note's resources; this value only has effect if flags also have <code>WithResourceMetadata</code> value enabled!

## 5.41.1.2 GetResourceOption

```
enum class quentier::LocalStorageManager::GetResourceOption [strong]
```

The `GetResourceOption` enum is a `QFlags` enum which allows to specify which resource fields should be included when `findEnResource` method is called.

Most resource data is included unconditionally but some specific data can be opted to not be included into the returned resource data - notably, binary data of the resource. If it is omitted, `findEnResource` method might work faster than otherwise

## Enumerator

WithBinaryData	WithBinaryData value specifies than <code>dataBody</code> and <code>alternateDataBody</code> should be included into the returned resource
----------------	--

## 5.41.1.3 ListObjectsOption

```
enum class quentier::LocalStorageManager::ListObjectsOption [strong]
```

The `ListObjectsOption` enum is a `QFlags` enum which allows to specify the desired local storage elements in calls to methods listing them from the database.

For example, one can either list all available elements of certain type from local storage or only elements marked as dirty (modified locally, not yet synchronized) or elements never synchronized with the remote storage or elements which are synchronizable with the remote storage etc.

## 5.41.1.4 StartupOption

```
enum class quentier::LocalStorageManager::StartupOption [strong]
```

The `StartupOption` enum is a `QFlags` enum which allows to specify some options to be applied to the local storage database on startup or on call to `switchUser` method.

## Enumerator

ClearDatabase	If <code>ClearDatabase</code> flag is active, <a href="#">LocalStorageManager</a> would wipe any existing database contents; the net effect would be as if no database existed for the given user before the creation of <a href="#">LocalStorageManager</a> or before the call to its <code>switchUser</code> method
OverrideLock	If <code>OverrideLock</code> flag is active, <a href="#">LocalStorageManager</a> would ignore the existing advisory lock (if any) put on the database file; if this flag is not active, the attempt to create <a href="#">LocalStorageManager</a> (or the attempt to call its <code>switchUser</code> method) with the advisory lock on the database file put by someone else would cause the throwing of <a href="#">DatabaseLockedException</a>

### 5.41.1.5 UpdateNoteOption

```
enum class quentier::LocalStorageManager::UpdateNoteOption [strong]
```

The UpdateNoteOption enum is a QFlags enum which allows to specify which note fields should be updated when updateNote method is called.

Most note data is updated unconditionally - note title, content, attributes (if any) etc. However, some specific data can be chosen to not update - notably, metadata of resources, binary data of resources or lists of note's tags

#### Enumerator

UpdateResourceMetadata	UpdateResourceMetadata value specifies that fields aside dataBody, dataSize, dataHash, alternateDataBody, alternateDataSize, alternateDataHash for each note's resource should be updated
UpdateResourceBinaryData	UpdateResourceBinaryData value specifies that dataBody, its size and hash and alternateDataBody, its size and hash should be updated for each of note's resources; this value only has effect if flags also have UpdateResourceMetadata value enabled!
UpdateTags	UpdateTags value specifies that note's tag lists should be updated

## 5.41.2 Constructor & Destructor Documentation

### 5.41.2.1 LocalStorageManager()

```
quentier::LocalStorageManager::LocalStorageManager (
    const Account & account,
    const StartupOptions options = {},
    QObject * parent = nullptr ) [explicit]
```

[LocalStorageManager](#) - constructor. Takes in the account for which the [LocalStorageManager](#) instance is created plus some other parameters determining the startup behaviour.

#### Parameters

<i>account</i>	The account for which the local storage is being created and initialized
<i>options</i>	Startup options for the local storage, none enabled by default
<i>parent</i>	Parent QObject

## 5.41.3 Member Function Documentation

### 5.41.3.1 accountHighUsn()

```
qint32 quentier::LocalStorageManager::accountHighUsn (
    const QString & linkedNotebookGuid,
    ErrorString & errorDescription )
```

accountHighUsn returns the highest update sequence number within the data elements stored in the local storage database, either for user's own account or for some linked notebook.

## Parameters

<i>linkedNotebookGuid</i>	The guid of the linked notebook for which the highest update sequence number is requested; if null or empty, the highest update sequence number for user's own account is returned
<i>errorDescription</i>	Error description if account's highest update sequence number could not be returned

## Returns

Either the highest update sequence number - a non-negative value - or a negative number in case of error

**5.41.3.2 addEnResource()**

```
bool quantier::LocalStorageManager::addEnResource (
    Resource & resource,
    ErrorString & errorDescription )
```

addEnResource adds passed in resource to the local storage database.

## Parameters

<i>resource</i>	<a href="#">Resource</a> to be added to the database, must have either note's local uid set or note's "remote" Evernote service's guid set; may be changed as a result of the call, filled with autogenerated fields like local uid if it was empty before the call
<i>errorDescription</i>	Error description if resource could not be added

## Returns

True if resource was added successfully, false otherwise

**5.41.3.3 addLinkedNotebook()**

```
bool quantier::LocalStorageManager::addLinkedNotebook (
    const LinkedNotebook & linkedNotebook,
    ErrorString & errorDescription )
```

addLinkedNotebook adds passed in [LinkedNotebook](#) to the local storage database; [LinkedNotebook](#) must have "remote" Evernote service's guid set. It is not possible to add a linked notebook in offline mode so it doesn't make sense for [LinkedNotebook](#) objects to not have guid.

## Parameters

<i>linkedNotebook</i>	<a href="#">LinkedNotebook</a> to be added to the local storage database
<i>errorDescription</i>	Error description if linked notebook could not be added

## Returns

True if linked notebook was added successfully, false otherwise

#### 5.41.3.4 addNote()

```
bool quantier::LocalStorageManager::addNote (
    Note & note,
    ErrorString & errorDescription )
```

addNote adds passed in [Note](#) to the local storage database.

##### Parameters

<i>note</i>	<a href="#">Note</a> to be added to local storage database; required to contain either "remote" notebook guid or local notebook uid; may be changed as a result of the call, filled with autogenerated fields like local uid if it was empty before the call; also tag guids are filled if the note passed in contained only tag local uids and tag local uids are filled if the note passed in contained only tag guids
<i>errorDescription</i>	Error description if note could not be added

##### Returns

True if note was added successfully, false otherwise

#### 5.41.3.5 addNotebook()

```
bool quantier::LocalStorageManager::addNotebook (
    Notebook & notebook,
    ErrorString & errorDescription )
```

addNotebook adds the passed in [Notebook](#) to the local storage database

If the notebook has "remote" Evernote service's guid set, it is identified by this guid in the local storage database. Otherwise it is identified by the local uid

##### Parameters

<i>notebook</i>	The notebook to be added to the local storage database; the object is passed by reference and may be changed as a result of the call (filled with autocompleted fields like local uid if it was empty before the call)
<i>errorDescription</i>	Error description if the notebook could not be added

##### Returns

True if the notebook was added successfully, false otherwise

#### 5.41.3.6 addSavedSearch()

```
bool quantier::LocalStorageManager::addSavedSearch (
    SavedSearch & search,
    ErrorString & errorDescription )
```

addSavedSearch adds passed in [SavedSearch](#) to the local storage database; if search has "remote" Evernote service's guid set, it is identified in the database by this guid. Otherwise it is identified by local uid.

## Parameters

<i>search</i>	<a href="#">SavedSearch</a> to be added to the local storage; may be changed as a result of the call, filled with autogenerated fields like local uid if it was empty before the call
<i>errorDescription</i>	Error description if <a href="#">SavedSearch</a> could not be added

## Returns

True if [SavedSearch](#) was added successfully, false otherwise

## 5.41.3.7 addTag()

```
bool quantier::LocalStorageManager::addTag (
    Tag & tag,
    ErrorString & errorDescription )
```

addTag adds passed in [Tag](#) to the local storage database. If tag has "remote" Evernote service's guid set, it is identified in the database by this guid. Otherwise it is identified by local uid.

## Parameters

<i>tag</i>	<a href="#">Tag</a> to be added to the local storage; may be changed as a result of the call, filled with autogenerated fields like local uid if it was empty before the call
<i>errorDescription</i>	Error description if <a href="#">Tag</a> could not be added

## Returns

True if [Tag](#) was added successfully, false otherwise

## 5.41.3.8 addUser()

```
bool quantier::LocalStorageManager::addUser (
    const User & user,
    ErrorString & errorDescription )
```

addUser adds the passed in [User](#) object to the local storage database

The table with Users is only involved in operations with notebooks which have "contact" field set which in turn is used with business accounts

## Parameters

<i>user</i>	The user to be added to the local storage database
<i>errorDescription</i>	Error description if the user could not be added

## Returns

True if the user was added successfully, false otherwise

### 5.41.3.9 deleteUser()

```
bool quantier::LocalStorageManager::deleteUser (
    const User & user,
    ErrorString & errorDescription )
```

deleteUser marks the user as deleted in local storage

#### Parameters

<i>user</i>	The user to be marked as deleted
<i>errorDescription</i>	Error description if the user could not be marked as deleted

#### Returns

True if the user was marked as deleted successfully, false otherwise

### 5.41.3.10 enResourceCount()

```
int quantier::LocalStorageManager::enResourceCount (
    ErrorString & errorDescription ) const
```

enResourceCount (the name is not [Resource](#) to prevent problems with macro defined on some versions of Windows) returns the number of resources currently stored in the local storage database.

#### Parameters

<i>errorDescription</i>	Error description if the number of resources could not be returned
-------------------------	--

#### Returns

Either non-negative value with the number of resources or -1 which means some error occurred

### 5.41.3.11 expungeEnResource()

```
bool quantier::LocalStorageManager::expungeEnResource (
    Resource & resource,
    ErrorString & errorDescription )
```

expungeResource permanently deletes resource from the local storage database.

#### Parameters

<i>resource</i>	<a href="#">Resource</a> to be expunged; may be changed as a result of the call, automatically filled with local uid and note local uid and/or guid if these were empty before the call
<i>errorDescription</i>	Error description if resource could not be expunged

**Returns**

True if resource was expunged successfully, false otherwise

**5.41.3.12 expungeLinkedNotebook()**

```
bool quantier::LocalStorageManager::expungeLinkedNotebook (
    const LinkedNotebook & linkedNotebook,
    ErrorString & errorDescription )
```

expungeLinkedNotebook permanently deletes specified linked notebook from the local storage database.

Evernote API doesn't allow to delete linked notebooks from the remote storage, it can only be done by official desktop client or web client. So this method should be called only during the synchronization with remote service, when some linked notebook is found to be deleted via either official desktop client or web client.

**Parameters**

<i>linkedNotebook</i>	Linked notebook to be expunged. Must have "remote" guid set
<i>errorDescription</i>	Error description if linked notebook could not be expunged

**Returns**

True if linked notebook was expunged successfully, false otherwise

**5.41.3.13 expungeNote()**

```
bool quantier::LocalStorageManager::expungeNote (
    Note & note,
    ErrorString & errorDescription )
```

expungeNote permanently deletes note from local storage.

Evernote API doesn't allow to delete notes from the remote storage, it can only be done by official desktop client or web client. So this method should be called only during the synchronization with remote database, when some note is found to be deleted via either official desktop client or web client.

**Parameters**

<i>note</i>	<a href="#">Note</a> to be expunged; may be changed as a result of the call, filled with fields like local uid or notebook guid or local uid
<i>errorDescription</i>	Error description if note could not be expunged

**Returns**

True if note was expunged successfully, false otherwise

**5.41.3.14 expungeNotebook()**

```
bool quantier::LocalStorageManager::expungeNotebook (
```

```
Notebook & notebook,  
ErrorString & errorDescription )
```

expungeNotebook permanently deletes the specified notebook from the local storage database.

Evernote API doesn't allow to delete the notebooks from the remote storage, it can only be done by the official desktop Evernote client or via its web client. So this method should be called only during the synchronization with the remote storage, when some notebook is found to be deleted via either the official desktop client or via the web client; also, this method can be called for local notebooks not synchronized with Evernote at all.

#### Parameters

<i>notebook</i>	The notebook to be expunged. Must have either "remote" guid or local uid set; the object is passed by reference and may be changed as a result of the call (filled with local uid if it was empty before the call)
<i>errorDescription</i>	Error description if the notebook could not be expunged

#### Returns

True if the notebook was expunged successfully, false otherwise

#### 5.41.3.15 expungeNotelessTagsFromLinkedNotebooks()

```
bool quantier::LocalStorageManager::expungeNotelessTagsFromLinkedNotebooks (  
    ErrorString & errorDescription )
```

expungeNotelessTagsFromLinkedNotebooks permanently deletes from the local storage database those tags which belong to some linked notebook and are not linked with any notes.

#### Parameters

<i>errorDescription</i>	Error description if tag could not be expunged
-------------------------	--

#### Returns

True if relevant tags were expunged successfully, false otherwise

#### 5.41.3.16 expungeSavedSearch()

```
bool quantier::LocalStorageManager::expungeSavedSearch (  
    SavedSearch & search,  
    ErrorString & errorDescription )
```

expungeSavedSearch permanently deletes saved search from the local storage database.

#### Parameters

<i>search</i>	Saved search to be expunged; may be changed as a result of the call filled local uid if it was empty before the call
<i>errorDescription</i>	Error description if saved search could not be expunged

**Returns**

True if saved search was expunged successfully, false otherwise

**5.41.3.17 expungeTag()**

```
bool quantier::LocalStorageManager::expungeTag (
    Tag & tag,
    QStringList & expungedChildTagLocalUids,
    ErrorString & errorDescription )
```

expungeTag permanently deletes tag from the local storage database.

Evernote API doesn't allow to delete tags from remote storage, it can only be done by official desktop client or web client. So this method should be called only during the synchronization with remote database, when some tag is found to be deleted via either official desktop client or web client.

**Parameters**

<i>tag</i>	Tag to be expunged; may be changed as a result of the call, automatically filled with local uid if it was empty before the call
<i>expungedChildTagLocalUids</i>	If the expunged tag was a parent of some other tags, these were expunged as well; this parameter would contain the local uids of expunged child tags
<i>errorDescription</i>	Error description if tag could not be expunged

**Returns**

True if tag was expunged successfully, false otherwise

**5.41.3.18 expungeUser()**

```
bool quantier::LocalStorageManager::expungeUser (
    const User & user,
    ErrorString & errorDescription )
```

expungeUser permanently deletes the user from the local storage database

**Parameters**

<i>user</i>	The user to be expunged
<i>errorDescription</i>	Error description if the user could not be expunged

**Returns**

True if the user was expunged successfully, false otherwise

**5.41.3.19 findDefaultNotebook()**

```
bool quantier::LocalStorageManager::findDefaultNotebook (
```

```
Notebook & notebook,  
ErrorString & errorDescription ) const
```

findDefaultNotebook attempts to find the default notebook in the local storage database.

#### Parameters

<i>notebook</i>	The default notebook to be found
<i>errorDescription</i>	Error description if the default notebook could not be found

#### Returns

True if the default notebook was found, false otherwise

#### 5.41.3.20 findDefaultOrLastUsedNotebook()

```
bool quantier::LocalStorageManager::findDefaultOrLastUsedNotebook (  
    Notebook & notebook,  
    ErrorString & errorDescription ) const
```

findDefaultOrLastUsedNotebook attempts to find either the default or the last used notebook in the local storage database.

#### Parameters

<i>notebook</i>	Either the default or the last used notebook to be found
<i>errorDescription</i>	Error description if the default or the last used notebook could not be found

#### Returns

True if the default or the last used notebook were found, false otherwise

#### 5.41.3.21 findEnResource()

```
bool quantier::LocalStorageManager::findEnResource (  
    Resource & resource,  
    const GetResourceOptions options,  
    ErrorString & errorDescription ) const
```

findEnResource method attempts to find resource in the local storage database

#### Parameters

<i>resource</i>	<a href="#">Resource</a> to be found in the local storage database. If it has the "remote" Evernote service's guid set, this guid is used to identify the resource in the local storage database. Otherwise resource's local uid is used
<i>options</i>	Options specifying which optionally includable fields of the resource should actually be included
<i>errorDescription</i>	Error description if resource could not be found

**Returns**

True if resource was found successfully, false otherwise

**5.41.3.22 findLastUsedNotebook()**

```
bool quantier::LocalStorageManager::findLastUsedNotebook (
    Notebook & notebook,
    ErrorString & errorDescription ) const
```

findLastUsedNotebook attempts to find the last used notebook in the local storage database.

**Parameters**

<i>notebook</i>	The last used notebook to be found
<i>errorDescription</i>	Error description if the last used notebook could not be found

**Returns**

True if the last used notebook was found, false otherwise

**5.41.3.23 findLinkedNotebook()**

```
bool quantier::LocalStorageManager::findLinkedNotebook (
    LinkedNotebook & linkedNotebook,
    ErrorString & errorDescription ) const
```

findLinkedNotebook attempts to find and set all found fields for passed in by reference [LinkedNotebook](#) object. For [LinkedNotebook](#) local uid doesn't mean anything because it can only be considered valid if it has "remote" Evernote service's guid set. So this passed in [LinkedNotebook](#) object must have guid set to identify the linked notebook in the local storage database.

**Parameters**

<i>linkedNotebook</i>	Linked notebook to be found. Must have "remote" guid set
<i>errorDescription</i>	Error description if linked notebook could not be found

**Returns**

True if linked notebook was found, false otherwise

**5.41.3.24 findNote()**

```
bool quantier::LocalStorageManager::findNote (
    Note & note,
    const GetNoteOptions options,
    ErrorString & errorDescription ) const
```

findNote - attempts to find note in the local storage database

## Parameters

<i>note</i>	- note to be found in the local storage database. Must have either local or "remote" Evernote service's guid set
<i>options</i>	- options specifying which optionally includable fields of the note should actually be included
<i>errorDescription</i>	- error description if note could not be found

## Returns

true if note was found successfully, false otherwise

**5.41.3.25 findNotebook()**

```
bool quantier::LocalStorageManager::findNotebook (
    Notebook & notebook,
    ErrorString & errorDescription ) const
```

findNotebook attempts to find and set all found fields of the passed in [Notebook](#) object

If "remote" Evernote service's guid for the notebook is set, it is used to identify the notebook in the local storage database. Otherwise the notebook is identified by its local uid. If it's empty, the search would attempt to find the notebook by its name. If the name is also not set, the search would attempt to find the notebook by linked notebook guid assuming that no more than one notebook corresponds to the linked notebook guid. If linked notebook guid is also not set, the search would fail.

Important! Due to the fact that the notebook name is only unique within the users's own account as well as within each linked notebook, the result of the search by name depends on the notebook's linked notebook guid: if it is not set, the search by name would only search for the notebook with the specified name within the user's own account. If it is set, the search would only consider the linked notebook with the corresponding guid.

## Parameters

<i>notebook</i>	The notebook to be found. Must have either "remote" or local uid or name or linked notebook guid set
<i>errorDescription</i>	Error description if the notebook could not be found

## Returns

True if the notebook was found, false otherwise

**5.41.3.26 findNoteLocalUidsWithSearchQuery()**

```
QStringList quantier::LocalStorageManager::findNoteLocalUidsWithSearchQuery (
    const NoteSearchQuery & noteSearchQuery,
    ErrorString & errorDescription ) const
```

findNoteLocalUidsWithSearchQuery attempts to find note local uids of notes corresponding to the passed in [NoteSearchQuery](#) object.

## Parameters

<i>noteSearchQuery</i>	Filled <a href="#">NoteSearchQuery</a> object used to filter the notes
<i>errorDescription</i>	Error description in case note local uids could not be listed

## Returns

The list of found notes' local uids or empty list in case of error

**5.41.3.27 findNotesWithSearchQuery()**

```
NoteList quentier::LocalStorageManager::findNotesWithSearchQuery (
    const NoteSearchQuery & noteSearchQuery,
    const GetNoteOptions options,
    ErrorString & errorDescription ) const
```

findNotesWithSearchQuery attempts to find notes corresponding to the passed in [NoteSearchQuery](#) object.

## Parameters

<i>noteSearchQuery</i>	Filled <a href="#">NoteSearchQuery</a> object used to filter the notes
<i>options</i>	Options specifying which optionally includable fields of the note should actually be included
<i>errorDescription</i>	Error description in case notes could not be listed

## Returns

Either list of notes per [NoteSearchQuery](#) or empty list in case of error or no notes presence for the given [NoteSearchQuery](#)

**5.41.3.28 findSavedSearch()**

```
bool quentier::LocalStorageManager::findSavedSearch (
    SavedSearch & search,
    ErrorString & errorDescription ) const
```

findSavedSearch attempts to find and fill the fields of passed in saved search object.

If "remote" Evernote services's guid for the saved search is set, it would be used to identify the saved search in the local storage. Otherwise the local uid would be used. If neither guid not local uid are set, saved search's name would be used. If the name is also not set, the search for saved search would fail.

## Parameters

<i>search</i>	<a href="#">SavedSearch</a> to be found in the local storage database
<i>errorDescription</i>	Error description if <a href="#">SavedSearch</a> could not be found

#### Returns

True if [SavedSearch](#) was found, false otherwise

#### 5.41.3.29 findTag()

```
bool quantier::LocalStorageManager::findTag (
    Tag & tag,
    ErrorString & errorDescription ) const
```

findTag attempts to find and fill the fields of passed in tag object.

If "remote" Evernote service's guid for the tag is set, it would be used to identify the tag in the local storage database. Otherwise the local uid would be used. If neither guid nor local uid are set, tag's name would be used. If the name is also not set, the search would fail.

Important! Due to the fact that the tag name is only unique within the users's own account as well as within each linked notebook, the result of the search by name depends on the tag's linked notebook guid: if it is not set, the search by name would only search for the tag with the specified name within the user's own account. If it is set, the search would only consider tags from a linked notebook with the corresponding guid.

#### Parameters

<i>tag</i>	<a href="#">Tag</a> to be found in the local storage database; must have either guid, local uid or name set
<i>errorDescription</i>	Error description in case tag could not be found

#### Returns

True if tag was found, false otherwise

#### 5.41.3.30 findUser()

```
bool quantier::LocalStorageManager::findUser (
    User & user,
    ErrorString & errorDescription ) const
```

findUser attempts to find and fill the fields of the passed in [User](#) object which must have "id" field set as this value is used as the identifier of [User](#) objects in the local storage database

#### Parameters

<i>user</i>	The user to be found. Must have "id" field set
<i>errorDescription</i>	Error description if the user could not be found

#### Returns

True if the user was found successfully, false otherwise

### 5.41.3.31 highestSupportedLocalStorageVersion()

```
qint32 quentier::LocalStorageManager::highestSupportedLocalStorageVersion ( ) const
```

highestSupportedLocalStorageVersion returns the highest version of local storage persistence which the current build of libquentier is capable of working with

#### Returns

Highest supported local storage version

### 5.41.3.32 isLocalStorageVersionTooHigh()

```
bool quentier::LocalStorageManager::isLocalStorageVersionTooHigh (
    ErrorString & errorDescription )
```

isLocalStorageVersionTooHigh method checks whether the existing local storage persistence has version which is too high for the currently run version of libquentier to work with i.e. whether the local storage has already been upgraded using a new version of libquentier.

NOTE: it is libquentier client code's responsibility to call this method and/or localStorageRequiresUpgrade method, libquentier won't call any of these on its own and will just attempt to work with the existing local storage, whatever version it is of. If version is too high, things can fail in most mysterious way, so the client code is obliged to call these methods to ensure the local storage version is checked properly.

#### Parameters

<i>errorDescription</i>	Textual description of the error if the method was unable to determine whether the local storage version is too high for the currently run version of libquentier to work with, otherwise this parameter is not touched by the method
-------------------------	---

#### Returns

True if local storage version is too high for the currently run version of libquentier to work with, false otherwise

### 5.41.3.33 linkedNotebookCount()

```
int quentier::LocalStorageManager::linkedNotebookCount (
    ErrorString & errorDescription ) const
```

linkedNotebookCount returns the number of linked notebooks stored in the local storage database.

#### Parameters

<i>errorDescription</i>	Error description if the number of linked notebooks count not be returned
-------------------------	---

#### Returns

Either non-negative number of linked notebooks or -1 if some error has occurred

### 5.41.3.34 listAllLinkedNotebooks()

```
QList< LinkedNotebook > quantier::LocalStorageManager::listAllLinkedNotebooks (
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListLinkedNotebooksOrder order = ListLinkedNotebooksOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending ) const
```

listAllLinkedNotebooks - attempts to list all linked notebooks within the account.

#### Parameters

<i>errorDescription</i>	Error description if linked notebooks could not be listed, otherwise this parameter is untouched
<i>limit</i>	Limit for the max number of linked notebooks in the result, zero by default which means no limit is set
<i>offset</i>	Number of linked notebooks to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of linked notebooks in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder

#### Returns

Either list of all linked notebooks or empty list in case of error or no linked notebooks presence within the account

### 5.41.3.35 listAllNotebooks()

```
QList< Notebook > quantier::LocalStorageManager::listAllNotebooks (
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListNotebooksOrder order = ListNotebooksOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending,
    const QString & linkedNotebookGuid = QString\(\) ) const
```

listAllNotebooks attempts to list all notebooks within the current account from the local storage database.

#### Parameters

<i>errorDescription</i>	Error description if all notebooks could not be listed; if no error happens, this parameter is untouched
<i>limit</i>	The limit for the max number of notebooks in the result, zero by default which means no limit is set
<i>offset</i>	The number of notebooks to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify a particular ordering of notebooks in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder
<i>linkedNotebookGuid</i>	If it's null, the method would list the notebooks ignoring their belonging to the current account or to some linked notebook; if it's empty, only the non-linked notebooks would be listed; otherwise, the only one notebook from the corresponding linked notebook would be listed

**Returns**

Either the list of all notebooks within the account or empty list in cases of error or no notebooks presence within the account

**5.41.3.36 listAllSavedSearches()**

```
QList< SavedSearch > quantier::LocalStorageManager::listAllSavedSearches (
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListSavedSearchesOrder order = ListSavedSearchesOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending ) const
```

listAllSavedSearches lists all saved searches within the account.

**Parameters**

<i>errorDescription</i>	Error description if all saved searches could not be listed; otherwise this parameter is untouched
<i>limit</i>	Limit for the max number of saved searches in the result, zero by default which means no limit is set
<i>offset</i>	Number of saved searches to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of saved searches in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder

**Returns**

Either the list of all saved searches within the account or empty list in case of error or if there are no saved searches within the account

**5.41.3.37 listAllSharedNotebooks()**

```
QList< SharedNotebook > quantier::LocalStorageManager::listAllSharedNotebooks (
    ErrorString & errorDescription ) const
```

listAllSharedNotebooks attempts to list all shared notebooks within the account.

**Parameters**

<i>errorDescription</i>	Error description if shared notebooks could not be listed; if no error happens, this parameter is untouched
-------------------------	---

**Returns**

Either the list of all shared notebooks within the account or empty list in cases of error or no shared notebooks presence within the account

### 5.41.3.38 listAllTags()

```
QList< Tag > quentier::LocalStorageManager::listAllTags (
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListTagsOrder order = ListTagsOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending,
    const QString & linkedNotebookGuid = QString() ) const
```

listAllTags lists all tags within the current user's account.

#### Parameters

<i>errorDescription</i>	Error description if tags were not listed successfully. In such case the returned list of tags would be empty and error description won't be empty. However, if, for example, the list of tags is empty and error description is empty too, it means the current account does not have any tags created.
<i>limit</i>	Limit for the max number of tags in the result, zero by default which means no limit is set
<i>offset</i>	Number of tags to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of tags in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder
<i>linkedNotebookGuid</i>	If it's null, the method would list tags ignoring their belonging to the current account or to some linked notebook; if it's empty, only the tags from user's own account would be listed; otherwise, only the tags corresponding to the certain linked notebook would be listed

#### Returns

The list of found tags within the account

### 5.41.3.39 listAllTagsPerNote()

```
QList< Tag > quentier::LocalStorageManager::listAllTagsPerNote (
    const Note & note,
    ErrorString & errorDescription,
    const ListObjectsOptions & flag = ListObjectsOption::ListAll,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListTagsOrder & order = ListTagsOrder::NoOrder,
    const OrderDirection & orderDirection = OrderDirection::Ascending ) const
```

listAllTagsPerNote lists all tags per given note

#### Parameters

<i>note</i>	<b>Note</b> for which the list of tags is requested. If it has "remote" Evernote service's guid set, it is used to identify the note in the local storage database. Otherwise its local uid is used for that.
<i>errorDescription</i>	Error description if tags were not listed successfully. In such case the returned list of tags would be empty and error description won't be empty. However, if, for example, the list of tags is empty and error description is empty too, it means the provided note does not have any tags assigned to it.

## Parameters

<i>flag</i>	Input parameter used to set the filter for the desired tags to be listed
<i>limit</i>	Limit for the max number of tags in the result, zero by default which means no limit is set
<i>offset</i>	Number of tags to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of tags in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used;

## Returns

The list of found tags per note

**5.41.3.40 listLinkedNotebooks()**

```
QList< LinkedNotebook > quentier::LocalStorageManager::listLinkedNotebooks (
    const ListObjectsOptions flag,
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListLinkedNotebooksOrder order = ListLinkedNotebooksOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending ) const
```

listLinkedNotebooks attempts to list linked notebooks within the account according to the specified input flag.

## Parameters

<i>flag</i>	Input parameter used to set the filter for the desired linked notebooks to be listed
<i>errorDescription</i>	Error description if linked notebooks within the account could not be listed; if no error happens, this parameter is untouched
<i>limit</i>	Limit for the max number of linked notebooks in the result, zero by default which means no limit is set
<i>offset</i>	Number of linked notebooks to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of linked notebooks in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder

## Returns

Either list of linked notebooks within the account conforming to the filter or empty list in cases of error or no linked notebooks conforming to the filter exist within the account

**5.41.3.41 listNotebooks()**

```
QList< Notebook > quentier::LocalStorageManager::listNotebooks (
    const ListObjectsOptions flag,
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
```

```
const ListNotebooksOrder order = ListNotebooksOrder::NoOrder,
const OrderDirection orderDirection = OrderDirection::Ascending,
const QString & linkedNotebookGuid = QString() ) const
```

listNotebooks attempts to list notebooks within the account according to the specified input flag

#### Parameters

<i>flag</i>	Input parameter used to set the filter for the desired notebooks to be listed
<i>errorDescription</i>	Error description if notebooks within the account could not be listed; if no error happens, this parameter is untouched
<i>limit</i>	The limit for the max number of notebooks in the result, zero by default which means no limit is set
<i>offset</i>	The number of notebooks to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify a particular ordering of notebooks in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder
<i>linkedNotebookGuid</i>	If it's null, the method would list notebooks ignoring their belonging to the current account or to some linked notebook; if it's empty, only the non-linked notebooks would be listed; otherwise, the only one notebook from the corresponding linked notebook would be listed

#### Returns

Either the list of notebooks within the account conforming to the filter or empty list in cases of error or no notebooks conforming to the filter exist within the account

#### 5.41.3.42 listNotes()

```
QList< Note > quantier::LocalStorageManager::listNotes (
    const ListObjectsOptions flag,
    const GetNoteOptions options,
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListNotesOrder order = ListNotesOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending,
    const QString & linkedNotebookGuid = QString() ) const
```

listNotes attempts to list notes within the account according to the specified input flag.

#### Parameters

<i>flag</i>	Input parameter used to set the filter for the desired notes to be listed
<i>options</i>	Options specifying which optionally includable fields of the note should actually be included
<i>errorDescription</i>	Error description if notes within the account could not be listed; if no error happens, this parameter is untouched
<i>limit</i>	Limit for the max number of notes in the result, zero by default which means no limit is set
<i>offset</i>	Number of notes to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of notes in the result, NoOrder by default

## Parameters

<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder
<i>linkedNotebookGuid</i>	If it's null, notes from both user's own notebooks and linked notebooks would be listed; if it's empty, only the notes from non-linked notebooks would be listed; otherwise, only the notes from the specified linked notebook would be listed

## Returns

Either list of notes within the account conforming to the filter or empty list in cases of error or no notes conforming to the filter exist within the account

## 5.41.3.43 listNotesByLocalUids()

```
QList< Note > quantier::LocalStorageManager::listNotesByLocalUids (
    const QStringList & noteLocalUids,
    const GetNoteOptions options,
    ErrorString & errorDescription,
    const ListObjectsOptions & flag = ListObjectsOption::ListAll,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListNotesOrder & order = ListNotesOrder::NoOrder,
    const OrderDirection & orderDirection = OrderDirection::Ascending ) const
```

listNotesByLocalUids attempts to list notes given their local uids

The method would only return notes which it managed to find within the local storage i.e. having an invalid local uid in the list won't result in an error, just in the corresponding note not returned within the result

Notes within the result can be additionally filtered with flag parameter

## Parameters

<i>noteLocalUids</i>	Local uids of notes to be listed
<i>options</i>	Options specifying which optionally includable fields of the note should actually be included
<i>errorDescription</i>	Error description in case notes could not be listed
<i>flag</i>	Input parameter used to set the filter for the desired notes to be listed
<i>limit</i>	Limit for the max number of notes in the result, zero by default which means no limit is set
<i>offset</i>	Number of notes to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of notes in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used;

## Returns

Either list of notes by local uids or empty list in case of error or no notes corresponding to given local uids presence

## 5.41.3.44 listNotesPerNotebook()

```
QList< Note > quantier::LocalStorageManager::listNotesPerNotebook (
```

```

const Notebook & notebook,
const GetNoteOptions options,
ErrorString & errorDescription,
const ListObjectsOptions & flag = ListObjectsOption::ListAll,
const size_t limit = 0,
const size_t offset = 0,
const ListNotesOrder & order = ListNotesOrder::NoOrder,
const OrderDirection & orderDirection = OrderDirection::Ascending ) const

```

listNotesPerNotebook attempts to list notes per given notebook

#### Parameters

<i>notebook</i>	Notebook for which the list of notes is requested. If it has the "remote" Evernote service's guid set, it would be used to identify the notebook in the local storage database, otherwise its local uid would be used
<i>options</i>	Options specifying which optionally includable fields of the note should actually be included
<i>errorDescription</i>	Error description in case notes could not be listed
<i>flag</i>	Input parameter used to set the filter for the desired notes to be listed
<i>limit</i>	Limit for the max number of notes in the result, zero by default which means no limit is set
<i>offset</i>	Number of notes to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of notes in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used;

#### Returns

Either list of notes per notebook or empty list in case of error or no notes presence in the given notebook

#### 5.41.3.45 listNotesPerNotebooksAndTags()

```

QList< Note > quantier::LocalStorageManager::listNotesPerNotebooksAndTags (
    const QStringList & notebookLocalUids,
    const QStringList & tagLocalUids,
    const GetNoteOptions options,
    ErrorString & errorDescription,
    const ListObjectsOptions & flag = ListObjectsOption::ListAll,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListNotesOrder & order = ListNotesOrder::NoOrder,
    const OrderDirection & orderDirection = OrderDirection::Ascending ) const

```

listNotesPerNotebooksAndTags attempts to list notes which are present within one of specified notebooks and are labeled with at least one of specified tags

#### Parameters

<i>notebookLocalUids</i>	Local uids of notebooks to which the listed notes might belong
<i>tagLocalUids</i>	Local uids of tags with which the listed notes might be labeled
<i>options</i>	Options specifying which optionally includable fields of the note should actually be included
<i>errorDescription</i>	Error description in case notes could not be listed
<i>flag</i>	Input parameter used to set the filter for the desired notes to be listed
<i>limit</i>	Limit for the max number of notes in the result, zero by default which means no limit is set
<i>offset</i>	Number of notes to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of notes in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used;

**Returns**

Either list of notes per notebooks and tags or empty list in case of error or no notes corresponding to given notebooks and tags presence

**5.41.3.46 listNotesPerTag()**

```
QList< Note > quantier::LocalStorageManager::listNotesPerTag (
    const Tag & tag,
    const GetNoteOptions options,
    ErrorString & errorDescription,
    const ListObjectsOptions & flag = ListObjectsOption::ListAll,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListNotesOrder & order = ListNotesOrder::NoOrder,
    const OrderDirection & orderDirection = OrderDirection::Ascending ) const
```

listNotesPerTag attempts to list notes labeled with a given tag

**Parameters**

<i>tag</i>	Tag for which the list of notes labeled with it is requested. If it has the "remote" Evernote service's guid set, it is used to identify the tag in the local storage database, otherwise its local uid is used
<i>options</i>	Options specifying which optionally includable fields of the note should actually be included
<i>errorDescription</i>	Error description in case notes could not be listed
<i>flag</i>	Input parameter used to set the filter for the desired notes to be listed
<i>limit</i>	Limit for the max number of notes in the result, zero by default which means no limit is set
<i>offset</i>	Number of notes to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of notes in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used;

**Returns**

Either list of notes per tag or empty list in case of error or no notes labeled with the given tag presence

**5.41.3.47 listSavedSearches()**

```
QList< SavedSearch > quantier::LocalStorageManager::listSavedSearches (
    const ListObjectsOptions flag,
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListSavedSearchesOrder order = ListSavedSearchesOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending ) const
```

listSavedSearches attempts to list saved searches within the account according to the specified input flag.

**Parameters**

<i>flag</i>	Input parameter used to set the filter for the desired saved searches to be listed
-------------	--

## Parameters

<i>errorDescription</i>	Error description if saved searches within the account could not be listed; if no error happens, this parameter is untouched
<i>limit</i>	Limit for the max number of saved searches in the result, zero by default which means no limit is set
<i>offset</i>	Number of saved searches to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of saved searches in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder

## Returns

Either list of saved searches within the account conforming to the filter or empty list in cases of error or no saved searches conforming to the filter exist within the account

**5.41.3.48 listSharedNotebooksPerNotebookGuid()**

```
QList< SharedNotebook > quantier::LocalStorageManager::listSharedNotebooksPerNotebookGuid (
    const QString & notebookGuid,
    ErrorString & errorDescription ) const
```

listSharedNotebooksPerNotebookGuid - attempts to list all shared notebooks per given notebook's remote guid (not local uid, it's important).

## Parameters

<i>notebookGuid</i>	Remote Evernote service's guid of the notebook for which the shared notebooks are requested
<i>errorDescription</i>	Error description if shared notebooks per notebook guid could not be listed; if no error happens, this parameter is untouched

## Returns

Either the list of shared notebooks per notebook guid or empty list in case of error or no shared notebooks presence per given notebook guid

**5.41.3.49 listTags()**

```
QList< Tag > quantier::LocalStorageManager::listTags (
    const ListObjectsOptions flag,
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListTagsOrder & order = ListTagsOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending,
    const QString & linkedNotebookGuid = QString() ) const
```

listTags attempts to list tags within the account according to the specified input flag.

## Parameters

<i>flag</i>	Input parameter used to set the filter for the desired tags to be listed
<i>errorDescription</i>	Error description if notes within the account could not be listed; if no error happens, this parameter is untouched
<i>limit</i>	Limit for the max number of tags in the result, zero by default which means no limit is set
<i>offset</i>	Number of tags to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of tags in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder
<i>linkedNotebookGuid</i>	If it's null, the method would list tags ignoring their belonging to the current account or to some linked notebook; if it's empty, only the tags from user's own account would be listed; otherwise, only the tags corresponding to the certain linked notebook would be listed

## Returns

Either list of tags within the account conforming to the filter or empty list in cases of error or no tags conforming to the filter exist within the account

## 5.41.3.50 listTagsWithNoteLocalUids()

```
QList< std::pair< Tag, QStringList > > quentier::LocalStorageManager::listTagsWithNoteLocalUids (
    const ListObjectsOptions flag,
    ErrorString & errorDescription,
    const size_t limit = 0,
    const size_t offset = 0,
    const ListTagsOrder & order = ListTagsOrder::NoOrder,
    const OrderDirection orderDirection = OrderDirection::Ascending,
    const QString & linkedNotebookGuid = QString() ) const
```

listTagsWithNoteLocalUids attempts to list tags and their corresponding local uids within the account according to the specified input flag

The method is very similar to listTags only for each listed tag it returns the list of note local uids corresponding to notes labeled with the respective tag.

## Parameters

<i>flag</i>	Input parameter used to set the filter for the desired tags to be listed
<i>errorDescription</i>	Error description if notes within the account could not be listed; if no error happens, this parameter is untouched
<i>limit</i>	Limit for the max number of tags in the result, zero by default which means no limit is set
<i>offset</i>	Number of tags to skip in the beginning of the result, zero by default
<i>order</i>	Allows to specify particular ordering of tags in the result, NoOrder by default
<i>orderDirection</i>	Specifies the direction of ordering, by default ascending direction is used; this parameter has no meaning if order is equal to NoOrder
<i>linkedNotebookGuid</i>	If it's null, the method would list tags ignoring their belonging to the current account or to some linked notebook; if it's empty, only the tags from user's own account would be listed; otherwise, only the tags corresponding to the certain linked notebook would be listed

### Returns

Either list of tags and note local uids within the account conforming to the filter or empty list in cases of error or no tags conforming to the filter exist within the account

#### 5.41.3.51 localStorageRequiresUpgrade()

```
bool quantier::LocalStorageManager::localStorageRequiresUpgrade (
    ErrorString & errorDescription )
```

localStorageRequiresUpgrade method checks whether the existing local storage persistence requires to be upgraded. The upgrades may be required sometimes when new version of libquantier is rolled out which changes something in the internals of local storage organization. This method only checks for changes which are backwards incompatible i.e. once the local storage is upgraded, previous version of libquantier won't be able to work with it properly!

NOTE: it is libquantier client code's responsibility to call this method and/or isLocalStorageVersionTooHigh method, libquantier won't call any of these on its own and will just attempt to work with the existing local storage, whatever version it is of. If version is too high, things can fail in most mysterious way, so the client code is obliged to call these methods to ensure the local storage version is checked properly.

### Parameters

<i>errorDescription</i>	Textual description of the error if the method was unable to determine whether the local storage requires upgrade, otherwise this parameter is not touched by the method
-------------------------	--

### Returns

True if local storage requires upgrade, false otherwise

#### 5.41.3.52 localStorageVersion()

```
qint32 quantier::LocalStorageManager::localStorageVersion (
    ErrorString & errorDescription )
```

localStorageVersion method fetches the current version of local storage persistence which can be used for informational purposes.

### Parameters

<i>errorDescription</i>	Textual description of the error if the method was unable to determine the current version of local storage persistence
-------------------------	---

### Returns

Positive number indication local storage version or negative number in case of error retrieving the local storage version

### 5.41.3.53 notebookCount()

```
int quentier::LocalStorageManager::notebookCount (
    ErrorString & errorDescription ) const
```

notebookCount returns the number of notebooks currently stored in the local storage database

#### Parameters

<i>errorDescription</i>	Error description if the number of notebooks could not be returned
-------------------------	--

#### Returns

Either non-negative value with the number of notebooks or -1 which means some error has occurred

### 5.41.3.54 noteCount()

```
int quentier::LocalStorageManager::noteCount (
    ErrorString & errorDescription,
    const NoteCountOptions options = NoteCountOption::IncludeNonDeletedNotes ) const
```

noteCount returns the number of notes currently stored in the local storage database.

#### Parameters

<i>errorDescription</i>	Error description if the number of notes could not be returned
<i>options</i>	Options clarifying which notes to list; by default only non-deleted notes are listed

#### Returns

Either non-negative value with the number of notes or -1 which means some error occurred

### 5.41.3.55 noteCountPerNotebook()

```
int quentier::LocalStorageManager::noteCountPerNotebook (
    const Notebook & notebook,
    ErrorString & errorDescription,
    const NoteCountOptions options = NoteCountOption::IncludeNonDeletedNotes ) const
```

noteCountPerNotebook returns the number of notes currently stored in the local storage database per given notebook.

#### Parameters

<i>notebook</i>	<a href="#">Notebook</a> for which the number of notes is requested. If its guid is set, it is used to identify the notebook, otherwise its local uid is used
<i>errorDescription</i>	Error description if the number of notes per given notebook could not be returned
<i>options</i>	Options clarifying which notes to list; by default only non-deleted notes are listed

**Returns**

Either non-negative value with the number of notes per given notebook or -1 which means some error occurred

**5.41.3.56 noteCountPerNotebooksAndTags()**

```
int quotientier::LocalStorageManager::noteCountPerNotebooksAndTags (
    const QStringList & notebookLocalUids,
    const QStringList & tagLocalUids,
    ErrorString & errorDescription,
    const NoteCountOptions options = NoteCountOption::IncludeNonDeletedNotes ) const
```

noteCountPerNotebooksAndTags returns the number of notes currently stored in local storage database belonging to one of notebooks corresponding to given notebook local uids and labeled by at least one of tags corresponding to given tag local uids

**Parameters**

<i>notebookLocalUids</i>	The list of notebook local uids used for filtering
<i>tagLocalUids</i>	The list of tag local uids used for filtering
<i>errorDescription</i>	Error description if the number of notes per notebooks and tags could not be returned
<i>options</i>	Options clarifying which notes to list; by default only non-deleted notes are listed

**Returns**

Either non-negative value with the number of notes per given tag or -1 which means some error occurred

**5.41.3.57 noteCountPerTag()**

```
int quotientier::LocalStorageManager::noteCountPerTag (
    const Tag & tag,
    ErrorString & errorDescription,
    const NoteCountOptions options = NoteCountOption::IncludeNonDeletedNotes ) const
```

noteCountPerTag returns the number of notes currently stored in local storage database labeled with given tag.

**Parameters**

<i>tag</i>	<a href="#">Tag</a> for which the number of notes labeled with it is requested. If its guid is set, it is used to identify the tag, otherwise its local uid is used
<i>errorDescription</i>	Error description if the number of notes per given tag could not be returned
<i>options</i>	Options clarifying which notes to list; by default only non-deleted notes are listed

**Returns**

Either non-negative value with the number of notes per given tag or -1 which means some error occurred

**5.41.3.58 noteCountsPerAllTags()**

```
bool quotientier::LocalStorageManager::noteCountsPerAllTags (
```

```
QHash< QString, int > & noteCountsPerTagLocalUid,
QString & errorDescription,
const NoteCountOptions options = NoteCountOption::IncludeNonDeletedNotes ) const
```

noteCountsPerAllTags returns the number of notes currently stored in local storage database labeled with each tag stored in the local storage database.

#### Parameters

<i>noteCountsPerTagLocalUid</i>	The result hash: note counts by tag local uids
<i>errorDescription</i>	Error description if the number of notes per all tags could not be returned
<i>options</i>	Options clarifying which notes to list; by default only non-deleted notes are listed

#### Returns

True if note counts for all tags were computed successfully, false otherwise

### 5.41.3.59 requiredLocalStoragePatches()

```
QVector< std::shared_ptr< ILocalStoragePatch > > quentier::LocalStorageManager::required←
LocalStoragePatches ( )
```

requiredLocalStoragePatches provides the client code with the list of patches which need to be applied to the current state of local storage in order to bring it to a state compatible with the current version of code. If no patches are required, an empty list of patches is returned.

The client code should apply each patch in the exact order in which they are returned by this method.

#### Returns

The vector of patches required to be applied to the current local storage version

### 5.41.3.60 savedSearchCount()

```
int quentier::LocalStorageManager::savedSearchCount (
    QString & errorDescription ) const
```

savedSearchCount returns the number of saved searches currently stored in local storage database.

#### Parameters

<i>errorDescription</i>	Error description if the number of saved searches could not be returned
-------------------------	---

#### Returns

Either non-negative value with the number of saved searches or -1 which means some error occurred

### 5.41.3.61 switchUser()

```
void quantier::LocalStorageManager::switchUser (
    const Account & account,
    const StartupOptions options = {} )
```

switchUser - switches to another local storage database file associated with the passed in account

If optional "startFromScratch" parameter is set to true (it is false by default), the database file would be erased and only then - opened. If optional "overrideLock" parameter is set to true, the advisory lock set on the database file (if any) would be forcefully removed; otherwise, if this parameter is set to false, the presence of advisory lock on the database file would cause the method to throw [DatabaseLockedException](#)

#### Parameters

<i>account</i>	The account to which the local storage is to be switched
<i>options</i>	Startup options for the local storage, none enabled by default

### 5.41.3.62 tagCount()

```
int quantier::LocalStorageManager::tagCount (
    ErrorString & errorDescription ) const
```

tagCount returns the number of non-deleted tags currently stored in the local storage database.

#### Parameters

<i>errorDescription</i>	Error description if the number of tags could not be returned
-------------------------	---

#### Returns

Either non-negative value with the number of tags or -1 which means some error occurred

### 5.41.3.63 updateEnResource()

```
bool quantier::LocalStorageManager::updateEnResource (
    Resource & resource,
    ErrorString & errorDescription )
```

updateEnResource updates passed in resource in the local storage database.

If the resource has "remote" Evernote service's guid set, it is identified by this guid in the local storage database. If no resource with such guid is found, the local uid is used to identify the resource in the local storage database. If the resource has no guid, the local uid is used to identify it in the local storage database.

#### Parameters

<i>resource</i>	<a href="#">Resource</a> to be updated; may be changed as a result of the call, automatically filled with local uid and note local uid and/or guid if these were empty before the call
<i>errorDescription</i>	Error description if resource could not be updated

**Returns**

True if resource was updated successfully, false otherwise

**5.41.3.64 updateLinkedNotebook()**

```
bool quantier::LocalStorageManager::updateLinkedNotebook (
    const LinkedNotebook & linkedNotebook,
    ErrorString & errorDescription )
```

updateLinkedNotebook updates passed in [LinkedNotebook](#) in the local storage database; [LinkedNotebook](#) must have "remote" Evernote service's guid set.

**Parameters**

<i>linkedNotebook</i>	<a href="#">LinkedNotebook</a> to be updated in the local storage database
<i>errorDescription</i>	Error description if linked notebook could not be updated

**Returns**

True if linked notebook was updated successfully, false otherwise

**5.41.3.65 updateNote()**

```
bool quantier::LocalStorageManager::updateNote (
    Note & note,
    const UpdateNoteOptions options,
    ErrorString & errorDescription )
```

updateNote updates passed in [Note](#) in the local storage database.

If the note has "remote" Evernote service's guid set, it is identified by this guid in the local storage database. If no note with such guid is found, the local uid is used to identify the note in the local storage database. If the note has no guid, the local uid is used to identify it in the local storage database.

A special way in which this method might be used is the update of a note which clears note's guid. This way is special because it imposes certain requirements onto the resources which the note might have. However, it is only relevant if options input parameter has UpdateResourceMetadata flag enabled. The requirements for this special case are as follows:

- each resource should not have noteGuid field set to a non-empty value
- each resource should not have guid field set to a non-empty value as it makes no sense for note without guid i.e. note not synchronized with Evernote to own a resource which has guid i.e. is synchronized with Evernote

## Parameters

<i>note</i>	<a href="#">Note</a> to be updated in the local storage database; required to contain either "remote" notebook guid or local notebook uid; may be changed as a result of the call, filled with fields like local uid or notebook guid or local uid if any of these were empty before the call; also tag guids are filled if the note passed in contained only tag local uids and tag local uids are filled if the note passed in contained only tag guids. Bear in mind that after the call the note may not have the representative resources if "updateNoteOptions" input parameter contained no "UpdateResourceMetadata" flag as well as it may not have the representative tags if "UpdateTags" flag was not set
<i>options</i>	Options specifying which optionally updatable fields of the note should actually be updated
<i>errorDescription</i>	Error description if note could not be updated

## Returns

True if note was updated successfully, false otherwise

### 5.41.3.66 updateNotebook()

```
bool quantier::LocalStorageManager::updateNotebook (
    Notebook & notebook,
    ErrorString & errorDescription )
```

updateNotebook updates the passed in [Notebook](#) in the local storage database

If the notebook has "remote" Evernote service's guid set, it is identified by this guid in the local storage database. Otherwise it is identified by the local uid.

## Parameters

<i>notebook</i>	<a href="#">Notebook</a> to be updated in the local storage database; the object is passed by reference and may be changed as a result of the call (filled with autocompleted fields like local uid if it was empty before the call)
<i>errorDescription</i>	Error description if the notebook could not be updated

## Returns

True if the notebook was updated successfully, false otherwise

### 5.41.3.67 updateSavedSearch()

```
bool quantier::LocalStorageManager::updateSavedSearch (
    SavedSearch & search,
    ErrorString & errorDescription )
```

updateSavedSearch updates passed in [SavedSearch](#) in the local storage database.

If search has "remote" Evernote service's guid set, it is identified in the database by this guid. If the saved search has no guid, the local uid is used to identify it in the local storage database.

## Parameters

<i>search</i>	<a href="#">SavedSearch</a> filled with values to be updated in the local storage database; may be changed as a result of the call filled local uid if it was empty before the call
<i>errorDescription</i>	Error description if <a href="#">SavedSearch</a> could not be updated

## Returns

True if [SavedSearch](#) was updated successfully, false otherwise

**5.41.3.68 updateTag()**

```
bool quantier::LocalStorageManager::updateTag (
    Tag & tag,
    ErrorString & errorDescription )
```

updateTag updates passed in [Tag](#) in the local storage database.

If the tag has "remote" Evernote service's guid set, it is identified by this guid in the local storage database. If the tag has no guid, the local uid is used to identify it in the local storage database.

## Parameters

<i>tag</i>	<a href="#">Tag</a> filled with values to be updated in the local storage database. <a href="#">Note</a> that it can be changed * as a result of the call: automatically filled with local uid if it was empty before the call
<i>errorDescription</i>	Error description if tag could not be updated

## Returns

True if tag was updated successfully, false otherwise

**5.41.3.69 updateUser()**

```
bool quantier::LocalStorageManager::updateUser (
    const User & user,
    ErrorString & errorDescription )
```

updateUser updates the passed in [User](#) object in the local storage database

The table with Users is only involved in operations with notebooks which have "contact" field set which in turn is used with business accounts

## Parameters

<i>user</i>	The user to be updated in the local storage database
<i>errorDescription</i>	Error description if the user could not be updated

**Returns**

True if the user was updated successfully, false otherwise

**5.41.3.70 upgradeProgress**

```
void quantier::LocalStorageManager::upgradeProgress (
    double progress ) [signal]
```

[LocalStorageManager](#) is capable of performing automatic database upgrades if/when it is necessary.

As the database upgrade can be a lengthy operation, this signal is meant to provide some feedback on the progress of the upgrade

**Parameters**

<i>progress</i>	The value from 0 to 1 denoting the database upgrade progress
-----------------	--

**5.41.3.71 userCount()**

```
int quantier::LocalStorageManager::userCount (
    QString & errorDescription ) const
```

userCount returns the number of non-deleted users currently stored in the local storage database

**Parameters**

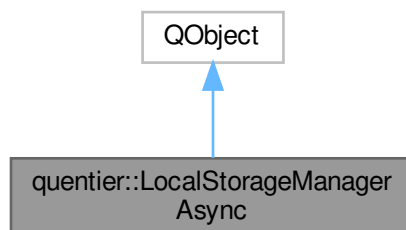
<i>errorDescription</i>	Error description if the number of users could not be returned
-------------------------	--

**Returns**

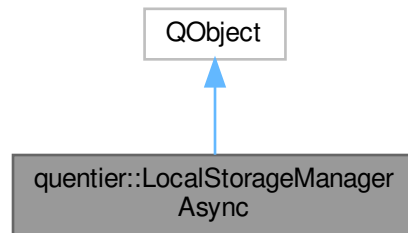
Either non-negative value with the number of users or -1 which means some error has occurred

**5.42 quantier::LocalStorageManagerAsync Class Reference**

Inheritance diagram for quantier::LocalStorageManagerAsync:



Collaboration diagram for `quentier::LocalStorageManagerAsync`:



### Public Slots

- void **init** ()
- void **onGetUserCountRequest** (QUuid requestId)
- void **onSwitchUserRequest** ([Account](#) account, LocalStorageManager::StartupOptions startupOptions, QUuid requestId)
- void **onAddUserRequest** ([User](#) user, QUuid requestId)
- void **onUpdateUserRequest** ([User](#) user, QUuid requestId)
- void **onFindUserRequest** ([User](#) user, QUuid requestId)
- void **onDeleteUserRequest** ([User](#) user, QUuid requestId)
- void **onExpungeUserRequest** ([User](#) user, QUuid requestId)
- void **onGetNotebookCountRequest** (QUuid requestId)
- void **onAddNotebookRequest** ([Notebook](#) notebook, QUuid requestId)
- void **onUpdateNotebookRequest** ([Notebook](#) notebook, QUuid requestId)
- void **onFindNotebookRequest** ([Notebook](#) notebook, QUuid requestId)
- void **onFindDefaultNotebookRequest** ([Notebook](#) notebook, QUuid requestId)
- void **onFindLastUsedNotebookRequest** ([Notebook](#) notebook, QUuid requestId)
- void **onFindDefaultOrLastUsedNotebookRequest** ([Notebook](#) notebook, QUuid requestId)
- void **onListAllNotebooksRequest** (size\_t limit, size\_t offset, [LocalStorageManager::ListNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QUuid requestId)
- void **onListAllSharedNotebooksRequest** (QUuid requestId)
- void **onListNotebooksRequest** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QUuid requestId)
- void **onListSharedNotebooksPerNotebookGuidRequest** (QString notebookGuid, QUuid requestId)
- void **onExpungeNotebookRequest** ([Notebook](#) notebook, QUuid requestId)
- void **onGetLinkedNotebookCountRequest** (QUuid requestId)
- void **onAddLinkedNotebookRequest** ([LinkedNotebook](#) linkedNotebook, QUuid requestId)
- void **onUpdateLinkedNotebookRequest** ([LinkedNotebook](#) linkedNotebook, QUuid requestId)
- void **onFindLinkedNotebookRequest** ([LinkedNotebook](#) linkedNotebook, QUuid requestId)
- void **onListAllLinkedNotebooksRequest** (size\_t limit, size\_t offset, [LocalStorageManager::ListLinkedNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onListLinkedNotebooksRequest** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListLinkedNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onExpungeLinkedNotebookRequest** ([LinkedNotebook](#) linkedNotebook, QUuid requestId)
- void **onGetNoteCountRequest** (LocalStorageManager::NoteCountOptions options, QUuid requestId)

- void **onGetNoteCountPerNotebookRequest** ([Notebook](#) notebook, LocalStorageManager::NoteCountOptions options, QUuid requestId)
- void **onGetNoteCountPerTagRequest** ([Tag](#) tag, LocalStorageManager::NoteCountOptions options, QUuid requestId)
- void **onGetNoteCountsPerAllTagsRequest** (LocalStorageManager::NoteCountOptions options, QUuid requestId)
- void **onGetNoteCountPerNotebooksAndTagsRequest** (QStringList notebookLocalUids, QStringList tagLocalUids, LocalStorageManager::NoteCountOptions options, QUuid requestId)
- void **onAddNoteRequest** ([Note](#) note, QUuid requestId)
- void **onUpdateNoteRequest** ([Note](#) note, LocalStorageManager::UpdateNoteOptions options, QUuid requestId)
- void **onFindNoteRequest** ([Note](#) note, LocalStorageManager::GetNoteOptions options, QUuid requestId)
- void **onListNotesPerNotebookRequest** ([Notebook](#) notebook, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onListNotesPerTagRequest** ([Tag](#) tag, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onListNotesPerNotebooksAndTagsRequest** (QStringList notebookLocalUids, QStringList tagLocalUids, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onListNotesByLocalUidsRequest** (QStringList noteLocalUids, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onListNotesRequest** (LocalStorageManager::ListObjectsOptions flag, LocalStorageManager::GetNoteOptions options, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QUuid requestId)
- void **onFindNoteLocalUidsWithSearchQuery** ([NoteSearchQuery](#) noteSearchQuery, QUuid requestId)
- void **onExpungeNoteRequest** ([Note](#) note, QUuid requestId)
- void **onGetTagCountRequest** (QUuid requestId)
- void **onAddTagRequest** ([Tag](#) tag, QUuid requestId)
- void **onUpdateTagRequest** ([Tag](#) tag, QUuid requestId)
- void **onFindTagRequest** ([Tag](#) tag, QUuid requestId)
- void **onListAllTagsPerNoteRequest** ([Note](#) note, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onListAllTagsRequest** (size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QUuid requestId)
- void **onListTagsRequest** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QUuid requestId)
- void **onListTagsWithNoteLocalUidsRequest** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QUuid requestId)
- void **onExpungeTagRequest** ([Tag](#) tag, QUuid requestId)
- void **onExpungeNotelessTagsFromLinkedNotebooksRequest** (QUuid requestId)
- void **onGetResourceCountRequest** (QUuid requestId)
- void **onAddResourceRequest** ([Resource](#) resource, QUuid requestId)
- void **onUpdateResourceRequest** ([Resource](#) resource, QUuid requestId)
- void **onFindResourceRequest** ([Resource](#) resource, LocalStorageManager::GetResourceOptions options, QUuid requestId)
- void **onExpungeResourceRequest** ([Resource](#) resource, QUuid requestId)
- void **onGetSavedSearchCountRequest** (QUuid requestId)
- void **onAddSavedSearchRequest** ([SavedSearch](#) search, QUuid requestId)
- void **onUpdateSavedSearchRequest** ([SavedSearch](#) search, QUuid requestId)

- void **onFindSavedSearchRequest** ([SavedSearch](#) search, QUuid requestId)
- void **onListAllSavedSearchesRequest** (size\_t limit, size\_t offset, [LocalStorageManager::ListSavedSearchesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onListSavedSearchesRequest** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListSavedSearchesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **onExpungeSavedSearchRequest** ([SavedSearch](#) search, QUuid requestId)
- void **onAccountHighUsnRequest** (QString linkedNotebookGuid, QUuid requestId)

## Signals

- void **initialized** ()
- void **getUserCountComplete** (int userCount, QUuid requestId)
- void **getUserCountFailed** ([ErrorString](#) errorDescription, QUuid requestId)
- void **switchUserComplete** ([Account](#) account, QUuid requestId)
- void **switchUserFailed** ([Account](#) account, [ErrorString](#) errorDescription, QUuid requestId)
- void **addUserComplete** ([User](#) user, QUuid requestId)
- void **addUserFailed** ([User](#) user, [ErrorString](#) errorDescription, QUuid requestId)
- void **updateUserComplete** ([User](#) user, QUuid requestId)
- void **updateUserFailed** ([User](#) user, [ErrorString](#) errorDescription, QUuid requestId)
- void **findUserComplete** ([User](#) foundUser, QUuid requestId)
- void **findUserFailed** ([User](#) user, [ErrorString](#) errorDescription, QUuid requestId)
- void **deleteUserComplete** ([User](#) user, QUuid requestId)
- void **deleteUserFailed** ([User](#) user, [ErrorString](#) errorDescription, QUuid requestId)
- void **expungeUserComplete** ([User](#) user, QUuid requestId)
- void **expungeUserFailed** ([User](#) user, [ErrorString](#) errorDescription, QUuid requestId)
- void **getNotebookCountComplete** (int notebookCount, QUuid requestId)
- void **getNotebookCountFailed** ([ErrorString](#) errorDescription, QUuid requestId)
- void **addNotebookComplete** ([Notebook](#) notebook, QUuid requestId)
- void **addNotebookFailed** ([Notebook](#) notebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **updateNotebookComplete** ([Notebook](#) notebook, QUuid requestId)
- void **updateNotebookFailed** ([Notebook](#) notebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **findNotebookComplete** ([Notebook](#) foundNotebook, QUuid requestId)
- void **findNotebookFailed** ([Notebook](#) notebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **findDefaultNotebookComplete** ([Notebook](#) foundNotebook, QUuid requestId)
- void **findDefaultNotebookFailed** ([Notebook](#) notebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **findLastUsedNotebookComplete** ([Notebook](#) foundNotebook, QUuid requestId)
- void **findLastUsedNotebookFailed** ([Notebook](#) notebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **findDefaultOrLastUsedNotebookComplete** ([Notebook](#) foundNotebook, QUuid requestId)
- void **findDefaultOrLastUsedNotebookFailed** ([Notebook](#) notebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **listAllNotebooksComplete** (size\_t limit, size\_t offset, [LocalStorageManager::ListNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QList< [Notebook](#) > foundNotebooks, QUuid requestId)
- void **listAllNotebooksFailed** (size\_t limit, size\_t offset, [LocalStorageManager::ListNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, [ErrorString](#) errorDescription, QUuid requestId)
- void **listNotebooksComplete** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QList< [Notebook](#) > foundNotebooks, QUuid requestId)
- void **listNotebooksFailed** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, [ErrorString](#) errorDescription, QUuid requestId)
- void **listAllSharedNotebooksComplete** (QList< [SharedNotebook](#) > foundSharedNotebooks, QUuid requestId)

- void **listAllSharedNotebooksFailed** ([ErrorString](#) errorDescription, QUuid requestId)
- void **listSharedNotebooksPerNotebookGuidComplete** (QString notebookGuid, QList< [SharedNotebook](#) > foundSharedNotebooks, QUuid requestId)
- void **listSharedNotebooksPerNotebookGuidFailed** (QString notebookGuid, [ErrorString](#) errorDescription, QUuid requestId)
- void **expungeNotebookComplete** ([Notebook](#) notebook, QUuid requestId)
- void **expungeNotebookFailed** ([Notebook](#) notebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **getLinkedNotebookCountComplete** (int linkedNotebookCount, QUuid requestId)
- void **getLinkedNotebookCountFailed** ([ErrorString](#) errorDescription, QUuid requestId)
- void **addLinkedNotebookComplete** ([LinkedNotebook](#) linkedNotebook, QUuid requestId)
- void **addLinkedNotebookFailed** ([LinkedNotebook](#) linkedNotebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **updateLinkedNotebookComplete** ([LinkedNotebook](#) linkedNotebook, QUuid requestId)
- void **updateLinkedNotebookFailed** ([LinkedNotebook](#) linkedNotebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **findLinkedNotebookComplete** ([LinkedNotebook](#) foundLinkedNotebook, QUuid requestId)
- void **findLinkedNotebookFailed** ([LinkedNotebook](#) linkedNotebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **listAllLinkedNotebooksComplete** (size\_t limit, size\_t offset, [LocalStorageManager::ListLinkedNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QList< [LinkedNotebook](#) > foundLinkedNotebooks, QUuid requestId)
- void **listAllLinkedNotebooksFailed** (size\_t limit, size\_t offset, [LocalStorageManager::ListLinkedNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, [ErrorString](#) errorDescription, QUuid requestId)
- void **listLinkedNotebooksComplete** ([LocalStorageManager::ListObjectsOptions](#) flag, size\_t limit, size\_t offset, [LocalStorageManager::ListLinkedNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QList< [LinkedNotebook](#) > foundLinkedNotebooks, QUuid requestId)
- void **listLinkedNotebooksFailed** ([LocalStorageManager::ListObjectsOptions](#) flag, size\_t limit, size\_t offset, [LocalStorageManager::ListLinkedNotebooksOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, [ErrorString](#) errorDescription, QUuid requestId)
- void **expungeLinkedNotebookComplete** ([LinkedNotebook](#) linkedNotebook, QUuid requestId)
- void **expungeLinkedNotebookFailed** ([LinkedNotebook](#) linkedNotebook, [ErrorString](#) errorDescription, QUuid requestId)
- void **getNoteCountComplete** (int noteCount, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountFailed** ([ErrorString](#) errorDescription, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountPerNotebookComplete** (int noteCount, [Notebook](#) notebook, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountPerNotebookFailed** ([ErrorString](#) errorDescription, [Notebook](#) notebook, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountPerTagComplete** (int noteCount, [Tag](#) tag, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountPerTagFailed** ([ErrorString](#) errorDescription, [Tag](#) tag, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountsPerAllTagsComplete** (QHash< QString, int > noteCountsPerTagLocalUids, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountsPerAllTagsFailed** ([ErrorString](#) errorDescription, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountPerNotebooksAndTagsComplete** (int noteCount, QStringList notebookLocalUids, QStringList tagLocalUids, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **getNoteCountPerNotebooksAndTagsFailed** ([ErrorString](#) errorDescription, QStringList notebookLocalUids, QStringList tagLocalUids, [LocalStorageManager::NoteCountOptions](#) options, QUuid requestId)
- void **addNoteComplete** ([Note](#) note, QUuid requestId)
- void **addNoteFailed** ([Note](#) note, [ErrorString](#) errorDescription, QUuid requestId)
- void **updateNoteComplete** ([Note](#) note, [LocalStorageManager::UpdateNoteOptions](#) options, QUuid requestId)

- void **updateNoteFailed** ([Note](#) note, LocalStorageManager::UpdateNoteOptions options, [ErrorString](#) error↔ Description, QUuid requestId)
- void **findNoteComplete** ([Note](#) foundNote, LocalStorageManager::GetNoteOptions options, QUuid requestId)
- void **findNoteFailed** ([Note](#) note, LocalStorageManager::GetNoteOptions options, [ErrorString](#) error↔ Description, QUuid requestId)
- void **listNotesPerNotebookComplete** ([Notebook](#) notebook, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QList< [Note](#) > foundNotes, QUuid requestId)
- void **listNotesPerNotebookFailed** ([Notebook](#) notebook, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, [ErrorString](#) errorDescription, QUuid requestId)
- void **listNotesPerTagComplete** ([Tag](#) tag, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QList< [Note](#) > foundNotes, QUuid requestId)
- void **listNotesPerTagFailed** ([Tag](#) tag, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, [ErrorString](#) errorDescription, QUuid requestId)
- void **listNotesPerNotebooksAndTagsComplete** (QStringList notebookLocalUids, QStringList tagLocalUids, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) order↔ Direction, QList< [Note](#) > foundNotes, QUuid requestId)
- void **listNotesPerNotebooksAndTagsFailed** (QStringList notebookLocalUids, QStringList tagLocalUids, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) order↔ Direction, [ErrorString](#) errorDescription, QUuid requestId)
- void **listNotesByLocalUidsComplete** (QStringList noteLocalUids, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QList< [Note](#) > foundNotes, QUuid requestId)
- void **listNotesByLocalUidsFailed** (QStringList noteLocalUids, LocalStorageManager::GetNoteOptions options, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, [ErrorString](#) errorDescription, QUuid requestId)
- void **listNotesComplete** (LocalStorageManager::ListObjectsOptions flag, LocalStorageManager↔::GetNoteOptions options, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QList< [Note](#) > found↔ Notes, QUuid requestId)
- void **listNotesFailed** (LocalStorageManager::ListObjectsOptions flag, LocalStorageManager::GetNote↔ Options options, size\_t limit, size\_t offset, [LocalStorageManager::ListNotesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, [ErrorString](#) errorDescription, QUuid requestId)
- void **findNoteLocalUidsWithSearchQueryComplete** (QStringList noteLocalUids, [NoteSearchQuery](#) note↔ SearchQuery, QUuid requestId)
- void **findNoteLocalUidsWithSearchQueryFailed** ([NoteSearchQuery](#) noteSearchQuery, [ErrorString](#) error↔ Description, QUuid requestId)
- void **expungeNoteComplete** ([Note](#) note, QUuid requestId)
- void **expungeNoteFailed** ([Note](#) note, [ErrorString](#) errorDescription, QUuid requestId)
- void **noteMovedToAnotherNotebook** (QString noteLocalUid, QString previousNotebookLocalUid, QString newNotebookLocalUid)
- void **noteTagListChanged** (QString noteLocalUid, QStringList previousNoteTagLocalUids, QStringList newNoteTagLocalUids)
- void **getTagCountComplete** (int tagCount, QUuid requestId)
- void **getTagCountFailed** ([ErrorString](#) errorDescription, QUuid requestId)
- void **addTagComplete** ([Tag](#) tag, QUuid requestId)
- void **addTagFailed** ([Tag](#) tag, [ErrorString](#) errorDescription, QUuid requestId)
- void **updateTagComplete** ([Tag](#) tag, QUuid requestId)
- void **updateTagFailed** ([Tag](#) tag, [ErrorString](#) errorDescription, QUuid requestId)
- void **linkTagWithNoteComplete** ([Tag](#) tag, [Note](#) note, QUuid requestId)
- void **linkTagWithNoteFailed** ([Tag](#) tag, [Note](#) note, [ErrorString](#) errorDescription, QUuid requestId)

- void **findTagComplete** ([Tag](#) tag, QUuid requestId)
- void **findTagFailed** ([Tag](#) tag, [ErrorString](#) errorDescription, QUuid requestId)
- void **listAllTagsPerNoteComplete** (QList< [Tag](#) > foundTags, [Note](#) note, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QUuid requestId)
- void **listAllTagsPerNoteFailed** ([Note](#) note, LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, [ErrorString](#) errorDescription, QUuid requestId)
- void **listAllTagsComplete** (size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QList< [Tag](#) > foundTags, QUuid requestId)
- void **listAllTagsFailed** (size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, [ErrorString](#) errorDescription, QUuid requestId)
- void **listTagsComplete** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QList< [Tag](#) > foundTags, QUuid requestId=QUuid())
- void **listTagsFailed** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, [ErrorString](#) errorDescription, QUuid requestId)
- void **listTagsWithNoteLocalUidsComplete** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, QList< std::pair< [Tag](#), QStringList > > foundTags, QUuid requestId)
- void **listTagsWithNoteLocalUidsFailed** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, [LocalStorageManager::ListTagsOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QString linkedNotebookGuid, [ErrorString](#) errorDescription, QUuid requestId)
- void **expungeTagComplete** ([Tag](#) tag, QStringList expungedChildTagLocalUids, QUuid requestId)
- void **expungeTagFailed** ([Tag](#) tag, [ErrorString](#) errorDescription, QUuid requestId)
- void **expungeNotelessTagsFromLinkedNotebooksComplete** (QUuid requestId)
- void **expungeNotelessTagsFromLinkedNotebooksFailed** ([ErrorString](#) errorDescription, QUuid requestId)
- void **getResourceCountComplete** (int resourceCount, QUuid requestId)
- void **getResourceCountFailed** ([ErrorString](#) errorDescription, QUuid requestId)
- void **addResourceComplete** ([Resource](#) resource, QUuid requestId)
- void **addResourceFailed** ([Resource](#) resource, [ErrorString](#) errorDescription, QUuid requestId)
- void **updateResourceComplete** ([Resource](#) resource, QUuid requestId)
- void **updateResourceFailed** ([Resource](#) resource, [ErrorString](#) errorDescription, QUuid requestId)
- void **findResourceComplete** ([Resource](#) resource, LocalStorageManager::GetResourceOptions options, QUuid requestId)
- void **findResourceFailed** ([Resource](#) resource, LocalStorageManager::GetResourceOptions options, [ErrorString](#) errorDescription, QUuid requestId)
- void **expungeResourceComplete** ([Resource](#) resource, QUuid requestId)
- void **expungeResourceFailed** ([Resource](#) resource, [ErrorString](#) errorDescription, QUuid requestId)
- void **getSavedSearchCountComplete** (int savedSearchCount, QUuid requestId)
- void **getSavedSearchCountFailed** ([ErrorString](#) errorDescription, QUuid requestId)
- void **addSavedSearchComplete** ([SavedSearch](#) search, QUuid requestId)
- void **addSavedSearchFailed** ([SavedSearch](#) search, [ErrorString](#) errorDescription, QUuid requestId)
- void **updateSavedSearchComplete** ([SavedSearch](#) search, QUuid requestId)
- void **updateSavedSearchFailed** ([SavedSearch](#) search, [ErrorString](#) errorDescription, QUuid requestId)
- void **findSavedSearchComplete** ([SavedSearch](#) search, QUuid requestId)
- void **findSavedSearchFailed** ([SavedSearch](#) search, [ErrorString](#) errorDescription, QUuid requestId)
- void **listAllSavedSearchesComplete** (size\_t limit, size\_t offset, [LocalStorageManager::ListSavedSearchesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, QList< [SavedSearch](#) > foundSearches, QUuid requestId)
- void **listAllSavedSearchesFailed** (size\_t limit, size\_t offset, [LocalStorageManager::ListSavedSearchesOrder](#) order, [LocalStorageManager::OrderDirection](#) orderDirection, [ErrorString](#) errorDescription, QUuid requestId)

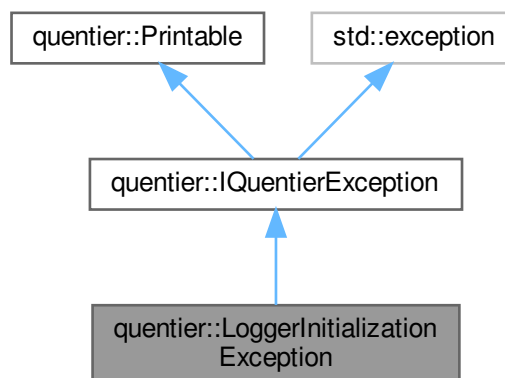
- void **listSavedSearchesComplete** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, LocalStorageManager::ListSavedSearchesOrder order, LocalStorageManager::OrderDirection order↔ Direction, QList< SavedSearch > foundSearches, QUuid requestId)
- void **listSavedSearchesFailed** (LocalStorageManager::ListObjectsOptions flag, size\_t limit, size\_t offset, LocalStorageManager::ListSavedSearchesOrder order, LocalStorageManager::OrderDirection order↔ Direction, ErrorString errorDescription, QUuid requestId)
- void **expungeSavedSearchComplete** (SavedSearch search, QUuid requestId)
- void **expungeSavedSearchFailed** (SavedSearch search, ErrorString errorDescription, QUuid requestId)
- void **accountHighUsnComplete** (qint32 usn, QString linkedNotebookGuid, QUuid requestId)
- void **accountHighUsnFailed** (QString linkedNotebookGuid, ErrorString errorDescription, QUuid requestId)

### Public Member Functions

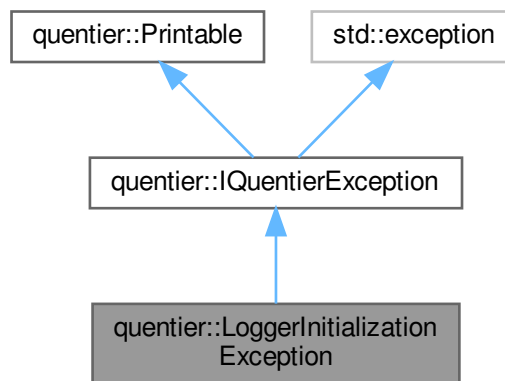
- **LocalStorageManagerAsync** (const Account &account, LocalStorageManager::StartupOptions options={}, QObject \*parent=nullptr)
- void **setUseCache** (const bool useCache)
- const LocalStorageCacheManager \* **localStorageCacheManager** () const
- bool **installCacheExpiryFunction** (const ILocalStorageCacheExpiryChecker &checker)
- const LocalStorageManager \* **localStorageManager** () const
- LocalStorageManager \* **localStorageManager** ()

## 5.43 quantier::LoggerInitializationException Class Reference

Inheritance diagram for quantier::LoggerInitializationException:



Collaboration diagram for quantier::LoggerInitializationException:



#### Public Member Functions

- **LoggerInitializationException** (const [ErrorString](#) &message)

#### Public Member Functions inherited from [quantier::IQuantierException](#)

- **IQuantierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

#### Public Member Functions inherited from [quantier::Printable](#)

- virtual const **QString toString** () const

#### Protected Member Functions

- const **QString exceptionDisplayName** () const override

#### Protected Member Functions inherited from [quantier::IQuantierException](#)

- **IQuantierException** (const [IQuantierException](#) &other)
- **IQuantierException & operator=** (const [IQuantierException](#) &other)

#### Protected Member Functions inherited from [quantier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable & operator=** (const [Printable](#) &other)

### 5.43.1 Member Function Documentation

#### 5.43.1.1 exceptionDisplayName()

```
const QString quotientier::LoggerInitializationException::exceptionDisplayName ( ) const [override],
[protected], [virtual]
```

Implements [quotientier::IQuentierException](#).

## 5.44 quotientier::LRUCache< Key, Value, Allocator > Class Template Reference

### Public Types

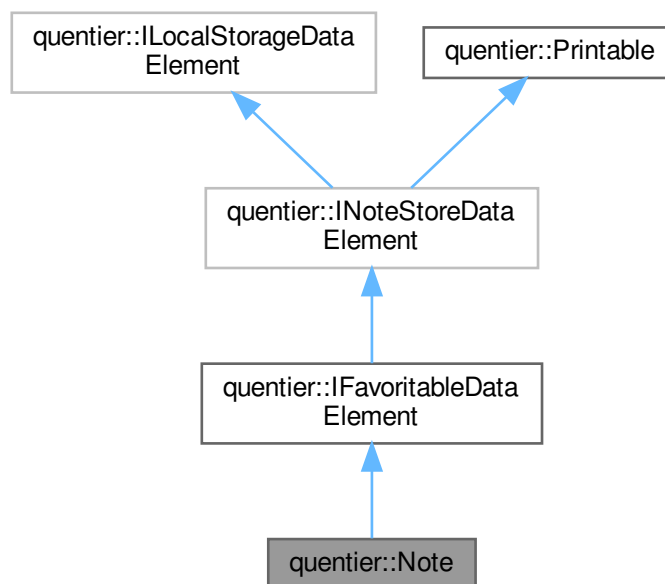
- using **key\_type** = Key
- using **mapped\_type** = Value
- using **allocator\_type** = Allocator
- using **value\_type** = std::pair< key\_type, mapped\_type >
- using **container\_type** = std::list< value\_type, allocator\_type >
- using **size\_type** = typename container\_type::size\_type
- using **difference\_type** = typename container\_type::difference\_type
- using **iterator** = typename container\_type::iterator
- using **const\_iterator** = typename container\_type::const\_iterator
- using **reverse\_iterator** = std::reverse\_iterator< iterator >
- using **const\_reverse\_iterator** = std::reverse\_iterator< const\_iterator >
- using **reference** = value\_type &
- using **const\_reference** = const value\_type &
- using **pointer** = typename std::allocator\_traits< allocator\_type >::pointer
- using **const\_pointer** = typename std::allocator\_traits< allocator\_type >::const\_pointer

### Public Member Functions

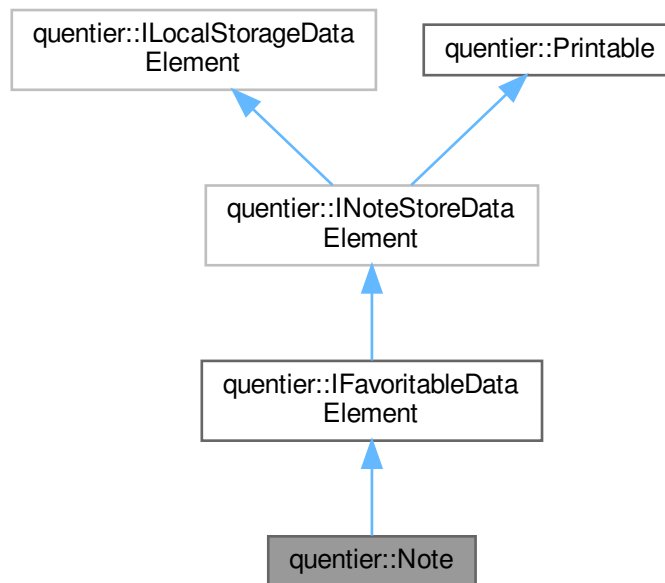
- **LRUCache** (const size\_t maxSize=100)
- iterator **begin** ()
- const\_iterator **begin** () const
- reverse\_iterator **rbegin** ()
- const\_reverse\_iterator **rbegin** () const
- iterator **end** ()
- const\_iterator **end** () const
- reverse\_iterator **rend** ()
- const\_reverse\_iterator **rend** () const
- bool **empty** () const
- size\_t **size** () const
- size\_t **max\_size** () const
- void **clear** ()
- void **put** (const key\_type &key, const mapped\_type &value)
- const mapped\_type \* **get** (const key\_type &key) const
- bool **exists** (const key\_type &key)
- bool **remove** (const key\_type &key)
- void **setMaxSize** (const size\_t maxSize)

## 5.45 quantier::Note Class Reference

Inheritance diagram for quantier::Note:



Collaboration diagram for `quentier::Note`:



### Public Member Functions

- **Note** (const [Note](#) &other)
- **Note** ([Note](#) &&other)
- [Note](#) & **operator=** (const [Note](#) &other)
- [Note](#) & **operator=** ([Note](#) &&other)
- **Note** (const qevercloud::Note &other)
- [Note](#) & **operator=** (const qevercloud::Note &other)
- bool **operator==** (const [Note](#) &other) const
- bool **operator!=** (const [Note](#) &other) const
- const qevercloud::Note & **qevercloudNote** () const
- qevercloud::Note & **qevercloudNote** ()
- virtual bool [hasGuid](#) () const override
- virtual const QString & [guid](#) () const override
- virtual void [setGuid](#) (const QString &guid) override
- virtual bool [hasUpdateSequenceNumber](#) () const override
- virtual qint32 [updateSequenceNumber](#) () const override
- virtual void [setUpdateSequenceNumber](#) (const qint32 usn) override
- virtual void [clear](#) () override
- virtual bool [checkParameters](#) ([ErrorString](#) &errorDescription) const override
- bool **hasTitle** () const
- const QString & **title** () const
- void **setTitle** (const QString &title)
- bool **hasContent** () const
- const QString & **content** () const
- void **setContent** (const QString &content)
- bool **hasContentHash** () const

- const QByteArray & **contentHash** () const
- void **setContentHash** (const QByteArray &contentHash)
- bool **hasContentLength** () const
- qint32 **contentLength** () const
- void **setContentLength** (const qint32 length)
- bool **hasCreationTimestamp** () const
- qint64 **creationTimestamp** () const
- void **setCreationTimestamp** (const qint64 timestamp)
- bool **hasModificationTimestamp** () const
- qint64 **modificationTimestamp** () const
- void **setModificationTimestamp** (const qint64 timestamp)
- bool **hasDeletionTimestamp** () const
- qint64 **deletionTimestamp** () const
- void **setDeletionTimestamp** (const qint64 timestamp)
- bool **hasActive** () const
- bool **active** () const
- void **setActive** (const bool active)
- bool **hasNotebookGuid** () const
- const QString & **notebookGuid** () const
- void **setNotebookGuid** (const QString &guid)
- bool **hasNotebookLocalUid** () const
- const QString & **notebookLocalUid** () const
- void **setNotebookLocalUid** (const QString &notebookLocalUid)
- bool **hasTagGuids** () const
- const QStringList & **tagGuids** () const
- void **setTagGuids** (const QStringList &guids)
- void **addTagGuid** (const QString &guid)
- void **removeTagGuid** (const QString &guid)
- bool **hasTagLocalUids** () const
- const QStringList & **tagLocalUids** () const
- void **setTagLocalUids** (const QStringList &localUids)
- void **addTagLocalUid** (const QString &localUid)
- void **removeTagLocalUid** (const QString &localUid)
- bool **hasResources** () const
- int **numResources** () const
- QList< [Resource](#) > **resources** () const
- void **setResources** (const QList< [Resource](#) > &resources)
- void **addResource** (const [Resource](#) &resource)
- bool **updateResource** (const [Resource](#) &resource)
- bool **removeResource** (const [Resource](#) &resource)
- bool **hasNoteAttributes** () const
- const qevercloud::NoteAttributes & **noteAttributes** () const
- qevercloud::NoteAttributes & **noteAttributes** ()
- void **clearNoteAttributes** ()
- bool **hasSharedNotes** () const
- QList< [SharedNote](#) > **sharedNotes** () const
- void **setSharedNotes** (const QList< [SharedNote](#) > &sharedNotes)
- void **addSharedNote** (const [SharedNote](#) &sharedNote)
- bool **updateSharedNote** (const [SharedNote](#) &sharedNote)
- bool **removeSharedNote** (const [SharedNote](#) &sharedNote)
- bool **hasNoteRestrictions** () const
- const qevercloud::NoteRestrictions & **noteRestrictions** () const
- qevercloud::NoteRestrictions & **noteRestrictions** ()
- void **setNoteRestrictions** (qevercloud::NoteRestrictions &&restrictions)
- bool **hasNoteLimits** () const

- const qevercloud::NoteLimits & **noteLimits** () const
- qevercloud::NoteLimits & **noteLimits** ()
- void **setNoteLimits** (qevercloud::NoteLimits &&limits)
- QByteArray **thumbnailData** () const
- void **setThumbnailData** (const QByteArray &thumbnailData)
- bool **isInkNote** () const
- QString **plainText** (ErrorString \*pErrorMessage=nullptr) const
- QStringList **listOfWords** (ErrorString \*pErrorMessage=nullptr) const
- std::pair< QString, QStringList > **plainTextAndListOfWords** (ErrorString \*pErrorMessage=nullptr) const
- bool **containsCheckedTodo** () const
- bool **containsUncheckedTodo** () const
- bool **containsTodo** () const
- bool **containsEncryption** () const
- virtual QTextStream & **print** (QTextStream &strm) const override

### Public Member Functions inherited from [quentier::IFavoritableDataElement](#)

- virtual bool **isFavorited** () const =0
- virtual void **setFavorited** (const bool favorited)=0

### Public Member Functions inherited from [quentier::INoteStoreDataElement](#)

- virtual bool **isDirty** () const =0
- virtual void **setDirty** (const bool dirty)=0
- virtual bool **isLocal** () const =0
- virtual void **setLocal** (const bool local)=0

### Public Member Functions inherited from [quentier::ILocalStorageDataElement](#)

- virtual const QString **localUuid** () const =0
- virtual void **setLocalUuid** (const QString &guid)=0
- virtual void **unsetLocalUuid** ()=0

### Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

### Static Public Member Functions

- static bool **validateTitle** (const QString &title, ErrorString \*pErrorDescription=nullptr)

### Additional Inherited Members

### Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const Printable &other)
- **Printable** & **operator=** (const Printable &other)

## 5.45.1 Member Function Documentation

### 5.45.1.1 checkParameters()

```
virtual bool quantier::Note::checkParameters (
    ErrorString & errorDescription ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.45.1.2 clear()

```
virtual void quantier::Note::clear ( ) [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.45.1.3 guid()

```
virtual const QString & quantier::Note::guid ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.45.1.4 hasGuid()

```
virtual bool quantier::Note::hasGuid ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.45.1.5 hasUpdateSequenceNumber()

```
virtual bool quantier::Note::hasUpdateSequenceNumber ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.45.1.6 print()

```
virtual QTextStream & quantier::Note::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quantier::Printable](#).

### 5.45.1.7 setGuid()

```
virtual void quantier::Note::setGuid (
    const QString & guid ) [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

#### 5.45.1.8 setUpdateSequenceNumber()

```
virtual void quantier::Note::setUpdateSequenceNumber (
    const qint32 usn ) [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

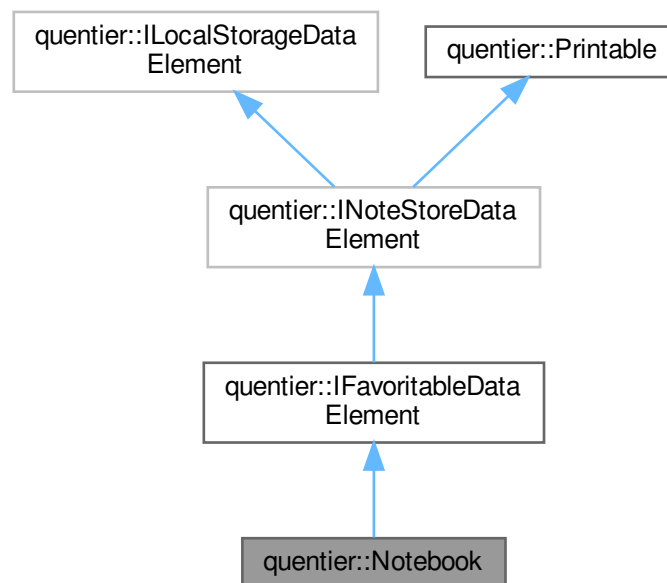
#### 5.45.1.9 updateSequenceNumber()

```
virtual qint32 quantier::Note::updateSequenceNumber ( ) const [override], [virtual]
```

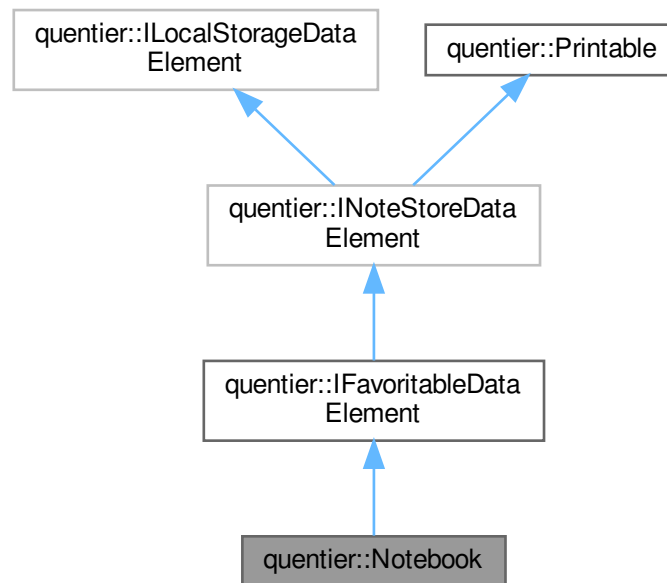
Implements [quantier::INoteStoreDataElement](#).

## 5.46 quantier::Notebook Class Reference

Inheritance diagram for quantier::Notebook:



Collaboration diagram for quantier::Notebook:



### Public Member Functions

- **Notebook** (const [Notebook](#) &other)
- **Notebook** ([Notebook](#) &&other)
- [Notebook](#) & **operator=** (const [Notebook](#) &other)
- [Notebook](#) & **operator=** ([Notebook](#) &&other)
- **Notebook** (const qevercloud::Notebook &other)
- **Notebook** (qevercloud::Notebook &&other)
- [Notebook](#) & **operator=** (const qevercloud::Notebook &other)
- [Notebook](#) & **operator=** (qevercloud::Notebook &&other)
- bool **operator==** (const [Notebook](#) &other) const
- bool **operator!=** (const [Notebook](#) &other) const
- const qevercloud::Notebook & **qevercloudNotebook** () const
- qevercloud::Notebook & **qevercloudNotebook** ()
- virtual void **clear** () override
- virtual bool **hasGuid** () const override
- virtual const QString & **guid** () const override
- virtual void **setGuid** (const QString &guid) override
- virtual bool **hasUpdateSequenceNumber** () const override
- virtual qint32 **updateSequenceNumber** () const override
- virtual void **setUpdateSequenceNumber** (const qint32 usn) override
- virtual bool **checkParameters** ([ErrorString](#) &errorDescription) const override
- bool **hasName** () const
- const QString & **name** () const
- void **setName** (const QString &name)
- bool **isDefaultNotebook** () const
- void **setDefaultNotebook** (const bool defaultNotebook)

- bool **hasLinkedNotebookGuid** () const
- const QString & **linkedNotebookGuid** () const
- void **setLinkedNotebookGuid** (const QString &linkedNotebookGuid)
- bool **hasCreationTimestamp** () const
- qint64 **creationTimestamp** () const
- void **setCreationTimestamp** (const qint64 timestamp)
- bool **hasModificationTimestamp** () const
- qint64 **modificationTimestamp** () const
- void **setModificationTimestamp** (const qint64 timestamp)
- bool **hasPublishingUri** () const
- const QString & **publishingUri** () const
- void **setPublishingUri** (const QString &uri)
- bool **hasPublishingOrder** () const
- qint8 **publishingOrder** () const
- void **setPublishingOrder** (const qint8 order)
- bool **hasPublishingAscending** () const
- bool **isPublishingAscending** () const
- void **setPublishingAscending** (const bool ascending)
- bool **hasPublishingPublicDescription** () const
- const QString & **publishingPublicDescription** () const
- void **setPublishingPublicDescription** (const QString &publishingPublicDescription)
- bool **hasPublished** () const
- bool **isPublished** () const
- void **setPublished** (const bool published)
- bool **hasStack** () const
- const QString & **stack** () const
- void **setStack** (const QString &stack)
- bool **hasSharedNotebooks** ()
- QList< [SharedNotebook](#) > **sharedNotebooks** () const
- void **setSharedNotebooks** (QList< qevercloud::SharedNotebook > sharedNotebooks)
- void **setSharedNotebooks** (QList< [SharedNotebook](#) > &&notebooks)
- void **addSharedNotebook** (const [SharedNotebook](#) &sharedNotebook)
- void **removeSharedNotebook** (const [SharedNotebook](#) &sharedNotebook)
- bool **hasBusinessNotebookDescription** () const
- const QString & **businessNotebookDescription** () const
- void **setBusinessNotebookDescription** (const QString &businessNotebookDescription)
- bool **hasBusinessNotebookPrivilegeLevel** () const
- qint8 **businessNotebookPrivilegeLevel** () const
- void **setBusinessNotebookPrivilegeLevel** (const qint8 privilegeLevel)
- bool **hasBusinessNotebookRecommended** () const
- bool **isBusinessNotebookRecommended** () const
- void **setBusinessNotebookRecommended** (const bool recommended)
- bool **hasContact** () const
- const [User](#) **contact** () const
- void **setContact** (const [User](#) &contact)
- bool **isLastUsed** () const
- void **setLastUsed** (const bool lastUsed)
- bool **canReadNotes** () const
- void **setCanReadNotes** (const bool canReadNotes)
- bool **canCreateNotes** () const
- void **setCanCreateNotes** (const bool canCreateNotes)
- bool **canUpdateNotes** () const
- void **setCanUpdateNotes** (const bool canUpdateNotes)
- bool **canExpungeNotes** () const
- void **setCanExpungeNotes** (const bool canExpungeNotes)

- bool **canShareNotes** () const
- void **setCanShareNotes** (const bool canShareNotes)
- bool **canEmailNotes** () const
- void **setCanEmailNotes** (const bool canEmailNotes)
- bool **canSendMessageToRecipients** () const
- void **setCanSendMessageToRecipients** (const bool canSendMessageToRecipients)
- bool **canUpdateNotebook** () const
- void **setCanUpdateNotebook** (const bool canUpdateNotebook)
- bool **canExpungeNotebook** () const
- void **setCanExpungeNotebook** (const bool canExpungeNotebook)
- bool **canSetDefaultNotebook** () const
- void **setCanSetDefaultNotebook** (const bool canSetDefaultNotebook)
- bool **canSetNotebookStack** () const
- void **setCanSetNotebookStack** (const bool canSetNotebookStack)
- bool **canPublishToPublic** () const
- void **setCanPublishToPublic** (const bool canPublishToPublic)
- bool **canPublishToBusinessLibrary** () const
- void **setCanPublishToBusinessLibrary** (const bool canPublishToBusinessLibrary)
- bool **canCreateTags** () const
- void **setCanCreateTags** (const bool canCreateTags)
- bool **canUpdateTags** () const
- void **setCanUpdateTags** (const bool canUpdateTags)
- bool **canExpungeTags** () const
- void **setCanExpungeTags** (const bool canExpungeTags)
- bool **canSetParentTag** () const
- void **setCanSetParentTag** (const bool canSetParentTag)
- bool **canCreateSharedNotebooks** () const
- void **setCanCreateSharedNotebooks** (const bool canCreateSharedNotebooks)
- bool **canShareNotesWithBusiness** () const
- void **setCanShareNotesWithBusiness** (const bool canShareNotesWithBusiness)
- bool **canRenameNotebook** () const
- void **setCanRenameNotebook** (const bool canRenameNotebook)
- bool **hasUpdateWhichSharedNotebookRestrictions** () const
- qint8 **updateWhichSharedNotebookRestrictions** () const
- void **setUpdateWhichSharedNotebookRestrictions** (const qint8 which)
- bool **hasExpungeWhichSharedNotebookRestrictions** () const
- qint8 **expungeWhichSharedNotebookRestrictions** () const
- void **setExpungeWhichSharedNotebookRestrictions** (const qint8 which)
- bool **hasRestrictions** () const
- const qevercloud::NotebookRestrictions & **restrictions** () const
- void **setNotebookRestrictions** (qevercloud::NotebookRestrictions &&restrictions)
- bool **hasRecipientReminderNotifyEmail** () const
- bool **recipientReminderNotifyEmail** () const
- void **setRecipientReminderNotifyEmail** (const bool notifyEmail)
- bool **hasRecipientReminderNotifyInApp** () const
- bool **recipientReminderNotifyInApp** () const
- void **setRecipientReminderNotifyInApp** (const bool notifyInApp)
- bool **hasRecipientInMyList** () const
- bool **recipientInMyList** () const
- void **setRecipientInMyList** (const bool inMyList)
- bool **hasRecipientStack** () const
- const QString & **recipientStack** () const
- void **setRecipientStack** (const QString &recipientString)
- bool **hasRecipientSettings** () const
- const qevercloud::NotebookRecipientSettings & **recipientSettings** () const
- void **setNotebookRecipientSettings** (qevercloud::NotebookRecipientSettings &&settings)
- virtual QTextStream & **print** (QTextStream &strm) const override

### Public Member Functions inherited from [quentier::IFavoritableDataElement](#)

- virtual bool **isFavorited** () const =0
- virtual void **setFavorited** (const bool favored)=0

### Public Member Functions inherited from [quentier::INoteStoreDataElement](#)

- virtual bool **isDirty** () const =0
- virtual void **setDirty** (const bool dirty)=0
- virtual bool **isLocal** () const =0
- virtual void **setLocal** (const bool local)=0

### Public Member Functions inherited from [quentier::ILocalStorageDataElement](#)

- virtual const QString **localUid** () const =0
- virtual void **setLocalUid** (const QString &guid)=0
- virtual void **unsetLocalUid** ()=0

### Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

### Static Public Member Functions

- static bool **validateName** (const QString &name, [ErrorString](#) \*pErrorDescription=nullptr)

### Additional Inherited Members

### Protected Member Functions inherited from [quentier::Printable](#)

- [Printable](#) (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

## 5.46.1 Member Function Documentation

### 5.46.1.1 checkParameters()

```
virtual bool quentier::Notebook::checkParameters (
    ErrorString & errorDescription ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

### 5.46.1.2 clear()

```
virtual void quentier::Notebook::clear ( ) [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

#### 5.46.1.3 guid()

```
virtual const QString & quantier::Notebook::guid ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

#### 5.46.1.4 hasGuid()

```
virtual bool quantier::Notebook::hasGuid ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

#### 5.46.1.5 hasUpdateSequenceNumber()

```
virtual bool quantier::Notebook::hasUpdateSequenceNumber ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

#### 5.46.1.6 print()

```
virtual QTextStream & quantier::Notebook::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quantier::Printable](#).

#### 5.46.1.7 setGuid()

```
virtual void quantier::Notebook::setGuid (
    const QString & guid ) [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

#### 5.46.1.8 setUpdateSequenceNumber()

```
virtual void quantier::Notebook::setUpdateSequenceNumber (
    const quint32 usn ) [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

#### 5.46.1.9 updateSequenceNumber()

```
virtual quint32 quantier::Notebook::updateSequenceNumber ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

## 5.47 quantier::ENMLConverter::NoteContentToHtmlExtraData Struct Reference

### Public Attributes

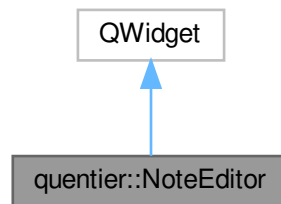
- quint64 `m_numEnToDoNodes` = 0
- quint64 `m_numHyperlinkNodes` = 0
- quint64 `m_numEnCryptNodes` = 0
- quint64 `m_numEnDecryptedNodes` = 0

## 5.48 quantier::NoteEditor Class Reference

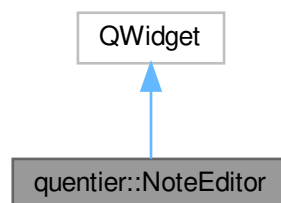
The [NoteEditor](#) class is a widget encapsulating all the functionality necessary for showing and editing notes.

```
#include <NoteEditor.h>
```

Inheritance diagram for quantier::NoteEditor:



Collaboration diagram for quantier::NoteEditor:



## Public Slots

- void [convertToNote](#) ()
- void [saveNoteToLocalStorage](#) ()
- void [setNoteTitle](#) (const QString &noteTitle)
- void [setTagIds](#) (const QStringList &tagLocalUids, const QStringList &tagGuids)
- void **undo** ()
- void **redo** ()
- void **cut** ()
- void **copy** ()
- void **paste** ()
- void **pasteUnformatted** ()
- void **selectAll** ()
- void **formatSelectionAsSourceCode** ()
- void **fontMenu** ()
- void **textBold** ()
- void **textItalic** ()
- void **textUnderline** ()
- void **textStrikethrough** ()
- void **textHighlight** ()
- void **alignLeft** ()
- void **alignCenter** ()
- void **alignRight** ()
- void **alignFull** ()
- void **findNext** (const QString &text, const bool matchCase) const
- void **findPrevious** (const QString &text, const bool matchCase) const
- void **replace** (const QString &textToReplace, const QString &replacementText, const bool matchCase)
- void **replaceAll** (const QString &textToReplace, const QString &replacementText, const bool matchCase)
- void **insertToDoCheckbox** ()
- void **insertInAppNoteLink** (const QString &userId, const QString &shardId, const QString &noteGuid, const QString &linkText)
- void **setSpellcheck** (const bool enabled)
- void **setFont** (const QFont &font)
- void **setFontHeight** (const int height)
- void **setFontColor** (const QColor &color)
- void **setBackgroundColor** (const QColor &color)
- void [setDefaultPalette](#) (const QPalette &pal)
- void [setDefaultFont](#) (const QFont &font)
- void **insertHorizontalLine** ()
- void **increaseFontSize** ()
- void **decreaseFontSize** ()
- void **increaseIndentation** ()
- void **decreaseIndentation** ()
- void **insertBulletedList** ()
- void **insertNumberedList** ()
- void **insertTableDialog** ()
- void **insertFixedWidthTable** (const int rows, const int columns, const int widthInPixels)
- void **insertRelativeWidthTable** (const int rows, const int columns, const double relativeWidth)
- void **insertTableRow** ()
- void **insertTableColumn** ()
- void **removeTableRow** ()
- void **removeTableColumn** ()
- void **addAttachmentDialog** ()
- void **saveAttachmentDialog** (const QByteArray &resourceHash)
- void **saveAttachmentUnderCursor** ()

- void **openAttachment** (const QByteArray &resourceHash)
- void **openAttachmentUnderCursor** ()
- void **copyAttachment** (const QByteArray &resourceHash)
- void **copyAttachmentUnderCursor** ()
- void **encryptSelectedText** ()
- void **decryptEncryptedTextUnderCursor** ()
- void **editHyperlinkDialog** ()
- void **copyHyperlink** ()
- void **removeHyperlink** ()
- void **onNoteLoadCancelled** ()

## Signals

- void **contentChanged** ()  
*contentChanged signal is emitted when the note's content (text) gets modified via manual editing (i.e. not any action like paste or cut)*
- void **noteAndNotebookFoundInLocalStorage** ([Note](#) note, [Notebook](#) notebook)  
*noteAndNotebookFoundInLocalStorage signal is emitted when note and its corresponding notebook were found within the local storage right before the note editor starts to load the note into the editor*
- void **noteNotFound** (QString noteLocalUid)  
*noteNotFound signal is emitted when the note could not be found within the local storage by the provided local uid*
- void **noteDeleted** (QString noteLocalUid)  
*noteDeleted signal is emitted when the note displayed within the note editor is deleted. The note editor stops displaying the note in this case shortly after emitting this signal*
- void **noteModified** ()  
*noteModified signal is emitted when the note's content within the editor gets modified via some way - either via manual editing or via some action (like paste or cut)*
- void **notifyError** ([ErrorString](#) error)  
*notifyError signal is emitted when [NoteEditor](#) encounters some problem worth letting the user to know about*
- void **inAppNoteLinkClicked** (QString userId, QString shardId, QString noteGuid)  
*inAppNoteLinkClicked signal is emitted when the in-app note link is clicked within the note editor*
- void **inAppNoteLinkPasteRequested** (QString url, QString userId, QString shardId, QString noteGuid)
- void **convertedToNote** ([Note](#) note)
- void **cantConvertToNote** ([ErrorString](#) error)
- void **noteEditorHtmlUpdated** (QString html)
- void **currentNoteChanged** ([Note](#) note)
- void **spellCheckerNotReady** ()
- void **spellCheckerReady** ()
- void **noteLoaded** ()
- void **noteSavedToLocalStorage** (QString noteLocalUid)  
*noteSavedToLocalStorage signal is emitted when the note has been saved within the local storage. [NoteEditor](#) doesn't do this on its own unless it's explicitly asked to do this via invoking its saveNoteToLocalStorage slot*
- void **failedToSaveNoteToLocalStorage** ([ErrorString](#) errorDescription, QString noteLocalUid)  
*failedToSaveNoteToLocalStorage signal is emitted in case of failure to save the note to local storage*
- void **textBoldState** (bool state)
- void **textItalicState** (bool state)
- void **textUnderlineState** (bool state)
- void **textStrikethroughState** (bool state)
- void **textAlignLeftState** (bool state)
- void **textAlignCenterState** (bool state)
- void **textAlignRightState** (bool state)
- void **textAlignFullState** (bool state)
- void **textInsideOrderedListState** (bool state)

- void **textInsideUnorderedListState** (bool state)
- void **textInsideTableState** (bool state)
- void **textFontFamilyChanged** (QString fontFamily)
- void **textFontSizeChanged** (int fontSize)
- void **insertTableDialogRequested** ()

## Public Member Functions

- **NoteEditor** (QWidget \*parent=nullptr, Qt::WindowFlags flags={})
- void **initialize** (LocalStorageManagerAsync &localStorageManager, SpellChecker &spellChecker, const Account &account, QThread \*pBackgroundJobsThread=nullptr)
- INoteEditorBackend \* **backend** ()
- void **setBackend** (INoteEditorBackend \*backend)
- void **setAccount** (const Account &account)
- const QUndoStack \* **undoStack** () const
- void **setUndoStack** (QUndoStack \*pUndoStack)
- void **setInitialPageHtml** (const QString &html)
- void **setNoteNotFoundPageHtml** (const QString &html)
- void **setNoteDeletedPageHtml** (const QString &html)
- void **setNoteLoadingPageHtml** (const QString &html)
- QString **currentNoteLocalUid** () const
- void **setCurrentNoteLocalUid** (const QString &noteLocalUid)
- void **clear** ()
- bool **isModified** () const
- bool **isEditorPageModified** () const
- bool **isNoteLoaded** () const
- qint64 **idleTime** () const
- void **setFocus** ()
- QString **selectedText** () const
- bool **hasSelection** () const
- bool **spellCheckEnabled** () const
- bool **print** (QPrinter &printer, ErrorString &errorDescription)
- bool **exportToPdf** (const QString &absoluteFilePath, ErrorString &errorDescription)
- bool **exportToEnex** (const QStringList &tagNames, QString &enex, ErrorString &errorDescription)
- QPalette **defaultPalette** () const
- const QFont \* **defaultFont** () const

## Protected Member Functions

- virtual void **dragMoveEvent** (QDragMoveEvent \*pEvent) override
- virtual void **dropEvent** (QDropEvent \*pEvent) override

### 5.48.1 Detailed Description

The `NoteEditor` class is a widget encapsulating all the functionality necessary for showing and editing notes.

## 5.48.2 Member Function Documentation

### 5.48.2.1 backend()

```
INoteEditorBackend * quantier::NoteEditor::backend ( )
```

#### Returns

the pointer to the note editor's backend

### 5.48.2.2 clear()

```
void quantier::NoteEditor::clear ( )
```

Clear the contents of the note editor

### 5.48.2.3 convertToNote

```
void quantier::NoteEditor::convertToNote ( ) [slot]
```

Invoke this slot to launch the asynchronous procedure of converting the current contents of the note editor to note; the convertedToNote signal would be emitted in response when the conversion is done

### 5.48.2.4 currentNoteLocalUid()

```
QString quantier::NoteEditor::currentNoteLocalUid ( ) const
```

Get the local uid of the note currently set to the note editor

### 5.48.2.5 defaultFont()

```
const QFont * quantier::NoteEditor::defaultFont ( ) const
```

#### Returns

pointer to the default font used by the note editor; if no such font was set to the editor previously, returns null pointer

### 5.48.2.6 defaultPalette()

```
QPalette quantier::NoteEditor::defaultPalette ( ) const
```

#### Returns

palette containing default colors used by the editor; the palette is composed of colors from note editor widget's native palette but some of them might be overridden by colors from the palette specified previously via set↔DefaultPalette method: those colors from the specified palette which were valid

### 5.48.2.7 idleTime()

```
qint64 quantier::NoteEditor::idleTime ( ) const
```

#### Returns

the number of milliseconds since the last user's interaction with the note editor or -1 if there was no interaction or if no note is loaded at the moment

### 5.48.2.8 inAppNoteLinkPasteRequested

```
void quantier::NoteEditor::inAppNoteLinkPasteRequested (
    QString url,
    QString userId,
    QString shardId,
    QString noteGuid ) [signal]
```

inAppNoteLinkPasteRequested signal is emitted when the note editor detects the attempt to paste the in-app note link into the note editor; the link would not be inserted right away, instead this signal would be emitted. Whatever party managing the note editor is expected to connect some slot to this signal and provide the optionally amended link information to the note editor by sending the signal connected to its insertInAppNoteLink slot - this slot accepts both the URL of the link and the link text and performs the actual link insertion into the note. If the link text is empty, the URL itself is used as the link text.

### 5.48.2.9 initialize()

```
void quantier::NoteEditor::initialize (
    LocalStorageManagerAsync & localStorageManager,
    SpellChecker & spellChecker,
    const Account & account,
    QThread * pBackgroundJobsThread = nullptr )
```

[NoteEditor](#) requires [LocalStorageManagerAsync](#), [SpellChecker](#) and [Account](#) for its work but due to the particularities of Qt's .ui files processing these can't be passed right inside the constructor, hence here's a special initialization method

#### Parameters

<i>localStorageManager</i>	The reference to <a href="#">LocalStorageManagerAsync</a> , to set up signal-slot connections with it
<i>spellChecker</i>	The spell checker to be used by note editor for, well, spell-checking
<i>account</i>	Currently active account
<i>pBackgroundJobsThread</i>	Pointer to the thread to be used for scheduling of background jobs of <a href="#">NoteEditor</a> ; if null, <a href="#">NoteEditor</a> 's background jobs would take place in GUI thread

### 5.48.2.10 isEditorPageModified()

```
bool quantier::NoteEditor::isEditorPageModified ( ) const
```

**Returns**

true if there's content within the editor not yet converted to note, false otherwise

**5.48.2.11 isModified()**

```
bool quantier::NoteEditor::isModified ( ) const
```

**Returns**

true if there's content within the editor not yet converted to note or not saved to local storage, false otherwise

**5.48.2.12 isNoteLoaded()**

```
bool quantier::NoteEditor::isNoteLoaded ( ) const
```

**Returns**

true if the note last set to the editor has been fully loaded already, false otherwise

**5.48.2.13 saveNoteToLocalStorage**

```
void quantier::NoteEditor::saveNoteToLocalStorage ( ) [slot]
```

Invoke this slot to launch the asynchronous procedure of saving the modified current note back to the local storage. If no note is set to the editor or if the note is not modified, no action would be performed. Otherwise noteSaved↵ToLocalStorage signal would be emitted in case of successful saving or failedToSaveNoteToLocalStorage would be emitted otherwise

**5.48.2.14 setAccount()**

```
void quantier::NoteEditor::setAccount (
    const Account & account )
```

Set the current account to the note editor

**5.48.2.15 setBackend()**

```
void quantier::NoteEditor::setBackend (
    INoteEditorBackend * backend )
```

This method can be used to set the backend to the note editor; the note editor has the default backend so this method is not obligatory to be called

**5.48.2.16 setCurrentNoteLocalUid()**

```
void quantier::NoteEditor::setCurrentNoteLocalUid (
    const QString & noteLocalUid )
```

Set note local uid to the note editor. The note is being searched for within the local storage, in case of no note being found noteNotFound signal is emitted. Otherwise note editor page starts loading.

## Parameters

<i>noteLocalUid</i>	The local uid of note
---------------------	-----------------------

**5.48.2.17 setDefaultFont**

```
void quantier::NoteEditor::setDefaultFont (
    const QFont & font ) [slot]
```

Sets the font which would be used by the editor by default

## Parameters

<i>font</i>	The font to be used by the editor by default
-------------	--

**5.48.2.18 setDefaultPalette**

```
void quantier::NoteEditor::setDefaultPalette (
    const QPalette & pal ) [slot]
```

Sets the palette with colors to be used by the editor. New colors are applied after the note is fully loaded. If no note is set to the editor, the palette is simply remembered for the next note to be loaded into it.

Colors within the palette and their usage:

1. WindowText - used as default font color
2. Base - used as default background color
3. HighlightedText - used as font color for selected text
4. Highlight - used as background color for selected text

## Parameters

<i>pal</i>	The palette to be set. Invalid colors from it are substituted by colors from widget's palette by the editor
------------	---

**5.48.2.19 setFocus()**

```
void quantier::NoteEditor::setFocus ( )
```

Sets the focus to the backend note editor widget

**5.48.2.20 setInitialPageHtml()**

```
void quantier::NoteEditor::setInitialPageHtml (
    const QString & html )
```

Set the html to be displayed when the note is not set to the editor

**5.48.2.21 setNoteDeletedPageHtml()**

```
void quentier::NoteEditor::setNoteDeletedPageHtml (
    const QString & html )
```

Set the html to be displayed when the note set to the editor was deleted from the local storage (either marked as deleted or deleted permanently i.e. expunged)

**5.48.2.22 setNoteLoadingPageHtml()**

```
void quentier::NoteEditor::setNoteLoadingPageHtml (
    const QString & html )
```

Set the html to be displayed when the note set to the editor is being loaded into it

**5.48.2.23 setNoteNotFoundPageHtml()**

```
void quentier::NoteEditor::setNoteNotFoundPageHtml (
    const QString & html )
```

Set the html to be displayed when the note attempted to be set to the editor was not found within the local storage

**5.48.2.24 setNoteTitle**

```
void quentier::NoteEditor::setNoteTitle (
    const QString & noteTitle ) [slot]
```

Invoke this slot to set the title to the note displayed via the note editor. The note editor itself doesn't manage the note title in any way so any external code using the note editor can set the title to the note editor's note which would be considered modified if the title is new and then eventually the note would be saved to local storage

**Parameters**

<i>noteTitle</i>	The title of the note
------------------	-----------------------

**5.48.2.25 setTagIds**

```
void quentier::NoteEditor::setTagIds (
    const QStringList & tagLocalUids,
    const QStringList & tagGuids ) [slot]
```

Invoke this slot to set tag local uids and/or tag guides to the note displayed via the note editor. The note editor itself doesn't manage the note tags in any way so any external code using the note editor can set the tag ids to the note editor's internal note which would be considered modified if the tag ids are new and then eventually the note would be saved to local storage

**Parameters**

<i>tagLocalUids</i>	The list of tag local uids for the note
<i>tagGuids</i>	The list of tag guides for the note

### 5.48.2.26 setUndoStack()

```
void quantier::NoteEditor::setUndoStack (
    QUndoStack * pUndoStack )
```

Set the undo stack for the note editor to use

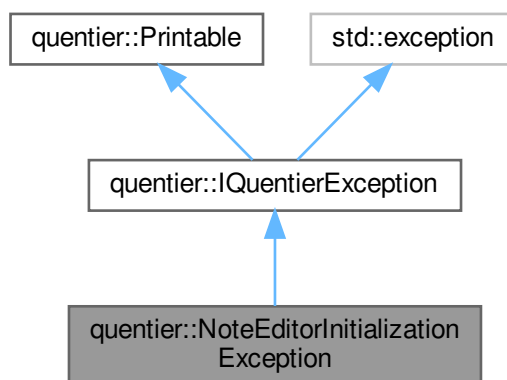
### 5.48.2.27 undoStack()

```
const QUndoStack * quantier::NoteEditor::undoStack ( ) const
```

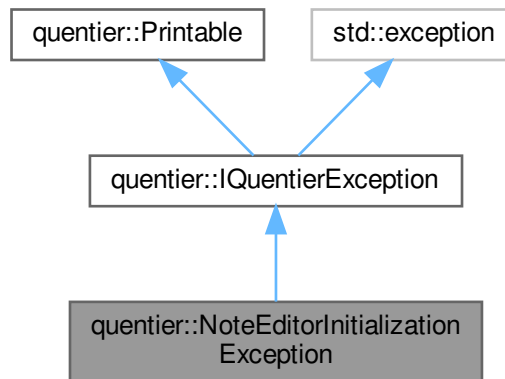
Get the undo stack serving to the note editor

## 5.49 quantier::NoteEditorInitializationException Class Reference

Inheritance diagram for quantier::NoteEditorInitializationException:



Collaboration diagram for `quentier::NoteEditorInitializationException`:



#### Public Member Functions

- **NoteEditorInitializationException** (const [ErrorString](#) &message)

#### Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

#### Public Member Functions inherited from [quentier::Printable](#)

- virtual const **QString toString** () const

#### Protected Member Functions

- virtual const **QString exceptionDisplayName** () const override

#### Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [IQuentierException](#) &other)
- **IQuentierException & operator=** (const [IQuentierException](#) &other)

#### Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable & operator=** (const [Printable](#) &other)

### 5.49.1 Member Function Documentation

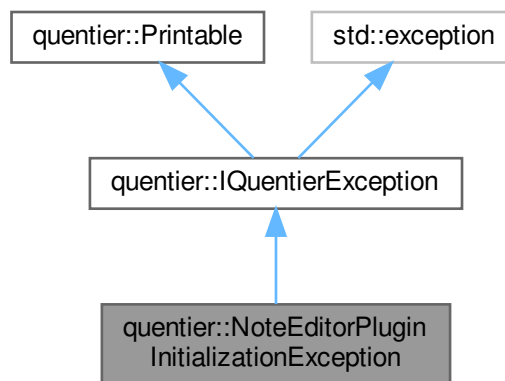
#### 5.49.1.1 exceptionDisplayName()

```
virtual const QString quantier::NoteEditorInitializationException::exceptionDisplayName ( )  
const [override], [protected], [virtual]
```

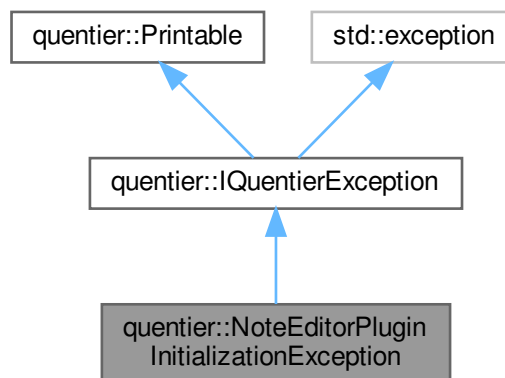
Implements [quantier::IQuantierException](#).

## 5.50 quantier::NoteEditorPluginInitializationException Class Reference

Inheritance diagram for quantier::NoteEditorPluginInitializationException:



Collaboration diagram for quantier::NoteEditorPluginInitializationException:



## Public Member Functions

- **NoteEditorPluginInitializationException** (const [ErrorString](#) &message)

## Public Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [ErrorString](#) &message)
- **QString localizedErrorMessage** () const
- **QString nonLocalizedErrorMessage** () const
- virtual const char \* **what** () const noexcept override
- virtual **QTextStream & print** (QTextStream &strm) const override

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const **QString toString** () const

## Protected Member Functions

- virtual const **QString exceptionDisplayName** () const override

## Protected Member Functions inherited from [quentier::IQuentierException](#)

- **IQuentierException** (const [IQuentierException](#) &other)
- [IQuentierException](#) & **operator=** (const [IQuentierException](#) &other)

## Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

## 5.50.1 Member Function Documentation

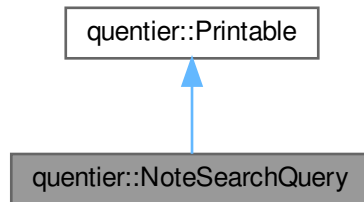
### 5.50.1.1 exceptionDisplayName()

```
virtual const QString quentier::NoteEditorPluginInitializationException::exceptionDisplayName  
( ) const [override], [protected], [virtual]
```

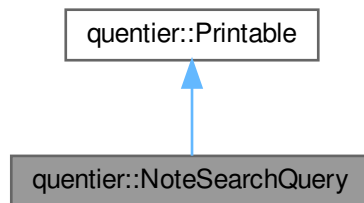
Implements [quentier::IQuentierException](#).

## 5.51 quantier::NoteSearchQuery Class Reference

Inheritance diagram for quantier::NoteSearchQuery:



Collaboration diagram for quantier::NoteSearchQuery:



### Public Member Functions

- **NoteSearchQuery** (const [NoteSearchQuery](#) &other)
- **NoteSearchQuery** ([NoteSearchQuery](#) &&other)
- [NoteSearchQuery](#) & **operator=** (const [NoteSearchQuery](#) &other)
- [NoteSearchQuery](#) & **operator=** ([NoteSearchQuery](#) &&other)
- bool **isEmpty** () const
- void **clear** ()
- const QString [queryString](#) () const
- bool **setQueryString** (const QString &[queryString](#), [ErrorString](#) &error)
- const QString [notebookModifier](#) () const
- bool **hasAnyModifier** () const
- const QStringList & **tagNames** () const
- const QStringList & **negatedTagNames** () const
- bool **hasAnyTag** () const
- bool **hasNegatedAnyTag** () const
- const QStringList & **titleNames** () const
- const QStringList & **negatedTitleNames** () const

- bool **hasAnyTitleName** () const
- bool **hasNegatedAnyTitleName** () const
- const QVector< qint64 > & **creationTimestamps** () const
- const QVector< qint64 > & **negatedCreationTimestamps** () const
- bool **hasAnyCreationTimestamp** () const
- bool **hasNegatedAnyCreationTimestamp** () const
- const QVector< qint64 > & **modificationTimestamps** () const
- const QVector< qint64 > & **negatedModificationTimestamps** () const
- bool **hasAnyModificationTimestamp** () const
- bool **hasNegatedAnyModificationTimestamp** () const
- const QStringList & **resourceMimeTypes** () const
- const QStringList & **negatedResourceMimeTypes** () const
- bool **hasAnyResourceMimeType** () const
- bool **hasNegatedAnyResourceMimeType** () const
- const QVector< qint64 > & **subjectDateTimestamps** () const
- const QVector< qint64 > & **negatedSubjectDateTimestamps** () const
- bool **hasAnySubjectDateTimestamp** () const
- bool **hasNegatedAnySubjectDateTimestamp** () const
- const QVector< double > & **latitudes** () const
- const QVector< double > & **negatedLatitudes** () const
- bool **hasAnyLatitude** () const
- bool **hasNegatedAnyLatitude** () const
- const QVector< double > & **longitudes** () const
- const QVector< double > & **negatedLongitudes** () const
- bool **hasAnyLongitude** () const
- bool **hasNegatedAnyLongitude** () const
- const QVector< double > & **altitudes** () const
- const QVector< double > & **negatedAltitudes** () const
- bool **hasAnyAltitude** () const
- bool **hasNegatedAnyAltitude** () const
- const QStringList & **authors** () const
- const QStringList & **negatedAuthors** () const
- bool **hasAnyAuthor** () const
- bool **hasNegatedAnyAuthor** () const
- const QStringList & **sources** () const
- const QStringList & **negatedSources** () const
- bool **hasAnySource** () const
- bool **hasNegatedAnySource** () const
- const QStringList & **sourceApplications** () const
- const QStringList & **negatedSourceApplications** () const
- bool **hasAnySourceApplication** () const
- bool **hasNegatedAnySourceApplication** () const
- const QStringList & **contentClasses** () const
- const QStringList & **negatedContentClasses** () const
- bool **hasAnyContentClass** () const
- bool **hasNegatedAnyContentClass** () const
- const QStringList & **placeNames** () const
- const QStringList & **negatedPlaceNames** () const
- bool **hasAnyPlaceName** () const
- bool **hasNegatedAnyPlaceName** () const
- const QStringList & **applicationData** () const
- const QStringList & **negatedApplicationData** () const
- bool **hasAnyApplicationData** () const
- bool **hasNegatedAnyApplicationData** () const
- const QVector< qint64 > & **reminderOrders** () const

- const QVector< quint64 > & **negatedReminderOrders** () const
- bool **hasAnyReminderOrder** () const
- bool **hasNegatedAnyReminderOrder** () const
- const QVector< quint64 > & **reminderTimes** () const
- const QVector< quint64 > & **negatedReminderTimes** () const
- bool **hasAnyReminderTime** () const
- bool **hasNegatedAnyReminderTime** () const
- const QVector< quint64 > & **reminderDoneTimes** () const
- const QVector< quint64 > & **negatedReminderDoneTimes** () const
- bool **hasAnyReminderDoneTime** () const
- bool **hasNegatedAnyReminderDoneTime** () const
- bool **hasUnfinishedToDo** () const
- bool **hasNegatedUnfinishedToDo** () const
- bool **hasFinishedToDo** () const
- bool **hasNegatedFinishedToDo** () const
- bool **hasAnyToDo** () const
- bool **hasNegatedAnyToDo** () const
- bool **hasEncryption** () const
- bool **hasNegatedEncryption** () const
- const QStringList & **contentSearchTerms** () const
- const QStringList & **negatedContentSearchTerms** () const
- bool **hasAnyContentSearchTerms** () const
- bool **isMatcheable** () const
- virtual QTextStream & **print** (QTextStream &strm) const override

## Public Member Functions inherited from [quantier::Printable](#)

- virtual const QString **toString** () const

## Additional Inherited Members

## Protected Member Functions inherited from [quantier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

### 5.51.1 Member Function Documentation

#### 5.51.1.1 notebookModifier()

```
const QString quantier::NoteSearchQuery::notebookModifier ( ) const
```

If query string has "notebook:<notebook name>" scope modifier, this method returns the name of the notebook, otherwise it returns empty string

#### 5.51.1.2 print()

```
virtual QTextStream & quantier::NoteSearchQuery::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quantier::Printable](#).

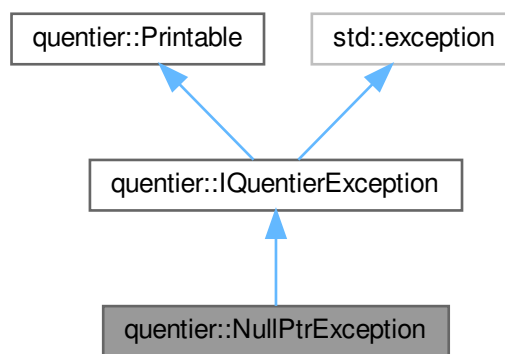
### 5.51.1.3 queryString()

```
const QString quentier::NoteSearchQuery::queryString ( ) const
```

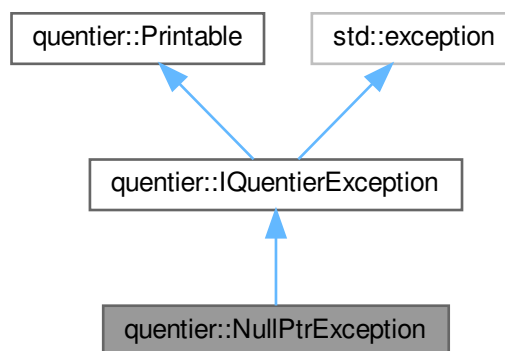
Returns the original non-parsed query string

## 5.52 quentier::NullPtrException Class Reference

Inheritance diagram for quentier::NullPtrException:



Collaboration diagram for quentier::NullPtrException:



### Public Member Functions

- **NullPtrException** (const [ErrorString](#) &message)

### Public Member Functions inherited from `quentier::IQuentierException`

- **IQuentierException** (const `ErrorString` &message)
- `QString localizedErrorMessage` () const
- `QString nonLocalizedErrorMessage` () const
- virtual const char \* **what** () const noexcept override
- virtual `QTextStream` & **print** (`QTextStream` &strm) const override

### Public Member Functions inherited from `quentier::Printable`

- virtual const `QString` **toString** () const

### Protected Member Functions

- virtual const `QString` **exceptionDisplayName** () const override

### Protected Member Functions inherited from `quentier::IQuentierException`

- **IQuentierException** (const `IQuentierException` &other)
- `IQuentierException` & **operator=** (const `IQuentierException` &other)

### Protected Member Functions inherited from `quentier::Printable`

- **Printable** (const `Printable` &other)
- `Printable` & **operator=** (const `Printable` &other)

## 5.52.1 Member Function Documentation

### 5.52.1.1 `exceptionDisplayName()`

```
virtual const QString quentier::NullPtrException::exceptionDisplayName ( ) const [override],  
[protected], [virtual]
```

Implements `quentier::IQuentierException`.

## 5.53 `quentier::ResourceRecognitionIndexItem::ObjectItem` Struct Reference

### Public Member Functions

- bool **operator==** (const `ObjectItem` &other) const

### Public Attributes

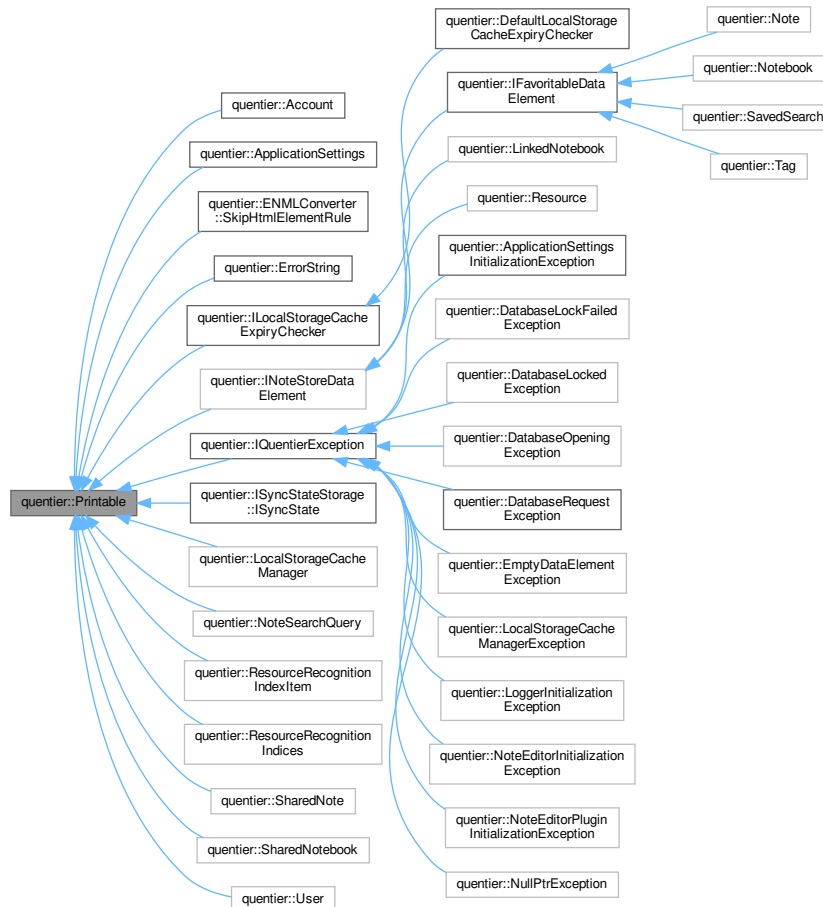
- `QString` **m\_objectType**
- int **m\_weight** = -1

## 5.54 quantier::Printable Class Reference

The [Printable](#) class is the interface for Quantier's internal classes which should be able to write themselves into QTextStream and/or convert to QString.

```
#include <Printable.h>
```

Inheritance diagram for quantier::Printable:



### Public Member Functions

- virtual QTextStream & [print](#) (QTextStream &strm) const =0
- virtual const QString [toString](#) () const

### Protected Member Functions

- [Printable](#) (const [Printable](#) &other)
- [Printable](#) & [operator=](#) (const [Printable](#) &other)

### Friends

- QUENTIER\_EXPORT QTextStream & **operator**<< (QTextStream &strm, const [Printable](#) &printable)
- QUENTIER\_EXPORT QDebug & **operator**<< (QDebug &debug, const [Printable](#) &printable)

## 5.54.1 Detailed Description

The [Printable](#) class is the interface for Quantier's internal classes which should be able to write themselves into QTextStream and/or convert to QString.

## 5.54.2 Member Function Documentation

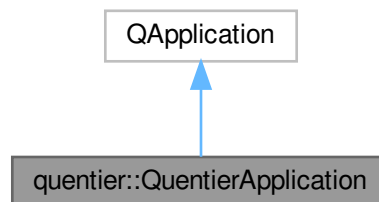
### 5.54.2.1 print()

```
virtual QTextStream & quantier::Printable::print (  
    QTextStream & strm ) const [pure virtual]
```

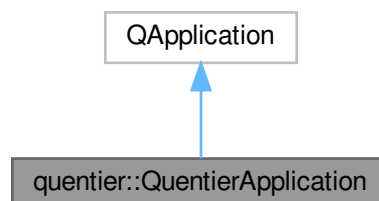
Implemented in [quantier::LocalStorageCacheExpiryChecker](#), and [quantier::DefaultLocalStorageCacheExpiryChecker](#).

## 5.55 quantier::QuantierApplication Class Reference

Inheritance diagram for quantier::QuantierApplication:



Collaboration diagram for quantier::QuantierApplication:



### Public Member Functions

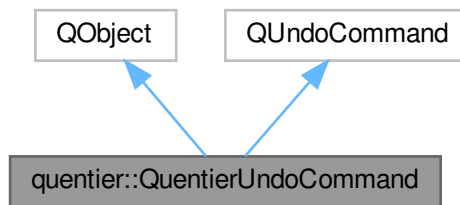
- **QuentierApplication** (int &argc, char \*argv[])
- virtual bool **notify** (QObject \*pObject, QEvent \*pEvent) override
- virtual bool **event** (QEvent \*pEvent) override

## 5.56 quentier::QuentierUndoCommand Class Reference

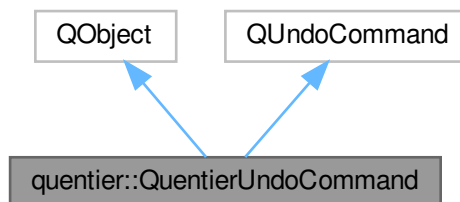
The [QuentierUndoCommand](#) class has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push QUndoCommand to QUndoStack, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that.

```
#include <QuentierUndoCommand.h>
```

Inheritance diagram for quentier::QuentierUndoCommand:



Collaboration diagram for quentier::QuentierUndoCommand:



### Signals

- void **notifyError** ([ErrorString](#) error)

**Public Member Functions**

- **QuantierUndoCommand** (QUndoCommand \*parent=nullptr)
- **QuantierUndoCommand** (const QString &text, QUndoCommand \*parent=nullptr)
- virtual void **undo** () override final
- virtual void **redo** () override final
- bool **onceUndoExecuted** () const

**Protected Member Functions**

- virtual void **undoImpl** ()=0
- virtual void **redoImpl** ()=0

**5.56.1 Detailed Description**

The [QuantierUndoCommand](#) class has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push QUndoCommand to QUndoStack, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that.

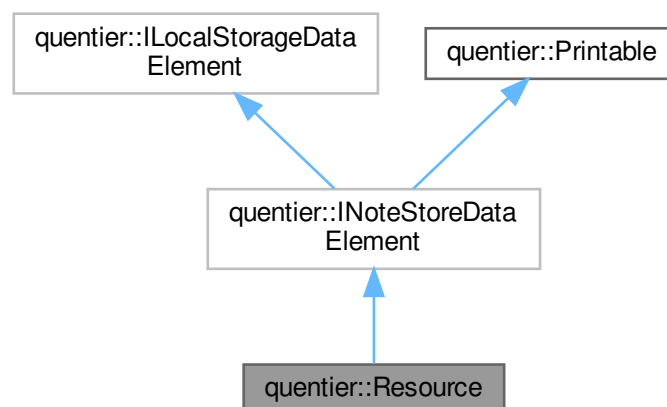
The rationale behind the current behaviour seems to be the compliance with "command pattern behaviour" when you create the command to execute the action instead of just executing it immediately. This design is enforced by Qt's undo/redo framework, there's no option to choose not to call "redo" when pushing to the stack.

One thing which this design fails to see is the fact that the command may be already executed externally by the moment the QUndoCommand can be created. Suppose we can get the information about how to undo (and then again redo) that command. We create the corresponding QUndoCommand, set up the stuff for its undo/redo methods and push it to QUndoStack for future use... But at the same time QUndoStack calls "redo" method of the command. Really not the behaviour you'd like to have.

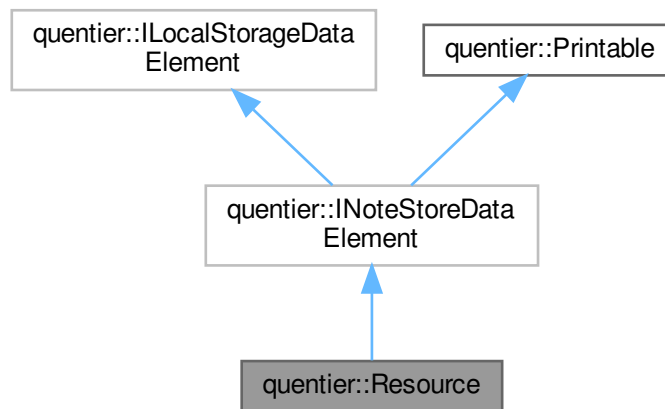
[QuantierUndoCommand](#) is also QObject, it is for error reporting via notifyError signal

**5.57 quantier::Resource Class Reference**

Inheritance diagram for quantier::Resource:



Collaboration diagram for `quentier::Resource`:



### Public Member Functions

- **Resource** (const [Resource](#) &other)
- **Resource** ([Resource](#) &&other)
- **Resource** (const `qevercloud::Resource` &resource)
- [Resource](#) & **operator=** (const [Resource](#) &other)
- [Resource](#) & **operator=** ([Resource](#) &&other)
- bool **operator==** (const [Resource](#) &other) const
- bool **operator!=** (const [Resource](#) &other) const
- const `qevercloud::Resource` & **qevercloudResource** () const
- `qevercloud::Resource` & **qevercloudResource** ()
- virtual void **clear** () override
- virtual bool **hasGuid** () const override
- virtual const `QString` & **guid** () const override
- virtual void **setGuid** (const `QString` &guid) override
- virtual bool **hasUpdateSequenceNumber** () const override
- virtual `qint32` **updateSequenceNumber** () const override
- virtual void **setUpdateSequenceNumber** (const `qint32` updateSequenceNumber) override
- virtual bool **checkParameters** (`QString` &errorDescription) const override
- `QString` **displayName** () const
- void **setDisplayName** (const `QString` &displayName)
- `QString` **preferredFileSuffix** () const
- int **indexInNote** () const
- void **setIndexInNote** (const int index)
- bool **hasNoteGuid** () const
- const `QString` & **noteGuid** () const
- void **setNoteGuid** (const `QString` &guid)
- bool **hasNoteLocalUid** () const
- const `QString` & **noteLocalUid** () const
- void **setNoteLocalUid** (const `QString` &guid)
- bool **hasData** () const
- bool **hasDataHash** () const

- `const QByteArray & dataHash () const`
- `void setDataHash (const QByteArray &hash)`
- `bool hasDataSize () const`
- `qint32 dataSize () const`
- `void setDataSize (const qint32 size)`
- `bool hasDataBody () const`
- `const QByteArray & dataBody () const`
- `void setDataBody (const QByteArray &body)`
- `bool hasMime () const`
- `const QString & mime () const`
- `void setMime (const QString &mime)`
- `bool hasWidth () const`
- `qint16 width () const`
- `void setWidth (const qint16 width)`
- `bool hasHeight () const`
- `qint16 height () const`
- `void setHeight (const qint16 height)`
- `bool hasRecognitionData () const`
- `bool hasRecognitionDataHash () const`
- `const QByteArray & recognitionDataHash () const`
- `void setRecognitionDataHash (const QByteArray &hash)`
- `bool hasRecognitionDataSize () const`
- `qint32 recognitionDataSize () const`
- `void setRecognitionDataSize (const qint32 size)`
- `bool hasRecognitionDataBody () const`
- `const QByteArray & recognitionDataBody () const`
- `void setRecognitionDataBody (const QByteArray &body)`
- `bool hasAlternateData () const`
- `bool hasAlternateDataHash () const`
- `const QByteArray & alternateDataHash () const`
- `void setAlternateDataHash (const QByteArray &hash)`
- `bool hasAlternateDataSize () const`
- `qint32 alternateDataSize () const`
- `void setAlternateDataSize (const qint32 size)`
- `bool hasAlternateDataBody () const`
- `const QByteArray & alternateDataBody () const`
- `void setAlternateDataBody (const QByteArray &body)`
- `bool hasResourceAttributes () const`
- `const qevercloud::ResourceAttributes & resourceAttributes () const`
- `qevercloud::ResourceAttributes & resourceAttributes ()`
- `void setResourceAttributes (const qevercloud::ResourceAttributes &attributes)`
- `void setResourceAttributes (qevercloud::ResourceAttributes &&attributes)`
- `virtual QTextStream & print (QTextStream &strm) const` override

## Public Member Functions inherited from `quentier::INoteStoreDataElement`

- `virtual bool isDirty () const` =0
- `virtual void setDirty (const bool dirty)` =0
- `virtual bool isLocal () const` =0
- `virtual void setLocal (const bool local)` =0

## Public Member Functions inherited from [quentier::ILocalStorageDataElement](#)

- virtual const QString **localUid** () const =0
- virtual void **setLocalUid** (const QString &guid)=0
- virtual void **unsetLocalUid** ()=0

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

## Friends

- class **Note**

## Additional Inherited Members

## Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable** & **operator=** (const [Printable](#) &other)

## 5.57.1 Member Function Documentation

### 5.57.1.1 checkParameters()

```
virtual bool quentier::Resource::checkParameters (
    ErrorString & errorDescription ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

### 5.57.1.2 clear()

```
virtual void quentier::Resource::clear ( ) [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

### 5.57.1.3 guid()

```
virtual const QString & quentier::Resource::guid ( ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

### 5.57.1.4 hasGuid()

```
virtual bool quentier::Resource::hasGuid ( ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

#### 5.57.1.5 `hasUpdateSequenceNumber()`

```
virtual bool quentier::Resource::hasUpdateSequenceNumber ( ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

#### 5.57.1.6 `print()`

```
virtual QTextStream & quentier::Resource::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quentier::Printable](#).

#### 5.57.1.7 `setGuid()`

```
virtual void quentier::Resource::setGuid (
    const QString & guid ) [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

#### 5.57.1.8 `setUpdateSequenceNumber()`

```
virtual void quentier::Resource::setUpdateSequenceNumber (
    const quint32 updateSequenceNumber ) [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

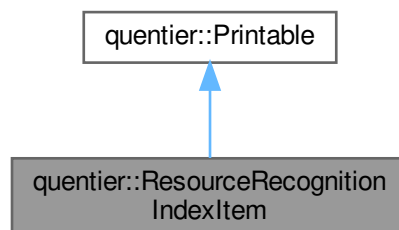
#### 5.57.1.9 `updateSequenceNumber()`

```
virtual quint32 quentier::Resource::updateSequenceNumber ( ) const [override], [virtual]
```

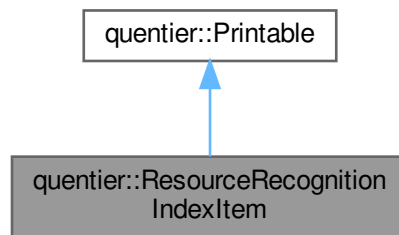
Implements [quentier::INoteStoreDataElement](#).

## 5.58 `quentier::ResourceRecognitionIndexItem` Class Reference

Inheritance diagram for `quentier::ResourceRecognitionIndexItem`:



Collaboration diagram for `quentier::ResourceRecognitionIndexItem`:



## Classes

- struct [BarcodeItem](#)
- struct [ObjectItem](#)
- struct [ShapelItem](#)
- struct [TextItem](#)

## Public Member Functions

- **ResourceRecognitionIndexItem** (const [ResourceRecognitionIndexItem](#) &other)
- [ResourceRecognitionIndexItem](#) & **operator=** (const [ResourceRecognitionIndexItem](#) &other)
- bool **isValid** () const
- int **x** () const
- void **setX** (const int x)
- int **y** () const
- void **setY** (const int y)
- int **h** () const
- void **setH** (const int h)
- int **w** () const
- void **setW** (const int w)
- int **offset** () const
- void **setOffset** (const int offset)
- int **duration** () const
- void **setDuration** (const int duration)
- [QVector](#)< int > **strokeList** () const
- int **numStrokes** () const
- bool **strokeAt** (const int strokeIndex, int &stroke) const
- bool **setStrokeAt** (const int strokeIndex, const int stroke)
- void **setStrokeList** (const [QVector](#)< int > &strokeList)
- void **reserveStrokeListSpace** (const int numItems)
- void **addStroke** (const int stroke)
- bool **removeStroke** (const int stroke)
- bool **removeStrokeAt** (const int strokeIndex)
- [QVector](#)< [TextItem](#) > **textItems** () const
- int **numTextItems** () const
- bool **textItemAt** (const int textItemIndex, [TextItem](#) &textItem) const

- bool **setTextItemAt** (const int textItemIndex, const [TextItem](#) &textItem)
- void **setTextItems** (const QVector< [TextItem](#) > &textItems)
- void **reserveTextItemsSpace** (const int numItems)
- void **addTextItem** (const [TextItem](#) &item)
- bool **removeTextItem** (const [TextItem](#) &item)
- bool **removeTextItemAt** (const int textItemIndex)
- QVector< [ObjectItem](#) > **objectItems** () const
- int **numObjectItems** () const
- bool **objectItemAt** (const int objectItemIndex, [ObjectItem](#) &objectItem) const
- bool **setObjectItemAt** (const int objectItemIndex, const [ObjectItem](#) &objectItem)
- void **setObjectItems** (const QVector< [ObjectItem](#) > &objectItems)
- void **reserveObjectItemsSpace** (const int numItems)
- void **addObjectItem** (const [ObjectItem](#) &item)
- bool **removeObjectItem** (const [ObjectItem](#) &item)
- bool **removeObjectItemAt** (const int objectItemIndex)
- QVector< [ShapelItem](#) > **shapelItems** () const
- int **numShapelItems** () const
- bool **shapelItemAt** (const int shapelItemIndex, [ShapelItem](#) &shapelItem) const
- bool **setShapelItemAt** (const int shapelItemIndex, const [ShapelItem](#) &shapelItem)
- void **setShapelItems** (const QVector< [ShapelItem](#) > &shapelItems)
- void **reserveShapelItemsSpace** (const int numItems)
- void **addShapelItem** (const [ShapelItem](#) &item)
- bool **removeShapelItem** (const [ShapelItem](#) &item)
- bool **removeShapelItemAt** (const int shapelItemIndex)
- QVector< [BarcodeItem](#) > **barcodeItems** () const
- int **numBarcodeItems** () const
- bool **barcodeItemAt** (const int barcodeItemIndex, [BarcodeItem](#) &barcodeItem) const
- bool **setBarcodeItemAt** (const int barcodeItemIndex, const [BarcodeItem](#) &barcodeItem)
- void **setBarcodeItems** (const QVector< [BarcodeItem](#) > &barcodeItems)
- void **reserveBarcodeItemsSpace** (const int numItems)
- void **addBarcodeItem** (const [BarcodeItem](#) &item)
- bool **removeBarcodeItem** (const [BarcodeItem](#) &item)
- bool **removeBarcodeItemAt** (const int barcodeItemIndex)
- virtual QTextStream & **print** (QTextStream &strm) const override

## Public Member Functions inherited from [quantier::Printable](#)

- virtual const QString **toString** () const

## Additional Inherited Members

## Protected Member Functions inherited from [quantier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

### 5.58.1 Member Function Documentation

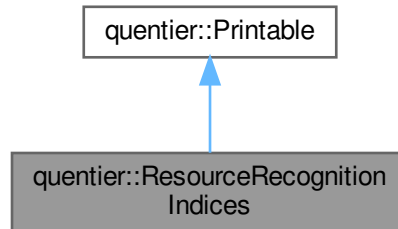
#### 5.58.1.1 print()

```
virtual QTextStream & quantier::ResourceRecognitionIndexItem::print (
    QTextStream & strm ) const [override], [virtual]
```

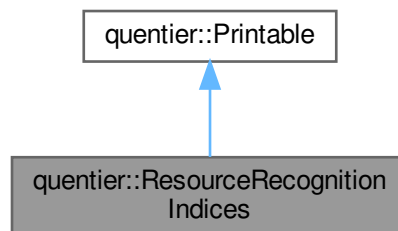
Implements [quantier::Printable](#).

## 5.59 quantier::ResourceRecognitionIndices Class Reference

Inheritance diagram for quantier::ResourceRecognitionIndices:



Collaboration diagram for quantier::ResourceRecognitionIndices:



### Public Member Functions

- **ResourceRecognitionIndices** (const [ResourceRecognitionIndices](#) &other)
- **ResourceRecognitionIndices** (const QByteArray &rawRecognitionIndicesData)
- [ResourceRecognitionIndices](#) & **operator=** (const [ResourceRecognitionIndices](#) &other)
- bool **isNull** () const
- bool **isValid** () const
- QString **objectId** () const
- QString **objectType** () const
- QString **recoType** () const
- QString **engineVersion** () const
- QString **docType** () const
- QString **lang** () const
- int **objectHeight** () const
- int **objectWidth** () const
- QVector< [ResourceRecognitionIndexItem](#) > **items** () const
- bool **setData** (const QByteArray &rawRecognitionIndicesData)
- virtual QTextStream & [print](#) (QTextStream &strm) const override

## Public Member Functions inherited from [quantier::Printable](#)

- virtual const QString **toString** () const

## Additional Inherited Members

## Protected Member Functions inherited from [quantier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

### 5.59.1 Member Function Documentation

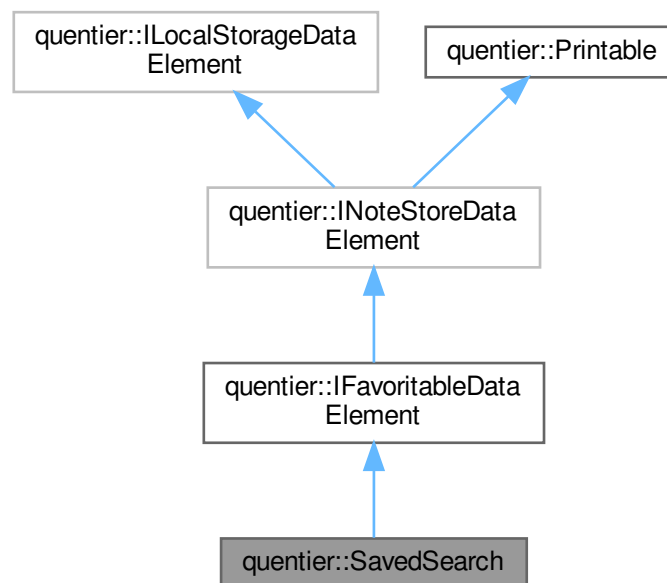
#### 5.59.1.1 print()

```
virtual QTextStream & quantier::ResourceRecognitionIndices::print (  
    QTextStream & strm ) const [override], [virtual]
```

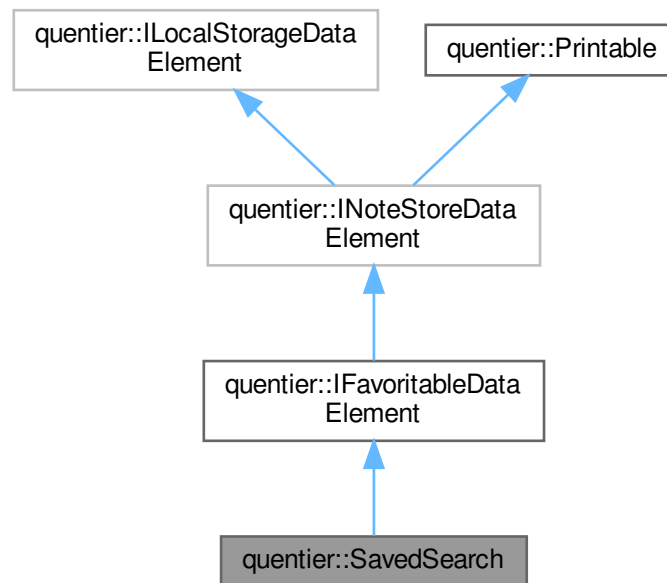
Implements [quantier::Printable](#).

## 5.60 quantier::SavedSearch Class Reference

Inheritance diagram for quantier::SavedSearch:



Collaboration diagram for `quentier::SavedSearch`:



## Public Types

- using **QueryFormat** = `qevercloud::QueryFormat`
- using **SavedSearchScope** = `qevercloud::SavedSearchScope`

## Public Member Functions

- **SavedSearch** (const [SavedSearch](#) &other)
- **SavedSearch** ([SavedSearch](#) &&other)
- [SavedSearch](#) & **operator=** (const [SavedSearch](#) &other)
- [SavedSearch](#) & **operator=** ([SavedSearch](#) &&other)
- **SavedSearch** (const `qevercloud::SavedSearch` &search)
- **SavedSearch** (`qevercloud::SavedSearch` &&search)
- const `qevercloud::SavedSearch` & **qevercloudSavedSearch** () const
- `qevercloud::SavedSearch` & **qevercloudSavedSearch** ()
- bool **operator==** (const [SavedSearch](#) &other) const
- bool **operator!=** (const [SavedSearch](#) &other) const
- virtual void **clear** () override
- virtual bool **hasGuid** () const override
- virtual const `QString` & **guid** () const override
- virtual void **setGuid** (const `QString` &guid) override
- virtual bool **hasUpdateSequenceNumber** () const override
- virtual `qint32` **updateSequenceNumber** () const override
- virtual void **setUpdateSequenceNumber** (const `qint32` usn) override
- virtual bool **checkParameters** ([ErrorString](#) &errorDescription) const override
- bool **hasName** () const

- const QString & **name** () const
- void **setName** (const QString &name)
- bool **hasQuery** () const
- const QString & **query** () const
- void **setQuery** (const QString &query)
- bool **hasQueryFormat** () const
- QueryFormat **queryFormat** () const
- void **setQueryFormat** (const quint8 queryFormat)
- bool **hasIncludeAccount** () const
- bool **includeAccount** () const
- void **setIncludeAccount** (const bool includeAccount)
- bool **hasIncludePersonalLinkedNotebooks** () const
- bool **includePersonalLinkedNotebooks** () const
- void **setIncludePersonalLinkedNotebooks** (const bool includePersonalLinkedNotebooks)
- bool **hasIncludeBusinessLinkedNotebooks** () const
- bool **includeBusinessLinkedNotebooks** () const
- void **setIncludeBusinessLinkedNotebooks** (const bool includeBusinessLinkedNotebooks)
- virtual QTextStream & **print** (QTextStream &strm) const override

### Public Member Functions inherited from [quantier::IFavoritableDataElement](#)

- virtual bool **isFavorited** () const =0
- virtual void **setFavorited** (const bool favorited)=0

### Public Member Functions inherited from [quantier::INoteStoreDataElement](#)

- virtual bool **isDirty** () const =0
- virtual void **setDirty** (const bool dirty)=0
- virtual bool **isLocal** () const =0
- virtual void **setLocal** (const bool local)=0

### Public Member Functions inherited from [quantier::ILocalStorageDataElement](#)

- virtual const QString **localUid** () const =0
- virtual void **setLocalUid** (const QString &guid)=0
- virtual void **unsetLocalUid** ()=0

### Public Member Functions inherited from [quantier::Printable](#)

- virtual const QString **toString** () const

### Static Public Member Functions

- static bool **validateName** (const QString &name, [ErrorString](#) \*pErrorDescription=nullptr)

### Additional Inherited Members

### Protected Member Functions inherited from [quantier::Printable](#)

- **Printable** (const [Printable](#) &other)
- **Printable** & **operator=** (const [Printable](#) &other)

## 5.60.1 Member Function Documentation

### 5.60.1.1 checkParameters()

```
virtual bool quantier::SavedSearch::checkParameters (
    ErrorString & errorDescription ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.60.1.2 clear()

```
virtual void quantier::SavedSearch::clear ( ) [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.60.1.3 guid()

```
virtual const QString & quantier::SavedSearch::guid ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.60.1.4 hasGuid()

```
virtual bool quantier::SavedSearch::hasGuid ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.60.1.5 hasUpdateSequenceNumber()

```
virtual bool quantier::SavedSearch::hasUpdateSequenceNumber ( ) const [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

### 5.60.1.6 print()

```
virtual QTextStream & quantier::SavedSearch::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quantier::Printable](#).

### 5.60.1.7 setGuid()

```
virtual void quantier::SavedSearch::setGuid (
    const QString & guid ) [override], [virtual]
```

Implements [quantier::INoteStoreDataElement](#).

#### 5.60.1.8 `setUpdateSequenceNumber()`

```
virtual void quentier::SavedSearch::setUpdateSequenceNumber (
    const qint32 usn ) [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

#### 5.60.1.9 `updateSequenceNumber()`

```
virtual qint32 quentier::SavedSearch::updateSequenceNumber ( ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

## 5.61 `quentier::ResourceRecognitionIndexItem::Shapeltem` Struct Reference

### Public Member Functions

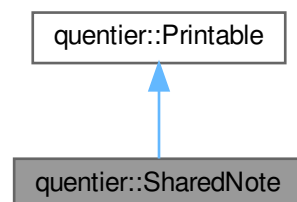
- `bool operator== (const Shapeltem &other) const`

### Public Attributes

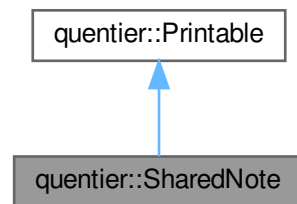
- `QString m_shapeType`
- `int m_weight = -1`

## 5.62 `quentier::SharedNote` Class Reference

Inheritance diagram for `quentier::SharedNote`:



Collaboration diagram for `quentier::SharedNote`:



### Public Types

- using **SharedNotePrivilegeLevel** = `qevercloud::SharedNotePrivilegeLevel`
- using **ContactType** = `qevercloud::ContactType`

### Public Member Functions

- **SharedNote** (const [SharedNote](#) &other)
- **SharedNote** ([SharedNote](#) &&other)
- **SharedNote** (const `qevercloud::SharedNote` &sharedNote)
- [SharedNote](#) & **operator=** (const [SharedNote](#) &other)
- [SharedNote](#) & **operator=** ([SharedNote](#) &&other)
- bool **operator==** (const [SharedNote](#) &other) const
- bool **operator!=** (const [SharedNote](#) &other) const
- const `qevercloud::SharedNote` & **qevercloudSharedNote** () const
- `qevercloud::SharedNote` & **qevercloudSharedNote** ()
- const `QString` & **noteGuid** () const
- void **setNoteGuid** (const `QString` &noteGuid)
- int **indexInNote** () const
- void **setIndexInNote** (const int index)
- bool **hasSharerUserId** () const
- `qint32` **sharerUserId** () const
- void **setSharerUserId** (const `qint32` id)
- bool **hasRecipientIdentityId** () const
- `qint64` **recipientIdentityId** () const
- void **setRecipientIdentityId** (const `qint64` identityId)
- bool **hasRecipientIdentityContactName** () const
- const `QString` & **recipientIdentityContactName** () const
- void **setRecipientIdentityContactName** (const `QString` &recipientIdentityContactName)
- bool **hasRecipientIdentityContactId** () const
- const `QString` & **recipientIdentityContactId** () const
- void **setRecipientIdentityContactId** (const `QString` &recipientIdentityContactId)
- bool **hasRecipientIdentityContactType** () const
- `ContactType` **recipientIdentityContactType** () const
- void **setRecipientIdentityContactType** (const `ContactType` recipientIdentityContactType)
- void **setRecipientIdentityContactType** (const `qint32` recipientIdentityContactType)
- bool **hasRecipientIdentityContactPhotoUrl** () const

- const QString & **recipientIdentityContactPhotoUrl** () const
- void **setRecipientIdentityContactPhotoUrl** (const QString &recipientIdentityPhotoUrl)
- bool **hasRecipientIdentityContactPhotoLastUpdated** () const
- qint64 **recipientIdentityContactPhotoLastUpdated** () const
- void **setRecipientIdentityContactPhotoLastUpdated** (const qint64 recipientIdentityPhotoLastUpdated)
- bool **hasRecipientIdentityContactMessagingPermit** () const
- const QByteArray & **recipientIdentityContactMessagingPermit** () const
- void **setRecipientIdentityContactMessagingPermit** (const QByteArray &messagingPermit)
- bool **hasRecipientIdentityContactMessagingPermitExpires** () const
- qint64 **recipientIdentityContactMessagingPermitExpires** () const
- void **setRecipientIdentityContactMessagingPermitExpires** (const qint64 timestamp)
- bool **hasRecipientIdentityUserId** () const
- qint32 **recipientIdentityUserId** () const
- void **setRecipientIdentityUserId** (const qint32 userId)
- bool **hasRecipientIdentityDeactivated** () const
- bool **recipientIdentityDeactivated** () const
- void **setRecipientIdentityDeactivated** (const bool deactivated)
- bool **hasRecipientIdentitySameBusiness** () const
- bool **recipientIdentitySameBusiness** () const
- void **setRecipientIdentitySameBusiness** (const bool sameBusiness)
- bool **hasRecipientIdentityBlocked** () const
- bool **recipientIdentityBlocked** () const
- void **setRecipientIdentityBlocked** (const bool blocked)
- bool **hasRecipientIdentityUserConnected** () const
- bool **recipientIdentityUserConnected** () const
- void **setRecipientIdentityUserConnected** (const bool userConnected)
- bool **hasRecipientIdentityEventId** () const
- qint64 **recipientIdentityEventId** () const
- void **setRecipientIdentityEventId** (const qint64 eventId)
- bool **hasRecipientIdentity** () const
- const qevercloud::Identity & **recipientIdentity** () const
- void **setRecipientIdentity** (qevercloud::Identity &&identity)
- bool **hasPrivilegeLevel** () const
- SharedNotePrivilegeLevel **privilegeLevel** () const
- void **setPrivilegeLevel** (const SharedNotePrivilegeLevel level)
- void **setPrivilegeLevel** (const qint8 level)
- bool **hasCreationTimestamp** () const
- qint64 **creationTimestamp** () const
- void **setCreationTimestamp** (const qint64 timestamp)
- bool **hasModificationTimestamp** () const
- qint64 **modificationTimestamp** () const
- void **setModificationTimestamp** (const qint64 timestamp)
- bool **hasAssignmentTimestamp** () const
- qint64 **assignmentTimestamp** () const
- void **setAssignmentTimestamp** (const qint64 timestamp)
- virtual QTextStream & **print** (QTextStream &strm) const override

## Public Member Functions inherited from [quantier::Printable](#)

- virtual const QString **toString** () const

## Additional Inherited Members

### Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

## 5.62.1 Member Function Documentation

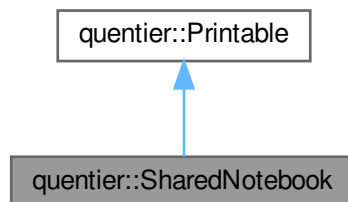
### 5.62.1.1 `print()`

```
virtual QTextStream & quentier::SharedNotebook::print (
    QTextStream & strm ) const    [override], [virtual]
```

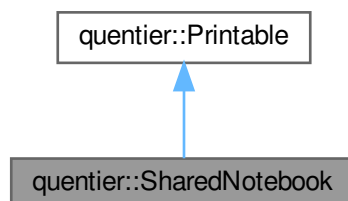
Implements [quentier::Printable](#).

## 5.63 [quentier::SharedNotebook](#) Class Reference

Inheritance diagram for [quentier::SharedNotebook](#):



Collaboration diagram for [quentier::SharedNotebook](#):



## Public Types

- using **SharedNotebookPrivilegeLevel** = qevercloud::SharedNotebookPrivilegeLevel

## Public Member Functions

- **SharedNotebook** (const [SharedNotebook](#) &other)
- **SharedNotebook** ([SharedNotebook](#) &&other)
- **SharedNotebook** (const qevercloud::SharedNotebook &qecSharedNotebook)
- [SharedNotebook](#) & **operator=** (const [SharedNotebook](#) &other)
- [SharedNotebook](#) & **operator=** ([SharedNotebook](#) &&other)
- bool **operator==** (const [SharedNotebook](#) &other) const
- bool **operator!=** (const [SharedNotebook](#) &other) const
- const qevercloud::SharedNotebook & **qevercloudSharedNotebook** () const
- qevercloud::SharedNotebook & **qevercloudSharedNotebook** ()
- int **indexInNotebook** () const
- void **setIndexInNotebook** (const int index)
- bool **hasId** () const
- qint64 **id** () const
- void **setId** (const qint64 id)
- bool **hasUserId** () const
- qint32 **userId** () const
- void **setUserId** (const qint32 userId)
- bool **hasNotebookGuid** () const
- const QString & **notebookGuid** () const
- void **setNotebookGuid** (const QString &notebookGuid)
- bool **hasEmail** () const
- const QString & **email** () const
- void **setEmail** (const QString &email)
- bool **hasCreationTimestamp** () const
- qint64 **creationTimestamp** () const
- void **setCreationTimestamp** (const qint64 timestamp)
- bool **hasModificationTimestamp** () const
- qint64 **modificationTimestamp** () const
- void **setModificationTimestamp** (const qint64 timestamp)
- bool **hasUsername** () const
- const QString & **username** () const
- void **setUsername** (const QString &username)
- bool **hasPrivilegeLevel** () const
- SharedNotebookPrivilegeLevel **privilegeLevel** () const
- void **setPrivilegeLevel** (const SharedNotebookPrivilegeLevel privilegeLevel)
- void **setPrivilegeLevel** (const qint8 privilegeLevel)
- bool **hasReminderNotifyEmail** () const
- bool **reminderNotifyEmail** () const
- void **setReminderNotifyEmail** (const bool notifyEmail)
- bool **hasReminderNotifyApp** () const
- bool **reminderNotifyApp** () const
- void **setReminderNotifyApp** (const bool notifyApp)
- bool **hasRecipientUsername** () const
- const QString & **recipientUsername** () const
- void **setRecipientUsername** (const QString &recipientUsername)
- bool **hasRecipientUserId** () const
- qint32 **recipientUserId** () const
- void **setRecipientUserId** (const qint32 userId)

- bool **hasRecipientIdentityId** () const
- qint64 **recipientIdentityId** () const
- void **setRecipientIdentityId** (const qint64 recipientIdentityId)
- bool **hasGlobalId** () const
- const QString & **globalId** () const
- void **setGlobalId** (const QString &globalId)
- bool **hasSharerUserId** () const
- qint32 **sharerUserId** () const
- void **setSharerUserId** (qint32 sharerUserId)
- bool **hasAssignmentTimestamp** () const
- qint64 **assignmentTimestamp** () const
- void **setAssignmentTimestamp** (const qint64 timestamp)
- virtual QTextStream & **print** (QTextStream &strm) const override

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

## Friends

- class **Notebook**

## Additional Inherited Members

## Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

## 5.63.1 Member Function Documentation

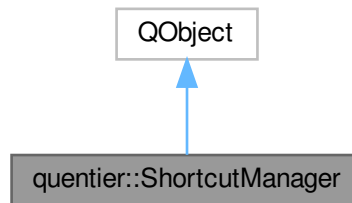
### 5.63.1.1 print()

```
virtual QTextStream & quentier::SharedNotebook::print (
    QTextStream & strm ) const [override], [virtual]
```

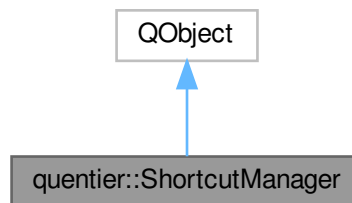
Implements [quentier::Printable](#).

## 5.64 quantier::ShortcutManager Class Reference

Inheritance diagram for quantier::ShortcutManager:



Collaboration diagram for quantier::ShortcutManager:



### Public Types

- enum **QuantierShortcutKey** {  
**NewNote** = 5000 , **NewTag** , **NewNotebook** , **NewSavedSearch** ,  
**AddAttachment** , **SaveAttachment** , **OpenAttachment** , **CopyAttachment** ,  
**CutAttachment** , **RemoveAttachment** , **RenameAttachment** , **AddAccount** ,  
**ExitAccount** , **SwitchAccount** , **AccountInfo** , **NoteSearch** ,  
**NewNoteSearch** , **ShowNotes** , **ShowNotebooks** , **ShowTags** ,  
**ShowSavedSearches** , **ShowDeletedNotes** , **ShowStatusBar** , **ShowToolBar** ,  
**PasteUnformatted** , **Font** , **UpperIndex** , **LowerIndex** ,  
**AlignLeft** , **AlignCenter** , **AlignRight** , **AlignFull** ,  
**IncreaseIndentation** , **DecreaseIndentation** , **IncreaseFontSize** , **DecreaseFontSize** ,  
**InsertNumberedList** , **InsertBulletedList** , **Strikethrough** , **Highlight** ,  
**InsertTable** , **InsertRow** , **InsertColumn** , **RemoveRow** ,  
**RemoveColumn** , **InsertHorizontalLine** , **InsertToDoTag** , **EditHyperlink** ,  
**CopyHyperlink** , **RemoveHyperlink** , **Encrypt** , **Decrypt** ,  
**DecryptPermanently** , **BackupLocalStorage** , **RestoreLocalStorage** , **UpgradeLocalStorage** ,  
**LocalStorageStatus** , **SpellCheck** , **SpellCheckIgnoreWord** , **SpellCheckAddWordToUserDictionary** ,  
**SaveImage** , **AnnotateImage** , **ImageRotateClockwise** , **ImageRotateCounterClockwise** ,  
**Synchronize** , **FullSync** , **ImportFolders** , **Preferences** ,  
**ReleaseNotes** , **ViewLogs** , **About** , **UnknownKey** = 100000 }

## Public Slots

- void **setUserShortcut** (int key, QKeySequence [shortcut](#), const [Account](#) &account, QString context={})
- void **setNonStandardUserShortcut** (QString nonStandardKey, QKeySequence [shortcut](#), const [Account](#) &account, QString context={})
- void **setDefaultShortcut** (int key, QKeySequence [shortcut](#), const [Account](#) &account, QString context={})
- void **setNonStandardDefaultShortcut** (QString nonStandardKey, QKeySequence [shortcut](#), const [Account](#) &account, QString context={})

## Signals

- void **shortcutChanged** (int key, QKeySequence [shortcut](#), const [Account](#) &account, QString context)
- void **nonStandardShortcutChanged** (QString nonStandardKey, QKeySequence [shortcut](#), const [Account](#) &account, QString context)

## Public Member Functions

- **ShortcutManager** (QObject \*parent=nullptr)
- QKeySequence [shortcut](#) (const int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [shortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const
- QKeySequence [defaultShortcut](#) (const int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [defaultShortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const
- QKeySequence [userShortcut](#) (const int key, const [Account](#) &account, const QString &context={}) const
- QKeySequence [userShortcut](#) (const QString &nonStandardKey, const [Account](#) &account, const QString &context={}) const

## 5.64.1 Member Function Documentation

### 5.64.1.1 defaultShortcut() [1/2]

```
QKeySequence quentier::ShortcutManager::defaultShortcut (
    const int key,
    const Account & account,
    const QString & context = {} ) const
```

#### Returns

Default shortcut for the standard key if present, otherwise empty key sequence

### 5.64.1.2 defaultShortcut() [2/2]

```
QKeySequence quentier::ShortcutManager::defaultShortcut (
    const QString & nonStandardKey,
    const Account & account,
    const QString & context = {} ) const
```

#### Returns

Default shortcut for the non-standard key if present, otherwise empty key sequence

### 5.64.1.3 `shortcut()` [1/2]

```
QKeySequence quentier::ShortcutManager::shortcut (
    const int key,
    const Account & account,
    const QString & context = {} ) const
```

#### Returns

Active shortcut for the standard key - either the user defined shortcut (if present) or the default one (if present as well)

### 5.64.1.4 `shortcut()` [2/2]

```
QKeySequence quentier::ShortcutManager::shortcut (
    const QString & nonStandardKey,
    const Account & account,
    const QString & context = {} ) const
```

#### Returns

Active shortcut for the non-standard key - either the user defined shortcut (if present) or the default one (if present as well)

### 5.64.1.5 `userShortcut()` [1/2]

```
QKeySequence quentier::ShortcutManager::userShortcut (
    const int key,
    const Account & account,
    const QString & context = {} ) const
```

#### Returns

User defined shortcut for the standard key if present, otherwise empty key sequence

### 5.64.1.6 `userShortcut()` [2/2]

```
QKeySequence quentier::ShortcutManager::userShortcut (
    const QString & nonStandardKey,
    const Account & account,
    const QString & context = {} ) const
```

#### Returns

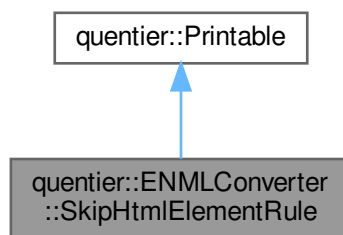
User defined shortcut for the non-standard key if present, otherwise empty key sequence

## 5.65 quantier::ENMLConverter::SkipHtmlElementRule Class Reference

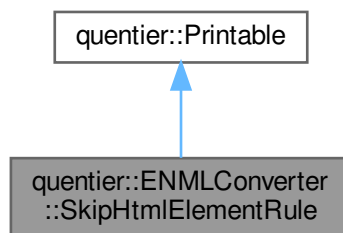
The [SkipHtmlElementRule](#) class describes the set of rules for HTML -> ENML conversion about the HTML elements that should not be actually converted to ENML due to their nature of being "helper" elements for the display or functioning of something within the note editor's page. The HTML -> ENML conversion would ignore tags and attributes forbidden by ENML even without these rules conditionally preserving or skipping the contents and nested elements of skipped elements.

```
#include <ENMLConverter.h>
```

Inheritance diagram for quantier::ENMLConverter::SkipHtmlElementRule:



Collaboration diagram for quantier::ENMLConverter::SkipHtmlElementRule:



### Public Types

- enum class **ComparisonRule** { **Equals** = 0 , **StartsWith** , **EndsWith** , **Contains** }

### Public Member Functions

- virtual QTextStream & [print](#) (QTextStream &strm) const override

## Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

## Public Attributes

- QString **m\_elementNameToSkip**
- ComparisonRule **m\_elementNameComparisonRule** = ComparisonRule::Equals
- Qt::CaseSensitivity **m\_elementNameCaseSensitivity** = Qt::CaseSensitive
- QString **m\_attributeNameToSkip**
- ComparisonRule **m\_attributeNameComparisonRule** = ComparisonRule::Equals
- Qt::CaseSensitivity **m\_attributeNameCaseSensitivity** = Qt::CaseSensitive
- QString **m\_attributeValueToSkip**
- ComparisonRule **m\_attributeValueComparisonRule** = ComparisonRule::Equals
- Qt::CaseSensitivity **m\_attributeValueCaseSensitivity** = Qt::CaseSensitive
- bool **m\_includeElementContents** = false

## Friends

- QUENTIER\_EXPORT QTextStream & **operator**<< (QTextStream &strm, const ComparisonRule rule)

## Additional Inherited Members

## Protected Member Functions inherited from [quentier::Printable](#)

- **Printable** (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

### 5.65.1 Detailed Description

The [SkipHtmlElementRule](#) class describes the set of rules for HTML -> ENML conversion about the HTML elements that should not be actually converted to ENML due to their nature of being "helper" elements for the display or functioning of something within the note editor's page. The HTML -> ENML conversion would ignore tags and attributes forbidden by ENML even without these rules conditionally preserving or skipping the contents and nested elements of skipped elements.

### 5.65.2 Member Function Documentation

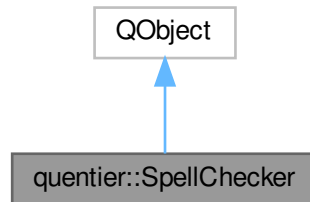
#### 5.65.2.1 print()

```
virtual QTextStream & quentier::ENMLConverter::SkipHtmlElementRule::print (
    QTextStream & strm ) const [override], [virtual]
```

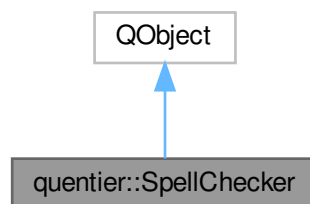
Implements [quentier::Printable](#).

## 5.66 quantier::SpellChecker Class Reference

Inheritance diagram for quantier::SpellChecker:



Collaboration diagram for quantier::SpellChecker:



### Signals

- void **ready** ()

### Public Member Functions

- **SpellChecker** ([FileIOProcessorAsync](#) \*pFileIOProcessorAsync, const [Account](#) &account, QObject \*parent=nullptr, const QString &userDictionaryPath={})
- QVector< std::pair< QString, bool > > **listAvailableDictionaries** () const
- void **setAccount** (const [Account](#) &account)
- void **enableDictionary** (const QString &language)
- void **disableDictionary** (const QString &language)
- bool **checkSpell** (const QString &word) const
- QStringList **spellCorrectionSuggestions** (const QString &misSpelledWord) const
- void **addToUserWordlist** (const QString &word)
- void **removeFromUserWordList** (const QString &word)
- void **ignoreWord** (const QString &word)
- void **removeWord** (const QString &word)
- bool **isReady** () const

## 5.67 `quentier::StringUtils::StringFilterPredicate` Struct Reference

### Public Member Functions

- **StringFilterPredicate** (`QSet< QString > &filteredStrings`)
- **bool operator()** (`const QString &str`) `const`

### Public Attributes

- `QSet< QString > & m_filteredStrings`

## 5.68 `quentier::StringUtils` Class Reference

### Classes

- struct [StringFilterPredicate](#)

### Public Member Functions

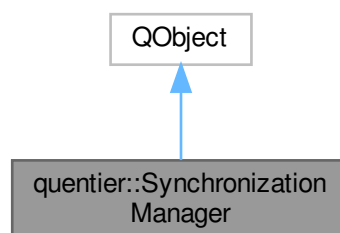
- **void removePunctuation** (`QString &str`, `const QVector< QChar > &charactersToPreserve={}`) `const`
- **void removeDiacritics** (`QString &str`) `const`
- **void removeNewlines** (`QString &str`) `const`

## 5.69 `quentier::SynchronizationManager` Class Reference

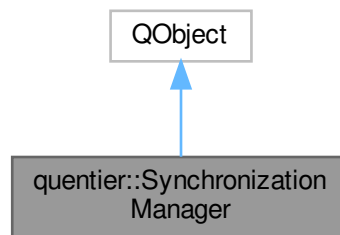
The [SynchronizationManager](#) class encapsulates methods and signals & slots required to perform the full or partial synchronization of data with remote Evernote servers. The class also deals with authentication with Evernote service through OAuth.

```
#include <SynchronizationManager.h>
```

Inheritance diagram for `quentier::SynchronizationManager`:



Collaboration diagram for `quentier::SynchronizationManager`:



### Public Slots

- void [setAccount](#) ([Account](#) account)
- void [authenticate](#) ()
- void [authenticateCurrentAccount](#) ()
- void [synchronize](#) ()
- void [stop](#) ()
- void [revokeAuthentication](#) (const `qevercloud::UserID` userId)
- void [setDownloadNoteThumbnails](#) (bool flag)
- void [setDownloadInkNoteImages](#) (bool flag)
- void [setInkNoteImagesStoragePath](#) (QString path)

### Signals

- void [started](#) ()
- void [stopped](#) ()
- void [failed](#) ([ErrorString](#) errorDescription)
- void [finished](#) ([Account](#) account, bool somethingDownloaded, bool somethingSent)
- void [authenticationRevoked](#) (bool success, [ErrorString](#) errorDescription, `qevercloud::UserID` userId)
- void [authenticationFinished](#) (bool success, [ErrorString](#) errorDescription, [Account](#) account)
- void [remoteToLocalSyncStopped](#) ()
- void [sendLocalChangesStopped](#) ()
- void [willRepeatRemoteToLocalSyncAfterSendingChanges](#) ()
- void [detectedConflictDuringLocalChangesSending](#) ()
- void [rateLimitExceeded](#) (qint32 secondsToWait)
- void [remoteToLocalSyncDone](#) (bool somethingDownloaded)
- void [syncChunksDownloadProgress](#) (qint32 highestDownloadedUsn, qint32 highestServerUsn, qint32 last← PreviousUsn)
- void [syncChunksDownloaded](#) ()
- void [linkedNotebookSyncChunksDownloadProgress](#) (qint32 highestDownloadedUsn, qint32 highestServer← Usn, qint32 lastPreviousUsn, [LinkedNotebook](#) linkedNotebook)
- void [linkedNotebooksSyncChunksDownloaded](#) ()
- void [notesDownloadProgress](#) (qint32 notesDownloaded, qint32 totalNotesToDownload)
- void [linkedNotebooksNotesDownloadProgress](#) (qint32 notesDownloaded, qint32 totalNotesToDownload)
- void [resourcesDownloadProgress](#) (qint32 resourcesDownloaded, qint32 totalResourcesToDownload)

- void [linkedNotebooksResourcesDownloadProgress](#) (quint32 resourcesDownloaded, quint32 total↔ ResourcesToDownload)
- void [preparedDirtyObjectsForSending](#) ()
- void [preparedLinkedNotebooksDirtyObjectsForSending](#) ()
- void [setAccountDone](#) (Account account)
- void [setDownloadNoteThumbnailsDone](#) (bool flag)
- void [setDownloadInkNoteImagesDone](#) (bool flag)
- void [setInkNoteImagesStoragePathDone](#) (QString path)

## Public Member Functions

- [SynchronizationManager](#) (QString host, [LocalStorageManagerAsync](#) &localStorageManagerAsync, [IAuthenticationManager](#) &authenticationManager, QObject \*parent=nullptr, INoteStorePtr pNoteStore={}, IUserStorePtr pUserStore={}, IKeychainServicePtr pKeychainService={}, ISyncStateStoragePtr pSync↔ StateStorage={})
- bool [active](#) () const
- bool [downloadNoteThumbnailsOption](#) () const

## 5.69.1 Detailed Description

The [SynchronizationManager](#) class encapsulates methods and signals & slots required to perform the full or partial synchronization of data with remote Evernote servers. The class also deals with authentication with Evernote service through OAuth.

## 5.69.2 Constructor & Destructor Documentation

### 5.69.2.1 SynchronizationManager()

```
quantier::SynchronizationManager::SynchronizationManager (
    QString host,
    LocalStorageManagerAsync & localStorageManagerAsync,
    IAuthenticationManager & authenticationManager,
    QObject * parent = nullptr,
    INoteStorePtr pNoteStore = {},
    IUserStorePtr pUserStore = {},
    IKeychainServicePtr pKeychainService = {},
    ISyncStateStoragePtr pSyncStateStorage = {} )
```

#### Parameters

<i>host</i>	The host to use for the connection with the Evernote service - typically <a href="#">www.evernote.com</a> but could also be <a href="#">sandbox.evernote.com</a> or some other one
<i>localStorageManagerAsync</i>	Local storage manager
<i>authenticationManager</i>	Reference to an object implementing <a href="#">IAuthenticationManager</a> interface; <a href="#">SynchronizationManager</a> does not store this reference, it only connects to the object via signals and slots during construction
<i>pNoteStore</i>	Pointer to an object implementing <a href="#">INoteStore</a> interface; if nullptr, SynchronizatrionManager would create and use its own instance; otherwise <a href="#">SynchronizationManager</a> would set itself as the parent of the passed in object
<i>pUserStore</i>	Pointer to an object implementing <a href="#">IUserStore</a> interface; if nullptr, <a href="#">SynchronizationManager</a> would create and use its own instance

## Parameters

<i>pKeychainService</i>	Pointer to an object implementing <a href="#">IKeychainService</a> interface; if nullptr, <a href="#">SynchronizationManager</a> would create and use its own default instance; otherwise <a href="#">SynchronizationManager</a> would set itself as the parent of the passed in object
<i>pSyncStateStorage</i>	Pointer to an object implementing <a href="#">ISyncStateStorage</a> interface; if nullptr, <a href="#">SynchronizationManager</a> would create and use its own instance; otherwise <a href="#">SynchronizationManager</a> would set itself as the parent of the passed in object

### 5.69.3 Member Function Documentation

#### 5.69.3.1 active()

```
bool quentier::SynchronizationManager::active ( ) const
```

## Returns

True if the synchronization is being performed at the moment, false otherwise

#### 5.69.3.2 authenticate

```
void quentier::SynchronizationManager::authenticate ( ) [slot]
```

Use this slot to authenticate the new user to do the synchronization with the Evernote service via the client app. The invocation of this slot would respond asynchronously with authenticationFinished signal but won't start the synchronization.

[Note](#) that this slot would always proceed to the actual OAuth.

#### 5.69.3.3 authenticateCurrentAccount

```
void quentier::SynchronizationManager::authenticateCurrentAccount ( ) [slot]
```

Use this slot to authenticate the current account to do the synchronization with the Evernote service via the client app. The invocation of this slot would respond asynchronously with authenticationFinished signal but won't start the synchronization

If no account was set to [SynchronizationManager](#) prior to this slot invocation, it would proceed to OAuth. Otherwise [SynchronizationManager](#) would first check whether the persistent authentication data is in place and actual. If yes, no OAuth would be performed.

#### 5.69.3.4 authenticationFinished

```
void quentier::SynchronizationManager::authenticationFinished (
    bool success,
    ErrorString errorDescription,
    Account account ) [signal]
```

This signal is emitted in response to the explicit attempt to authenticate the new user of the client app to the Evernote service. NOTE: this signal is not emitted if the authentication was requested automatically during sync attempt, it is only emitted in response to the explicit invocation of authenticate slot.

## Parameters

<i>success</i>	True if the authentication was successful, false otherwise
<i>errorDescription</i>	The textual explanation of the failure to authenticate the new user
<i>account</i>	The account of the authenticated user

**5.69.3.5 authenticationRevoked**

```
void quantier::SynchronizationManager::authenticationRevoked (
    bool success,
    ErrorString errorDescription,
    qevercloud::UserID userId ) [signal]
```

This signal is emitted in response to the attempt to revoke the authentication for a given user ID

## Parameters

<i>success</i>	True if the authentication was revoked successfully, false otherwise
<i>errorDescription</i>	The textual explanation of the failure to revoke the authentication
<i>userId</i>	The ID of the user for which the revoke of the authentication was requested

**5.69.3.6 detectedConflictDuringLocalChangesSending**

```
void quantier::SynchronizationManager::detectedConflictDuringLocalChangesSending ( ) [signal]
```

This signal is emitted if during the "send local changes" synchronization step it was found out that new changes from the Evernote service are available AND some of them conflict with the local changes being sent.

Such situation can rarely happen in case of changes introduced concurrently with the running synchronization - perhaps via another client. The algorithm will handle it by repeating the "remote to local" incremental synchronization step, the signal is just for the sake of diagnostic.

**5.69.3.7 downloadNoteThumbnailsOption()**

```
bool quantier::SynchronizationManager::downloadNoteThumbnailsOption ( ) const
```

## Returns

True or false depending on the option to download the thumbnails for notes containing resources during sync; by default no thumbnails are downloaded

**5.69.3.8 failed**

```
void quantier::SynchronizationManager::failed (
    ErrorString errorDescription ) [signal]
```

This signal is emitted when the synchronization fails; at this moment there is no error code explaining the reason of the failure programmatically so the only explanation available is the textual one for the end user

### 5.69.3.9 finished

```
void quantier::SynchronizationManager::finished (
    Account account,
    bool somethingDownloaded,
    bool somethingSent ) [signal]
```

This signal is emitted when the synchronization is finished

#### Parameters

<i>account</i>	Represents the latest version of <a href="#">Account</a> structure filled during the synchronization procedure
<i>somethingDownloaded</i>	Boolean parameter telling the receiver whether any data items were actually downloaded during remote to local synchronization step; if there was nothing to sync up from the remote storage, this boolean would be false, otherwise it would be true
<i>somethingSent</i>	Boolean parameter telling the receiver whether any data items were actually sent during the send local changes synchronization step; if there was nothing to send to the remote storage, this boolean would be false, otherwise it would be true

### 5.69.3.10 linkedNotebooksNotesDownloadProgress

```
void quantier::SynchronizationManager::linkedNotebooksNotesDownloadProgress (
    quint32 notesDownloaded,
    quint32 totalNotesToDownload ) [signal]
```

This signal is emitted on each successful download of full note data from linked notebooks.

#### Parameters

<i>notesDownloaded</i>	The number of notes downloaded by the moment
<i>totalNotesToDownload</i>	The total number of notes that need to be downloaded

### 5.69.3.11 linkedNotebooksResourcesDownloadProgress

```
void quantier::SynchronizationManager::linkedNotebooksResourcesDownloadProgress (
    quint32 resourcesDownloaded,
    quint32 totalResourcesToDownload ) [signal]
```

This signal is emitted on each successful download of full resource data from linked notebooks during the incremental sync (as individual resources are downloaded along with their notes during full sync).

#### Parameters

<i>resourcesDownloaded</i>	The number of resources downloaded by the moment
<i>totalResourcesToDownload</i>	The total number of resources that need to be downloaded

### 5.69.3.12 linkedNotebooksSyncChunksDownloaded

```
void quantier::SynchronizationManager::linkedNotebooksSyncChunksDownloaded ( ) [signal]
```

This signal is emitted when the sync chunks for the stuff from linked notebooks are downloaded during "remote to local" synchronization step

### 5.69.3.13 linkedNotebookSyncChunksDownloadProgress

```
void quantier::SynchronizationManager::linkedNotebookSyncChunksDownloadProgress (
    qint32 highestDownloadedUsn,
    qint32 highestServerUsn,
    qint32 lastPreviousUsn,
    LinkedNotebook linkedNotebook ) [signal]
```

This signal is emitted during linked notebooks sync chunks downloading and denotes the progress of that step, individually for each linked notebook. The percentage of completeness can be computed roughly as  $(\text{highestDownloadedUsn} - \text{lastPreviousUsn}) / (\text{highestServerUsn} - \text{lastPreviousUsn}) * 100\%$ . The sync chunks for each linked notebook are downloaded sequentially so the signals for one linked notebook should not intermix with signals for other linked notebooks, however, it is within hands of Qt's slot schedulers

#### Parameters

<i>highestDownloadedUsn</i>	The highest update sequence number within data items from linked notebook sync chunks downloaded so far
<i>highestServerUsn</i>	The current highest update sequence number within the linked notebook
<i>lastPreviousUsn</i>	The last update sequence number from previous sync of the given linked notebook; if current sync is the first one, this value is zero
<i>linkedNotebook</i>	The linked notebook which sync chunks download progress is reported

### 5.69.3.14 notesDownloadProgress

```
void quantier::SynchronizationManager::notesDownloadProgress (
    quint32 notesDownloaded,
    quint32 totalNotesToDownload ) [signal]
```

This signal is emitted on each successful download of full note data from use 's own account.

#### Parameters

<i>notesDownloaded</i>	The number of notes downloaded by the moment
<i>totalNotesToDownload</i>	The total number of notes that need to be downloaded

### 5.69.3.15 preparedDirtyObjectsForSending

```
void quantier::SynchronizationManager::preparedDirtyObjectsForSending ( ) [signal]
```

This signal is emitted during "send local changes" synchronization step when all the relevant data elements from user's own account were prepared for sending to the Evernote service

### 5.69.3.16 preparedLinkedNotebooksDirtyObjectsForSending

```
void quantier::SynchronizationManager::preparedLinkedNotebooksDirtyObjectsForSending ( ) [signal]
```

This signal is emitted during "send local changes" synchronization step when all the relevant data elements from linked notebooks were prepared for sending to the Evernote service

### 5.69.3.17 rateLimitExceeded

```
void quantier::SynchronizationManager::rateLimitExceeded (
    quint32 secondsToWait ) [signal]
```

This signal is emitted when the Evernote API rate limit is breached during the synchronization; the algorithm will handle it by auto-pausing itself until the time necessary to wait passes and then automatically continuing the synchronization.

#### Parameters

<i>secondsToWait</i>	The amount of time (in seconds) necessary to wait before the synchronization will continue
----------------------	--

### 5.69.3.18 remoteToLocalSyncDone

```
void quantier::SynchronizationManager::remoteToLocalSyncDone (
    bool somethingDownloaded ) [signal]
```

This signal is emitted when the "remote to local" synchronization step is finished; once that step is done, the algorithm switches to sending the local changes back to the Evernote service.

#### Parameters

<i>somethingDownloaded</i>	Boolean parameter telling the receiver whether any data items were actually downloaded during remote to local synchronization step; if there was nothing to sync up from the remote storage, this boolean would be false, otherwise it would be true
----------------------------	--

### 5.69.3.19 remoteToLocalSyncStopped

```
void quantier::SynchronizationManager::remoteToLocalSyncStopped ( ) [signal]
```

This signal is emitted when the "remote to local" synchronization step is stopped

### 5.69.3.20 resourcesDownloadProgress

```
void quantier::SynchronizationManager::resourcesDownloadProgress (
    quint32 resourcesDownloaded,
    quint32 totalResourcesToDownload ) [signal]
```

This signal is emitted on each successful download of full resource data from user's own account during the incremental sync (as individual resources are downloaded along with their notes during full sync).

## Parameters

<i>resourcesDownloaded</i>	The number of resources downloaded by the moment
<i>totalResourcesToDownload</i>	The total number of resources that need to be downloaded

**5.69.3.21 revokeAuthentication**

```
void quantier::SynchronizationManager::revokeAuthentication (
    const qevercloud::UserID userId ) [slot]
```

Use this slot to remove any previously cached authentication tokens (and shard ids) for a given user ID. After the call of this method the next attempt to synchronize the data for this user ID would cause the launch of OAuth to get the new authentication token

**5.69.3.22 sendLocalChangesStopped**

```
void quantier::SynchronizationManager::sendLocalChangesStopped ( ) [signal]
```

This signal is emitted when the "send local changes" synchronization step is stopped

**5.69.3.23 setAccount**

```
void quantier::SynchronizationManager::setAccount (
    Account account ) [slot]
```

Use this slot to set the current account for the synchronization manager. If the slot is called during the synchronization running, it would stop, any internal caches belonging to previously selected account (if any) would be purged (but persistent settings like the authentication token saved in the system keychain would remain). Setting the current account won't automatically start the synchronization for it, use synchronize slot for this.

The attempt to set the current account of "Local" type would just clean up the synchronization manager as if it was just created

After the method finishes its job, setAccountDone signal is emitted

**5.69.3.24 setAccountDone**

```
void quantier::SynchronizationManager::setAccountDone (
    Account account ) [signal]
```

This signal is emitted in response to invoking the setAccount slot after all the activities involved in switching the account inside [SynchronizationManager](#) are finished

**5.69.3.25 setDownloadInkNoteImages**

```
void quantier::SynchronizationManager::setDownloadInkNoteImages (
    bool flag ) [slot]
```

Use this slot to switch the option whether the synchronization of notes would download the plain images corresponding to ink notes or not. By default the downloading of ink note images is disabled.

After the method finishes its job, setDownloadInkNoteImagesDone signal is emitted

### 5.69.3.26 setDownloadInkNoteImagesDone

```
void quentier::SynchronizationManager::setDownloadInkNoteImagesDone (
    bool flag ) [signal]
```

This signal is emitted in response to invoking the setDownloadInkNoteImages slot after the setting is accepted

### 5.69.3.27 setDownloadNoteThumbnails

```
void quentier::SynchronizationManager::setDownloadNoteThumbnails (
    bool flag ) [slot]
```

Use this slot to switch the option whether the synchronization of notes would download the note thumbnails or not. By default the downloading of thumbnails for notes containing resources is disabled.

NOTE: even if thumbnails downloading is enabled, the thumbnails would be downloaded during sync only for notes containing resources

After the method finishes its job, setDownloadNoteThumbnailsDone signal is emitted

### 5.69.3.28 setDownloadNoteThumbnailsDone

```
void quentier::SynchronizationManager::setDownloadNoteThumbnailsDone (
    bool flag ) [signal]
```

This signal is emitted in response to invoking the setDownloadNoteThumbnails slot after the setting is accepted

### 5.69.3.29 setInkNoteImagesStoragePath

```
void quentier::SynchronizationManager::setInkNoteImagesStoragePath (
    QString path ) [slot]
```

Use this slot to specify the path to folder at which the downloaded ink note images should be stored. Each ink note image would be stored in a separate PNG file which name would be the same as the guid of the corresponding resource and the file extension would be PNG

The default storage path would be the folder "inkNoteImages" within the folder returned by applicationPersistentStoragePath function found in quentier/StandardPaths.h header

WARNING: if the passed in path cannot be used (either it doesn't exist and cannot be created or exists but is not writable), the default path is silently restored. So make sure you're setting a valid path

After the method finishes its job, setInkNoteImagesStoragePathDone signal is emitted

### 5.69.3.30 setInkNoteImagesStoragePathDone

```
void quentier::SynchronizationManager::setInkNoteImagesStoragePathDone (
    QString path ) [signal]
```

This signal is emitted in response to invoking the setInkNoteImagesStoragePath slot after the setting is accepted

**5.69.3.31 started**

```
void quantier::SynchronizationManager::started ( ) [signal]
```

This signal is emitted when the synchronization is started (authentication is not considered a part of synchronization so this signal is only emitted when the authentication is completed)

**5.69.3.32 stop**

```
void quantier::SynchronizationManager::stop ( ) [slot]
```

Use this slot to stop the running synchronization; if no synchronization is running by the moment of this slot call, it has no effect

**5.69.3.33 stopped**

```
void quantier::SynchronizationManager::stopped ( ) [signal]
```

This signal is emitted in response to invoking the stop slot, whether it was invoked manually or from within the [SynchronizationManager](#) itself (due to sync failure, for example)

**5.69.3.34 syncChunksDownloaded**

```
void quantier::SynchronizationManager::syncChunksDownloaded ( ) [signal]
```

This signal is emitted when the sync chunks for the stuff from user's own account are downloaded during "remote to local" synchronization step

**5.69.3.35 syncChunksDownloadProgress**

```
void quantier::SynchronizationManager::syncChunksDownloadProgress (
    qint32 highestDownloadedUsn,
    qint32 highestServerUsn,
    qint32 lastPreviousUsn ) [signal]
```

This signal is emitted during user own account's sync chunks downloading and denotes the progress of that step. The percentage of completeness can be computed roughly as  $(\text{highestDownloadedUsn} - \text{lastPreviousUsn}) / (\text{highestServerUsn} - \text{lastPreviousUsn}) * 100\%$

**Parameters**

<i>highestDownloadedUsn</i>	The highest update sequence number within data items from sync chunks downloaded so far
<i>highestServerUsn</i>	The current highest update sequence number within the account
<i>lastPreviousUsn</i>	The last update sequence number from previous sync; if current sync is the first one, this value is zero

### 5.69.3.36 synchronize

```
void quantier::SynchronizationManager::synchronize ( ) [slot]
```

Use this slot to launch the synchronization of data

### 5.69.3.37 willRepeatRemoteToLocalSyncAfterSendingChanges

```
void quantier::SynchronizationManager::willRepeatRemoteToLocalSyncAfterSendingChanges ( )  
[signal]
```

This signal is emitted if during the "send local changes" synchronization step it was found out that new changes from the Evernote service are available yet no conflict between remote and local changes was found yet.

Such situation can rarely happen in case of changes introduced concurrently with the running synchronization - perhaps via another client. The algorithm will handle it, the signal is just for the sake of diagnostic.

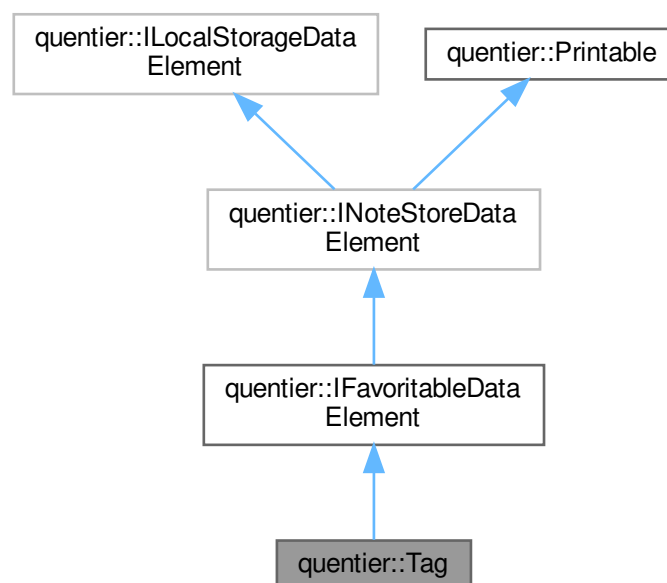
## 5.70 quantier::SysInfo Class Reference

### Public Member Functions

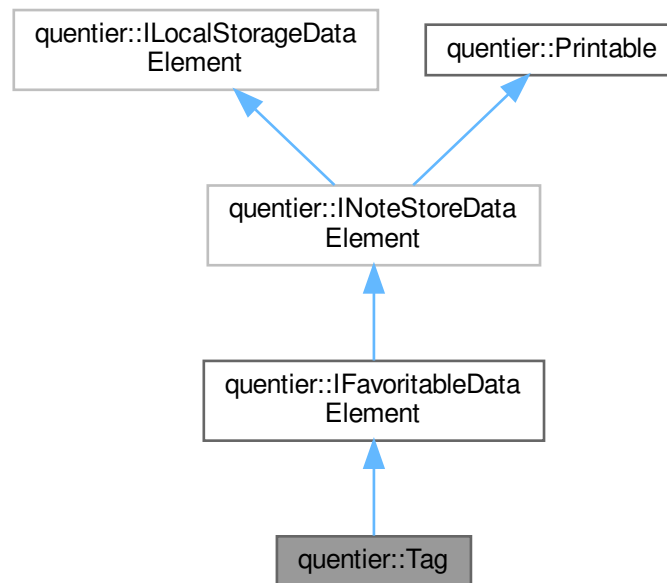
- qint64 **pageSize** ()
- qint64 **totalMemory** ()
- qint64 **freeMemory** ()
- QString **stackTrace** ()
- QString **platformName** ()

## 5.71 quantier::Tag Class Reference

Inheritance diagram for quantier::Tag:



Collaboration diagram for quantier::Tag:



### Public Member Functions

- **Tag** (const [Tag](#) &other)
- **Tag** ([Tag](#) &&other)
- [Tag](#) & **operator=** (const [Tag](#) &other)
- [Tag](#) & **operator=** ([Tag](#) &&other)
- **Tag** (const qevercloud::Tag &other)
- **Tag** (qevercloud::Tag &&other)
- bool **operator==** (const [Tag](#) &other) const
- bool **operator!=** (const [Tag](#) &other) const
- const qevercloud::Tag & **qevercloudTag** () const
- qevercloud::Tag & **qevercloudTag** ()
- virtual void **clear** () override
- virtual bool **hasGuid** () const override
- virtual const QString & **guid** () const override
- virtual void **setGuid** (const QString &guid) override
- virtual bool **hasUpdateSequenceNumber** () const override
- virtual qint32 **updateSequenceNumber** () const override
- virtual void **setUpdateSequenceNumber** (const qint32 usn) override
- virtual bool **checkParameters** ([ErrorString](#) &errorDescription) const override
- bool **hasName** () const
- const QString & **name** () const
- void **setName** (const QString &name)
- bool **hasParentGuid** () const
- const QString & **parentGuid** () const
- void **setParentGuid** (const QString &parentGuid)
- bool **hasParentLocalUid** () const

- const QString & **parentLocalUid** () const
- void **setParentLocalUid** (const QString &parentLocalUid)
- bool **hasLinkedNotebookGuid** () const
- const QString & **linkedNotebookGuid** () const
- void **setLinkedNotebookGuid** (const QString &linkedNotebookGuid)
- virtual QTextStream & **print** (QTextStream &strm) const override

### Public Member Functions inherited from [quentier::IFavoritableDataElement](#)

- virtual bool **isFavorited** () const =0
- virtual void **setFavorited** (const bool favorited)=0

### Public Member Functions inherited from [quentier::INoteStoreDataElement](#)

- virtual bool **isDirty** () const =0
- virtual void **setDirty** (const bool dirty)=0
- virtual bool **isLocal** () const =0
- virtual void **setLocal** (const bool local)=0

### Public Member Functions inherited from [quentier::ILocalStorageDataElement](#)

- virtual const QString **localUid** () const =0
- virtual void **setLocalUid** (const QString &guid)=0
- virtual void **unsetLocalUid** ()=0

### Public Member Functions inherited from [quentier::Printable](#)

- virtual const QString **toString** () const

### Static Public Member Functions

- static bool **validateName** (const QString &name, [ErrorString](#) \*pErrorDescription=nullptr)

### Additional Inherited Members

### Protected Member Functions inherited from [quentier::Printable](#)

- [Printable](#) (const [Printable](#) &other)
- [Printable](#) & **operator=** (const [Printable](#) &other)

## 5.71.1 Member Function Documentation

### 5.71.1.1 checkParameters()

```
virtual bool quentier::Tag::checkParameters (
    ErrorString & errorDescription ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

#### 5.71.1.2 clear()

```
virtual void quotientier::Tag::clear ( ) [override], [virtual]
```

Implements [quotientier::INoteStoreDataElement](#).

#### 5.71.1.3 guid()

```
virtual const QString & quotientier::Tag::guid ( ) const [override], [virtual]
```

Implements [quotientier::INoteStoreDataElement](#).

#### 5.71.1.4 hasGuid()

```
virtual bool quotientier::Tag::hasGuid ( ) const [override], [virtual]
```

Implements [quotientier::INoteStoreDataElement](#).

#### 5.71.1.5 hasUpdateSequenceNumber()

```
virtual bool quotientier::Tag::hasUpdateSequenceNumber ( ) const [override], [virtual]
```

Implements [quotientier::INoteStoreDataElement](#).

#### 5.71.1.6 print()

```
virtual QTextStream & quotientier::Tag::print (
    QTextStream & strm ) const [override], [virtual]
```

Implements [quotientier::Printable](#).

#### 5.71.1.7 setGuid()

```
virtual void quotientier::Tag::setGuid (
    const QString & guid ) [override], [virtual]
```

Implements [quotientier::INoteStoreDataElement](#).

#### 5.71.1.8 setUpdateSequenceNumber()

```
virtual void quotientier::Tag::setUpdateSequenceNumber (
    const quint32 usn ) [override], [virtual]
```

Implements [quotientier::INoteStoreDataElement](#).

#### 5.71.1.9 updateSequenceNumber()

```
virtual qint32 quentier::Tag::updateSequenceNumber ( ) const [override], [virtual]
```

Implements [quentier::INoteStoreDataElement](#).

## 5.72 quentier::ResourceRecognitionIndexItem::TextItem Struct Reference

### Public Member Functions

- bool **operator==** (const [TextItem](#) &other) const

### Public Attributes

- QString **m\_text**
- int **m\_weight** = -1

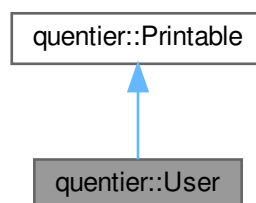
## 5.73 quentier::UidGenerator Class Reference

### Static Public Member Functions

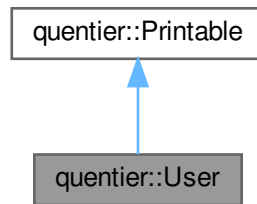
- static QString **Generate** ()
- static QString **UidToString** (const QUuid &uid)

## 5.74 quentier::User Class Reference

Inheritance diagram for quentier::User:



Collaboration diagram for quantier::User:



### Public Types

- using **PrivilegeLevel** = qevercloud::PrivilegeLevel
- using **ServiceLevel** = qevercloud::ServiceLevel

### Public Member Functions

- **User** (const qevercloud::User &user)
- **User** (qevercloud::User &&user)
- **User** (const [User](#) &other)
- **User** ([User](#) &&other)
- [User](#) & **operator=** (const [User](#) &other)
- [User](#) & **operator=** ([User](#) &&other)
- bool **operator==** (const [User](#) &other) const
- bool **operator!=** (const [User](#) &other) const
- const qevercloud::User & **qevercloudUser** () const
- qevercloud::User & **qevercloudUser** ()
- void **clear** ()
- bool **isDirty** () const
- void **setDirty** (const bool dirty)
- bool **isLocal** () const
- void **setLocal** (const bool local)
- bool **checkParameters** ([ErrorString](#) &errorDescription) const
- bool **hasId** () const
- qint32 **id** () const
- void **setId** (const qint32 id)
- bool **hasUsername** () const
- const QString & **username** () const
- void **setUsername** (const QString &username)
- bool **hasEmail** () const
- const QString & **email** () const
- void **setEmail** (const QString &email)
- bool **hasName** () const
- const QString & **name** () const
- void **setName** (const QString &name)
- bool **hasTimezone** () const
- const QString & **timezone** () const

- void **setTimezone** (const QString &timezone)
- bool **hasPrivilegeLevel** () const
- PrivilegeLevel **privilegeLevel** () const
- void **setPrivilegeLevel** (const quint8 level)
- bool **hasServiceLevel** () const
- ServiceLevel **serviceLevel** () const
- void **setServiceLevel** (const quint8 level)
- bool **hasCreationTimestamp** () const
- quint64 **creationTimestamp** () const
- void **setCreationTimestamp** (const quint64 timestamp)
- bool **hasModificationTimestamp** () const
- quint64 **modificationTimestamp** () const
- void **setModificationTimestamp** (const quint64 timestamp)
- bool **hasDeletionTimestamp** () const
- quint64 **deletionTimestamp** () const
- void **setDeletionTimestamp** (const quint64 timestamp)
- bool **hasActive** () const
- bool **active** () const
- void **setActive** (const bool active)
- bool **hasShardId** () const
- const QString & **shardId** () const
- void **setShardId** (const QString &shardId)
- bool **hasUserAttributes** () const
- const qevercloud::UserAttributes & **userAttributes** () const
- void **setUserAttributes** (qevercloud::UserAttributes &&attributes)
- bool **hasAccounting** () const
- const qevercloud::Accounting & **accounting** () const
- void **setAccounting** (qevercloud::Accounting &&accounting)
- bool **hasBusinessUserInfo** () const
- const qevercloud::BusinessUserInfo & **businessUserInfo** () const
- void **setBusinessUserInfo** (qevercloud::BusinessUserInfo &&info)
- bool **hasPhotoUrl** () const
- QString **photoUrl** () const
- void **setPhotoUrl** (const QString &photoUrl)
- bool **hasPhotoLastUpdateTimestamp** () const
- quint64 **photoLastUpdateTimestamp** () const
- void **setPhotoLastUpdateTimestamp** (const quint64 timestamp)
- bool **hasAccountLimits** () const
- const qevercloud::AccountLimits & **accountLimits** () const
- void **setAccountLimits** (qevercloud::AccountLimits &&limits)
- virtual QTextStream & **print** (QTextStream &strm) const override

## Public Member Functions inherited from [qumentier::Printable](#)

- virtual const QString **toString** () const

## Friends

- class **Notebook**

### Additional Inherited Members

### Protected Member Functions inherited from `quentier::Printable`

- **Printable** (const `Printable` &other)
- `Printable` & **operator=** (const `Printable` &other)

## 5.74.1 Member Function Documentation

### 5.74.1.1 `print()`

```
virtual QTextStream & quentier::User::print (  
    QTextStream & strm ) const    [override], [virtual]
```

Implements `quentier::Printable`.



## Chapter 6

# File Documentation

### 6.1 DecryptedTextManager.h

```
00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_ENML_DECRYPTED_TEXT_MANAGER_H
00020 #define LIB_QUENTIER_ENML_DECRYPTED_TEXT_MANAGER_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QtGlobal>
00025
00026 namespace quentier {
00027
00028 QT_FORWARD_DECLARE_CLASS(DecryptedTextManagerPrivate)
00029
00030 class QUENTIER_EXPORT DecryptedTextManager
00031 {
00032 public:
00033     DecryptedTextManager();
00034     virtual ~DecryptedTextManager();
00035
00036     void addEntry(
00037         const QString & hash, const QString & decryptedText,
00038         const bool rememberForSession, const QString & passphrase,
00039         const QString & cipher, const size_t keyLength);
00040
00041     void removeEntry(const QString & hash);
00042
00043     void clearNonRememberedForSessionEntries();
00044
00045     bool findDecryptedTextByEncryptedText(
00046         const QString & encryptedText, QString & decryptedText,
00047         bool & rememberForSession) const;
00048
00049     bool modifyDecryptedText(
00050         const QString & originalEncryptedText, const QString & newDecryptedText,
00051         QString & newEncryptedText);
00052
00053 private:
00054     Q_DISABLE_COPY(DecryptedTextManager)
00055
00056     DecryptedTextManagerPrivate * const d_ptr;
00057     Q_DECLARE_PRIVATE(DecryptedTextManager)
00058 };
```

```

00059
00060 } // namespace quantier
00061
00062 #endif // LIB_QUENTIER_ENML_DECRYPTED_TEXT_MANAGER_H

```

## 6.2 ENMLConverter.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_ENML_ENML_CONVERTER_H
00020 #define LIB_QUENTIER_ENML_ENML_CONVERTER_H
00021
00022 #include <quantier/types/ErrorMessage.h>
00023 #include <quantier/types/Note.h>
00024 #include <quantier/utility/Linkage.h>
00025 #include <quantier/utility/Printable.h>
00026
00027 #include <QHash>
00028 #include <QSet>
00029 #include <QString>
00030 #include <QTextDocument>
00031
00032 namespace quantier {
00033
00034 QT_FORWARD_DECLARE_CLASS(DecryptedTextManager)
00035 QT_FORWARD_DECLARE_CLASS(ENMLConverterPrivate)
00036 QT_FORWARD_DECLARE_CLASS(Resource)
00037
00038
00039 class QUENTIER_EXPORT ENMLConverter
00040 {
00041 public:
00042     ENMLConverter();
00043
00044     virtual ~ENMLConverter();
00045
00046     class QUENTIER_EXPORT SkipHtmlElementRule : public Printable
00047     {
00048     public:
00049         enum class ComparisonRule
00050         {
00051             Equals = 0,
00052             StartsWith,
00053             EndsWith,
00054             Contains
00055         };
00056
00057         friend QUENTIER_EXPORT QTextStream & operator<<(
00058             QTextStream & strm, const ComparisonRule rule);
00059
00060         virtual QTextStream & print(QTextStream & strm) const override;
00061
00062         QString m_elementNameToSkip;
00063         ComparisonRule m_elementNameComparisonRule = ComparisonRule::Equals;
00064         Qt::CaseSensitivity m_elementNameCaseSensitivity = Qt::CaseSensitive;
00065
00066         QString m_attributeNameToSkip;
00067         ComparisonRule m_attributeNameComparisonRule = ComparisonRule::Equals;
00068         Qt::CaseSensitivity m_attributeNameCaseSensitivity = Qt::CaseSensitive;
00069
00070         QString m_attributeValueToSkip;
00071         ComparisonRule m_attributeValueComparisonRule = ComparisonRule::Equals;
00072         Qt::CaseSensitivity m_attributeValueCaseSensitivity = Qt::CaseSensitive;
00073
00074         bool m_includeElementContents = false;
00075     };
00076
00077 };
00078
00079

```

```

00091     bool htmlToNoteContent(
00092         const QString & html, QString & noteContent,
00093         DecryptedTextManager & decryptedTextManager,
00094         ErrorString & errorDescription,
00095         const QVector<SkipHtmlElementRule> & skipRules = {}) const;
00096
00110     bool cleanupExternalHtml(
00111         const QString & inputHtml, QString & cleanedUpHtml,
00112         ErrorString & errorDescription) const;
00113
00127     bool htmlToQTextDocument(
00128         const QString & html, QTextDocument & doc,
00129         ErrorString & errorDescription,
00130         const QVector<SkipHtmlElementRule> & skipRules = {}) const;
00131
00132     struct NoteContentToHtmlExtraData
00133     {
00134         quint64 m_numEnToDoNodes = 0;
00135         quint64 m_numHyperlinkNodes = 0;
00136         quint64 m_numEnCryptNodes = 0;
00137         quint64 m_numEnDecryptedNodes = 0;
00138     };
00139
00140     bool noteContentToHtml(
00141         const QString & noteContent, QString & html,
00142         ErrorString & errorDescription,
00143         DecryptedTextManager & decryptedTextManager,
00144         NoteContentToHtmlExtraData & extraData) const;
00145
00146     bool validateEnml(
00147         const QString & enml, ErrorString & errorDescription) const;
00148
00149     bool validateAndFixupEnml(
00150         QString & enml, ErrorString & errorDescription) const;
00151
00152     static bool noteContentToPlainText(
00153         const QString & noteContent, QString & plainText,
00154         ErrorString & errorMessage);
00155
00156     static bool noteContentToListOfWords(
00157         const QString & noteContent, QStringList & listOfWorks,
00158         ErrorString & errorMessage, QString * plainText = nullptr);
00159
00160     static QStringList plainTextToListOfWords(const QString & plainText);
00161
00162     static QString toCheckBoxHtml(const bool checked, const quint64 idNumber);
00163
00164     static QString encryptedTextHtml(
00165         const QString & encryptedText, const QString & hint,
00166         const QString & cipher, const size_t keyLength,
00167         const quint64 enCryptIndex);
00168
00169     static QString decryptedTextHtml(
00170         const QString & decryptedText, const QString & encryptedText,
00171         const QString & hint, const QString & cipher, const size_t keyLength,
00172         const quint64 enDecryptedIndex);
00173
00174     static QString resourceHtml(
00175         const Resource & resource, ErrorString & errorDescription);
00176
00177     static void escapeString(QString & string, const bool simplify = true);
00178
00183     enum class EnexExportTags
00184     {
00185         Yes = 0,
00186         No
00187     };
00188
00217     bool exportNotesToEnex(
00218         const QVector<Note> & notes,
00219         const QHash<QString, QString> & tagNamesByTagLocalUids,
00220         const EnexExportTags exportTagsOption, QString & enex,
00221         ErrorString & errorDescription, const QString & version = {}) const;
00222
00241     bool importEnex(
00242         const QString & enex, QVector<Note> & notes,
00243         QHash<QString, QStringList> & tagNamesByNoteLocalUid,
00244         ErrorString & errorDescription) const;
00245
00246 private:
00247     Q_DISABLE_COPY(ENMLConverter)
00248
00249 private:
00250     ENMLConverterPrivate * const d_ptr;
00251     Q_DECLARE_PRIVATE(ENMLConverter)
00252 };
00253

```

```

00254 } // namespace quentier
00255
00256 #endif // LIB_QUENTIER_ENML_ENML_CONVERTER_H

```

## 6.3 HTMLCleaner.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_ENML_HTML_CLEANER_H
00020 #define LIB_QUENTIER_ENML_HTML_CLEANER_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025
00026 namespace quentier {
00027
00028 class QUENTIER_EXPORT HTMLCleaner
00029 {
00030 public:
00031     HTMLCleaner();
00032     virtual ~HTMLCleaner();
00033
00034     bool htmlToXml(
00035         const QString & html, QString & output, QString & errorDescription);
00036
00037     bool htmlToXhtml(
00038         const QString & html, QString & output, QString & errorDescription);
00039
00040     bool cleanupHtml(QString & html, QString & errorDescription);
00041
00042 private:
00043     Q_DISABLE_COPY(HTMLCleaner)
00044
00045 private:
00046     class Impl;
00047     Impl * m_impl;
00048 };
00049
00050 } // namespace quentier
00051
00052 #endif // LIB_QUENTIER_ENML_HTML_CLEANER_H

```

## 6.4 ApplicationSettingsInitializationException.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018

```

```

00019 #ifndef LIB_QUENTIER_EXCEPTION_APPLICATION_SETTINGS_INITIALIZATION_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_APPLICATION_SETTINGS_INITIALIZATION_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT ApplicationSettingsInitializationException :
00027     public IQuentierException
00028 {
00029 public:
00030     explicit ApplicationSettingsInitializationException(
00031         const ErrorString & message);
00032
00033 protected:
00034     virtual const QString exceptionDisplayName() const override;
00035 };
00036
00037 } // namespace quentier
00038
00039 #endif // LIB_QUENTIER_EXCEPTION_APPLICATION_SETTINGS_INITIALIZATION_EXCEPTION_H

```

## 6.5 DatabaseLockedException.h

```

00001 /*
00002  * Copyright 2016-2019 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_DATABASE_LOCKED_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_DATABASE_LOCKED_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT DatabaseLockedException : public IQuentierException
00027 {
00028 public:
00029     explicit DatabaseLockedException(const ErrorString & message);
00030
00031 protected:
00032     virtual const QString exceptionDisplayName() const override;
00033 };
00034
00035 } // namespace quentier
00036
00037 #endif // LIB_QUENTIER_EXCEPTION_DATABASE_LOCKED_EXCEPTION_H

```

## 6.6 DatabaseLockFailedException.h

```

00001 /*
00002  * Copyright 2016-2019 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License

```

```

00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_DATABASE_LOCK_FAILED_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_DATABASE_LOCK_FAILED_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT DatabaseLockFailedException : public IQuentierException
00027 {
00028 public:
00029     explicit DatabaseLockFailedException(const ErrorString & message);
00030
00031 protected:
00032     virtual const QString exceptionDisplayName() const override;
00033 };
00034
00035 } // namespace quentier
00036
00037 #endif // LIB_QUENTIER_EXCEPTION_DATABASE_LOCK_FAILED_EXCEPTION_H

```

## 6.7 DatabaseOpeningException.h

```

00001 /*
00002  * Copyright 2016-2019 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_DATABASE_OPENING_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_DATABASE_OPENING_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT DatabaseOpeningException : public IQuentierException
00027 {
00028 public:
00029     explicit DatabaseOpeningException(const ErrorString & message);
00030
00031 protected:
00032     virtual const QString exceptionDisplayName() const override;
00033 };
00034
00035 } // namespace quentier
00036
00037 #endif // LIB_QUENTIER_EXCEPTION_DATABASE_OPENING_EXCEPTION_H

```

## 6.8 DatabaseRequestException.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *

```

```

00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_DATABASE_SQL_ERROR_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_DATABASE_SQL_ERROR_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00030 class QUENTIER_EXPORT DatabaseRequestException : public IQuentierException
00031 {
00032 public:
00033     explicit DatabaseRequestException(const ErrorString & message);
00034
00035 protected:
00036     virtual const QString exceptionDisplayName() const override;
00037 };
00038
00039 } // namespace quentier
00040
00041 #endif // LIB_QUENTIER_EXCEPTION_DATABASE_SQL_ERROR_EXCEPTION_H

```

## 6.9 EmptyDataElementException.h

```

00001 /*
00002  * Copyright 2016-2019 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_EMPTY_DATA_ELEMENT_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_EMPTY_DATA_ELEMENT_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT EmptyDataElementException : public IQuentierException
00027 {
00028 public:
00029     explicit EmptyDataElementException(const ErrorString & message);
00030
00031 protected:
00032     virtual const QString exceptionDisplayName() const override;
00033 };
00034
00035 } // namespace quentier
00036
00037 #endif // LIB_QUENTIER_EXCEPTION_EMPTY_DATA_ELEMENT_EXCEPTION_H

```

## 6.10 IQuentierException.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.

```

```

00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_I_QUENTIER_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_I_QUENTIER_EXCEPTION_H
00021
00022 #include <exception>
00023 #include <quentier/types/ErrorMessage.h>
00024 #include <quentier/utility/Printable.h>
00025
00026 namespace quentier {
00027
00036 class QUINTIER_EXPORT IQuentierException :
00037     public Printable,
00038     public std::exception
00039 {
00040 public:
00041     explicit IQuentierException(const ErrorMessage & message);
00042
00043     virtual ~IQuentierException() noexcept override;
00044
00045     QString localizedErrorMessage() const;
00046     QString nonLocalizedErrorMessage() const;
00047
00048     virtual const char * what() const noexcept override;
00049
00050     virtual QTextStream & print(QTextStream & strm) const override;
00051
00052 protected:
00053     IQuentierException(const IQuentierException & other);
00054     IQuentierException & operator=(const IQuentierException & other);
00055
00056     virtual const QString exceptionDisplayName() const = 0;
00057
00058 private:
00059     IQuentierException() = delete;
00060
00061     ErrorMessage m_message;
00062     char * m_whatMessage;
00063 };
00064
00065 } // namespace quentier
00066
00067 #endif // LIB_QUENTIER_EXCEPTION_I_QUENTIER_EXCEPTION_H

```

## 6.11 LocalStorageCacheManagerException.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_LOCAL_STORAGE_CACHE_MANAGER_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_LOCAL_STORAGE_CACHE_MANAGER_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUINTIER_EXPORT LocalStorageCacheManagerException :
00027     public IQuentierException
00028 {
00029 public:
00030     explicit LocalStorageCacheManagerException(const ErrorMessage & message);
00031
00032 protected:
00033     virtual const QString exceptionDisplayName() const override;
00034 };

```

```

00035
00036 } // namespace quentier
00037
00038 #endif // LIB_QUENTIER_EXCEPTION_LOCAL_STORAGE_CACHE_MANAGER_EXCEPTION_H

```

## 6.12 LoggerInitializationException.h

```

00001 /*
00002  * Copyright 2016-2019 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_LOGGER_INITIALIZATION_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_LOGGER_INITIALIZATION_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT LoggerInitializationException : public IQuentierException
00027 {
00028 public:
00029     explicit LoggerInitializationException(const ErrorString & message);
00030
00031 protected:
00032     const QString exceptionDisplayName() const override;
00033 };
00034
00035 } // namespace quentier
00036
00037 #endif // LIB_QUENTIER_EXCEPTION_LOGGER_INITIALIZATION_EXCEPTION_H

```

## 6.13 NoteEditorInitializationException.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_NOTE_EDITOR_INITIALIZATION_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_NOTE_EDITOR_INITIALIZATION_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT NoteEditorInitializationException :
00027     public IQuentierException
00028 {
00029 public:
00030     explicit NoteEditorInitializationException(const ErrorString & message);
00031
00032 protected:

```

```

00033     virtual const QString exceptionDisplayName() const override;
00034 };
00035
00036 } // namespace quentier
00037
00038 #endif // LIB_QUENTIER_EXCEPTION_NOTE_EDITOR_INITIALIZATION_EXCEPTION_H

```

## 6.14 NoteEditorPluginInitializationException.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_NOTE_EDITOR_PLUGIN_INITIALIZATION_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_NOTE_EDITOR_PLUGIN_INITIALIZATION_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT NoteEditorPluginInitializationException :
00027     public IQuentierException
00028 {
00029 public:
00030     explicit NoteEditorPluginInitializationException(
00031         const ErrorString & message);
00032
00033 protected:
00034     virtual const QString exceptionDisplayName() const override;
00035 };
00036
00037 } // namespace quentier
00038
00039 #endif // LIB_QUENTIER_EXCEPTION_NOTE_EDITOR_PLUGIN_INITIALIZATION_EXCEPTION_H

```

## 6.15 NullPtrException.h

```

00001 /*
00002  * Copyright 2016-2019 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_EXCEPTION_NULL_PTR_EXCEPTION_H
00020 #define LIB_QUENTIER_EXCEPTION_NULL_PTR_EXCEPTION_H
00021
00022 #include <quentier/exception/IQuentierException.h>
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT NullPtrException : public IQuentierException
00027 {
00028 public:

```

```

00029     explicit NullPointerException(const ErrorString & message);
00030
00031 protected:
00032     virtual const QString exceptionDisplayName() const override;
00033 };
00034
00035 } // namespace quentier
00036
00037 #endif // LIB_QUENTIER_EXCEPTION_NULL_PTR_EXCEPTION_H

```

## 6.16 DefaultLocalStorageCacheExpiryChecker.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOCAL_STORAGE_DEFAULT_LOCAL_STORAGE_CACHE_EXPIRY_CHECKER_H
00020 #define LIB_QUENTIER_LOCAL_STORAGE_DEFAULT_LOCAL_STORAGE_CACHE_EXPIRY_CHECKER_H
00021
00022 #include <quentier/local_storage/ILocalStorageCacheExpiryChecker.h>
00023
00024 namespace quentier {
00025
00032 class QUENTIER_EXPORT DefaultLocalStorageCacheExpiryChecker :
00033     public ILocalStorageCacheExpiryChecker
00034 {
00035 public:
00036     DefaultLocalStorageCacheExpiryChecker(
00037         const LocalStorageCacheManager & cacheManager);
00038
00039     virtual ~DefaultLocalStorageCacheExpiryChecker();
00040
00045     virtual DefaultLocalStorageCacheExpiryChecker * clone() const override;
00046
00051     virtual bool checkNotes() const override;
00052
00057     virtual bool checkResources() const override;
00058
00063     virtual bool checkNotebooks() const override;
00064
00069     virtual bool checkTags() const override;
00070
00075     virtual bool checkLinkedNotebooks() const override;
00076
00081     virtual bool checkSavedSearches() const override;
00082
00088     virtual QTextStream & print(QTextStream & strm) const override;
00089
00090 private:
00091     Q_DISABLE_COPY(DefaultLocalStorageCacheExpiryChecker)
00092 };
00093
00094 } // namespace quentier
00095
00096 #endif // LIB_QUENTIER_LOCAL_STORAGE_DEFAULT_LOCAL_STORAGE_CACHE_EXPIRY_CHECKER_H

```

## 6.17 ILocalStorageCacheExpiryChecker.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by

```

```

00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOCAL_STORAGE_I_LOCAL_STORAGE_CACHE_EXPIRY_CHECKER_H
00020 #define LIB_QUENTIER_LOCAL_STORAGE_I_LOCAL_STORAGE_CACHE_EXPIRY_CHECKER_H
00021
00022 #include <quentier/utility/Printable.h>
00023
00024 namespace quentier {
00025
00026 QT_FORWARD_DECLARE_CLASS(LocalStorageCacheManager)
00027
00028
00029 class QUINTIER_EXPORT ILocalStorageCacheExpiryChecker : public Printable
00030 {
00031 public:
00032     virtual ~ILocalStorageCacheExpiryChecker();
00033
00034     virtual ILocalStorageCacheExpiryChecker * clone() const = 0;
00035
00036     virtual bool checkNotes() const = 0;
00037
00038     virtual bool checkResources() const = 0;
00039
00040     virtual bool checkNotebooks() const = 0;
00041
00042     virtual bool checkTags() const = 0;
00043
00044     virtual bool checkLinkedNotebooks() const = 0;
00045
00046     virtual bool checkSavedSearches() const = 0;
00047
00048     virtual QTextStream & print(QTextStream & strm) const = 0;
00049
00050 protected:
00051     ILocalStorageCacheExpiryChecker(
00052         const LocalStorageCacheManager & cacheManager);
00053
00054     const LocalStorageCacheManager & m_localStorageCacheManager;
00055
00056 private:
00057     ILocalStorageCacheExpiryChecker() = delete;
00058     Q_DISABLE_COPY(ILocalStorageCacheExpiryChecker)
00059 };
00060
00061 } // namespace quentier
00062
00063 #endif // LIB_QUENTIER_LOCAL_STORAGE_I_LOCAL_STORAGE_CACHE_EXPIRY_CHECKER_H

```

## 6.18 ILocalStoragePatch.h

```

00001 /*
00002  * Copyright 2018-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOCAL_STORAGE_I_LOCAL_STORAGE_PATCH_H
00020 #define LIB_QUENTIER_LOCAL_STORAGE_I_LOCAL_STORAGE_PATCH_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QObject>

```

```

00025
00026 namespace quentier {
00027
00028 QT_FORWARD_DECLARE_CLASS(ErrorString)
00029 QT_FORWARD_DECLARE_CLASS(LocalStorageDatabaseUpgrader)
00030
00031
00037 class QUENTIER_EXPORT ILocalStoragePatch : public QObject
00038 {
00039     Q_OBJECT
00040 protected:
00041     explicit ILocalStoragePatch(QObject * parent = nullptr);
00042
00043 public:
00044     virtual ~ILocalStoragePatch();
00045
00050     virtual int fromVersion() const = 0;
00051
00056     virtual int toVersion() const = 0;
00057
00061     virtual QString patchShortDescription() const = 0;
00062
00066     virtual QString patchLongDescription() const = 0;
00067
00078     virtual bool backupLocalStorage(ErrorString & errorDescription) = 0;
00079
00091     virtual bool restoreLocalStorageFromBackup(
00092         ErrorString & errorDescription) = 0;
00093
00106     virtual bool removeLocalStorageBackup(ErrorString & errorDescription) = 0;
00107
00116     virtual bool apply(ErrorString & errorDescription) = 0;
00117
00118     friend class LocalStorageDatabaseUpgrader;
00119
00120 Q_SIGNALS:
00126     void progress(double progress);
00127
00133     void backupProgress(double progress);
00134
00141     void restoreBackupProgress(double progress);
00142 };
00143
00144 } // namespace quentier
00145
00146 #endif // LIB_QUENTIER_LOCAL_STORAGE_I_LOCAL_STORAGE_PATCH_H

```

## 6.19 Lists.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOCAL_STORAGE_LISTS_H
00020 #define LIB_QUENTIER_LOCAL_STORAGE_LISTS_H
00021
00022 #include <QVector>
00023
00024 namespace quentier {
00025
00026 QT_FORWARD_DECLARE_CLASS(LinkedNotebook)
00027 QT_FORWARD_DECLARE_CLASS(Note)
00028 QT_FORWARD_DECLARE_CLASS(Notebook)
00029 QT_FORWARD_DECLARE_CLASS(Resource)
00030 QT_FORWARD_DECLARE_CLASS(SavedSearch)
00031 QT_FORWARD_DECLARE_CLASS(SharedNotebook)
00032 QT_FORWARD_DECLARE_CLASS(Tag)
00033 QT_FORWARD_DECLARE_CLASS(User)
00034

```

```

00035 using LinkedNotebookList = QList<LinkedNotebook>;
00036 using NoteList = QList<Note>;
00037 using NotebookList = QList<Notebook>;
00038 using ResourceList = QList<Resource>;
00039 using SavedSearchList = QList<SavedSearch>;
00040 using SharedNotebookList = QList<SharedNotebook>;
00041 using TagList = QList<Tag>;
00042 using UserList = QList<User>;
00043
00044 } // namespace quantier
00045
00046 #endif // LIB_QUENTIER_LOCAL_STORAGE_LISTS_H

```

## 6.20 LocalStorageCacheManager.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_CACHE_MANAGER_H
00020 #define LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_CACHE_MANAGER_H
00021
00022 #include <quantier/utility/Printable.h>
00023
00024 #include <memory>
00025
00026 namespace quantier {
00027
00028 QT_FORWARD_DECLARE_CLASS(LinkedNotebook)
00029 QT_FORWARD_DECLARE_CLASS(Note)
00030 QT_FORWARD_DECLARE_CLASS(Notebook)
00031 QT_FORWARD_DECLARE_CLASS(Resource)
00032 QT_FORWARD_DECLARE_CLASS(SavedSearch)
00033 QT_FORWARD_DECLARE_CLASS(Tag)
00034
00035 QT_FORWARD_DECLARE_CLASS(ILocalStorageCacheExpiryChecker)
00036
00037 QT_FORWARD_DECLARE_CLASS(LocalStorageCacheManagerPrivate)
00038 class QUENTIER_EXPORT LocalStorageCacheManager : public Printable
00039 {
00040 public:
00041     LocalStorageCacheManager();
00042     virtual ~LocalStorageCacheManager();
00043
00044     enum WhichUid
00045     {
00046         LocalUid,
00047         Guid
00048     };
00049
00050     void clear();
00051     bool empty() const;
00052
00053     // Notes cache
00054     size_t numCachedNotes() const;
00055     void cacheNote(const Note & note);
00056     void expungeNote(const Note & note);
00057
00058     const Note * findNote(const QString & uid, const WhichUid whichUid) const;
00059
00060     void clearAllNotes();
00061
00062     // Resources cache
00063     size_t numCachedResources() const;
00064     void cacheResource(const Resource & resource);
00065     void expungeResource(const Resource & resource);
00066
00067     const Resource * findResource(
00068         const QString & id, const WhichUid whichUid) const;

```

```

00069
00070     void clearAllResources();
00071
00072     // Notebooks cache
00073     size_t numCachedNotebooks() const;
00074     void cacheNotebook(const Notebook & notebook);
00075     void expungeNotebook(const Notebook & notebook);
00076
00077     const Notebook * findNotebook(
00078         const QString & uid, const WhichUid whichUid) const;
00079
00080     const Notebook * findNotebookByName(const QString & name) const;
00081     void clearAllNotebooks();
00082
00083     // Tags cache
00084     size_t numCachedTags() const;
00085     void cacheTag(const Tag & tag);
00086     void expungeTag(const Tag & tag);
00087     const Tag * findTag(const QString & uid, const WhichUid whichUid) const;
00088     const Tag * findTagByName(const QString & name) const;
00089     void clearAllTags();
00090
00091     // Linked notebooks cache
00092     size_t numCachedLinkedNotebooks() const;
00093     void cacheLinkedNotebook(const LinkedNotebook & linkedNotebook);
00094     void expungeLinkedNotebook(const LinkedNotebook & linkedNotebook);
00095     const LinkedNotebook * findLinkedNotebook(const QString & guid) const;
00096     void clearAllLinkedNotebooks();
00097
00098     // Saved searches cache
00099     size_t numCachedSavedSearches() const;
00100     void cacheSavedSearch(const SavedSearch & savedSearch);
00101     void expungeSavedSearch(const SavedSearch & savedSearch);
00102
00103     const SavedSearch * findSavedSearch(
00104         const QString & uid, const WhichUid whichUid) const;
00105
00106     const SavedSearch * findSavedSearchByName(const QString & name) const;
00107     void clearAllSavedSearches();
00108
00109     void installCacheExpiryFunction(
00110         const ILocalStorageCacheExpiryChecker & checker);
00111
00112     virtual QTextStream & print(QTextStream & strm) const override;
00113
00114 private:
00115     Q_DISABLE_COPY(LocalStorageCacheManager)
00116
00117     LocalStorageCacheManagerPrivate * const d_ptr;
00118     Q_DECLARE_PRIVATE(LocalStorageCacheManager)
00119 };
00120
00121 } // namespace quantier
00122
00123 #endif // LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_CACHE_MANAGER_H

```

## 6.21 LocalStorageManager.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_MANAGER_H
00020 #define LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_MANAGER_H
00021
00022 #include <quantier/local_storage/Lists.h>
00023 #include <quantier/local_storage/NoteSearchQuery.h>
00024 #include <quantier/types/Account.h>
00025 #include <quantier/types/ErrorString.h>

```

```

00026 #include <quentier/utility/Linkage.h>
00027
00028 #include <QHash>
00029 #include <QString>
00030 #include <QVector>
00031
00032 #include <stdint>
00033 #include <memory>
00034 #include <utility>
00035
00036 namespace qevercloud {
00037
00038 QT_FORWARD_DECLARE_STRUCT(Accounting)
00039 QT_FORWARD_DECLARE_STRUCT(BusinessUserInfo)
00040 QT_FORWARD_DECLARE_STRUCT(NoteAttributes)
00041 QT_FORWARD_DECLARE_STRUCT(NotebookRestrictions)
00042 QT_FORWARD_DECLARE_STRUCT(ResourceAttributes)
00043 QT_FORWARD_DECLARE_STRUCT(PremiumInfo)
00044 QT_FORWARD_DECLARE_STRUCT(SharedNotebook)
00045 QT_FORWARD_DECLARE_STRUCT(UserAttributes)
00046
00047 } // namespace qevercloud
00048
00049 namespace quentier {
00050
00051 QT_FORWARD_DECLARE_CLASS(ILocalStoragePatch)
00052 QT_FORWARD_DECLARE_CLASS(LocalStorageManagerPrivate)
00053
00054 class QUENTIER_EXPORT LocalStorageManager : public QObject
00055 {
00056     Q_OBJECT
00057 public:
00063     enum class StartupOption
00064     {
00072         ClearDatabase = 1,
00081         OverrideLock = 2
00082     };
00083     Q_DECLARE_FLAGS(StartupOptions, StartupOption)
00084
00085     friend QUENTIER_EXPORT QTextStream & operator<<(
00086         QTextStream & strm, const StartupOption option);
00087
00088     friend QUENTIER_EXPORT QDebug & operator<<(
00089         QDebug & dbg, const StartupOption option);
00090
00091     friend QUENTIER_EXPORT QTextStream & operator<<(
00092         QTextStream & strm, const StartupOptions options);
00093
00094     friend QUENTIER_EXPORT QDebug & operator<<(
00095         QDebug & dbg, const StartupOptions options);
00096
00108     explicit LocalStorageManager(
00109         const Account & account,
00110 #if QT_VERSION >= QT_VERSION_CHECK(5, 15, 0)
00111         const StartupOptions options = {},
00112 #else
00113         const StartupOptions options = 0,
00114 #endif
00115         QObject * parent = nullptr);
00116
00117     virtual ~LocalStorageManager() override;
00118
00119     Q_SIGNALS:
00130     void upgradeProgress(double progress);
00131
00132 public:
00143     enum class ListObjectsOption
00144     {
00145         ListAll = 0,
00146         ListDirty = 1,
00147         ListNonDirty = 2,
00148         ListElementsWithoutGuid = 4,
00149         ListElementsWithGuid = 8,
00150         ListLocal = 16,
00151         ListNonLocal = 32,
00152         ListFavoritedElements = 64,
00153         ListNonFavoritedElements = 128
00154     };
00155     Q_DECLARE_FLAGS(ListObjectsOptions, ListObjectsOption)
00156
00157     friend QUENTIER_EXPORT QTextStream & operator<<(
00158         QTextStream & strm, const ListObjectsOption option);
00159
00160     friend QUENTIER_EXPORT QDebug & operator<<(
00161         QDebug & dbg, const ListObjectsOption option);
00162
00163     friend QUENTIER_EXPORT QTextStream & operator<<(

```

```

00164         QTextStream & strm, const ListObjectsOptions options);
00165
00166     friend QUENTIER_EXPORT QDebug & operator«(
00167         QDebug & dbg, const ListObjectsOptions options);
00168
00185     void switchUser(
00186         const Account & account,
00187     #if QT_VERSION >= QT_VERSION_CHECK(5, 15, 0)
00188         const StartupOptions options = {});
00189     #else
00190         const StartupOptions options = 0);
00191     #endif
00192
00216     bool isLocalStorageVersionTooHigh(ErrorString & errorDescription);
00217
00242     bool localStorageRequiresUpgrade(ErrorString & errorDescription);
00243
00256     QVector<std::shared_ptr<ILocalStoragePatch>> requiredLocalStoragePatches();
00257
00270     qint32 localStorageVersion(ErrorString & errorDescription);
00271
00279     qint32 highestSupportedLocalStorageVersion() const;
00280
00291     int userCount(ErrorString & errorDescription) const;
00292
00307     bool addUser(const User & user, ErrorString & errorDescription);
00308
00323     bool updateUser(const User & user, ErrorString & errorDescription);
00324
00337     bool findUser(User & user, ErrorString & errorDescription) const;
00338
00348     bool deleteUser(const User & user, ErrorString & errorDescription);
00349
00360     bool expungeUser(const User & user, ErrorString & errorDescription);
00361
00372     int notebookCount(ErrorString & errorDescription) const;
00373
00393     bool addNotebook(Notebook & notebook, ErrorString & errorDescription);
00394
00413     bool updateNotebook(Notebook & notebook, ErrorString & errorDescription);
00414
00444     bool findNotebook(
00445         Notebook & notebook, ErrorString & errorDescription) const;
00446
00457     bool findDefaultNotebook(
00458         Notebook & notebook, ErrorString & errorDescription) const;
00459
00470     bool findLastUsedNotebook(
00471         Notebook & notebook, ErrorString & errorDescription) const;
00472
00484     bool findDefaultOrLastUsedNotebook(
00485         Notebook & notebook, ErrorString & errorDescription) const;
00486
00492     enum class OrderDirection
00493     {
00494         Ascending = 0,
00495         Descending
00496     };
00497
00498     friend QUENTIER_EXPORT QTextStream & operator«(
00499         QTextStream & strm, const OrderDirection orderDirection);
00500
00501     friend QUENTIER_EXPORT QDebug & operator«(
00502         QDebug & dbg, const OrderDirection orderDirection);
00503
00508     enum class ListNotebooksOrder
00509     {
00510         ByUpdateSequenceNumber = 0,
00511         ByNotebookName,
00512         ByCreationTimestamp,
00513         ByModificationTimestamp,
00514         NoOrder
00515     };
00516
00517     friend QUENTIER_EXPORT QTextStream & operator«(
00518         QTextStream & strm, const ListNotebooksOrder order);
00519
00520     friend QUENTIER_EXPORT QDebug & operator«(
00521         QDebug & dbg, const ListNotebooksOrder order);
00522
00556     QList<Notebook> listAllNotebooks(
00557         ErrorString & errorDescription, const size_t limit = 0,
00558         const size_t offset = 0,
00559         const ListNotebooksOrder order = ListNotebooksOrder::NoOrder,
00560         const OrderDirection orderDirection = OrderDirection::Ascending,
00561         const QString & linkedNotebookGuid = QString()) const;
00562

```

```

00596     QList<Notebook> listNotebooks(
00597         const ListObjectsOptions flag, QString & errorDescription,
00598         const size_t limit = 0, const size_t offset = 0,
00599         const ListNotebooksOrder order = ListNotebooksOrder::NoOrder,
00600         const OrderDirection orderDirection = OrderDirection::Ascending,
00601         const QString & linkedNotebookGuid = QString()) const;
00602
00615     QList<SharedNotebook> listAllSharedNotebooks(
00616         QString & errorDescription) const;
00617
00634     QList<SharedNotebook> listSharedNotebooksPerNotebookGuid(
00635         const QString & notebookGuid, QString & errorDescription) const;
00636
00660     bool expungeNotebook(Notebook & notebook, QString & errorDescription);
00661
00671     int linkedNotebookCount(QString & errorDescription) const;
00672
00686     bool addLinkedNotebook(
00687         const LinkedNotebook & linkedNotebook, QString & errorDescription);
00688
00701     bool updateLinkedNotebook(
00702         const LinkedNotebook & linkedNotebook, QString & errorDescription);
00703
00719     bool findLinkedNotebook(
00720         LinkedNotebook & linkedNotebook, QString & errorDescription) const;
00721
00726     enum class ListLinkedNotebooksOrder
00727     {
00728         ByUpdateSequenceNumber = 0,
00729         ByShareName,
00730         ByUsername,
00731         NoOrder
00732     };
00733
00734     friend QUINTIER_EXPORT QTextStream & operator<<(
00735         QTextStream & strm, const ListLinkedNotebooksOrder order);
00736
00737     friend QUINTIER_EXPORT QDebug & operator<<(
00738         QDebug & strm, const ListLinkedNotebooksOrder order);
00739
00763     QList<LinkedNotebook> listAllLinkedNotebooks(
00764         QString & errorDescription, const size_t limit = 0,
00765         const size_t offset = 0,
00766         const ListLinkedNotebooksOrder order =
00767             ListLinkedNotebooksOrder::NoOrder,
00768         const OrderDirection orderDirection = OrderDirection::Ascending) const;
00769
00797     QList<LinkedNotebook> listLinkedNotebooks(
00798         const ListObjectsOptions flag, QString & errorDescription,
00799         const size_t limit = 0, const size_t offset = 0,
00800         const ListLinkedNotebooksOrder order =
00801             ListLinkedNotebooksOrder::NoOrder,
00802         const OrderDirection orderDirection = OrderDirection::Ascending) const;
00803
00821     bool expungeLinkedNotebook(
00822         const LinkedNotebook & linkedNotebook, QString & errorDescription);
00823
00828     enum class NoteCountOption
00829     {
00830         IncludeNonDeletedNotes = 1,
00831         IncludeDeletedNotes = 2
00832     };
00833     Q_DECLARE_FLAGS(NoteCountOptions, NoteCountOption)
00834
00835     friend QUINTIER_EXPORT QTextStream & operator<<(
00836         QTextStream & strm, const NoteCountOption option);
00837
00838     friend QUINTIER_EXPORT QDebug & operator<<(
00839         QDebug & dbg, const NoteCountOption option);
00840
00841     friend QUINTIER_EXPORT QTextStream & operator<<(
00842         QTextStream & strm, const NoteCountOptions options);
00843
00844     friend QUINTIER_EXPORT QDebug & operator<<(
00845         QDebug & strm, const NoteCountOptions options);
00846
00859     int noteCount(
00860         QString & errorDescription,
00861         const NoteCountOptions options =
00862             NoteCountOption::IncludeNonDeletedNotes) const;
00863
00880     int noteCountPerNotebook(
00881         const Notebook & notebook, QString & errorDescription,
00882         const NoteCountOptions options =
00883             NoteCountOption::IncludeNonDeletedNotes) const;
00884
00901     int noteCountPerTag(

```

```

00902         const Tag & tag, ErrorString & errorDescription,
00903         const NoteCountOptions options =
00904             NoteCountOption::IncludeNonDeletedNotes) const;
00905
00923     bool noteCountsPerAllTags(
00924         QHash<QString, int> & noteCountsPerTagLocalUid,
00925         ErrorString & errorDescription,
00926         const NoteCountOptions options =
00927             NoteCountOption::IncludeNonDeletedNotes) const;
00928
00947     int noteCountPerNotebooksAndTags(
00948         const QStringList & notebookLocalUids, const QStringList & tagLocalUids,
00949         ErrorString & errorDescription,
00950         const NoteCountOptions options =
00951             NoteCountOption::IncludeNonDeletedNotes) const;
00952
00970     bool addNote(Note & note, ErrorString & errorDescription);
00971
00981     enum class UpdateNoteOption
00982     {
00983         UpdateResourceMetadata = 1,
00984         UpdateResourceBinaryData = 2,
00985         UpdateTags = 4
00986     };
01000     Q_DECLARE_FLAGS(UpdateNoteOptions, UpdateNoteOption)
01001
01003     friend QUINTIER_EXPORT QTextStream & operator<<(
01004         QTextStream & strm, const UpdateNoteOption option);
01005
01006     friend QUINTIER_EXPORT QDebug & operator<<(
01007         QDebug & strm, const UpdateNoteOption option);
01008
01009     friend QUINTIER_EXPORT QTextStream & operator<<(
01010         QTextStream & strm, const UpdateNoteOptions options);
01011
01012     friend QUINTIER_EXPORT QDebug & operator<<(
01013         QDebug & strm, const UpdateNoteOptions options);
01014
01056     bool updateNote(
01057         Note & note, const UpdateNoteOptions options,
01058         ErrorString & errorDescription);
01059
01071     enum class GetNoteOption
01072     {
01073         WithResourceMetadata = 1,
01074         WithResourceBinaryData = 2
01075     };
01086     Q_DECLARE_FLAGS(GetNoteOptions, GetNoteOption)
01087
01089     friend QUINTIER_EXPORT QTextStream & operator<<(
01090         QTextStream & strm, const GetNoteOption option);
01091
01092     friend QUINTIER_EXPORT QDebug & operator<<(
01093         QDebug & dbg, const GetNoteOption option);
01094
01095     friend QUINTIER_EXPORT QTextStream & operator<<(
01096         QTextStream & strm, const GetNoteOptions options);
01097
01098     friend QUINTIER_EXPORT QDebug & operator<<(
01099         QDebug & strm, const GetNoteOptions options);
01100
01110     bool findNote(
01111         Note & note, const GetNoteOptions options,
01112         ErrorString & errorDescription) const;
01113
01118     enum class ListNotesOrder
01119     {
01120         ByUpdateSequenceNumber = 0,
01121         ByTitle,
01122         ByCreationTimestamp,
01123         ByModificationTimestamp,
01124         ByDeletionTimestamp,
01125         ByAuthor,
01126         BySource,
01127         BySourceApplication,
01128         ByReminderTime,
01129         ByPlaceName,
01130         NoOrder
01131     };
01132
01133     friend QUINTIER_EXPORT QTextStream & operator<<(
01134         QTextStream & strm, const ListNotesOrder order);
01135
01136     friend QUINTIER_EXPORT QDebug & operator<<(
01137         QDebug & strm, const ListNotesOrder order);
01138
01167     QList<Note> listNotesPerNotebook(

```

```

01168     const Notebook & notebook, const GetNoteOptions options,
01169     ErrorString & errorDescription,
01170     const ListObjectsOptions & flag = ListObjectsOption::ListAll,
01171     const size_t limit = 0, const size_t offset = 0,
01172     const ListNotesOrder & order = ListNotesOrder::NoOrder,
01173     const OrderDirection & orderDirection =
01174         OrderDirection::Ascending) const;
01175
01203 QList<Note> listNotesPerTag(
01204     const Tag & tag, const GetNoteOptions options,
01205     ErrorString & errorDescription,
01206     const ListObjectsOptions & flag = ListObjectsOption::ListAll,
01207     const size_t limit = 0, const size_t offset = 0,
01208     const ListNotesOrder & order = ListNotesOrder::NoOrder,
01209     const OrderDirection & orderDirection =
01210         OrderDirection::Ascending) const;
01211
01242 QList<Note> listNotesPerNotebooksAndTags(
01243     const QStringList & notebookLocalUids, const QStringList & tagLocalUids,
01244     const GetNoteOptions options, ErrorString & errorDescription,
01245     const ListObjectsOptions & flag = ListObjectsOption::ListAll,
01246     const size_t limit = 0, const size_t offset = 0,
01247     const ListNotesOrder & order = ListNotesOrder::NoOrder,
01248     const OrderDirection & orderDirection =
01249         OrderDirection::Ascending) const;
01250
01282 QList<Note> listNotesByLocalUids(
01283     const QStringList & noteLocalUids, const GetNoteOptions options,
01284     ErrorString & errorDescription,
01285     const ListObjectsOptions & flag = ListObjectsOption::ListAll,
01286     const size_t limit = 0, const size_t offset = 0,
01287     const ListNotesOrder & order = ListNotesOrder::NoOrder,
01288     const OrderDirection & orderDirection =
01289         OrderDirection::Ascending) const;
01290
01325 QList<Note> listNotes(
01326     const ListObjectsOptions flag, const GetNoteOptions options,
01327     ErrorString & errorDescription, const size_t limit = 0,
01328     const size_t offset = 0,
01329     const ListNotesOrder order = ListNotesOrder::NoOrder,
01330     const OrderDirection orderDirection = OrderDirection::Ascending,
01331     const QString & linkedNotebookGuid = QString()) const;
01332
01344 QStringList findNoteLocalUidsWithSearchQuery(
01345     const NoteSearchQuery & noteSearchQuery,
01346     ErrorString & errorDescription) const;
01347
01363 NoteList findNotesWithSearchQuery(
01364     const NoteSearchQuery & noteSearchQuery, const GetNoteOptions options,
01365     ErrorString & errorDescription) const;
01366
01384 bool expungeNote(Note & note, ErrorString & errorDescription);
01385
01395 int tagCount(ErrorString & errorDescription) const;
01396
01410 bool addTag(Tag & tag, ErrorString & errorDescription);
01411
01429 bool updateTag(Tag & tag, ErrorString & errorDescription);
01430
01455 bool findTag(Tag & tag, ErrorString & errorDescription) const;
01456
01461 enum class ListTagsOrder
01462 {
01463     ByUpdateSequenceNumber,
01464     ByName,
01465     NoOrder
01466 };
01467
01468 friend QUINTIER_EXPORT QTextStream & operator<<(
01469     QTextStream & strm, const ListTagsOrder order);
01470
01471 friend QUINTIER_EXPORT QDebug & operator<<(
01472     QDebug & strm, const ListTagsOrder order);
01473
01502 QList<Tag> listAllTagsPerNote(
01503     const Note & note, ErrorString & errorDescription,
01504     const ListObjectsOptions & flag = ListObjectsOption::ListAll,
01505     const size_t limit = 0, const size_t offset = 0,
01506     const ListTagsOrder & order = ListTagsOrder::NoOrder,
01507     const OrderDirection & orderDirection =
01508         OrderDirection::Ascending) const;
01509
01540 QList<Tag> listAllTags(
01541     ErrorString & errorDescription, const size_t limit = 0,
01542     const size_t offset = 0,
01543     const ListTagsOrder order = ListTagsOrder::NoOrder,
01544     const OrderDirection orderDirection = OrderDirection::Ascending,

```

```

01545         const QString & linkedNotebookGuid = QString()) const;
01546
01579     QList<Tag> listTags(
01580         const ListObjectsOptions flag, QString & errorDescription,
01581         const size_t limit = 0, const size_t offset = 0,
01582         const ListTagsOrder & order = ListTagsOrder::NoOrder,
01583         const OrderDirection orderDirection = OrderDirection::Ascending,
01584         const QString & linkedNotebookGuid = QString()) const;
01585
01624     QList<std::pair<Tag, QStringList>> listTagsWithNoteLocalUids(
01625         const ListObjectsOptions flag, QString & errorDescription,
01626         const size_t limit = 0, const size_t offset = 0,
01627         const ListTagsOrder & order = ListTagsOrder::NoOrder,
01628         const OrderDirection orderDirection = OrderDirection::Ascending,
01629         const QString & linkedNotebookGuid = QString()) const;
01630
01655     bool expungeTag(
01656         Tag & tag, QStringList & expungedChildTagLocalUids,
01657         QString & errorDescription);
01658
01669     bool expungeNotelessTagsFromLinkedNotebooks(QString & errorDescription);
01670
01682     int enResourceCount(QString & errorDescription) const;
01683
01700     bool addEnResource(Resource & resource, QString & errorDescription);
01701
01722     bool updateEnResource(Resource & resource, QString & errorDescription);
01723
01734     enum class GetResourceOption
01735     {
01740         WithBinaryData = 1
01741     };
01742     Q_DECLARE_FLAGS(GetResourceOptions, GetResourceOption)
01743
01744     friend QUINTIER_EXPORT QTextStream & operator<<(
01745         QTextStream & strm, const GetResourceOption option);
01746
01747     friend QUINTIER_EXPORT QDebug & operator<<(
01748         QDebug & strm, const GetResourceOption option);
01749
01750     friend QUINTIER_EXPORT QTextStream & operator<<(
01751         QTextStream & strm, const GetResourceOptions options);
01752
01753     friend QUINTIER_EXPORT QDebug & operator<<(
01754         QDebug & strm, const GetResourceOptions options);
01755
01774     bool findEnResource(
01775         Resource & resource, const GetResourceOptions options,
01776         QString & errorDescription) const;
01777
01792     bool expungeEnResource(Resource & resource, QString & errorDescription);
01793
01803     int savedSearchCount(QString & errorDescription) const;
01804
01821     bool addSavedSearch(SavedSearch & search, QString & errorDescription);
01822
01841     bool updateSavedSearch(
01842         SavedSearch & search, QString & errorDescription);
01843
01861     bool findSavedSearch(
01862         SavedSearch & search, QString & errorDescription) const;
01863
01868     enum class ListSavedSearchesOrder
01869     {
01870         ByUpdateSequenceNumber = 0,
01871         ByName,
01872         ByFormat,
01873         NoOrder
01874     };
01875
01876     friend QUINTIER_EXPORT QTextStream & operator<<(
01877         QTextStream & strm, const ListSavedSearchesOrder order);
01878
01879     friend QUINTIER_EXPORT QDebug & operator<<(
01880         QDebug & strm, const ListSavedSearchesOrder order);
01881
01904     QList<SavedSearch> listAllSavedSearches(
01905         QString & errorDescription, const size_t limit = 0,
01906         const size_t offset = 0,
01907         const ListSavedSearchesOrder order = ListSavedSearchesOrder::NoOrder,
01908         const OrderDirection orderDirection = OrderDirection::Ascending) const;
01909
01939     QList<SavedSearch> listSavedSearches(
01940         const ListObjectsOptions flag, QString & errorDescription,
01941         const size_t limit = 0, const size_t offset = 0,
01942         const ListSavedSearchesOrder order = ListSavedSearchesOrder::NoOrder,
01943         const OrderDirection orderDirection = OrderDirection::Ascending) const;

```

```

01944
01958     bool expungeSavedSearch(
01959         SavedSearch & search, QString & errorDescription);
01960
01978     qint32 accountHighUsn(
01979         const QString & linkedNotebookGuid, QString & errorDescription);
01980
01981 private:
01982     Q_DISABLE_COPY(LocalStorageManager)
01983
01984     LocalStorageManagerPrivate * const d_ptr;
01985     Q_DECLARE_PRIVATE(LocalStorageManager)
01986 };
01987
01988 Q_DECLARE_OPERATORS_FOR_FLAGS(LocalStorageManager::GetNoteOptions)
01989 Q_DECLARE_OPERATORS_FOR_FLAGS(LocalStorageManager::ListObjectsOptions)
01990 Q_DECLARE_OPERATORS_FOR_FLAGS(LocalStorageManager::StartupOptions)
01991 Q_DECLARE_OPERATORS_FOR_FLAGS(LocalStorageManager::UpdateNoteOptions)
01992
01993 } // namespace quantier
01994
01995 #endif // LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_MANAGER_H

```

## 6.22 LocalStorageManagerAsync.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_MANAGER_ASYNC_H
00020 #define LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_MANAGER_ASYNC_H
00021
00022 #include <quantier/local_storage/ILocalStorageCacheExpiryChecker.h>
00023 #include <quantier/local_storage/LocalStorageCacheManager.h>
00024 #include <quantier/local_storage/LocalStorageManager.h>
00025 #include <quantier/types/ErrorMessage.h>
00026 #include <quantier/types/LinkedNotebook.h>
00027 #include <quantier/types/Note.h>
00028 #include <quantier/types/Notebook.h>
00029 #include <quantier/types/Resource.h>
00030 #include <quantier/types/SavedSearch.h>
00031 #include <quantier/types/SharedNotebook.h>
00032 #include <quantier/types/Tag.h>
00033 #include <quantier/types/User.h>
00034
00035 #include <QObject>
00036
00037 #include <memory>
00038
00039 namespace quantier {
00040
00041 QT_FORWARD_DECLARE_CLASS(LocalStorageManagerAsyncPrivate)
00042
00043 class QUENTIER_EXPORT LocalStorageManagerAsync : public QObject
00044 {
00045     Q_OBJECT
00046 public:
00047     explicit LocalStorageManagerAsync(
00048         const Account & account,
00049         #if QT_VERSION >= QT_VERSION_CHECK(5, 15, 0)
00050         LocalStorageManager::StartupOptions options = {},
00051         #else
00052         LocalStorageManager::StartupOptions options = 0,
00053         #endif
00054         QObject * parent = nullptr);
00055
00056     virtual ~LocalStorageManagerAsync();
00057
00058     void setUseCache(const bool useCache);

```

```

00059
00060     const LocalStorageCacheManager * localStorageCacheManager() const;
00061
00062     bool installCacheExpiryFunction(
00063         const ILocalStorageCacheExpiryChecker & checker);
00064
00065     const LocalStorageManager * localStorageManager() const;
00066     LocalStorageManager * localStorageManager();
00067
00068     Q_SIGNALS:
00069     // Sent when the initialization is complete
00070     void initialized();
00071
00072     // User-related signals:
00073     void getUserCountComplete(int userCount, QUuid requestId);
00074     void getUserCountFailed(ErrorString errorDescription, QUuid requestId);
00075     void switchUserComplete(Account account, QUuid requestId);
00076
00077     void switchUserFailed(
00078         Account account, ErrorString errorDescription, QUuid requestId);
00079
00080     void addUserComplete(User user, QUuid requestId);
00081
00082     void addUserFailed(
00083         User user, ErrorString errorDescription, QUuid requestId);
00084
00085     void updateUserComplete(User user, QUuid requestId);
00086
00087     void updateUserFailed(
00088         User user, ErrorString errorDescription, QUuid requestId);
00089
00090     void findUserComplete(User foundUser, QUuid requestId);
00091
00092     void findUserFailed(
00093         User user, ErrorString errorDescription, QUuid requestId);
00094
00095     void deleteUserComplete(User user, QUuid requestId);
00096
00097     void deleteUserFailed(
00098         User user, ErrorString errorDescription, QUuid requestId);
00099
00100     void expungeUserComplete(User user, QUuid requestId);
00101
00102     void expungeUserFailed(
00103         User user, ErrorString errorDescription, QUuid requestId);
00104
00105     // Notebook-related signals:
00106     void getNotebookCountComplete(int notebookCount, QUuid requestId);
00107     void getNotebookCountFailed(ErrorString errorDescription, QUuid requestId);
00108     void addNotebookComplete(Notebook notebook, QUuid requestId);
00109
00110     void addNotebookFailed(
00111         Notebook notebook, ErrorString errorDescription, QUuid requestId);
00112
00113     void updateNotebookComplete(Notebook notebook, QUuid requestId);
00114
00115     void updateNotebookFailed(
00116         Notebook notebook, ErrorString errorDescription, QUuid requestId);
00117
00118     void findNotebookComplete(Notebook foundNotebook, QUuid requestId);
00119
00120     void findNotebookFailed(
00121         Notebook notebook, ErrorString errorDescription, QUuid requestId);
00122
00123     void findDefaultNotebookComplete(Notebook foundNotebook, QUuid requestId);
00124
00125     void findDefaultNotebookFailed(
00126         Notebook notebook, ErrorString errorDescription, QUuid requestId);
00127
00128     void findLastUsedNotebookComplete(Notebook foundNotebook, QUuid requestId);
00129
00130     void findLastUsedNotebookFailed(
00131         Notebook notebook, ErrorString errorDescription, QUuid requestId);
00132
00133     void findDefaultOrLastUsedNotebookComplete(
00134         Notebook foundNotebook, QUuid requestId);
00135
00136     void findDefaultOrLastUsedNotebookFailed(
00137         Notebook notebook, ErrorString errorDescription, QUuid requestId);
00138
00139     void listAllNotebooksComplete(
00140         size_t limit, size_t offset,
00141         LocalStorageManager::ListNotebooksOrder order,
00142         LocalStorageManager::OrderDirection orderDirection,
00143         QString linkedNotebookGuid, QList<Notebook> foundNotebooks,
00144         QUuid requestId);
00145

```

```

00146 void listAllNotebooksFailed(
00147     size_t limit, size_t offset,
00148     LocalStorageManager::ListNotebooksOrder order,
00149     LocalStorageManager::OrderDirection orderDirection,
00150     QString linkedNotebookGuid, ErrorString errorDescription,
00151     QUuid requestId);
00152
00153 void listNotebooksComplete(
00154     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00155     size_t offset, LocalStorageManager::ListNotebooksOrder order,
00156     LocalStorageManager::OrderDirection orderDirection,
00157     QString linkedNotebookGuid, QList<Notebook> foundNotebooks,
00158     QUuid requestId);
00159
00160 void listNotebooksFailed(
00161     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00162     size_t offset, LocalStorageManager::ListNotebooksOrder order,
00163     LocalStorageManager::OrderDirection orderDirection,
00164     QString linkedNotebookGuid, ErrorString errorDescription,
00165     QUuid requestId);
00166
00167 void listAllSharedNotebooksComplete(
00168     QList<SharedNotebook> foundSharedNotebooks, QUuid requestId);
00169
00170 void listAllSharedNotebooksFailed(
00171     ErrorString errorDescription, QUuid requestId);
00172
00173 void listSharedNotebooksPerNotebookGuidComplete(
00174     QString notebookGuid, QList<SharedNotebook> foundSharedNotebooks,
00175     QUuid requestId);
00176
00177 void listSharedNotebooksPerNotebookGuidFailed(
00178     QString notebookGuid, ErrorString errorDescription, QUuid requestId);
00179
00180 void expungeNotebookComplete(Notebook notebook, QUuid requestId);
00181
00182 void expungeNotebookFailed(
00183     Notebook notebook, ErrorString errorDescription, QUuid requestId);
00184
00185 // Linked notebook-related signals:
00186 void getLinkedNotebookCountComplete(
00187     int linkedNotebookCount, QUuid requestId);
00188
00189 void getLinkedNotebookCountFailed(
00190     ErrorString errorDescription, QUuid requestId);
00191
00192 void addLinkedNotebookComplete(
00193     LinkedNotebook linkedNotebook, QUuid requestId);
00194
00195 void addLinkedNotebookFailed(
00196     LinkedNotebook linkedNotebook, ErrorString errorDescription,
00197     QUuid requestId);
00198
00199 void updateLinkedNotebookComplete(
00200     LinkedNotebook linkedNotebook, QUuid requestId);
00201
00202 void updateLinkedNotebookFailed(
00203     LinkedNotebook linkedNotebook, ErrorString errorDescription,
00204     QUuid requestId);
00205
00206 void findLinkedNotebookComplete(
00207     LinkedNotebook foundLinkedNotebook, QUuid requestId);
00208
00209 void findLinkedNotebookFailed(
00210     LinkedNotebook linkedNotebook, ErrorString errorDescription,
00211     QUuid requestId);
00212
00213 void listAllLinkedNotebooksComplete(
00214     size_t limit, size_t offset,
00215     LocalStorageManager::ListLinkedNotebooksOrder order,
00216     LocalStorageManager::OrderDirection orderDirection,
00217     QList<LinkedNotebook> foundLinkedNotebooks, QUuid requestId);
00218
00219 void listAllLinkedNotebooksFailed(
00220     size_t limit, size_t offset,
00221     LocalStorageManager::ListLinkedNotebooksOrder order,
00222     LocalStorageManager::OrderDirection orderDirection,
00223     ErrorString errorDescription, QUuid requestId);
00224
00225 void listLinkedNotebooksComplete(
00226     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00227     size_t offset, LocalStorageManager::ListLinkedNotebooksOrder order,
00228     LocalStorageManager::OrderDirection orderDirection,
00229     QList<LinkedNotebook> foundLinkedNotebooks, QUuid requestId);
00230
00231 void listLinkedNotebooksFailed(
00232     LocalStorageManager::ListObjectsOptions flag, size_t limit,

```

```
00233         size_t offset, LocalStorageManager::ListLinkedNotebooksOrder order,
00234         LocalStorageManager::OrderDirection orderDirection,
00235         ErrorString errorDescription, QUuid requestId);
00236
00237     void expungeLinkedNotebookComplete(
00238         LinkedNotebook linkedNotebook, QUuid requestId);
00239
00240     void expungeLinkedNotebookFailed(
00241         LinkedNotebook linkedNotebook, ErrorString errorDescription,
00242         QUuid requestId);
00243
00244     // Note-related signals:
00245     void getNoteCountComplete(
00246         int noteCount, LocalStorageManager::NoteCountOptions options,
00247         QUuid requestId);
00248
00249     void getNoteCountFailed(
00250         ErrorString errorDescription,
00251         LocalStorageManager::NoteCountOptions options, QUuid requestId);
00252
00253     void getNoteCountPerNotebookComplete(
00254         int noteCount, Notebook notebook,
00255         LocalStorageManager::NoteCountOptions options, QUuid requestId);
00256
00257     void getNoteCountPerNotebookFailed(
00258         ErrorString errorDescription, Notebook notebook,
00259         LocalStorageManager::NoteCountOptions options, QUuid requestId);
00260
00261     void getNoteCountPerTagComplete(
00262         int noteCount, Tag tag, LocalStorageManager::NoteCountOptions options,
00263         QUuid requestId);
00264
00265     void getNoteCountPerTagFailed(
00266         ErrorString errorDescription, Tag tag,
00267         LocalStorageManager::NoteCountOptions options, QUuid requestId);
00268
00269     void getNoteCountsPerAllTagsComplete(
00270         QHash<QString, int> noteCountsPerTagLocalUids,
00271         LocalStorageManager::NoteCountOptions options, QUuid requestId);
00272
00273     void getNoteCountsPerAllTagsFailed(
00274         ErrorString errorDescription,
00275         LocalStorageManager::NoteCountOptions options, QUuid requestId);
00276
00277     void getNoteCountPerNotebooksAndTagsComplete(
00278         int noteCount, QStringList notebookLocalUids, QStringList tagLocalUids,
00279         LocalStorageManager::NoteCountOptions options, QUuid requestId);
00280
00281     void getNoteCountPerNotebooksAndTagsFailed(
00282         ErrorString errorDescription, QStringList notebookLocalUids,
00283         QStringList tagLocalUids, LocalStorageManager::NoteCountOptions options,
00284         QUuid requestId);
00285
00286     void addNoteComplete(Note note, QUuid requestId);
00287
00288     void addNoteFailed(
00289         Note note, ErrorString errorDescription, QUuid requestId);
00290
00291     void updateNoteComplete(
00292         Note note, LocalStorageManager::UpdateNoteOptions options,
00293         QUuid requestId);
00294
00295     void updateNoteFailed(
00296         Note note, LocalStorageManager::UpdateNoteOptions options,
00297         ErrorString errorDescription, QUuid requestId);
00298
00299     void findNoteComplete(
00300         Note foundNote, LocalStorageManager::GetNoteOptions options,
00301         QUuid requestId);
00302
00303     void findNoteFailed(
00304         Note note, LocalStorageManager::GetNoteOptions options,
00305         ErrorString errorDescription, QUuid requestId);
00306
00307     void listNotesPerNotebookComplete(
00308         Notebook notebook, LocalStorageManager::GetNoteOptions options,
00309         LocalStorageManager::ListObjectsOptions flag, size_t limit,
00310         size_t offset, LocalStorageManager::ListNotesOrder order,
00311         LocalStorageManager::OrderDirection orderDirection,
00312         QList<Note> foundNotes, QUuid requestId);
00313
00314     void listNotesPerNotebookFailed(
00315         Notebook notebook, LocalStorageManager::GetNoteOptions options,
00316         LocalStorageManager::ListObjectsOptions flag, size_t limit,
00317         size_t offset, LocalStorageManager::ListNotesOrder order,
00318         LocalStorageManager::OrderDirection orderDirection,
00319         ErrorString errorDescription, QUuid requestId);
```

```

00320
00321 void listNotesPerTagComplete(
00322     Tag tag, LocalStorageManager::GetNoteOptions options,
00323     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00324     size_t offset, LocalStorageManager::ListNotesOrder order,
00325     LocalStorageManager::OrderDirection orderDirection,
00326     QList<Note> foundNotes, QUuid requestId);
00327
00328 void listNotesPerTagFailed(
00329     Tag tag, LocalStorageManager::GetNoteOptions options,
00330     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00331     size_t offset, LocalStorageManager::ListNotesOrder order,
00332     LocalStorageManager::OrderDirection orderDirection,
00333     ErrorString errorDescription, QUuid requestId);
00334
00335 void listNotesPerNotebooksAndTagsComplete(
00336     QStringList notebookLocalUids, QStringList tagLocalUids,
00337     LocalStorageManager::GetNoteOptions options,
00338     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00339     size_t offset, LocalStorageManager::ListNotesOrder order,
00340     LocalStorageManager::OrderDirection orderDirection,
00341     QList<Note> foundNotes, QUuid requestId);
00342
00343 void listNotesPerNotebooksAndTagsFailed(
00344     QStringList notebookLocalUids, QStringList tagLocalUids,
00345     LocalStorageManager::GetNoteOptions options,
00346     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00347     size_t offset, LocalStorageManager::ListNotesOrder order,
00348     LocalStorageManager::OrderDirection orderDirection,
00349     ErrorString errorDescription, QUuid requestId);
00350
00351 void listNotesByLocalUidsComplete(
00352     QStringList noteLocalUids, LocalStorageManager::GetNoteOptions options,
00353     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00354     size_t offset, LocalStorageManager::ListNotesOrder order,
00355     LocalStorageManager::OrderDirection orderDirection,
00356     QList<Note> foundNotes, QUuid requestId);
00357
00358 void listNotesByLocalUidsFailed(
00359     QStringList noteLocalUids, LocalStorageManager::GetNoteOptions options,
00360     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00361     size_t offset, LocalStorageManager::ListNotesOrder order,
00362     LocalStorageManager::OrderDirection orderDirection,
00363     ErrorString errorDescription, QUuid requestId);
00364
00365 void listNotesComplete(
00366     LocalStorageManager::ListObjectsOptions flag,
00367     LocalStorageManager::GetNoteOptions options, size_t limit,
00368     size_t offset, LocalStorageManager::ListNotesOrder order,
00369     LocalStorageManager::OrderDirection orderDirection,
00370     QString linkedNotebookGuid, QList<Note> foundNotes, QUuid requestId);
00371
00372 void listNotesFailed(
00373     LocalStorageManager::ListObjectsOptions flag,
00374     LocalStorageManager::GetNoteOptions options, size_t limit,
00375     size_t offset, LocalStorageManager::ListNotesOrder order,
00376     LocalStorageManager::OrderDirection orderDirection,
00377     QString linkedNotebookGuid, ErrorString errorDescription,
00378     QUuid requestId);
00379
00380 void findNoteLocalUidsWithSearchQueryComplete(
00381     QStringList noteLocalUids, NoteSearchQuery noteSearchQuery,
00382     QUuid requestId);
00383
00384 void findNoteLocalUidsWithSearchQueryFailed(
00385     NoteSearchQuery noteSearchQuery, ErrorString errorDescription,
00386     QUuid requestId);
00387
00388 void expungeNoteComplete(Note note, QUuid requestId);
00389
00390 void expungeNoteFailed(
00391     Note note, ErrorString errorDescription, QUuid requestId);
00392
00393 // Specialized signal emitted alongside updateNoteComplete (after it)
00394 // if the update of a note causes the change of its notebook
00395 void noteMovedToAnotherNotebook(
00396     QString noteLocalUid, QString previousNotebookLocalUid,
00397     QString newNotebookLocalUid);
00398
00399 // Specialized signal emitted alongside updateNoteComplete (after it)
00400 // if the update of a note causes the change of its set of tags
00401 void noteTagListChanged(
00402     QString noteLocalUid, QStringList previousNoteTagLocalUids,
00403     QStringList newNoteTagLocalUids);
00404
00405 // Tag-related signals:
00406 void getTagCountComplete(int tagCount, QUuid requestId);

```

```

00407 void getTagCountFailed(ErrorString errorDescription, QUuid requestId);
00408 void addTagComplete(Tag tag, QUuid requestId);
00409 void addTagFailed(Tag tag, ErrorString errorDescription, QUuid requestId);
00410 void updateTagComplete(Tag tag, QUuid requestId);
00411
00412 void updateTagFailed(
00413     Tag tag, ErrorString errorDescription, QUuid requestId);
00414
00415 void linkTagWithNoteComplete(Tag tag, Note note, QUuid requestId);
00416
00417 void linkTagWithNoteFailed(
00418     Tag tag, Note note, ErrorString errorDescription, QUuid requestId);
00419
00420 void findTagComplete(Tag tag, QUuid requestId);
00421 void findTagFailed(Tag tag, ErrorString errorDescription, QUuid requestId);
00422
00423 void listAllTagsPerNoteComplete(
00424     QList<Tag> foundTags, Note note,
00425     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00426     size_t offset, LocalStorageManager::ListTagsOrder order,
00427     LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00428
00429 void listAllTagsPerNoteFailed(
00430     Note note, LocalStorageManager::ListObjectsOptions flag, size_t limit,
00431     size_t offset, LocalStorageManager::ListTagsOrder order,
00432     LocalStorageManager::OrderDirection orderDirection,
00433     ErrorString errorDescription, QUuid requestId);
00434
00435 void listAllTagsComplete(
00436     size_t limit, size_t offset, LocalStorageManager::ListTagsOrder order,
00437     LocalStorageManager::OrderDirection orderDirection,
00438     QString linkedNotebookGuid, QList<Tag> foundTags, QUuid requestId);
00439
00440 void listAllTagsFailed(
00441     size_t limit, size_t offset, LocalStorageManager::ListTagsOrder order,
00442     LocalStorageManager::OrderDirection orderDirection,
00443     QString linkedNotebookGuid, ErrorString errorDescription,
00444     QUuid requestId);
00445
00446 void listTagsComplete(
00447     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00448     size_t offset, LocalStorageManager::ListTagsOrder order,
00449     LocalStorageManager::OrderDirection orderDirection,
00450     QString linkedNotebookGuid, QList<Tag> foundTags,
00451     QUuid requestId = QUuid());
00452
00453 void listTagsFailed(
00454     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00455     size_t offset, LocalStorageManager::ListTagsOrder order,
00456     LocalStorageManager::OrderDirection orderDirection,
00457     QString linkedNotebookGuid, ErrorString errorDescription,
00458     QUuid requestId);
00459
00460 void listTagsWithNoteLocalUidsComplete(
00461     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00462     size_t offset, LocalStorageManager::ListTagsOrder order,
00463     LocalStorageManager::OrderDirection orderDirection,
00464     QString linkedNotebookGuid,
00465     QList<std::pair<Tag, QStringList>> foundTags, QUuid requestId);
00466
00467 void listTagsWithNoteLocalUidsFailed(
00468     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00469     size_t offset, LocalStorageManager::ListTagsOrder order,
00470     LocalStorageManager::OrderDirection orderDirection,
00471     QString linkedNotebookGuid, ErrorString errorDescription,
00472     QUuid requestId);
00473
00474 void expungeTagComplete(
00475     Tag tag, QStringList expungedChildTagLocalUids, QUuid requestId);
00476
00477 void expungeTagFailed(
00478     Tag tag, ErrorString errorDescription, QUuid requestId);
00479
00480 void expungeNotelessTagsFromLinkedNotebooksComplete(QUuid requestId);
00481
00482 void expungeNotelessTagsFromLinkedNotebooksFailed(
00483     ErrorString errorDescription, QUuid requestId);
00484
00485 // Resource-related signals:
00486 void getResourceCountComplete(int resourceCount, QUuid requestId);
00487 void getResourceCountFailed(ErrorString errorDescription, QUuid requestId);
00488 void addResourceComplete(Resource resource, QUuid requestId);
00489
00490 void addResourceFailed(
00491     Resource resource, ErrorString errorDescription, QUuid requestId);
00492
00493 void updateResourceComplete(Resource resource, QUuid requestId);

```

```

00494
00495 void updateResourceFailed(
00496     Resource resource, ErrorString errorDescription, QUuid requestId);
00497
00498 void findResourceComplete(
00499     Resource resource, LocalStorageManager::GetResourceOptions options,
00500     QUuid requestId);
00501
00502 void findResourceFailed(
00503     Resource resource, LocalStorageManager::GetResourceOptions options,
00504     ErrorString errorDescription, QUuid requestId);
00505
00506 void expungeResourceComplete(Resource resource, QUuid requestId);
00507
00508 void expungeResourceFailed(
00509     Resource resource, ErrorString errorDescription, QUuid requestId);
00510
00511 // Saved search-related signals:
00512 void getSavedSearchCountComplete(int savedSearchCount, QUuid requestId);
00513
00514 void getSavedSearchCountFailed(
00515     ErrorString errorDescription, QUuid requestId);
00516
00517 void addSavedSearchComplete(SavedSearch search, QUuid requestId);
00518
00519 void addSavedSearchFailed(
00520     SavedSearch search, ErrorString errorDescription, QUuid requestId);
00521
00522 void updateSavedSearchComplete(SavedSearch search, QUuid requestId);
00523
00524 void updateSavedSearchFailed(
00525     SavedSearch search, ErrorString errorDescription, QUuid requestId);
00526
00527 void findSavedSearchComplete(SavedSearch search, QUuid requestId);
00528
00529 void findSavedSearchFailed(
00530     SavedSearch search, ErrorString errorDescription, QUuid requestId);
00531
00532 void listAllSavedSearchesComplete(
00533     size_t limit, size_t offset,
00534     LocalStorageManager::ListSavedSearchesOrder order,
00535     LocalStorageManager::OrderDirection orderDirection,
00536     QList<SavedSearch> foundSearches, QUuid requestId);
00537
00538 void listAllSavedSearchesFailed(
00539     size_t limit, size_t offset,
00540     LocalStorageManager::ListSavedSearchesOrder order,
00541     LocalStorageManager::OrderDirection orderDirection,
00542     ErrorString errorDescription, QUuid requestId);
00543
00544 void listSavedSearchesComplete(
00545     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00546     size_t offset, LocalStorageManager::ListSavedSearchesOrder order,
00547     LocalStorageManager::OrderDirection orderDirection,
00548     QList<SavedSearch> foundSearches, QUuid requestId);
00549
00550 void listSavedSearchesFailed(
00551     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00552     size_t offset, LocalStorageManager::ListSavedSearchesOrder order,
00553     LocalStorageManager::OrderDirection orderDirection,
00554     ErrorString errorDescription, QUuid requestId);
00555
00556 void expungeSavedSearchComplete(SavedSearch search, QUuid requestId);
00557
00558 void expungeSavedSearchFailed(
00559     SavedSearch search, ErrorString errorDescription, QUuid requestId);
00560
00561 void accountHighUsnComplete(
00562     qint32 usn, QString linkedNotebookGuid, QUuid requestId);
00563
00564 void accountHighUsnFailed(
00565     QString linkedNotebookGuid, ErrorString errorDescription,
00566     QUuid requestId);
00567
00568 public Q_SLOTS:
00569     void init();
00570
00571 // User-related slots:
00572 void onGetUserCountRequest(QUuid requestId);
00573
00574 void onSwitchUserRequest(
00575     Account account, LocalStorageManager::StartupOptions startupOptions,
00576     QUuid requestId);
00577
00578 void onAddUserRequest(User user, QUuid requestId);
00579 void onUpdateUserRequest(User user, QUuid requestId);
00580 void onFindUserRequest(User user, QUuid requestId);

```

```
00581 void onDeleteUserRequest(User user, QUuid requestId);
00582 void onExpungeUserRequest(User user, QUuid requestId);
00583
00584 // Notebook-related slots:
00585 void onGetNotebookCountRequest(QUuid requestId);
00586 void onAddNotebookRequest(Notebook notebook, QUuid requestId);
00587 void onUpdateNotebookRequest(Notebook notebook, QUuid requestId);
00588 void onFindNotebookRequest(Notebook notebook, QUuid requestId);
00589 void onFindDefaultNotebookRequest(Notebook notebook, QUuid requestId);
00590 void onFindLastUsedNotebookRequest(Notebook notebook, QUuid requestId);
00591
00592 void onFindDefaultOrLastUsedNotebookRequest(
00593     Notebook notebook, QUuid requestId);
00594
00595 void onListAllNotebooksRequest(
00596     size_t limit, size_t offset,
00597     LocalStorageManager::ListNotebooksOrder order,
00598     LocalStorageManager::OrderDirection orderDirection,
00599     QString linkedNotebookGuid, QUuid requestId);
00600
00601 void onListAllSharedNotebooksRequest(QUuid requestId);
00602
00603 void onListNotebooksRequest(
00604     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00605     size_t offset, LocalStorageManager::ListNotebooksOrder order,
00606     LocalStorageManager::OrderDirection orderDirection,
00607     QString linkedNotebookGuid, QUuid requestId);
00608
00609 void onListSharedNotebooksPerNotebookGuidRequest(
00610     QString notebookGuid, QUuid requestId);
00611
00612 void onExpungeNotebookRequest(Notebook notebook, QUuid requestId);
00613
00614 // Linked notebook-related slots:
00615 void onGetLinkedNotebookCountRequest(QUuid requestId);
00616
00617 void onAddLinkedNotebookRequest(
00618     LinkedNotebook linkedNotebook, QUuid requestId);
00619
00620 void onUpdateLinkedNotebookRequest(
00621     LinkedNotebook linkedNotebook, QUuid requestId);
00622
00623 void onFindLinkedNotebookRequest(
00624     LinkedNotebook linkedNotebook, QUuid requestId);
00625
00626 void onListAllLinkedNotebooksRequest(
00627     size_t limit, size_t offset,
00628     LocalStorageManager::ListLinkedNotebooksOrder order,
00629     LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00630
00631 void onListLinkedNotebooksRequest(
00632     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00633     size_t offset, LocalStorageManager::ListLinkedNotebooksOrder order,
00634     LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00635
00636 void onExpungeLinkedNotebookRequest(
00637     LinkedNotebook linkedNotebook, QUuid requestId);
00638
00639 // Note-related slots:
00640 void onGetNoteCountRequest(
00641     LocalStorageManager::NoteCountOptions options, QUuid requestId);
00642
00643 void onGetNoteCountPerNotebookRequest(
00644     Notebook notebook, LocalStorageManager::NoteCountOptions options,
00645     QUuid requestId);
00646
00647 void onGetNoteCountPerTagRequest(
00648     Tag tag, LocalStorageManager::NoteCountOptions options,
00649     QUuid requestId);
00650
00651 void onGetNoteCountsPerAllTagsRequest(
00652     LocalStorageManager::NoteCountOptions options, QUuid requestId);
00653
00654 void onGetNoteCountPerNotebooksAndTagsRequest(
00655     QStringList notebookLocalUids, QStringList tagLocalUids,
00656     LocalStorageManager::NoteCountOptions options, QUuid requestId);
00657
00658 void onAddNoteRequest(Note note, QUuid requestId);
00659
00660 void onUpdateNoteRequest(
00661     Note note, LocalStorageManager::UpdateNoteOptions options,
00662     QUuid requestId);
00663
00664 void onFindNoteRequest(
00665     Note note, LocalStorageManager::GetNoteOptions options,
00666     QUuid requestId);
00667
```

```

00668 void onListNotesPerNotebookRequest (
00669     Notebook notebook, LocalStorageManager::GetNoteOptions options,
00670     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00671     size_t offset, LocalStorageManager::ListNotesOrder order,
00672     LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00673
00674 void onListNotesPerTagRequest (
00675     Tag tag, LocalStorageManager::GetNoteOptions options,
00676     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00677     size_t offset, LocalStorageManager::ListNotesOrder order,
00678     LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00679
00680 void onListNotesPerNotebooksAndTagsRequest (
00681     QStringList notebookLocalUids, QStringList tagLocalUids,
00682     LocalStorageManager::GetNoteOptions options,
00683     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00684     size_t offset, LocalStorageManager::ListNotesOrder order,
00685     LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00686
00687 void onListNotesByLocalUidsRequest (
00688     QStringList noteLocalUids, LocalStorageManager::GetNoteOptions options,
00689     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00690     size_t offset, LocalStorageManager::ListNotesOrder order,
00691     LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00692
00693 void onListNotesRequest (
00694     LocalStorageManager::ListObjectsOptions flag,
00695     LocalStorageManager::GetNoteOptions options, size_t limit,
00696     size_t offset, LocalStorageManager::ListNotesOrder order,
00697     LocalStorageManager::OrderDirection orderDirection,
00698     QString linkedNotebookGuid, QUuid requestId);
00699
00700 void onFindNoteLocalUidsWithSearchQuery (
00701     NoteSearchQuery noteSearchQuery, QUuid requestId);
00702
00703 void onExpungeNoteRequest (Note note, QUuid requestId);
00704
00705 // Tag-related slots:
00706 void onGetTagCountRequest (QUuid requestId);
00707 void onAddTagRequest (Tag tag, QUuid requestId);
00708 void onUpdateTagRequest (Tag tag, QUuid requestId);
00709 void onFindTagRequest (Tag tag, QUuid requestId);
00710
00711 void onListAllTagsPerNoteRequest (
00712     Note note, LocalStorageManager::ListObjectsOptions flag, size_t limit,
00713     size_t offset, LocalStorageManager::ListTagsOrder order,
00714     LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00715
00716 void onListAllTagsRequest (
00717     size_t limit, size_t offset, LocalStorageManager::ListTagsOrder order,
00718     LocalStorageManager::OrderDirection orderDirection,
00719     QString linkedNotebookGuid, QUuid requestId);
00720
00721 void onListTagsRequest (
00722     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00723     size_t offset, LocalStorageManager::ListTagsOrder order,
00724     LocalStorageManager::OrderDirection orderDirection,
00725     QString linkedNotebookGuid, QUuid requestId);
00726
00727 void onListTagsWithNoteLocalUidsRequest (
00728     LocalStorageManager::ListObjectsOptions flag, size_t limit,
00729     size_t offset, LocalStorageManager::ListTagsOrder order,
00730     LocalStorageManager::OrderDirection orderDirection,
00731     QString linkedNotebookGuid, QUuid requestId);
00732
00733 void onExpungeTagRequest (Tag tag, QUuid requestId);
00734 void onExpungeNotelessTagsFromLinkedNotebooksRequest (QUuid requestId);
00735
00736 // Resource-related slots:
00737 void onGetResourceCountRequest (QUuid requestId);
00738 void onAddResourceRequest (Resource resource, QUuid requestId);
00739 void onUpdateResourceRequest (Resource resource, QUuid requestId);
00740
00741 void onFindResourceRequest (
00742     Resource resource, LocalStorageManager::GetResourceOptions options,
00743     QUuid requestId);
00744
00745 void onExpungeResourceRequest (Resource resource, QUuid requestId);
00746
00747 // Saved search-related slots:
00748 void onGetSavedSearchCountRequest (QUuid requestId);
00749 void onAddSavedSearchRequest (SavedSearch search, QUuid requestId);
00750 void onUpdateSavedSearchRequest (SavedSearch search, QUuid requestId);
00751 void onFindSavedSearchRequest (SavedSearch search, QUuid requestId);
00752
00753 void onListAllSavedSearchesRequest (
00754     size_t limit, size_t offset,

```

```

00755         LocalStorageManager::ListSavedSearchesOrder order,
00756         LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00757
00758     void onListSavedSearchesRequest (
00759         LocalStorageManager::ListObjectsOptions flag, size_t limit,
00760         size_t offset, LocalStorageManager::ListSavedSearchesOrder order,
00761         LocalStorageManager::OrderDirection orderDirection, QUuid requestId);
00762
00763     void onExpungeSavedSearchRequest(SavedSearch search, QUuid requestId);
00764
00765     void onAccountHighUsnRequest(QString linkedNotebookGuid, QUuid requestId);
00766
00767 private:
00768     LocalStorageManagerAsync() = delete;
00769     Q_DISABLE_COPY(LocalStorageManagerAsync)
00770
00771     LocalStorageManagerAsyncPrivate * const d_ptr;
00772     Q_DECLARE_PRIVATE(LocalStorageManagerAsync)
00773 };
00774
00775 } // namespace quentier
00776
00777 #endif // LIB_QUENTIER_LOCAL_STORAGE_LOCAL_STORAGE_MANAGER_ASYNC_H

```

## 6.23 NoteSearchQuery.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOCAL_STORAGE_NOTE_SEARCH_QUERY_H
00020 #define LIB_QUENTIER_LOCAL_STORAGE_NOTE_SEARCH_QUERY_H
00021
00022 #include <quentier/types/ErrorString.h>
00023
00024 #include <QSharedDataPointer>
00025
00026 namespace quentier {
00027
00028     QT_FORWARD_DECLARE_CLASS(NoteSearchQueryData)
00029
00030     class QUINTIER_EXPORT NoteSearchQuery : public Printable
00031     {
00032     public:
00033         NoteSearchQuery();
00034         NoteSearchQuery(const NoteSearchQuery & other);
00035         NoteSearchQuery(NoteSearchQuery && other);
00036         NoteSearchQuery & operator=(const NoteSearchQuery & other);
00037         NoteSearchQuery & operator=(NoteSearchQuery && other);
00038         virtual ~NoteSearchQuery();
00039
00040         bool isEmpty() const;
00041
00042         void clear();
00043
00044         const QString queryString() const;
00045
00046         bool setQueryString(const QString & queryString, ErrorString & error);
00047
00048         const QString notebookModifier() const;
00049
00050         bool hasAnyModifier() const;
00051
00052         const QStringList & tagNames() const;
00053         const QStringList & negatedTagNames() const;
00054         bool hasAnyTag() const;
00055         bool hasNegatedAnyTag() const;
00056
00057         const QStringList & titleNames() const;

```

```

00066     const QStringList & negatedTitles() const;
00067     bool hasAnyTitleName() const;
00068     bool hasNegatedAnyTitleName() const;
00069
00070     const QVector<qint64> & creationTimestamps() const;
00071     const QVector<qint64> & negatedCreationTimestamps() const;
00072     bool hasAnyCreationTimestamp() const;
00073     bool hasNegatedAnyCreationTimestamp() const;
00074
00075     const QVector<qint64> & modificationTimestamps() const;
00076     const QVector<qint64> & negatedModificationTimestamps() const;
00077     bool hasAnyModificationTimestamp() const;
00078     bool hasNegatedAnyModificationTimestamp() const;
00079
00080     const QStringList & resourceMimeTypes() const;
00081     const QStringList & negatedResourceMimeTypes() const;
00082     bool hasAnyResourceMimeType() const;
00083     bool hasNegatedAnyResourceMimeType() const;
00084
00085     const QVector<qint64> & subjectDateTimestamps() const;
00086     const QVector<qint64> & negatedSubjectDateTimestamps() const;
00087     bool hasAnySubjectDateTimestamp() const;
00088     bool hasNegatedAnySubjectDateTimestamp() const;
00089
00090     const QVector<double> & latitudes() const;
00091     const QVector<double> & negatedLatitudes() const;
00092     bool hasAnyLatitude() const;
00093     bool hasNegatedAnyLatitude() const;
00094
00095     const QVector<double> & longitudes() const;
00096     const QVector<double> & negatedLongitudes() const;
00097     bool hasAnyLongitude() const;
00098     bool hasNegatedAnyLongitude() const;
00099
00100     const QVector<double> & altitudes() const;
00101     const QVector<double> & negatedAltitudes() const;
00102     bool hasAnyAltitude() const;
00103     bool hasNegatedAnyAltitude() const;
00104
00105     const QStringList & authors() const;
00106     const QStringList & negatedAuthors() const;
00107     bool hasAnyAuthor() const;
00108     bool hasNegatedAnyAuthor() const;
00109
00110     const QStringList & sources() const;
00111     const QStringList & negatedSources() const;
00112     bool hasAnySource() const;
00113     bool hasNegatedAnySource() const;
00114
00115     const QStringList & sourceApplications() const;
00116     const QStringList & negatedSourceApplications() const;
00117     bool hasAnySourceApplication() const;
00118     bool hasNegatedAnySourceApplication() const;
00119
00120     const QStringList & contentClasses() const;
00121     const QStringList & negatedContentClasses() const;
00122     bool hasAnyContentClass() const;
00123     bool hasNegatedAnyContentClass() const;
00124
00125     const QStringList & placeNames() const;
00126     const QStringList & negatedPlaceNames() const;
00127     bool hasAnyPlaceName() const;
00128     bool hasNegatedAnyPlaceName() const;
00129
00130     const QStringList & applicationData() const;
00131     const QStringList & negatedApplicationData() const;
00132     bool hasAnyApplicationData() const;
00133     bool hasNegatedAnyApplicationData() const;
00134
00135     const QVector<qint64> & reminderOrders() const;
00136     const QVector<qint64> & negatedReminderOrders() const;
00137     bool hasAnyReminderOrder() const;
00138     bool hasNegatedAnyReminderOrder() const;
00139
00140     const QVector<qint64> & reminderTimes() const;
00141     const QVector<qint64> & negatedReminderTimes() const;
00142     bool hasAnyReminderTime() const;
00143     bool hasNegatedAnyReminderTime() const;
00144
00145     const QVector<qint64> & reminderDoneTimes() const;
00146     const QVector<qint64> & negatedReminderDoneTimes() const;
00147     bool hasAnyReminderDoneTime() const;
00148     bool hasNegatedAnyReminderDoneTime() const;
00149
00150     bool hasUnfinishedToDo() const;
00151     bool hasNegatedUnfinishedToDo() const;
00152

```

```

00153     bool hasFinishedToDo() const;
00154     bool hasNegatedFinishedToDo() const;
00155
00156     bool hasAnyToDo() const;
00157     bool hasNegatedAnyToDo() const;
00158
00159     bool hasEncryption() const;
00160     bool hasNegatedEncryption() const;
00161
00162     const QStringList & contentSearchTerms() const;
00163     const QStringList & negatedContentSearchTerms() const;
00164     bool hasAnyContentSearchTerms() const;
00165
00166     bool isMatcheable() const;
00167
00168     virtual QTextStream & print(QTextStream & strm) const override;
00169
00170 private:
00171     QSharedDataPointer<NoteSearchQueryData> d;
00172 };
00173
00174 } // namespace quentier
00175
00176 #endif // LIB_QUENTIER_LOCAL_STORAGE_NOTE_SEARCH_QUERY_H

```

## 6.24 QuentierLogger.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_LOGGING_QUENTIER_LOGGER_H
00020 #define LIB_QUENTIER_LOGGING_QUENTIER_LOGGER_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QDebug>
00025 #include <QRegularExpression>
00026 #include <QString>
00027 #include <QTextStream>
00028
00029 namespace quentier {
00030
00035     enum class LogLevel
00036     {
00037         Trace,
00038         Debug,
00039         Info,
00040         Warning,
00041         Error
00042     };
00043
00044     QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, const LogLevel logLevel);
00045
00046     QUENTIER_EXPORT QTextStream & operator<<(
00047         QTextStream & strm, const LogLevel logLevel);
00048
00056     void QUENTIER_EXPORT QuentierInitializeLogging();
00057
00061     void QUENTIER_EXPORT QuentierAddLogEntry(
00062         const QString & sourceFileName, const int sourceFileLineNumber,
00063         const QString & component, const QString & message,
00064         const LogLevel logLevel);
00065
00071     LogLevel QUENTIER_EXPORT QuentierMinLogLevel();
00072
00076     void QUENTIER_EXPORT QuentierSetMinLogLevel(const LogLevel logLevel);
00077
00082     void QUENTIER_EXPORT QuentierAddStdOutLogDestination();

```

```

00083
00088 bool QUENTIER_EXPORT QuantierIsLogLevelActive(const LogLevel logLevel);
00089
00093 QString QUENTIER_EXPORT QuantierLogFilesDirPath();
00094
00098 void QUENTIER_EXPORT QuantierRestartLogging();
00099
00103 QRegularExpression QUENTIER_EXPORT QuantierLogComponentFilter();
00104
00108 void QUENTIER_EXPORT
00109 QuantierSetLogComponentFilter(const QRegularExpression & filter);
00110
00111 } // namespace quantier
00112
00113 #define __QNLOG_BASE(component, message, level)           \
00114     if (quantier::QuantierIsLogLevelActive(quantier::LogLevel::level)) { \
00115         QString msg; \
00116         QDebug dbg(&msg); \
00117         dbg.nospace(); \
00118         dbg.noquote(); \
00119         dbg < message; \
00120         quantier::QuantierAddLogEntry( \
00121             QStringLiteral(__FILE__), __LINE__, QString::fromUtf8(component), \
00122             msg, quantier::LogLevel::level); \
00123     } \
00124     // __QNLOG_BASE
00125
00126 #define QNTRACE(component, message) \
00127     __QNLOG_BASE(component, message, Trace) \
00128     // QNTRACE
00129
00130 #define QNDEBUG(component, message) \
00131     __QNLOG_BASE(component, message, Debug) \
00132     // QNDEBUG
00133
00134 #define QNINFO(component, message) \
00135     __QNLOG_BASE(component, message, Info) \
00136     // QNINFO
00137
00138 #define QNWARNING(component, message) \
00139     __QNLOG_BASE(component, message, Warning) \
00140     // QNWARNING
00141
00142 #define QNERROR(component, message) \
00143     __QNLOG_BASE(component, message, Error) \
00144     // QNERROR
00145
00146 #define QUENTIER_SET_MIN_LOG_LEVEL(level) \
00147     quantier::QuantierSetMinLogLevel( \
00148         quantier::LogLevel::level) // QUENTIER_SET_MIN_LOG_LEVEL
00149
00150 #define QUENTIER_INITIALIZE_LOGGING() \
00151     quantier::QuantierInitializeLogging() // QUENTIER_INITIALIZE_LOGGING
00152
00153 #define QUENTIER_ADD_STDOUT_LOG_DESTINATION() \
00154     quantier:: \
00155     QuantierAddStdOutLogDestination() // QUENTIER_ADD_STDOUT_LOG_DESTINATION
00156
00157 #define QNLOG_FILE_LINENUMBER_DELIMITER ":"
00158
00159 #endif // LIB_QUENTIER_LOGGING_QUENTIER_LOGGER_H

```

## 6.25 INoteEditorBackend.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_NOTE_EDITOR_I_NOTE_EDITOR_BACKEND_H

```

```

00020 #define LIB_QUENTIER_NOTE_EDITOR_I_NOTE_EDITOR_BACKEND_H
00021
00022 #include <quentier/types/Note.h>
00023 #include <quentier/utility/Linkage.h>
00024 #include <quentier/utility/Printable.h>
00025
00026 #include <QPalette>
00027 #include <QPrinter>
00028 #include <QStringList>
00029 #include <QThread>
00030 #include <QWidget>
00031
00032 QT_FORWARD_DECLARE_CLASS(QUndoStack)
00033
00034 namespace quentier {
00035
00036 QT_FORWARD_DECLARE_CLASS(Account)
00037 QT_FORWARD_DECLARE_CLASS(LocalStorageManagerAsync)
00038 QT_FORWARD_DECLARE_CLASS(NoteEditor)
00039 QT_FORWARD_DECLARE_CLASS(SpellChecker)
00040
00041 class QUENTIER_EXPORT INoteEditorBackend
00042 {
00043 public:
00044     virtual ~INoteEditorBackend();
00045
00046     virtual void initialize(
00047         LocalStorageManagerAsync & localStorageManager,
00048         SpellChecker & spellChecker, const Account & account,
00049         QThread * pBackgroundJobsThread) = 0;
00050
00051     virtual QObject * object() = 0; // provide QObject interface
00052     virtual QWidget * widget() = 0; // provide QWidget interface
00053
00054     virtual void setAccount(const Account & account) = 0;
00055     virtual void setUndoStack(QUndoStack * pUndoStack) = 0;
00056
00057     virtual void setInitialPageHtml(const QString & html) = 0;
00058     virtual void setNoteNotFoundPageHtml(const QString & html) = 0;
00059     virtual void setNoteDeletedPageHtml(const QString & html) = 0;
00060     virtual void setNoteLoadingPageHtml(const QString & html) = 0;
00061
00062     virtual bool isNoteLoaded() const = 0;
00063     virtual qint64 idleTime() const = 0;
00064
00065     virtual void convertToNote() = 0;
00066     virtual void saveNoteToLocalStorage() = 0;
00067     virtual void setNoteTitle(const QString & noteTitle) = 0;
00068
00069     virtual void setTagIds(
00070         const QStringList & tagLocalUids, const QStringList & tagGuids) = 0;
00071
00072     virtual void undo() = 0;
00073     virtual void redo() = 0;
00074     virtual void cut() = 0;
00075     virtual void copy() = 0;
00076     virtual void paste() = 0;
00077     virtual void pasteUnformatted() = 0;
00078     virtual void selectAll() = 0;
00079
00080     virtual void formatSelectionAsSourceCode() = 0;
00081
00082     virtual void fontMenu() = 0;
00083     virtual void textBold() = 0;
00084     virtual void textItalic() = 0;
00085     virtual void textUnderline() = 0;
00086     virtual void textStrikethrough() = 0;
00087     virtual void textHighlight() = 0;
00088
00089     virtual void alignLeft() = 0;
00090     virtual void alignCenter() = 0;
00091     virtual void alignRight() = 0;
00092     virtual void alignFull() = 0;
00093
00094     virtual QString selectedText() const = 0;
00095     virtual bool hasSelection() const = 0;
00096
00097     virtual void findNext(const QString & text, const bool matchCase) const = 0;
00098
00099     virtual void findPrevious(
00100         const QString & text, const bool matchCase) const = 0;
00101
00102     virtual void replace(
00103         const QString & textToReplace, const QString & replacementText,
00104         const bool matchCase) = 0;
00105
00106     virtual void replaceAll(

```

```

00107         const QString & textToReplace, const QString & replacementText,
00108         const bool matchCase) = 0;
00109
00110     virtual void insertToDoCheckbox() = 0;
00111
00112     virtual void insertInAppNoteLink(
00113         const QString & userId, const QString & shardId,
00114         const QString & noteGuid, const QString & linkText) = 0;
00115
00116     virtual void setSpellcheck(const bool enabled) = 0;
00117     virtual bool spellCheckEnabled() const = 0;
00118
00119     virtual void setFont(const QFont & font) = 0;
00120     virtual void setFontHeight(const int height) = 0;
00121     virtual void setFontColor(const QColor & color) = 0;
00122     virtual void setBackgroundColor(const QColor & color) = 0;
00123
00124     virtual QPalette defaultPalette() const = 0;
00125     virtual void setDefaultPalette(const QPalette & pal) = 0;
00126
00127     virtual const QFont * defaultFont() const = 0;
00128     virtual void setDefaultFont(const QFont & font) = 0;
00129
00130     virtual void insertHorizontalLine() = 0;
00131
00132     virtual void increaseFontSize() = 0;
00133     virtual void decreaseFontSize() = 0;
00134
00135     virtual void increaseIndentation() = 0;
00136     virtual void decreaseIndentation() = 0;
00137
00138     virtual void insertBulletedList() = 0;
00139     virtual void insertNumberedList() = 0;
00140
00141     virtual void insertTableDialog() = 0;
00142
00143     virtual void insertFixedWidthTable(
00144         const int rows, const int columns, const int widthInPixels) = 0;
00145
00146     virtual void insertRelativeWidthTable(
00147         const int rows, const int columns, const double relativeWidth) = 0;
00148
00149     virtual void insertTableRow() = 0;
00150     virtual void insertTableColumn() = 0;
00151     virtual void removeTableRow() = 0;
00152     virtual void removeTableColumn() = 0;
00153
00154     virtual void addAttachmentDialog() = 0;
00155     virtual void saveAttachmentDialog(const QByteArray & resourceHash) = 0;
00156     virtual void saveAttachmentUnderCursor() = 0;
00157     virtual void openAttachment(const QByteArray & resourceHash) = 0;
00158     virtual void openAttachmentUnderCursor() = 0;
00159     virtual void copyAttachment(const QByteArray & resourceHash) = 0;
00160     virtual void copyAttachmentUnderCursor() = 0;
00161     virtual void removeAttachment(const QByteArray & resourceHash) = 0;
00162     virtual void removeAttachmentUnderCursor() = 0;
00163     virtual void renameAttachment(const QByteArray & resourceHash) = 0;
00164     virtual void renameAttachmentUnderCursor() = 0;
00165
00166     enum class Rotation
00167     {
00168         Clockwise = 0,
00169         Counterclockwise
00170     };
00171
00172     friend QUENTIER_EXPORT QTextStream & operator<<(
00173         QTextStream & strm, const Rotation rotation);
00174
00175     friend QUENTIER_EXPORT QDebug & operator<<(
00176         QDebug & dbg, const Rotation rotation);
00177
00178     virtual void rotateImageAttachment(
00179         const QByteArray & resourceHash, const Rotation rotationDirection) = 0;
00180
00181     virtual void rotateImageAttachmentUnderCursor(
00182         const Rotation rotationDirection) = 0;
00183
00184     virtual void encryptSelectedText() = 0;
00185
00186     virtual void decryptEncryptedTextUnderCursor() = 0;
00187
00188     virtual void decryptEncryptedText(
00189         QString encryptedText, QString cipher, QString keyLength, QString hint,
00190         QString encryptIndex) = 0;
00191
00192     virtual void hideDecryptedTextUnderCursor() = 0;
00193

```

```

00194     virtual void hideDecryptedText (
00195         QString encryptedText, QString decryptedText, QString cipher,
00196         QString keyLength, QString hint, QString enDecryptedIndex) = 0;
00197
00198     virtual void editHyperlinkDialog() = 0;
00199     virtual void copyHyperlink() = 0;
00200     virtual void removeHyperlink() = 0;
00201
00202     virtual void onNoteLoadCancelled() = 0;
00203
00204     virtual bool print(QPrinter & printer, ErrorString & errorDescription) = 0;
00205
00206     virtual bool exportToPdf(
00207         const QString & absoluteFilePath, ErrorString & errorDescription) = 0;
00208
00209     virtual bool exportToEnex(
00210         const QStringList & tagNames, QString & enex,
00211         ErrorString & errorDescription) = 0;
00212
00213     virtual QString currentNoteLocalUid() const = 0;
00214     virtual void setCurrentNoteLocalUid(const QString & noteLocalUid) = 0;
00215
00216     virtual void clear() = 0;
00217
00218     virtual bool isModified() const = 0;
00219     virtual bool isEditorPageModified() const = 0;
00220
00221     virtual void setFocusToEditor() = 0;
00222
00223 protected:
00224     INoteEditorBackend(NoteEditor * parent);
00225     NoteEditor * m_pNoteEditor;
00226 };
00227
00228 } // namespace quantier
00229
00230 #endif // LIB_QUENTIER_NOTE_EDITOR_I_NOTE_EDITOR_BACKEND_H

```

## 6.26 NoteEditor.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_NOTE_EDITOR_NOTE_EDITOR_H
00020 #define LIB_QUENTIER_NOTE_EDITOR_NOTE_EDITOR_H
00021
00022 #include <quentier/types/ErrorString.h>
00023 #include <quentier/types/Note.h>
00024 #include <quentier/types/Notebook.h>
00025 #include <quentier/utility/Linkage.h>
00026
00027 #include <QPrinter>
00028 #include <QStringList>
00029 #include <QThread>
00030 #include <QWidget>
00031
00032 QT_FORWARD_DECLARE_CLASS(QUndoStack)
00033
00034 namespace quantier {
00035
00036 QT_FORWARD_DECLARE_CLASS(Account)
00037 QT_FORWARD_DECLARE_CLASS(INoteEditorBackend)
00038 QT_FORWARD_DECLARE_CLASS(LocalStorageManagerAsync)
00039 QT_FORWARD_DECLARE_CLASS(SpellChecker)
00040
00041
00042 class QUENTIER_EXPORT NoteEditor : public QWidget
00043 {

```

```

00047     Q_OBJECT
00048 public:
00049     explicit NoteEditor(
00050         QWidget * parent = nullptr,
00051         #if QT_VERSION >= QT_VERSION_CHECK(5, 15, 0)
00052         Qt::WindowFlags flags = {});
00053     #else
00054         Qt::WindowFlags flags = 0);
00055     #endif
00056
00057     virtual ~NoteEditor() override;
00058
00077     void initialize(
00078         LocalStorageManagerAsync & localStorageManager,
00079         SpellChecker & spellChecker, const Account & account,
00080         QThread * pBackgroundJobsThread = nullptr);
00081
00085     INoteEditorBackend * backend();
00086
00092     void setBackend(INoteEditorBackend * backend);
00093
00097     void setAccount(const Account & account);
00098
00102     const QUndoStack * undoStack() const;
00103
00107     void setUndoStack(QUndoStack * pUndoStack);
00108
00112     void setInitialPageHtml(const QString & html);
00113
00118     void setNoteNotFoundPageHtml(const QString & html);
00119
00125     void setNoteDeletedPageHtml(const QString & html);
00126
00131     void setNoteLoadingPageHtml(const QString & html);
00132
00136     QString currentNoteLocalUid() const;
00137
00145     void setCurrentNoteLocalUid(const QString & noteLocalUid);
00146
00150     void clear();
00151
00156     bool isModified() const;
00157
00162     bool isEditorPageModified() const;
00163
00168     bool isNoteLoaded() const;
00169
00175     qint64 idleTime() const;
00176
00180     void setFocus();
00181
00182     QString selectedText() const;
00183     bool hasSelection() const;
00184
00185     bool spellCheckEnabled() const;
00186
00187     bool print(QPrinter & printer, ErrorString & errorDescription);
00188
00189     bool exportToPdf(
00190         const QString & absoluteFilePath, ErrorString & errorDescription);
00191
00192     bool exportToEnex(
00193         const QStringList & tagNames, QString & enex,
00194         ErrorString & errorDescription);
00195
00203     QPalette defaultPalette() const;
00204
00209     const QFont * defaultFont() const;
00210
00211 Q_SIGNALS:
00216     void contentChanged();
00217
00223     void noteAndNotebookFoundInLocalStorage(Note note, Notebook notebook);
00224
00229     void noteNotFound(QString noteLocalUid);
00230
00236     void noteDeleted(QString noteLocalUid);
00237
00243     void noteModified();
00244
00249     void notifyError(ErrorString error);
00250
00255     void inAppNoteLinkClicked(
00256         QString userId, QString shardId, QString noteGuid);
00257
00269     void inAppNoteLinkPasteRequested(
00270         QString url, QString userId, QString shardId, QString noteGuid);

```

```

00271
00272     void convertedToNote(Note note);
00273     void cantConvertToNote(ErrorString error);
00274
00275     void noteEditorHtmlUpdated(QString html);
00276
00277     void currentNoteChanged(Note note);
00278
00279     void spellCheckerNotReady();
00280     void spellCheckerReady();
00281
00282     void noteLoaded();
00283
00284     void noteSavedToLocalStorage(QString noteLocalUid);
00285
00286     void failedToSaveNoteToLocalStorage(
00287         ErrorString errorDescription, QString noteLocalUid);
00288
00289     // Signals to notify anyone interested of the formatting at the current
00290     // cursor position
00291     void textBoldState(bool state);
00292     void textItalicState(bool state);
00293     void textUnderlineState(bool state);
00294     void textStrikethroughState(bool state);
00295     void textAlignLeftState(bool state);
00296     void textAlignCenterState(bool state);
00297     void textAlignRightState(bool state);
00298     void textAlignFullState(bool state);
00299     void textInsideOrderedListState(bool state);
00300     void textInsideUnorderedListState(bool state);
00301     void textInsideTableState(bool state);
00302
00303     void textFontFamilyChanged(QString fontFamily);
00304     void textFontSizeChanged(int fontSize);
00305
00306     void insertTableDialogRequested();
00307
00308 public Q_SLOTS:
00309     void convertToNote();
00310
00311     void saveNoteToLocalStorage();
00312
00313     void setNoteTitle(const QString & noteTitle);
00314
00315     void setTagIds(
00316         const QStringList & tagLocalUids, const QStringList & tagGuids);
00317
00318     void undo();
00319     void redo();
00320     void cut();
00321     void copy();
00322     void paste();
00323     void pasteUnformatted();
00324     void selectAll();
00325
00326     void formatSelectionAsSourceCode();
00327
00328     void fontMenu();
00329     void textBold();
00330     void textItalic();
00331     void textUnderline();
00332     void textStrikethrough();
00333     void textHighlight();
00334
00335     void alignLeft();
00336     void alignCenter();
00337     void alignRight();
00338     void alignFull();
00339
00340     void findNext(const QString & text, const bool matchCase) const;
00341     void findPrevious(const QString & text, const bool matchCase) const;
00342
00343     void replace(
00344         const QString & textToReplace, const QString & replacementText,
00345         const bool matchCase);
00346
00347     void replaceAll(
00348         const QString & textToReplace, const QString & replacementText,
00349         const bool matchCase);
00350
00351     void insertToDoCheckbox();
00352
00353     void insertInAppNoteLink(
00354         const QString & userId, const QString & shardId,
00355         const QString & noteGuid, const QString & linkText);
00356
00357     void setSpellcheck(const bool enabled);

```

```

00401
00402     void setFont(const QFont & font);
00403     void setFontHeight(const int height);
00404     void setFontColor(const QColor & color);
00405     void setBackgroundColor(const QColor & color);
00406
00422     void setDefaultPalette(const QPalette & pal);
00423
00429     void setDefaultFont(const QFont & font);
00430
00431     void insertHorizontalLine();
00432
00433     void increaseFontSize();
00434     void decreaseFontSize();
00435
00436     void increaseIndentation();
00437     void decreaseIndentation();
00438
00439     void insertBulletedList();
00440     void insertNumberedList();
00441
00442     void insertTableDialog();
00443
00444     void insertFixedWidthTable(
00445         const int rows, const int columns, const int widthInPixels);
00446
00447     void insertRelativeWidthTable(
00448         const int rows, const int columns, const double relativeWidth);
00449
00450     void insertTableRow();
00451     void insertTableColumn();
00452     void removeTableRow();
00453     void removeTableColumn();
00454
00455     void addAttachmentDialog();
00456     void saveAttachmentDialog(const QByteArray & resourceHash);
00457     void saveAttachmentUnderCursor();
00458     void openAttachment(const QByteArray & resourceHash);
00459     void openAttachmentUnderCursor();
00460     void copyAttachment(const QByteArray & resourceHash);
00461     void copyAttachmentUnderCursor();
00462
00463     void encryptSelectedText();
00464     void decryptEncryptedTextUnderCursor();
00465
00466     void editHyperlinkDialog();
00467     void copyHyperlink();
00468     void removeHyperlink();
00469
00470     void onNoteLoadCancelled();
00471
00472 protected:
00473     virtual void dragMoveEvent(QDragMoveEvent * pEvent) override;
00474     virtual void dropEvent(QDropEvent * pEvent) override;
00475
00476 private:
00477     INoteEditorBackend * m_backend;
00478 };
00479
00480 } // namespace quentier
00481
00482 #endif // LIB_QUENTIER_NOTE_EDITOR_NOTE_EDITOR_H

```

## 6.27 SpellChecker.h

```

00001 /*
00002  * Copyright 2017-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018

```

```

00019 #ifndef LIB_QUENTIER_NOTE_EDITOR_SPELL_CHECKER_H
00020 #define LIB_QUENTIER_NOTE_EDITOR_SPELL_CHECKER_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QObject>
00025 #include <QVector>
00026
00027 #include <utility>
00028
00029 namespace quentier {
00030
00031 QT_FORWARD_DECLARE_CLASS(Account)
00032 QT_FORWARD_DECLARE_CLASS(FileIOProcessorAsync)
00033 QT_FORWARD_DECLARE_CLASS(SpellCheckerPrivate)
00034
00035 class QUENTIER_EXPORT SpellChecker : public QObject
00036 {
00037     Q_OBJECT
00038 public:
00039     SpellChecker(
00040         FileIOProcessorAsync * pFileIOProcessorAsync, const Account & account,
00041         QObject * parent = nullptr, const QString & userDictionaryPath = {});
00042
00043     // The second bool in the pair indicates whether the dictionary
00044     // is enabled or disabled
00045     QVector<std::pair<QString, bool>> listAvailableDictionaries() const;
00046
00047     void setAccount(const Account & account);
00048
00049     void enableDictionary(const QString & language);
00050     void disableDictionary(const QString & language);
00051
00052     bool checkSpell(const QString & word) const;
00053
00054     QStringList spellCorrectionSuggestions(
00055         const QString & missSpelledWord) const;
00056
00057     void addToUserWordlist(const QString & word);
00058     void removeFromUserWordList(const QString & word);
00059     void ignoreWord(const QString & word);
00060     void removeWord(const QString & word);
00061
00062     bool isReady() const;
00063
00064 Q_SIGNALS:
00065     void ready();
00066
00067 private:
00068     SpellCheckerPrivate * const d_ptr;
00069     Q_DECLARE_PRIVATE(SpellChecker)
00070 };
00071
00072 } // namespace quentier
00073
00074 #endif // LIB_QUENTIER_NOTE_EDITOR_SPELL_CHECKER_H

```

## 6.28 AuthenticationManager.h

```

00001 /*
00002  * Copyright 2017-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_SYNCHRONIZATION_AUTHENTICATION_MANAGER_H
00020 #define LIB_QUENTIER_SYNCHRONIZATION_AUTHENTICATION_MANAGER_H
00021
00022 #include <quentier/synchronization/IAuthenticationManager.h>
00023
00024 namespace quentier {

```

```

00025
00026 QT_FORWARD_DECLARE_CLASS(AuthenticationManagerPrivate)
00027
00028
00033 class QUENTIER_EXPORT AuthenticationManager : public IAuthenticationManager
00034 {
00035     Q_OBJECT
00036 public:
00037     explicit AuthenticationManager(
00038         const QString & consumerKey, const QString & consumerSecret,
00039         const QString & host, QObject * parent = nullptr);
00040
00041     virtual ~AuthenticationManager();
00042
00043 public Q_SLOTS:
00044     virtual void onAuthenticationRequest() override;
00045
00046 private:
00047     AuthenticationManager() = delete;
00048     Q_DISABLE_COPY(AuthenticationManager)
00049
00050 private:
00051     AuthenticationManagerPrivate * const d_ptr;
00052     Q_DECLARE_PRIVATE(AuthenticationManager)
00053 };
00054
00055 } // namespace quentier
00056
00057 #endif // LIB_QUENTIER_SYNCHRONIZATION_AUTHENTICATION_MANAGER_H

```

## 6.29 synchronization/ForwardDeclarations.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_SYNCHRONIZATION_FORWARD_DECLARATIONS_H
00020 #define LIB_QUENTIER_SYNCHRONIZATION_FORWARD_DECLARATIONS_H
00021
00022 #include <memory>
00023
00024 namespace quentier {
00025
00026 class IAuthenticationManager;
00027
00028 class INoteStore;
00029 using INoteStorePtr = std::shared_ptr<INoteStore>;
00030
00031 class ISyncStateStorage;
00032 using ISyncStateStoragePtr = std::shared_ptr<ISyncStateStorage>;
00033
00034 class IUserStore;
00035 using IUserStorePtr = std::shared_ptr<IUserStore>;
00036
00037 } // namespace quentier
00038
00039 #endif // LIB_QUENTIER_SYNCHRONIZATION_FORWARD_DECLARATIONS_H

```

## 6.30 utility/ForwardDeclarations.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *

```

```

00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_FORWARD_DECLARATIONS_H
00020 #define LIB_QUENTIER_UTILITY_FORWARD_DECLARATIONS_H
00021
00022 #include <memory>
00023
00024 namespace quentier {
00025
00026 class IKeychainService;
00027 using IKeychainServicePtr = std::shared_ptr<IKeychainService>;
00028
00029 } // namespace quentier
00030
00031 #endif // LIB_QUENTIER_UTILITY_FORWARD_DECLARATIONS_H

```

## 6.31 IAuthenticationManager.h

```

00001 /*
00002  * Copyright 2017-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_SYNCHRONIZATION_I_AUTHENTICATION_MANAGER_H
00020 #define LIB_QUENTIER_SYNCHRONIZATION_I_AUTHENTICATION_MANAGER_H
00021
00022 #include <quentier/synchronization/ForwardDeclarations.h>
00023 #include <quentier/types/ErrorMessage.h>
00024 #include <quentier/utility/Linkage.h>
00025
00026 #include <qt5qevercloud/QEverCloud.h>
00027
00028 #include <QHash>
00029 #include <QList>
00030 #include <QNetworkCookie>
00031 #include <QObject>
00032 #include <QVector>
00033
00034 namespace quentier {
00035
00036 class QUENTIER_EXPORT IAuthenticationManager : public QObject
00037 {
00038     Q_OBJECT
00039 protected:
00040     explicit IAuthenticationManager(QObject * parent = nullptr);
00041
00042 public:
00043     virtual ~IAuthenticationManager();
00044
00045     Q_SIGNALS:
00046         void sendAuthenticationResult(
00047             bool success, qevercloud::UserID userId, QString authToken,
00048             qevercloud::Timestamp authTokenExpirationTime, QString shardId,
00049             QString noteStoreUrl, QString webApiUrlPrefix,
00050             QList<QNetworkCookie> userStoreCookies, ErrorMessage errorDescription);
00051
00052     public Q_SLOTS:
00053         virtual void onAuthenticationRequest() = 0;
00054 };

```

```

00055
00056 } // namespace quentier
00057
00058 #endif // LIB_QUENTIER_SYNCHRONIZATION_I_AUTHENTICATION_MANAGER_H

```

## 6.32 INoteStore.h

```

00001 /*
00002  * Copyright 2018-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_SYNCHRONIZATION_I_NOTE_STORE_H
00020 #define LIB_QUENTIER_SYNCHRONIZATION_I_NOTE_STORE_H
00021
00022 #include <quentier/synchronization/ForwardDeclarations.h>
00023 #include <quentier/types/ErrorMessage.h>
00024 #include <quentier/types/Note.h>
00025 #include <quentier/types/Notebook.h>
00026 #include <quentier/types/SavedSearch.h>
00027 #include <quentier/types/Tag.h>
00028 #include <quentier/utility/Linkage.h>
00029
00030 #include <qt5qevercloud/QEverCloud.h>
00031
00032 #include <QObject>
00033
00034 #include <memory>
00035
00036 namespace quentier {
00037
00038 class QUENTIER_EXPORT INoteStore : public QObject
00039 {
00040     Q_OBJECT
00041 protected:
00042     explicit INoteStore(QObject * parent = nullptr);
00043
00044 public:
00045     virtual ~INoteStore() = default;
00046
00047     /*
00048      * Factory method, create a new INoteStore subclass object
00049      */
00050     virtual INoteStore * create() const = 0;
00051
00052     virtual QString noteStoreUrl() const = 0;
00053
00054     virtual void setNoteStoreUrl(QString noteStoreUrl) = 0;
00055
00056     virtual void setAuthData(
00057         QString authenticationToken, QList<QNetworkCookie> cookies) = 0;
00058
00059     virtual void stop() = 0;
00060
00061     virtual qint32 createNotebook(
00062         Notebook & notebook, ErrorMessage & errorDescription,
00063         qint32 & rateLimitSeconds, QString linkedNotebookAuthToken = {}) = 0;
00064
00065     virtual qint32 updateNotebook(
00066         Notebook & notebook, ErrorMessage & errorDescription,
00067         qint32 & rateLimitSeconds, QString linkedNotebookAuthToken = {}) = 0;
00068
00069     virtual qint32 createNote(
00070         Note & note, ErrorMessage & errorDescription, qint32 & rateLimitSeconds,
00071         QString linkedNotebookAuthToken = {}) = 0;
00072
00073     virtual qint32 updateNote(
00074         Note & note, ErrorMessage & errorDescription, qint32 & rateLimitSeconds,
00075         QString linkedNotebookAuthToken = {}) = 0;
00076
00077
00078
00079
00080
00081
00082
00083
00084
00085
00086
00087
00088
00089
00090
00091
00092
00093
00094
00095
00096
00097
00098
00099
00100
00101
00102
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119

```

```

00201     virtual qint32 createTag(
00202         Tag & tag, ErrorString & errorDescription, qint32 & rateLimitSeconds,
00203         QString linkedNotebookAuthToken = {}) = 0;
00204
00226     virtual qint32 updateTag(
00227         Tag & tag, ErrorString & errorDescription, qint32 & rateLimitSeconds,
00228         QString linkedNotebookAuthToken = {}) = 0;
00229
00247     virtual qint32 createSavedSearch(
00248         SavedSearch & savedSearch, ErrorString & errorDescription,
00249         qint32 & rateLimitSeconds) = 0;
00250
00267     virtual qint32 updateSavedSearch(
00268         SavedSearch & savedSearch, ErrorString & errorDescription,
00269         qint32 & rateLimitSeconds) = 0;
00270
00287     virtual qint32 getSyncState(
00288         qevercloud::SyncState & syncState, ErrorString & errorDescription,
00289         qint32 & rateLimitSeconds) = 0;
00290
00313     virtual qint32 getSyncChunk(
00314         const qint32 afterUSN, const qint32 maxEntries,
00315         const qevercloud::SyncChunkFilter & filter,
00316         qevercloud::SyncChunk & syncChunk, ErrorString & errorDescription,
00317         qint32 & rateLimitSeconds) = 0;
00318
00343     virtual qint32 getLinkedNotebookSyncState(
00344         const qevercloud::LinkedNotebook & linkedNotebook,
00345         const QString & authToken, qevercloud::SyncState & syncState,
00346         ErrorString & errorDescription, qint32 & rateLimitSeconds) = 0;
00347
00384     virtual qint32 getLinkedNotebookSyncChunk(
00385         const qevercloud::LinkedNotebook & linkedNotebook,
00386         const qint32 afterUSN, const qint32 maxEntries,
00387         const QString & linkedNotebookAuthToken, const bool fullSyncOnly,
00388         qevercloud::SyncChunk & syncChunk, ErrorString & errorDescription,
00389         qint32 & rateLimitSeconds) = 0;
00390
00421     virtual qint32 getNote(
00422         const bool withContent, const bool withResourcesData,
00423         const bool withResourcesRecognition,
00424         const bool withResourceAlternateData, Note & note,
00425         ErrorString & errorDescription, qint32 & rateLimitSeconds) = 0;
00426
00464     virtual bool getNoteAsync(
00465         const bool withContent, const bool withResourceData,
00466         const bool withResourcesRecognition,
00467         const bool withResourceAlternateData, const bool withSharedNotes,
00468         const bool withNoteAppDataValues, const bool withResourceAppDataValues,
00469         const bool withNoteLimits, const QString & noteGuid,
00470         const QString & authToken, ErrorString & errorDescription) = 0;
00471
00503     virtual qint32 getResource(
00504         const bool withDataBody, const bool withRecognitionDataBody,
00505         const bool withAlternateDataBody, const bool withAttributes,
00506         const QString & authToken, Resource & resource,
00507         ErrorString & errorDescription, qint32 & rateLimitSeconds) = 0;
00508
00533     virtual bool getResourceAsync(
00534         const bool withDataBody, const bool withRecognitionDataBody,
00535         const bool withAlternateDataBody, const bool withAttributes,
00536         const QString & resourceGuid, const QString & authToken,
00537         ErrorString & errorDescription) = 0;
00538
00558     virtual qint32 authenticateToSharedNotebook(
00559         const QString & shareKey, qevercloud::AuthenticationResult & authResult,
00560         ErrorString & errorDescription, qint32 & rateLimitSeconds) = 0;
00561
00562     Q_SIGNALS:
00563     void getNoteAsyncFinished(
00564         qint32 errorCode, qevercloud::Note note, qint32 rateLimitSeconds,
00565         ErrorString errorDescription);
00566
00567     void getResourceAsyncFinished(
00568         qint32 errorCode, qevercloud::Resource resource,
00569         qint32 rateLimitSeconds, ErrorString errorDescription);
00570
00571 private:
00572     Q_DISABLE_COPY(INoteStore)
00573 };
00574
00575 QUENTIER_EXPORT INoteStorePtr newNoteStore(QObject * parent = nullptr);
00576
00577 } // namespace quantier
00578
00579 #endif // LIB_QUENTIER_SYNCHRONIZATION_I_NOTE_STORE_H

```

## 6.33 ISyncStateStorage.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_SYNCHRONIZATION_I_SYNC_STATE_STORAGE_H
00020 #define LIB_QUENTIER_SYNCHRONIZATION_I_SYNC_STATE_STORAGE_H
00021
00022 #include <quentier/synchronization/ForwardDeclarations.h>
00023 #include <quentier/types/Account.h>
00024 #include <quentier/utility/Linkage.h>
00025
00026 #include <qt5qevercloud/QEverCloud.h>
00027
00028 #include <QHash>
00029 #include <QObject>
00030 #include <QString>
00031
00032 #include <memory>
00033
00034 namespace quentier {
00035
00041 class QUENTIER_EXPORT ISyncStateStorage : public QObject
00042 {
00043     Q_OBJECT
00044 public:
00049     class QUENTIER_EXPORT ISyncState : public Printable
00050     {
00051     public:
00052         virtual qint32 userDataUpdateCount() const = 0;
00053         virtual qevercloud::Timestamp userDataLastSyncTime() const = 0;
00054         virtual QHash<QString, qint32> linkedNotebookUpdateCounts() const = 0;
00055
00056         virtual QHash<QString, qevercloud::Timestamp>
00057             linkedNotebookLastSyncTimes() const = 0;
00058
00059         virtual QTextStream & print(QTextStream & strm) const override;
00060     };
00061
00062     using ISyncStatePtr = std::shared_ptr<ISyncState>;
00063
00064 public:
00065     explicit ISyncStateStorage(QObject * parent = nullptr) : QObject(parent) {}
00066
00067     virtual ~ISyncStateStorage() = default;
00068
00069     virtual ISyncStatePtr getSyncState(const Account & account) = 0;
00070
00071     virtual void setSyncState(
00072         const Account & account, ISyncStatePtr syncState) = 0;
00073
00074     Q_SIGNALS:
00080         void notifySyncStateUpdated(Account account, ISyncStatePtr syncState);
00081     };
00082
00083 QUENTIER_EXPORT ISyncStateStoragePtr
00084 newSyncStateStorage(QObject * parent = nullptr);
00085
00086 } // namespace quentier
00087
00088 #endif // LIB_QUENTIER_SYNCHRONIZATION_I_SYNC_STATE_STORAGE_H

```

## 6.34 IUserStore.h

```

00001 /*
00002  * Copyright 2018-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier

```

```

00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_SYNCHRONIZATION_I_USER_STORE_H
00020 #define LIB_QUENTIER_SYNCHRONIZATION_I_USER_STORE_H
00021
00022 #include <quentier/synchronization/ForwardDeclarations.h>
00023 #include <quentier/types/ErrorMessage.h>
00024 #include <quentier/utility/Linkage.h>
00025
00026 #include <QList>
00027 #include <QNetworkCookie>
00028
00029 #include <qt5gevercloud/QEverCloud.h>
00030
00031 #include <memory>
00032
00033 namespace quentier {
00034
00035 QT_FORWARD_DECLARE_CLASS(User)
00036
00037
00042 class QUENTIER_EXPORT IUserStore
00043 {
00044 public:
00045     virtual ~IUserStore() = default;
00046
00050     virtual void setAuthData(
00051         QString authenticationToken, QList<QNetworkCookie> cookies) = 0;
00052
00071     virtual bool checkVersion(
00072         const QString & clientName, quint16 edamVersionMajor,
00073         quint16 edamVersionMinor, ErrorMessage & errorDescription) = 0;
00074
00093     virtual quint32 getUser(
00094         User & user, ErrorMessage & errorDescription,
00095         quint32 & rateLimitSeconds) = 0;
00096
00117     virtual quint32 getAccountLimits(
00118         const qevercloud::ServiceLevel serviceLevel,
00119         qevercloud::AccountLimits & limits, ErrorMessage & errorDescription,
00120         quint32 & rateLimitSeconds) = 0;
00121 };
00122
00123 QUENTIER_EXPORT IUserStorePtr newUserStore(QString evernoteHost);
00124
00125 } // namespace quentier
00126
00127 #endif // LIB_QUENTIER_SYNCHRONIZATION_I_USER_STORE_H

```

## 6.35 SynchronizationManager.h

```

00001  /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_SYNCHRONIZATION_SYNCHRONIZATION_MANAGER_H
00020 #define LIB_QUENTIER_SYNCHRONIZATION_SYNCHRONIZATION_MANAGER_H

```

```

00021
00022 #include <quentier/synchronization/ForwardDeclarations.h>
00023 #include <quentier/types/Account.h>
00024 #include <quentier/types/ErrorMessage.h>
00025 #include <quentier/types/LinkedNotebook.h>
00026 #include <quentier/utility/ForwardDeclarations.h>
00027 #include <quentier/utility/Linkage.h>
00028
00029 #include <QObject>
00030
00031 namespace quentier {
00032
00033 QT_FORWARD_DECLARE_CLASS(LocalStorageManagerAsync)
00034 QT_FORWARD_DECLARE_CLASS(SynchronizationManagerPrivate)
00035
00036
00042 class QUENTIER_EXPORT SynchronizationManager : public QObject
00043 {
00044     Q_OBJECT
00045 public:
00081     SynchronizationManager(
00082         QString host, LocalStorageManagerAsync & localStorageManagerAsync,
00083         IAuthenticationManager & authenticationManager,
00084         QObject * parent = nullptr, INoteStorePtr pNoteStore = {},
00085         IUserStorePtr pUserStore = {},
00086         IKeychainServicePtr pKeychainService = {},
00087         ISyncStateStoragePtr pSyncStateStorage = {});
00088
00089     virtual ~SynchronizationManager();
00090
00095     bool active() const;
00096
00103     bool downloadNoteThumbnailsOption() const;
00104
00105 public Q_SLOTS:
00120     void setAccount(Account account);
00121
00130     void authenticate();
00131
00143     void authenticateCurrentAccount();
00144
00148     void synchronize();
00149
00154     void stop();
00155
00162     void revokeAuthentication(const qevercloud::UserID userId);
00163
00175     void setDownloadNoteThumbnails(bool flag);
00176
00185     void setDownloadInkNoteImages(bool flag);
00186
00204     void setInkNoteImagesStoragePath(QString path);
00205
00206 Q_SIGNALS:
00212     void started();
00213
00219     void stopped();
00220
00227     void failed(ErrorMessage errorDescription);
00228
00249     void finished(
00250         Account account, bool somethingDownloaded, bool somethingSent);
00251
00263     void authenticationRevoked(
00264         bool success, ErrorMessage errorDescription, qevercloud::UserID userId);
00265
00279     void authenticationFinished(
00280         bool success, ErrorMessage errorDescription, Account account);
00281
00286     void remoteToLocalSyncStopped();
00287
00292     void sendLocalChangesStopped();
00293
00304     void willRepeatRemoteToLocalSyncAfterSendingChanges();
00305
00317     void detectedConflictDuringLocalChangesSending();
00318
00328     void rateLimitExceeded(qint32 secondsToWait);
00329
00343     void remoteToLocalSyncDone(bool somethingDownloaded);
00344
00360     void syncChunksDownloadProgress(
00361         qint32 highestDownloadedUsn, qint32 highestServerUsn,
00362         qint32 lastPreviousUsn);
00363
00368     void syncChunksDownloaded();
00369

```

```

00393     void linkedNotebookSyncChunksDownloadProgress(
00394         qint32 highestDownloadedUsn, qint32 highestServerUsn,
00395         qint32 lastPreviousUsn, LinkedNotebook linkedNotebook);
00396
00401     void linkedNotebooksSyncChunksDownloaded();
00402
00411     void notesDownloadProgress(
00412         quint32 notesDownloaded, quint32 totalNotesToDownload);
00413
00422     void linkedNotebooksNotesDownloadProgress(
00423         quint32 notesDownloaded, quint32 totalNotesToDownload);
00424
00435     void resourcesDownloadProgress(
00436         quint32 resourcesDownloaded, quint32 totalResourcesToDownload);
00437
00448     void linkedNotebooksResourcesDownloadProgress(
00449         quint32 resourcesDownloaded, quint32 totalResourcesToDownload);
00450
00456     void preparedDirtyObjectsForSending();
00457
00463     void preparedLinkedNotebooksDirtyObjectsForSending();
00464
00470     void setAccountDone(Account account);
00471
00476     void setDownloadNoteThumbnailsDone(bool flag);
00477
00482     void setDownloadInkNoteImagesDone(bool flag);
00483
00488     void setInkNoteImagesStoragePathDone(QString path);
00489
00490 private:
00491     SynchronizationManager() = delete;
00492     Q_DISABLE_COPY(SynchronizationManager)
00493
00494     SynchronizationManagerPrivate * d_ptr;
00495     Q_DECLARE_PRIVATE(SynchronizationManager)
00496 };
00497
00498 } // namespace quentier
00499
00500 #endif // LIB_QUENTIER_SYNCHRONIZATION_SYNCHRONIZATION_MANAGER_H

```

## 6.36 Account.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_ACCOUNT_H
00020 #define LIB_QUENTIER_TYPES_ACCOUNT_H
00021
00022 #include <quentier/utility/Printable.h>
00023
00024 #include <qt5qevercloud/QEverCloud.h>
00025
00026 #include <QSharedDataPointer>
00027 #include <QString>
00028
00029 namespace quentier {
00030
00031 QT_FORWARD_DECLARE_CLASS(AccountData)
00032
00033 class QUINTIER_EXPORT Account : public Printable
00034 {
00035 public:
00036     enum class Type
00037     {
00038         Local = 0,

```

```

00044     Evernote
00045 };
00046
00047 friend QUINTIER_EXPORT QTextStream & operator<<
00048     QTextStream & strm, const Type type);
00049
00050 friend QUINTIER_EXPORT QDebug & operator<<(QDebug & dbg, const Type type);
00051
00052 enum class EvernoteAccountType
00053 {
00054     Free = 0,
00055     Plus,
00056     Premium,
00057     Business
00058 };
00059
00060 friend QUINTIER_EXPORT QTextStream & operator<<
00061     QTextStream & strm, const EvernoteAccountType type);
00062
00063 friend QUINTIER_EXPORT QDebug & operator<<
00064     QDebug & dbg, const EvernoteAccountType type);
00065
00066 public:
00067     explicit Account();
00068
00069     explicit Account(
00070         QString name, const Type type, const qevercloud::UserID userId = -1,
00071         const EvernoteAccountType evernoteAccountType =
00072             EvernoteAccountType::Free,
00073         QString evernoteHost = {}, QString shardId = {});
00074
00075     Account(const Account & other);
00076     Account & operator=(const Account & other);
00077     virtual ~Account() override;
00078
00079     bool operator==(const Account & other) const;
00080     bool operator!=(const Account & other) const;
00081
00082     bool isEmpty() const;
00083
00084     QString name() const;
00085
00086     void setName(QString name);
00087
00088     QString displayName() const;
00089
00090     void setDisplayName(QString displayName);
00091
00092     Type type() const;
00093
00094     qevercloud::UserID id() const;
00095
00096     EvernoteAccountType evernoteAccountType() const;
00097
00098     QString evernoteHost() const;
00099
00100     QString shardId() const;
00101
00102     void setEvernoteAccountType(const EvernoteAccountType evernoteAccountType);
00103     void setEvernoteHost(QString evernoteHost);
00104     void setShardId(QString shardId);
00105
00106     qint32 mailLimitDaily() const;
00107     qint64 noteSizeMax() const;
00108     qint64 resourceSizeMax() const;
00109     qint32 linkedNotebookMax() const;
00110     qint32 noteCountMax() const;
00111     qint32 notebookCountMax() const;
00112     qint32 tagCountMax() const;
00113     qint32 noteTagCountMax() const;
00114     qint32 savedSearchCountMax() const;
00115     qint32 noteResourceCountMax() const;
00116     void setEvernoteAccountLimits(const qevercloud::AccountLimits & limits);
00117
00118     virtual QTextStream & print(QTextStream & strm) const override;
00119
00120 private:
00121     QSharedDataPointer<AccountData> d;
00122 };
00123
00124 } // namespace quentier
00125
00126 #endif // LIB_QUENTIER_TYPES_ACCOUNT_H

```

## 6.37 ErrorString.h

```

00001 /*
00002  * Copyright 2017-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_ERROR_STRING_H
00020 #define LIB_QUENTIER_TYPES_ERROR_STRING_H
00021
00022 #include <quentier/utility/Printable.h>
00023
00024 #include <QSharedDataPointer>
00025
00026 namespace quentier {
00027
00028 QT_FORWARD_DECLARE_CLASS(ErrorStringData)
00029
00030
00043 class QUENTIER_EXPORT ErrorString : public Printable
00044 {
00045 public:
00046     explicit ErrorString(const char * error = nullptr);
00047     explicit ErrorString(const QString & error);
00048     ErrorString(const ErrorString & other);
00049     ErrorString & operator=(const ErrorString & other);
00050     virtual ~ErrorString() override;
00051
00052     const QString & base() const;
00053     QString & base();
00054
00055     const QStringList & additionalBases() const;
00056     QStringList & additionalBases();
00057
00058     const QString & details() const;
00059     QString & details();
00060
00061     void setBase(const QString & error);
00062     void setBase(const char * error);
00063
00064     void appendBase(const QString & error);
00065     void appendBase(const QStringList & errors);
00066     void appendBase(const char * error);
00067
00068     void setDetails(const QString & error);
00069     void setDetails(const char * error);
00070
00071     bool isEmpty() const;
00072     void clear();
00073
00074     QString localizedString() const;
00075     QString nonLocalizedString() const;
00076
00077     virtual QTextStream & print(QTextStream & strm) const override;
00078
00079 private:
00080     QSharedDataPointer<ErrorStringData> d;
00081 };
00082
00083 } // namespace quentier
00084
00085 #endif // LIB_QUENTIER_TYPES_ERROR_STRING_H

```

## 6.38 IFavoritableDataElement.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *

```

```

00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_I_FAVORITABLE_DATA_ELEMENT_H
00020 #define LIB_QUENTIER_TYPES_I_FAVORITABLE_DATA_ELEMENT_H
00021
00022 #include "INoteStoreDataElement.h"
00023
00024 namespace quentier {
00025
00026 class QUENTIER_EXPORT IFavoritableDataElement :
00027     public virtual INoteStoreDataElement
00028 {
00029 public:
00030     virtual bool isFavorited() const = 0;
00031     virtual void setFavorited(const bool favorited) = 0;
00032
00033     virtual ~IFavoritableDataElement() {}
00034 };
00035
00036 #define DECLARE_IS_FAVORITED                                \
00037     virtual bool isFavorited() const override;              \
00038     // DECLARE_IS_FAVORITED
00039
00040 #define DECLARE_SET_FAVORITED                                \
00041     virtual void setFavorited(const bool favorited) override; \
00042     // DECLARE_SET_FAVORITED
00043
00044 #define QN_DECLARE_FAVORITED                                \
00045     DECLARE_IS_FAVORITED                                    \
00046     DECLARE_SET_FAVORITED                                    \
00047     // QN_DECLARE_FAVORITED
00048
00049 #define DEFINE_IS_FAVORITED(type)                            \
00050     bool type::isFavorited() const                          \
00051     {                                                         \
00052         return d->m_isFavorited;                             \
00053     }                                                         \
00054     // DEFINE_IS_FAVORITED
00055
00056 #define DEFINE_SET_FAVORITED(type)                            \
00057     void type::setFavorited(const bool favorited)           \
00058     {                                                         \
00059         d->m_isFavorited = favorited;                         \
00060     }                                                         \
00061     // DEFINE_SET_FAVORITED
00062
00063 #define QN_DEFINE_FAVORITED(type)                            \
00064     DEFINE_IS_FAVORITED(type)                                \
00065     DEFINE_SET_FAVORITED(type)                                \
00066     // QN_DEFINE_FAVORITED
00067
00068 } // namespace quentier
00069
00070 #endif // LIB_QUENTIER_TYPES_I_FAVORITABLE_DATA_ELEMENT_H

```

## 6.39 ILocalStorageDataElement.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License

```

```

00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_I_LOCAL_STORAGE_DATA_ELEMENT_H
00020 #define LIB_QUENTIER_TYPES_I_LOCAL_STORAGE_DATA_ELEMENT_H
00021
00022 #include <quentier/utility/Linkage.h>
00023 #include <quentier/utility/UidGenerator.h>
00024
00025 #include <QString>
00026 #include <QUuid>
00027
00028 namespace quentier {
00029
00030 class QUENTIER_EXPORT ILocalStorageDataElement
00031 {
00032 public:
00033     virtual const QString localUid() const = 0;
00034     virtual void setLocalUid(const QString & guid) = 0;
00035     virtual void unsetLocalUid() = 0;
00036
00037     virtual ~ILocalStorageDataElement() {}
00038 };
00039
00040 #define DEFINE_LOCAL_UID_GETTER(type) //
00041     const QString type::localUid() const //
00042     { //
00043         return UidGenerator::UidToString(d->m_localUid); //
00044     } //
00045     // DEFINE_LOCAL_UID_GETTER
00046
00047 #define DEFINE_LOCAL_UID_SETTER(type) //
00048     void type::setLocalUid(const QString & uid) //
00049     { //
00050         d->m_localUid = uid; //
00051     } //
00052     // DEFINE_LOCAL_UID_SETTER
00053
00054 #define DEFINE_LOCAL_UID_UNSETTER(type) //
00055     void type::unsetLocalUid() //
00056     { //
00057         d->m_localUid = QUuid(); //
00058     } //
00059     // DEFINE_LOCAL_UID_UNSETTER
00060
00061 #define QN_DECLARE_LOCAL_UID //
00062     virtual const QString localUid() const override; //
00063     virtual void setLocalUid(const QString & guid) override; //
00064     virtual void unsetLocalUid() override; //
00065     // QN_DECLARE_LOCAL_UID
00066
00067 #define QN_DEFINE_LOCAL_UID(type) //
00068     DEFINE_LOCAL_UID_GETTER(type) //
00069     DEFINE_LOCAL_UID_SETTER(type) //
00070     DEFINE_LOCAL_UID_UNSETTER(type) //
00071     // QN_DEFINE_LOCAL_UID
00072
00073 } // namespace quentier
00074
00075 #endif // LIB_QUENTIER_TYPES_I_LOCAL_STORAGE_DATA_ELEMENT_H

```

## 6.40 INoteStoreDataElement.h

```

00001  /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_I_NOTE_STORE_DATA_ELEMENT_H
00020 #define LIB_QUENTIER_TYPES_I_NOTE_STORE_DATA_ELEMENT_H

```

```

00021
00022 #include "ILocalStorageDataElement.h"
00023
00024 #include <quentier/types/ErrorString.h>
00025 #include <quentier/utility/Printable.h>
00026
00027 #include <QUuid>
00028 #include <QtGlobal>
00029
00030 namespace quentier {
00031
00032 class QUENTIER_EXPORT INoteStoreDataElement :
00033     public ILocalStorageDataElement,
00034     public Printable
00035 {
00036 public:
00037     virtual void clear() = 0;
00038
00039     virtual bool hasGuid() const = 0;
00040     virtual const QString & guid() const = 0;
00041     virtual void setGuid(const QString & guid) = 0;
00042
00043     virtual bool hasUpdateSequenceNumber() const = 0;
00044     virtual quint32 updateSequenceNumber() const = 0;
00045     virtual void setUpdateSequenceNumber(const quint32 usn) = 0;
00046
00047     virtual bool checkParameters(ErrorString & errorDescription) const = 0;
00048
00049     virtual bool isDirty() const = 0;
00050     virtual void setDirty(const bool dirty) = 0;
00051
00052     virtual bool isLocal() const = 0;
00053     virtual void setLocal(const bool local) = 0;
00054
00055     virtual ~INoteStoreDataElement() {}
00056 };
00057
00058 #define DECLARE_IS_DIRTY
00059     virtual bool isDirty() const override;
00060     // DECLARE_IS_DIRTY
00061
00062 #define DECLARE_SET_DIRTY
00063     virtual void setDirty(const bool isDirty) override;
00064     // DECLARE_SET_DIRTY
00065
00066 #define QN_DECLARE_DIRTY
00067     DECLARE_IS_DIRTY
00068     DECLARE_SET_DIRTY
00069     // QN_DECLARE_DIRTY
00070
00071 #define DEFINE_IS_DIRTY(type)
00072     bool type::isDirty() const
00073     {
00074         return d->m_isDirty;
00075     }
00076     // DEFINE_IS_DIRTY
00077
00078 #define DEFINE_SET_DIRTY(type)
00079     void type::setDirty(const bool dirty)
00080     {
00081         d->m_isDirty = dirty;
00082     }
00083     // DEFINE_SET_DIRTY
00084
00085 #define QN_DEFINE_DIRTY(type)
00086     DEFINE_IS_DIRTY(type)
00087     DEFINE_SET_DIRTY(type)
00088     // QN_DEFINE_DIRTY
00089
00090 #define DECLARE_IS_LOCAL
00091     virtual bool isLocal() const override;
00092     // DECLARE_IS_LOCAL
00093
00094 #define DECLARE_SET_LOCAL
00095     virtual void setLocal(const bool isLocal) override;
00096     // DECLARE_SET_LOCAL
00097
00098 #define QN_DECLARE_LOCAL
00099     DECLARE_IS_LOCAL
00100     DECLARE_SET_LOCAL
00101     // QN_DECLARE_LOCAL
00102
00103 #define DEFINE_IS_LOCAL(type)
00104     bool type::isLocal() const
00105     {
00106         return d->m_isLocal;
00107     }

```

```

00108     // DEFINE_IS_LOCAL
00109
00110 #define DEFINE_SET_LOCAL(type)
00111     void type::setLocal(const bool local)
00112     {
00113         d->m_isLocal = local;
00114     }
00115     // DEFINE_SET_LOCAL
00116
00117 #define QN_DEFINE_LOCAL(type)
00118     DEFINE_IS_LOCAL(type)
00119     DEFINE_SET_LOCAL(type)
00120     // QN_DEFINE_LOCAL
00121
00122 } // namespace quantier
00123
00124 #endif // LIB_QUENTIER_TYPES_I_NOTE_STORE_DATA_ELEMENT_H

```

## 6.41 LinkedNotebook.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_LINKED_NOTEBOOK_H
00020 #define LIB_QUENTIER_TYPES_LINKED_NOTEBOOK_H
00021
00022 #include "INoteStoreDataElement.h"
00023
00024 #include <qt5qevercloud/QEverCloud.h>
00025
00026 #include <QSharedDataPointer>
00027
00028 namespace quantier {
00029
00030 QT_FORWARD_DECLARE_CLASS(LinkedNotebookData)
00031
00032 class QUENTIER_EXPORT LinkedNotebook : public INoteStoreDataElement
00033 {
00034 public:
00035     QN_DECLARE_DIRTY
00036
00037 public:
00038     explicit LinkedNotebook();
00039     LinkedNotebook(const LinkedNotebook & other);
00040     LinkedNotebook(LinkedNotebook && other);
00041     LinkedNotebook & operator=(const LinkedNotebook & other);
00042     LinkedNotebook & operator=(LinkedNotebook && other);
00043
00044     explicit LinkedNotebook(const qevercloud::LinkedNotebook & linkedNotebook);
00045     explicit LinkedNotebook(qevercloud::LinkedNotebook && linkedNotebook);
00046
00047     virtual ~LinkedNotebook() override;
00048
00049     const qevercloud::LinkedNotebook & qevercloudLinkedNotebook() const;
00050     qevercloud::LinkedNotebook & qevercloudLinkedNotebook();
00051
00052     bool operator==(const LinkedNotebook & other) const;
00053     bool operator!=(const LinkedNotebook & other) const;
00054
00055     virtual void clear() override;
00056
00057     virtual bool hasGuid() const override;
00058     virtual const QString & guid() const override;
00059     virtual void setGuid(const QString & guid) override;
00060
00061     virtual bool hasUpdateSequenceNumber() const override;
00062     virtual qint32 updateSequenceNumber() const override;
00063     virtual void setUpdateSequenceNumber(const qint32 usn) override;

```

```

00064
00065     virtual bool checkParameters(ErrorString & errorDescription) const override;
00066
00067     bool hasShareName() const;
00068     const QString & shareName() const;
00069     void setShareName(const QString & shareName);
00070
00071     bool hasUsername() const;
00072     const QString & username() const;
00073     void setUsername(const QString & username);
00074
00075     bool hasShardId() const;
00076     const QString & shardId() const;
00077     void setShardId(const QString & shardId);
00078
00079     bool hasSharedNotebookGlobalId() const;
00080     const QString & sharedNotebookGlobalId() const;
00081     void setSharedNotebookGlobalId(const QString & sharedNotebookGlobalId);
00082
00083     bool hasUri() const;
00084     const QString & uri() const;
00085     void setUri(const QString & uri);
00086
00087     bool hasNoteStoreUrl() const;
00088     const QString & noteStoreUrl() const;
00089     void setNoteStoreUrl(const QString & noteStoreUrl);
00090
00091     bool hasWebApiUrlPrefix() const;
00092     const QString & webApiUrlPrefix() const;
00093     void setWebApiUrlPrefix(const QString & webApiUrlPrefix);
00094
00095     bool hasStack() const;
00096     const QString & stack() const;
00097     void setStack(const QString & stack);
00098
00099     bool hasBusinessId() const;
00100     quint32 businessId() const;
00101     void setBusinessId(const quint32 businessId);
00102
00103     virtual QTextStream & print(QTextStream & strm) const override;
00104
00105 private:
00106     // hide useless methods inherited from the base class from public interface
00107     virtual const QString localUid() const override
00108     {
00109         return QString();
00110     }
00111     virtual void setLocalUid(const QString &) override {}
00112     virtual void unsetLocalUid() override {}
00113
00114     virtual bool isLocal() const override
00115     {
00116         return false;
00117     }
00118     virtual void setLocal(const bool) override {}
00119
00120 private:
00121     QSharedDataPointer<LinkedNotebookData> d;
00122 };
00123
00124 } // namespace quentier
00125
00126 Q_DECLARE_METATYPE(quentier::LinkedNotebook)
00127
00128 #endif // LIB_QUENTIER_TYPES_LINKED_NOTEBOOK_H

```

## 6.42 Note.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License

```

```

00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_NOTE_H
00020 #define LIB_QUENTIER_TYPES_NOTE_H
00021
00022 #include "IFavoritableDataElement.h"
00023 #include "SharedNote.h"
00024
00025 #include <qt5qevercloud/QEverCloud.h>
00026
00027 #include <QSharedDataPointer>
00028
00029 namespace quentier {
00030
00031 QT_FORWARD_DECLARE_CLASS(Resource)
00032 QT_FORWARD_DECLARE_CLASS(NoteData)
00033
00034 class QUENTIER_EXPORT Note : public IFavoritableDataElement
00035 {
00036 public:
00037     QN_DECLARE_LOCAL_UID
00038     QN_DECLARE_DIRTY
00039     QN_DECLARE_FAVORITED
00040     QN_DECLARE_LOCAL
00041
00042 public:
00043     explicit Note();
00044     Note(const Note & other);
00045     Note(Note && other);
00046     Note & operator=(const Note & other);
00047     Note & operator=(Note && other);
00048
00049     explicit Note(const qevercloud::Note & other);
00050     Note & operator=(const qevercloud::Note & other);
00051
00052     virtual ~Note() override;
00053
00054     bool operator==(const Note & other) const;
00055     bool operator!=(const Note & other) const;
00056
00057     const qevercloud::Note & qevercloudNote() const;
00058     qevercloud::Note & qevercloudNote();
00059
00060     virtual bool hasGuid() const override;
00061     virtual const QString & guid() const override;
00062     virtual void setGuid(const QString & guid) override;
00063
00064     virtual bool hasUpdateSequenceNumber() const override;
00065     virtual quint32 updateSequenceNumber() const override;
00066     virtual void setUpdateSequenceNumber(const quint32 usn) override;
00067
00068     virtual void clear() override;
00069
00070     static bool validateTitle(
00071         const QString & title, ErrorString * pErrorDescription = nullptr);
00072
00073     virtual bool checkParameters(ErrorString & errorDescription) const override;
00074
00075     bool hasTitle() const;
00076     const QString & title() const;
00077     void setTitle(const QString & title);
00078
00079     bool hasContent() const;
00080     const QString & content() const;
00081     void setContent(const QString & content);
00082
00083     bool hasContentHash() const;
00084     const QByteArray & contentHash() const;
00085     void setContentHash(const QByteArray & contentHash);
00086
00087     bool hasContentLength() const;
00088     quint32 contentLength() const;
00089     void setContentLength(const quint32 length);
00090
00091     bool hasCreationTimestamp() const;
00092     quint64 creationTimestamp() const;
00093     void setCreationTimestamp(const quint64 timestamp);
00094
00095     bool hasModificationTimestamp() const;
00096     quint64 modificationTimestamp() const;
00097     void setModificationTimestamp(const quint64 timestamp);
00098
00099     bool hasDeletionTimestamp() const;
00100     quint64 deletionTimestamp() const;
00101     void setDeletionTimestamp(const quint64 timestamp);
00102

```

```

00103     bool hasActive() const;
00104     bool active() const;
00105     void setActive(const bool active);
00106
00107     bool hasNotebookGuid() const;
00108     const QString & notebookGuid() const;
00109     void setNotebookGuid(const QString & guid);
00110
00111     bool hasNotebookLocalUid() const;
00112     const QString & notebookLocalUid() const;
00113     void setNotebookLocalUid(const QString & notebookLocalUid);
00114
00115     bool hasTagGuids() const;
00116     const QStringList tagGuids() const;
00117     void setTagGuids(const QStringList & guids);
00118     void addTagGuid(const QString & guid);
00119     void removeTagGuid(const QString & guid);
00120
00121     bool hasTagLocalUids() const;
00122     const QStringList & tagLocalUids() const;
00123     void setTagLocalUids(const QStringList & localUids);
00124     void addTagLocalUid(const QString & localUid);
00125     void removeTagLocalUid(const QString & localUid);
00126
00127     bool hasResources() const;
00128     int numResources() const;
00129     QList<Resource> resources() const;
00130     void setResources(const QList<Resource> & resources);
00131     void addResource(const Resource & resource);
00132     bool updateResource(const Resource & resource);
00133     bool removeResource(const Resource & resource);
00134
00135     bool hasNoteAttributes() const;
00136     const qevercloud::NoteAttributes & noteAttributes() const;
00137     qevercloud::NoteAttributes & noteAttributes();
00138     void clearNoteAttributes();
00139
00140     bool hasSharedNotes() const;
00141     QList<SharedNote> sharedNotes() const;
00142     void setSharedNotes(const QList<SharedNote> & sharedNotes);
00143     void addSharedNote(const SharedNote & sharedNote);
00144
00145     // NOTE: the shared note is recognized by its index in note
00146     // in the following two methods
00147     bool updateSharedNote(const SharedNote & sharedNote);
00148     bool removeSharedNote(const SharedNote & sharedNote);
00149
00150     bool hasNoteRestrictions() const;
00151     const qevercloud::NoteRestrictions & noteRestrictions() const;
00152     qevercloud::NoteRestrictions & noteRestrictions();
00153     void setNoteRestrictions(qevercloud::NoteRestrictions && restrictions);
00154
00155     bool hasNoteLimits() const;
00156     const qevercloud::NoteLimits & noteLimits() const;
00157     qevercloud::NoteLimits & noteLimits();
00158     void setNoteLimits(qevercloud::NoteLimits && limits);
00159
00160     QByteArray thumbnailData() const;
00161     void setThumbnailData(const QByteArray & thumbnailData);
00162
00163     bool isInkNote() const;
00164
00165     QString plainText(ErrorString * pErrorMessage = nullptr) const;
00166     QStringList listOfWords(ErrorString * pErrorMessage = nullptr) const;
00167
00168     std::pair<QString, QStringList> plainTextAndListOfWords(
00169         ErrorString * pErrorMessage = nullptr) const;
00170
00171     bool containsCheckedTodo() const;
00172     bool containsUncheckedTodo() const;
00173     bool containsTodo() const;
00174     bool containsEncryption() const;
00175
00176     virtual QTextStream & print(QTextStream & strm) const override;
00177 private:
00178     QSharedDataPointer<NoteData> d;
00179 };
00180
00181
00182 } // namespace quentier
00183
00184 Q_DECLARE_METATYPE(quentier::Note)
00185
00186 #endif // LIB_QUENTIER_TYPES_NOTE_H

```

## 6.43 Notebook.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_NOTEBOOK_H
00020 #define LIB_QUENTIER_TYPES_NOTEBOOK_H
00021
00022 #include "IFavoritableDataElement.h"
00023
00024 #include <qt5qevercloud/QEverCloud.h>
00025
00026 #include <QSharedDataPointer>
00027
00028 namespace quentier {
00029
00030 QT_FORWARD_DECLARE_CLASS(SharedNotebook)
00031 QT_FORWARD_DECLARE_CLASS(User)
00032 QT_FORWARD_DECLARE_CLASS(NotebookData)
00033
00034 class QUINTIER_EXPORT Notebook : public IFavoritableDataElement
00035 {
00036 public:
00037     QN_DECLARE_LOCAL_UID
00038     QN_DECLARE_DIRTY
00039     QN_DECLARE_LOCAL
00040     QN_DECLARE_FAVORITED
00041
00042 public:
00043     explicit Notebook();
00044     Notebook(const Notebook & other);
00045     Notebook(Notebook && other);
00046     Notebook & operator=(const Notebook & other);
00047     Notebook & operator=(Notebook && other);
00048
00049     explicit Notebook(const qevercloud::Notebook & other);
00050     explicit Notebook(qevercloud::Notebook && other);
00051     Notebook & operator=(const qevercloud::Notebook & other);
00052     Notebook & operator=(qevercloud::Notebook && other);
00053
00054     virtual ~Notebook() override;
00055
00056     bool operator==(const Notebook & other) const;
00057     bool operator!=(const Notebook & other) const;
00058
00059     const qevercloud::Notebook & qevercloudNotebook() const;
00060     qevercloud::Notebook & qevercloudNotebook();
00061
00062     virtual void clear() override;
00063
00064     static bool validateName(
00065         const QString & name, ErrorString * pErrorDescription = nullptr);
00066
00067     virtual bool hasGuid() const override;
00068     virtual const QString & guid() const override;
00069     virtual void setGuid(const QString & guid) override;
00070
00071     virtual bool hasUpdateSequenceNumber() const override;
00072     virtual quint32 updateSequenceNumber() const override;
00073     virtual void setUpdateSequenceNumber(const quint32 usn) override;
00074
00075     virtual bool checkParameters(ErrorString & errorDescription) const override;
00076
00077     bool hasName() const;
00078     const QString & name() const;
00079     void setName(const QString & name);
00080
00081     bool isDefaultNotebook() const;
00082     void setDefaultNotebook(const bool defaultNotebook);
00083
00084     bool hasLinkedNotebookGuid() const;
00085     const QString & linkedNotebookGuid() const;

```

```

00086     void setLinkedNotebookGuid(const QString & linkedNotebookGuid);
00087
00088     bool hasCreationTimestamp() const;
00089     qint64 creationTimestamp() const;
00090     void setCreationTimestamp(const qint64 timestamp);
00091
00092     bool hasModificationTimestamp() const;
00093     qint64 modificationTimestamp() const;
00094     void setModificationTimestamp(const qint64 timestamp);
00095
00096     bool hasPublishingUri() const;
00097     const QString & publishingUri() const;
00098     void setPublishingUri(const QString & uri);
00099
00100     bool hasPublishingOrder() const;
00101     qint8 publishingOrder() const;
00102     void setPublishingOrder(const qint8 order);
00103
00104     bool hasPublishingAscending() const;
00105     bool isPublishingAscending() const;
00106     void setPublishingAscending(const bool ascending);
00107
00108     bool hasPublishingPublicDescription() const;
00109     const QString & publishingPublicDescription() const;
00110     void setPublishingPublicDescription(
00111         const QString & publishingPublicDescription);
00112
00113     bool hasPublished() const;
00114     bool isPublished() const;
00115     void setPublished(const bool published);
00116
00117     bool hasStack() const;
00118     const QString & stack() const;
00119     void setStack(const QString & stack);
00120
00121     bool hasSharedNotebooks();
00122     QList<SharedNotebook> sharedNotebooks() const;
00123     void setSharedNotebooks(QList<qevercloud::SharedNotebook> sharedNotebooks);
00124     void setSharedNotebooks(QList<SharedNotebook> && notebooks);
00125     void addSharedNotebook(const SharedNotebook & sharedNotebook);
00126     void removeSharedNotebook(const SharedNotebook & sharedNotebook);
00127
00128     bool hasBusinessNotebookDescription() const;
00129     const QString & businessNotebookDescription() const;
00130
00131     void setBusinessNotebookDescription(
00132         const QString & businessNotebookDescription);
00133
00134     bool hasBusinessNotebookPrivilegeLevel() const;
00135     qint8 businessNotebookPrivilegeLevel() const;
00136     void setBusinessNotebookPrivilegeLevel(const qint8 privilegeLevel);
00137
00138     bool hasBusinessNotebookRecommended() const;
00139     bool isBusinessNotebookRecommended() const;
00140     void setBusinessNotebookRecommended(const bool recommended);
00141
00142     bool hasContact() const;
00143     const User contact() const;
00144     void setContact(const User & contact);
00145
00146     bool isLastUsed() const;
00147     void setLastUsed(const bool lastUsed);
00148
00149     bool canReadNotes() const;
00150     void setCanReadNotes(const bool canReadNotes);
00151
00152     bool canCreateNotes() const;
00153     void setCanCreateNotes(const bool canCreateNotes);
00154
00155     bool canUpdateNotes() const;
00156     void setCanUpdateNotes(const bool canUpdateNotes);
00157
00158     bool canExpungeNotes() const;
00159     void setCanExpungeNotes(const bool canExpungeNotes);
00160
00161     bool canShareNotes() const;
00162     void setCanShareNotes(const bool canShareNotes);
00163
00164     bool canEmailNotes() const;
00165     void setCanEmailNotes(const bool canEmailNotes);
00166
00167     bool canSendMessageToRecipients() const;
00168     void setCanSendMessageToRecipients(const bool canSendMessageToRecipients);
00169
00170     bool canUpdateNotebook() const;
00171     void setCanUpdateNotebook(const bool canUpdateNotebook);
00172

```

```

00173     bool canExpungeNotebook() const;
00174     void setCanExpungeNotebook(const bool canExpungeNotebook);
00175
00176     bool canSetDefaultNotebook() const;
00177     void setCanSetDefaultNotebook(const bool canSetDefaultNotebook);
00178
00179     bool canSetNotebookStack() const;
00180     void setCanSetNotebookStack(const bool canSetNotebookStack);
00181
00182     bool canPublishToPublic() const;
00183     void setCanPublishToPublic(const bool canPublishToPublic);
00184
00185     bool canPublishToBusinessLibrary() const;
00186     void setCanPublishToBusinessLibrary(const bool canPublishToBusinessLibrary);
00187
00188     bool canCreateTags() const;
00189     void setCanCreateTags(const bool canCreateTags);
00190
00191     bool canUpdateTags() const;
00192     void setCanUpdateTags(const bool canUpdateTags);
00193
00194     bool canExpungeTags() const;
00195     void setCanExpungeTags(const bool canExpungeTags);
00196
00197     bool canSetParentTag() const;
00198     void setCanSetParentTag(const bool canSetParentTag);
00199
00200     bool canCreateSharedNotebooks() const;
00201     void setCanCreateSharedNotebooks(const bool canCreateSharedNotebooks);
00202
00203     bool canShareNotesWithBusiness() const;
00204     void setCanShareNotesWithBusiness(const bool canShareNotesWithBusiness);
00205
00206     bool canRenameNotebook() const;
00207     void setCanRenameNotebook(const bool canRenameNotebook);
00208
00209     bool hasUpdateWhichSharedNotebookRestrictions() const;
00210     quint8 updateWhichSharedNotebookRestrictions() const;
00211     void setUpdateWhichSharedNotebookRestrictions(const quint8 which);
00212
00213     bool hasExpungeWhichSharedNotebookRestrictions() const;
00214     quint8 expungeWhichSharedNotebookRestrictions() const;
00215     void setExpungeWhichSharedNotebookRestrictions(const quint8 which);
00216
00217     bool hasRestrictions() const;
00218     const qevercloud::NotebookRestrictions & restrictions() const;
00219
00220     void setNotebookRestrictions(
00221         const qevercloud::NotebookRestrictions && restrictions);
00222
00223     bool hasRecipientReminderNotifyEmail() const;
00224     bool recipientReminderNotifyEmail() const;
00225     void setRecipientReminderNotifyEmail(const bool notifyEmail);
00226
00227     bool hasRecipientReminderNotifyInApp() const;
00228     bool recipientReminderNotifyInApp() const;
00229     void setRecipientReminderNotifyInApp(const bool notifyInApp);
00230
00231     bool hasRecipientInMyList() const;
00232     bool recipientInMyList() const;
00233     void setRecipientInMyList(const bool inMyList);
00234
00235     bool hasRecipientStack() const;
00236     const QString & recipientStack() const;
00237     void setRecipientStack(const QString & recipientString);
00238
00239     bool hasRecipientSettings() const;
00240     const qevercloud::NotebookRecipientSettings & recipientSettings() const;
00241     void setNotebookRecipientSettings(
00242         const qevercloud::NotebookRecipientSettings && settings);
00243
00244     virtual QTextStream & print(QTextStream & strm) const override;
00245
00246 private:
00247     QSharedDataPointer<NotebookData> d;
00248 };
00249
00250 } // namespace quantier
00251
00252 Q_DECLARE_METATYPE(quantier::Notebook)
00253
00254 #endif // LIB_QUENTIER_TYPES_NOTEBOOK_H

```

## 6.44 RegisterMetatypes.h

```

00001 /*
00002  * Copyright 2016-2019 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_REGISTER_METATYPES_H
00020 #define LIB_QUENTIER_TYPES_REGISTER_METATYPES_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 namespace quentier {
00025
00026 void QUENTIER_EXPORT registerMetatypes();
00027
00028 } // namespace quentier
00029
00030 #endif // LIB_QUENTIER_TYPES_REGISTER_METATYPES_H

```

## 6.45 Resource.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_RESOURCE_H
00020 #define LIB_QUENTIER_TYPES_RESOURCE_H
00021
00022 #include "INoteStoreDataElement.h"
00023 #include "Note.h"
00024
00025 namespace quentier {
00026
00027 QT_FORWARD_DECLARE_CLASS(ResourceData)
00028
00029 class QUENTIER_EXPORT Resource : public INoteStoreDataElement
00030 {
00031 public:
00032     QN_DECLARE_LOCAL_UID
00033     QN_DECLARE_DIRTY
00034     QN_DECLARE_LOCAL
00035
00036 public:
00037     explicit Resource();
00038     Resource(const Resource & other);
00039     Resource(Resource && other);
00040     explicit Resource(const qevercloud::Resource & resource);
00041     Resource & operator=(const Resource & other);
00042     Resource & operator=(Resource && other);
00043     virtual ~Resource() override;
00044
00045     bool operator==(const Resource & other) const;
00046     bool operator!=(const Resource & other) const;
00047
00048     const qevercloud::Resource & qevercloudResource() const;

```

```
00049     qevercloud::Resource & qevercloudResource();
00050
00051     virtual void clear() override;
00052
00053     virtual bool hasGuid() const override;
00054     virtual const QString & guid() const override;
00055     virtual void setGuid(const QString & guid) override;
00056
00057     virtual bool hasUpdateSequenceNumber() const override;
00058     virtual quint32 updateSequenceNumber() const override;
00059
00060     virtual void setUpdateSequenceNumber(
00061         const quint32 updateSequenceNumber) override;
00062
00063     virtual bool checkParameters(ErrorString & errorDescription) const override;
00064
00065     QString displayName() const;
00066     void setDisplayName(const QString & displayName);
00067
00068     QString preferredFileSuffix() const;
00069
00070     int indexInNote() const;
00071     void setIndexInNote(const int index);
00072
00073     bool hasNoteGuid() const;
00074     const QString & noteGuid() const;
00075     void setNoteGuid(const QString & guid);
00076
00077     bool hasNoteLocalUid() const;
00078     const QString & noteLocalUid() const;
00079     void setNoteLocalUid(const QString & guid);
00080
00081     bool hasData() const;
00082
00083     bool hasDataHash() const;
00084     const QByteArray & dataHash() const;
00085     void setDataHash(const QByteArray & hash);
00086
00087     bool hasDataSize() const;
00088     quint32 dataSize() const;
00089     void setDataSize(const quint32 size);
00090
00091     bool hasDataBody() const;
00092     const QByteArray & dataBody() const;
00093     void setDataBody(const QByteArray & body);
00094
00095     bool hasMime() const;
00096     const QString & mime() const;
00097     void setMime(const QString & mime);
00098
00099     bool hasWidth() const;
00100     quint16 width() const;
00101     void setWidth(const quint16 width);
00102
00103     bool hasHeight() const;
00104     quint16 height() const;
00105     void setHeight(const quint16 height);
00106
00107     bool hasRecognitionData() const;
00108
00109     bool hasRecognitionDataHash() const;
00110     const QByteArray & recognitionDataHash() const;
00111     void setRecognitionDataHash(const QByteArray & hash);
00112
00113     bool hasRecognitionDataSize() const;
00114     quint32 recognitionDataSize() const;
00115     void setRecognitionDataSize(const quint32 size);
00116
00117     bool hasRecognitionDataBody() const;
00118     const QByteArray & recognitionDataBody() const;
00119     void setRecognitionDataBody(const QByteArray & body);
00120
00121     bool hasAlternateData() const;
00122
00123     bool hasAlternateDataHash() const;
00124     const QByteArray & alternateDataHash() const;
00125     void setAlternateDataHash(const QByteArray & hash);
00126
00127     bool hasAlternateDataSize() const;
00128     quint32 alternateDataSize() const;
00129     void setAlternateDataSize(const quint32 size);
00130
00131     bool hasAlternateDataBody() const;
00132     const QByteArray & alternateDataBody() const;
00133     void setAlternateDataBody(const QByteArray & body);
00134
00135     bool hasResourceAttributes() const;
```

```

00136     const qevercloud::ResourceAttributes & resourceAttributes() const;
00137     qevercloud::ResourceAttributes & resourceAttributes();
00138
00139     void setResourceAttributes(
00140         const qevercloud::ResourceAttributes & attributes);
00141
00142     void setResourceAttributes(qevercloud::ResourceAttributes && attributes);
00143
00144     friend class Note;
00145
00146     virtual QTextStream & print(QTextStream & strm) const override;
00147
00148 private:
00149     QSharedDataPointer<ResourceData> d;
00150 };
00151
00152 } // namespace quantier
00153
00154 #endif // LIB_QUENTIER_TYPES_RESOURCE_H

```

## 6.46 ResourceRecognitionIndexItem.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_RESOURCE_RECOGNITION_INDEX_ITEM_H
00020 #define LIB_QUENTIER_TYPES_RESOURCE_RECOGNITION_INDEX_ITEM_H
00021
00022 #include <quentier/utility/Linkage.h>
00023 #include <quentier/utility/Printable.h>
00024
00025 #include <QByteArray>
00026 #include <QSharedDataPointer>
00027
00028 namespace quantier {
00029
00030 QT_FORWARD_DECLARE_CLASS(ResourceRecognitionIndexItemData)
00031
00032 class QUENTIER_EXPORT ResourceRecognitionIndexItem : public Printable
00033 {
00034 public:
00035     explicit ResourceRecognitionIndexItem();
00036     ResourceRecognitionIndexItem(const ResourceRecognitionIndexItem & other);
00037
00038     ResourceRecognitionIndexItem & operator=(
00039         const ResourceRecognitionIndexItem & other);
00040
00041     virtual ~ResourceRecognitionIndexItem() override;
00042
00043     bool isValid() const;
00044
00045     int x() const;
00046     void setX(const int x);
00047
00048     int y() const;
00049     void setY(const int y);
00050
00051     int h() const;
00052     void setH(const int h);
00053
00054     int w() const;
00055     void setW(const int w);
00056
00057     int offset() const;
00058     void setOffset(const int offset);
00059
00060     int duration() const;
00061     void setDuration(const int duration);

```

```

00062
00063     QVector<int> strokeList() const;
00064     int numStrokes() const;
00065     bool strokeAt(const int strokeIndex, int & stroke) const;
00066     bool setStrokeAt(const int strokeIndex, const int stroke);
00067     void setStrokeList(const QVector<int> & strokeList);
00068     void reserveStrokeListSpace(const int numItems);
00069     void addStroke(const int stroke);
00070     bool removeStroke(const int stroke);
00071     bool removeStrokeAt(const int strokeIndex);
00072
00073     struct TextItem
00074     {
00075         bool operator==(const TextItem & other) const
00076         {
00077             return (m_text == other.m_text) && (m_weight == other.m_weight);
00078         }
00079
00080         QString m_text;
00081         int m_weight = -1;
00082     };
00083
00084     QVector<TextItem> textItems() const;
00085     int numTextItems() const;
00086     bool textItemAt(const int textItemIndex, TextItem & textItem) const;
00087     bool setTextItemAt(const int textItemIndex, const TextItem & textItem);
00088     void setTextItems(const QVector<TextItem> & textItems);
00089     void reserveTextItemsSpace(const int numItems);
00090     void addTextItem(const TextItem & item);
00091     bool removeTextItem(const TextItem & item);
00092     bool removeTextItemAt(const int textItemIndex);
00093
00094     struct ObjectItem
00095     {
00096         bool operator==(const ObjectItem & other) const
00097         {
00098             return (m_objectType == other.m_objectType) &&
00099                 (m_weight == other.m_weight);
00100         }
00101
00102         QString m_objectType;
00103         int m_weight = -1;
00104     };
00105
00106     QVector<ObjectItem> objectItems() const;
00107     int numObjectItems() const;
00108     bool objectItemAt(const int objectItemIndex, ObjectItem & objectItem) const;
00109
00110     bool setObjectItemAt(
00111         const int objectItemIndex, const ObjectItem & objectItem);
00112
00113     void setObjectItems(const QVector<ObjectItem> & objectItems);
00114     void reserveObjectItemsSpace(const int numItems);
00115     void addObjectItem(const ObjectItem & item);
00116     bool removeObjectItem(const ObjectItem & item);
00117     bool removeObjectItemAt(const int objectItemIndex);
00118
00119     struct ShapeItem
00120     {
00121         bool operator==(const ShapeItem & other) const
00122         {
00123             return (m_shapeType == other.m_shapeType) &&
00124                 (m_weight == other.m_weight);
00125         }
00126
00127         QString m_shapeType;
00128         int m_weight = -1;
00129     };
00130
00131     QVector<ShapeItem> shapeItems() const;
00132     int numShapeItems() const;
00133     bool shapeItemAt(const int shapeItemIndex, ShapeItem & shapeItem) const;
00134     bool setShapeItemAt(const int shapeItemIndex, const ShapeItem & shapeItem);
00135     void setShapeItems(const QVector<ShapeItem> & shapeItems);
00136     void reserveShapeItemsSpace(const int numItems);
00137     void addShapeItem(const ShapeItem & item);
00138     bool removeShapeItem(const ShapeItem & item);
00139     bool removeShapeItemAt(const int shapeItemIndex);
00140
00141     struct BarcodeItem
00142     {
00143         bool operator==(const BarcodeItem & other) const
00144         {
00145             return (m_barcode == other.m_barcode) &&
00146                 (m_weight == other.m_weight);
00147         }
00148

```

```

00149         QString m_barcode;
00150         int m_weight = -1;
00151     };
00152
00153     QVector<BarcodeItem> barcodeItems() const;
00154     int numBarcodeItems() const;
00155
00156     bool barcodeItemAt(
00157         const int barcodeItemIndex, BarcodeItem & barcodeItem) const;
00158
00159     bool setBarcodeItemAt(
00160         const int barcodeItemIndex, const BarcodeItem & barcodeItem);
00161
00162     void setBarcodeItems(const QVector<BarcodeItem> & barcodeItems);
00163     void reserveBarcodeItemsSpace(const int numItems);
00164     void addBarcodeItem(const BarcodeItem & item);
00165     bool removeBarcodeItem(const BarcodeItem & item);
00166     bool removeBarcodeItemAt(const int barcodeItemIndex);
00167
00168     virtual QTextStream & print(QTextStream & strm) const override;
00169
00170 private:
00171     QSharedDataPointer<ResourceRecognitionIndexItemData> d;
00172 };
00173
00174 } // namespace quantier
00175
00176 #endif // LIB_QUENTIER_TYPES_RESOURCE_RECOGNITION_INDEX_ITEM_H

```

## 6.47 ResourceRecognitionIndices.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_RESOURCE_RECOGNITION_INDICES_H
00020 #define LIB_QUENTIER_TYPES_RESOURCE_RECOGNITION_INDICES_H
00021
00022 #include <quentier/types/ResourceRecognitionIndexItem.h>
00023
00024 #include <QByteArray>
00025 #include <QSharedDataPointer>
00026 #include <QVector>
00027
00028 namespace quantier {
00029
00030     QT_FORWARD_DECLARE_CLASS(ResourceRecognitionIndicesData)
00031
00032     class QUENTIER_EXPORT ResourceRecognitionIndices : public Printable
00033     {
00034     public:
00035         explicit ResourceRecognitionIndices();
00036         ResourceRecognitionIndices(const ResourceRecognitionIndices & other);
00037
00038         explicit ResourceRecognitionIndices(
00039             const QByteArray & rawRecognitionIndicesData);
00040
00041         ResourceRecognitionIndices & operator=(
00042             const ResourceRecognitionIndices & other);
00043
00044         virtual ~ResourceRecognitionIndices() override;
00045
00046         bool isNull() const;
00047         bool isValid() const;
00048
00049         QString objectId() const;
00050         QString objectType() const;
00051         QString recoType() const;
00052         QString engineVersion() const;

```

```

00053     QString docType() const;
00054     QString lang() const;
00055
00056     int objectHeight() const;
00057     int objectWidth() const;
00058
00059     QVector<ResourceRecognitionIndexItem> items() const;
00060
00061     bool setData(const QByteArray & rawRecognitionIndicesData);
00062
00063     virtual QTextStream & print(QTextStream & strm) const override;
00064
00065 private:
00066     QSharedDataPointer<ResourceRecognitionIndicesData> d;
00067 };
00068
00069 } // namespace quantier
00070
00071 #endif // LIB_QUENTIER_TYPES_RESOURCE_RECOGNITION_INDICES_H

```

## 6.48 SavedSearch.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_SAVED_SEARCH_H
00020 #define LIB_QUENTIER_TYPES_SAVED_SEARCH_H
00021
00022 #include "IFavoritableDataElement.h"
00023
00024 #include <qt5qevercloud/QEverCloud.h>
00025
00026 #include <QSharedDataPointer>
00027
00028 namespace quantier {
00029
00030     QT_FORWARD_DECLARE_CLASS(SavedSearchData)
00031
00032     class QUENTIER_EXPORT SavedSearch : public IFavoritableDataElement
00033     {
00034     public:
00035         QN_DECLARE_LOCAL_UID
00036         QN_DECLARE_DIRTY
00037         QN_DECLARE_LOCAL
00038         QN_DECLARE_FAVORITED
00039
00040     public:
00041         using QueryFormat = qevercloud::QueryFormat;
00042         using SavedSearchScope = qevercloud::SavedSearchScope;
00043
00044     public:
00045         explicit SavedSearch();
00046         SavedSearch(const SavedSearch & other);
00047         SavedSearch(SavedSearch && other);
00048         SavedSearch & operator=(const SavedSearch & other);
00049         SavedSearch & operator=(SavedSearch && other);
00050
00051         explicit SavedSearch(const qevercloud::SavedSearch & search);
00052         explicit SavedSearch(qevercloud::SavedSearch && search);
00053
00054         virtual ~SavedSearch() override;
00055
00056         const qevercloud::SavedSearch & qevercloudSavedSearch() const;
00057         qevercloud::SavedSearch & qevercloudSavedSearch();
00058
00059         bool operator==(const SavedSearch & other) const;
00060         bool operator!=(const SavedSearch & other) const;
00061

```

```

00062     virtual void clear() override;
00063
00064     static bool validateName(
00065         const QString & name, ErrorString * pErrorDescription = nullptr);
00066
00067     virtual bool hasGuid() const override;
00068     virtual const QString & guid() const override;
00069     virtual void setGuid(const QString & guid) override;
00070
00071     virtual bool hasUpdateSequenceNumber() const override;
00072     virtual quint32 updateSequenceNumber() const override;
00073     virtual void setUpdateSequenceNumber(const quint32 usn) override;
00074
00075     virtual bool checkParameters(ErrorString & errorDescription) const override;
00076
00077     bool hasName() const;
00078     const QString & name() const;
00079     void setName(const QString & name);
00080
00081     bool hasQuery() const;
00082     const QString & query() const;
00083     void setQuery(const QString & query);
00084
00085     bool hasQueryFormat() const;
00086     QueryFormat queryFormat() const;
00087     void setQueryFormat(const quint8 queryFormat);
00088
00089     bool hasIncludeAccount() const;
00090     bool includeAccount() const;
00091     void setIncludeAccount(const bool includeAccount);
00092
00093     bool hasIncludePersonalLinkedNotebooks() const;
00094     bool includePersonalLinkedNotebooks() const;
00095
00096     void setIncludePersonalLinkedNotebooks(
00097         const bool includePersonalLinkedNotebooks);
00098
00099     bool hasIncludeBusinessLinkedNotebooks() const;
00100     bool includeBusinessLinkedNotebooks() const;
00101
00102     void setIncludeBusinessLinkedNotebooks(
00103         const bool includeBusinessLinkedNotebooks);
00104
00105     virtual QTextStream & print(QTextStream & strm) const override;
00106
00107 private:
00108     QSharedDataPointer<SavedSearchData> d;
00109 };
00110
00111 } // namespace quantier
00112
00113 Q_DECLARE_METATYPE(quantier::SavedSearch)
00114
00115 #endif // LIB_QUENTIER_TYPES_SAVED_SEARCH_H

```

## 6.49 SharedNote.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_SHARED_NOTE_H
00020 #define LIB_QUENTIER_TYPES_SHARED_NOTE_H
00021
00022 #include <quantier/utility/Linkage.h>
00023 #include <quantier/utility/Printable.h>
00024
00025 #include <qt5qevercloud/QEverCloud.h>
00026

```

```

00027 #include <QSharedDataPointer>
00028
00029 namespace quantier {
00030
00031 QT_FORWARD_DECLARE_CLASS(SharedNoteData)
00032
00033 class QUENTIER_EXPORT SharedNote : public Printable
00034 {
00035 public:
00036     using SharedNotePrivilegeLevel = qevercloud::SharedNotePrivilegeLevel;
00037     using ContactType = qevercloud::ContactType;
00038
00039 public:
00040     explicit SharedNote();
00041     SharedNote(const SharedNote & other);
00042     SharedNote(SharedNote && other);
00043     explicit SharedNote(const qevercloud::SharedNote & sharedNote);
00044     SharedNote & operator=(const SharedNote & other);
00045     SharedNote & operator=(SharedNote && other);
00046     virtual ~SharedNote() override;
00047
00048     bool operator==(const SharedNote & other) const;
00049     bool operator!=(const SharedNote & other) const;
00050
00051     const qevercloud::SharedNote & qevercloudSharedNote() const;
00052     qevercloud::SharedNote & qevercloudSharedNote();
00053
00054     const QString & noteGuid() const;
00055     void setNoteGuid(const QString & noteGuid);
00056
00057     int indexInNote() const;
00058     void setIndexInNote(const int index);
00059
00060     bool hasSharerUserId() const;
00061     quint32 sharerUserId() const;
00062     void setSharerUserId(const quint32 id);
00063
00064     bool hasRecipientIdentityId() const;
00065     quint64 recipientIdentityId() const;
00066     void setRecipientIdentityId(const quint64 identityId);
00067
00068     bool hasRecipientIdentityContactName() const;
00069     const QString & recipientIdentityContactName() const;
00070
00071     void setRecipientIdentityContactName(
00072         const QString & recipientIdentityContactName);
00073
00074     bool hasRecipientIdentityContactId() const;
00075     const QString & recipientIdentityContactId() const;
00076
00077     void setRecipientIdentityContactId(
00078         const QString & recipientIdentityContactId);
00079
00080     bool hasRecipientIdentityContactType() const;
00081     ContactType recipientIdentityContactType() const;
00082
00083     void setRecipientIdentityContactType(
00084         const ContactType recipientIdentityContactType);
00085
00086     void setRecipientIdentityContactType(
00087         const quint32 recipientIdentityContactType);
00088
00089     bool hasRecipientIdentityContactPhotoUrl() const;
00090     const QString & recipientIdentityContactPhotoUrl() const;
00091
00092     void setRecipientIdentityContactPhotoUrl(
00093         const QString & recipientIdentityPhotoUrl);
00094
00095     bool hasRecipientIdentityContactPhotoLastUpdated() const;
00096     quint64 recipientIdentityContactPhotoLastUpdated() const;
00097
00098     void setRecipientIdentityContactPhotoLastUpdated(
00099         const quint64 recipientIdentityPhotoLastUpdated);
00100
00101     bool hasRecipientIdentityContactMessagingPermit() const;
00102     const QByteArray & recipientIdentityContactMessagingPermit() const;
00103
00104     void setRecipientIdentityContactMessagingPermit(
00105         const QByteArray & messagingPermit);
00106
00107     bool hasRecipientIdentityContactMessagingPermitExpires() const;
00108     quint64 recipientIdentityContactMessagingPermitExpires() const;
00109
00110     void setRecipientIdentityContactMessagingPermitExpires(
00111         const quint64 timestamp);
00112
00113     bool hasRecipientIdentityUserId() const;

```

```

00114     qint32 recipientIdentityUserId() const;
00115     void setRecipientIdentityUserId(const qint32 userId);
00116
00117     bool hasRecipientIdentityDeactivated() const;
00118     bool recipientIdentityDeactivated() const;
00119     void setRecipientIdentityDeactivated(const bool deactivated);
00120
00121     bool hasRecipientIdentitySameBusiness() const;
00122     bool recipientIdentitySameBusiness() const;
00123     void setRecipientIdentitySameBusiness(const bool sameBusiness);
00124
00125     bool hasRecipientIdentityBlocked() const;
00126     bool recipientIdentityBlocked() const;
00127     void setRecipientIdentityBlocked(const bool blocked);
00128
00129     bool hasRecipientIdentityUserConnected() const;
00130     bool recipientIdentityUserConnected() const;
00131     void setRecipientIdentityUserConnected(const bool userConnected);
00132
00133     bool hasRecipientIdentityEventId() const;
00134     qint64 recipientIdentityEventId() const;
00135     void setRecipientIdentityEventId(const qint64 eventId);
00136
00137     bool hasRecipientIdentity() const;
00138     const qevercloud::Identity & recipientIdentity() const;
00139     void setRecipientIdentity(qevercloud::Identity && identity);
00140
00141     bool hasPrivilegeLevel() const;
00142     SharedNotePrivilegeLevel privilegeLevel() const;
00143     void setPrivilegeLevel(const SharedNotePrivilegeLevel level);
00144     void setPrivilegeLevel(const qint8 level);
00145
00146     bool hasCreationTimestamp() const;
00147     qint64 creationTimestamp() const;
00148     void setCreationTimestamp(const qint64 timestamp);
00149
00150     bool hasModificationTimestamp() const;
00151     qint64 modificationTimestamp() const;
00152     void setModificationTimestamp(const qint64 timestamp);
00153
00154     bool hasAssignmentTimestamp() const;
00155     qint64 assignmentTimestamp() const;
00156     void setAssignmentTimestamp(const qint64 timestamp);
00157
00158     virtual QTextStream & print(QTextStream & strm) const override;
00159
00160 private:
00161     QSharedDataPointer<SharedNoteData> d;
00162 };
00163
00164 } // namespace quentier
00165
00166 #endif // LIB_QUENTIER_TYPES_SHARED_NOTE_H

```

## 6.50 SharedNotebook.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_SHARED_NOTEBOOK_H
00020 #define LIB_QUENTIER_TYPES_SHARED_NOTEBOOK_H
00021
00022 #include <quentier/utility/Printable.h>
00023
00024 #include <qt5qevercloud/QEverCloud.h>
00025
00026 #include <QSharedDataPointer>
00027

```

```

00028 namespace quantier {
00029
00030 QT_FORWARD_DECLARE_CLASS(SharedNotebookData)
00031
00032 class QUENTIER_EXPORT SharedNotebook : public Printable
00033 {
00034 public:
00035     using SharedNotebookPrivilegeLevel =
00036         qevercloud::SharedNotebookPrivilegeLevel;
00037
00038 public:
00039     explicit SharedNotebook();
00040     SharedNotebook(const SharedNotebook & other);
00041     SharedNotebook(SharedNotebook && other);
00042
00043     explicit SharedNotebook(
00044         const qevercloud::SharedNotebook & qecSharedNotebook);
00045
00046     SharedNotebook & operator=(const SharedNotebook & other);
00047     SharedNotebook & operator=(SharedNotebook && other);
00048
00049     virtual ~SharedNotebook() override;
00050
00051     bool operator==(const SharedNotebook & other) const;
00052     bool operator!=(const SharedNotebook & other) const;
00053
00054     const qevercloud::SharedNotebook & qevercloudSharedNotebook() const;
00055     qevercloud::SharedNotebook & qevercloudSharedNotebook();
00056
00057     int indexInNotebook() const;
00058     void setIndexInNotebook(const int index);
00059
00060     bool hasId() const;
00061     quint64 id() const;
00062     void setId(const quint64 id);
00063
00064     bool hasUserId() const;
00065     quint32 userId() const;
00066     void setUserId(const quint32 userId);
00067
00068     bool hasNotebookGuid() const;
00069     const QString & notebookGuid() const;
00070     void setNotebookGuid(const QString & notebookGuid);
00071
00072     bool hasEmail() const;
00073     const QString & email() const;
00074     void setEmail(const QString & email);
00075
00076     bool hasCreationTimestamp() const;
00077     quint64 creationTimestamp() const;
00078     void setCreationTimestamp(const quint64 timestamp);
00079
00080     bool hasModificationTimestamp() const;
00081     quint64 modificationTimestamp() const;
00082     void setModificationTimestamp(const quint64 timestamp);
00083
00084     bool hasUsername() const;
00085     const QString & username() const;
00086     void setUsername(const QString & username);
00087
00088     bool hasPrivilegeLevel() const;
00089     SharedNotebookPrivilegeLevel privilegeLevel() const;
00090     void setPrivilegeLevel(const SharedNotebookPrivilegeLevel privilegeLevel);
00091     void setPrivilegeLevel(const quint8 privilegeLevel);
00092
00093     bool hasReminderNotifyEmail() const;
00094     bool reminderNotifyEmail() const;
00095     void setReminderNotifyEmail(const bool notifyEmail);
00096
00097     bool hasReminderNotifyApp() const;
00098     bool reminderNotifyApp() const;
00099     void setReminderNotifyApp(const bool notifyApp);
00100
00101     bool hasRecipientUsername() const;
00102     const QString & recipientUsername() const;
00103     void setRecipientUsername(const QString & recipientUsername);
00104
00105     bool hasRecipientUserId() const;
00106     quint32 recipientUserId() const;
00107     void setRecipientUserId(const quint32 userId);
00108
00109     bool hasRecipientIdentityId() const;
00110     quint64 recipientIdentityId() const;
00111     void setRecipientIdentityId(const quint64 recipientIdentityId);
00112
00113     bool hasGlobalId() const;
00114     const QString & globalId() const;

```

```

00115     void setGlobalId(const QString & globalId);
00116
00117     bool hasSharerUserId() const;
00118     quint32 sharerUserId() const;
00119     void setSharerUserId(quint32 sharerUserId);
00120
00121     bool hasAssignmentTimestamp() const;
00122     quint64 assignmentTimestamp() const;
00123     void setAssignmentTimestamp(const quint64 timestamp);
00124
00125     virtual QTextStream & print(QTextStream & strm) const override;
00126
00127     friend class Notebook;
00128
00129 private:
00130     QSharedDataPointer<SharedNotebookData> d;
00131 };
00132
00133 } // namespace quentier
00134
00135 #endif // LIB_QUENTIER_TYPES_SHARED_NOTEBOOK_H

```

## 6.51 Tag.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_TAG_H
00020 #define LIB_QUENTIER_TYPES_TAG_H
00021
00022 #include "IFavoritableDataElement.h"
00023
00024 #include <qt5qevercloud/QEverCloud.h>
00025
00026 #include <QSharedDataPointer>
00027
00028 namespace quentier {
00029
00030 QT_FORWARD_DECLARE_CLASS(TagData)
00031
00032 class QUENTIER_EXPORT Tag : public IFavoritableDataElement
00033 {
00034 public:
00035     QN_DECLARE_LOCAL_UID
00036     QN_DECLARE_DIRTY
00037     QN_DECLARE_LOCAL
00038     QN_DECLARE_FAVORITED
00039
00040 public:
00041     explicit Tag();
00042     Tag(const Tag & other);
00043     Tag(Tag && other);
00044     Tag & operator=(const Tag & other);
00045     Tag & operator=(Tag && other);
00046
00047     explicit Tag(const qevercloud::Tag & other);
00048     explicit Tag(qevercloud::Tag && other);
00049
00050     virtual ~Tag() override;
00051
00052     bool operator==(const Tag & other) const;
00053     bool operator!=(const Tag & other) const;
00054
00055     const qevercloud::Tag & qevercloudTag() const;
00056     qevercloud::Tag & qevercloudTag();
00057
00058     virtual void clear() override;
00059

```

```

00060     static bool validateName(
00061         const QString & name, ErrorString * pErrorDescription = nullptr);
00062
00063     virtual bool hasGuid() const override;
00064     virtual const QString & guid() const override;
00065     virtual void setGuid(const QString & guid) override;
00066
00067     virtual bool hasUpdateSequenceNumber() const override;
00068     virtual quint32 updateSequenceNumber() const override;
00069     virtual void setUpdateSequenceNumber(const quint32 usn) override;
00070
00071     virtual bool checkParameters(ErrorString & errorDescription) const override;
00072
00073     bool hasName() const;
00074     const QString & name() const;
00075     void setName(const QString & name);
00076
00077     bool hasParentGuid() const;
00078     const QString & parentGuid() const;
00079     void setParentGuid(const QString & parentGuid);
00080
00081     bool hasParentLocalUid() const;
00082     const QString & parentLocalUid() const;
00083     void setParentLocalUid(const QString & parentLocalUid);
00084
00085     bool hasLinkedNotebookGuid() const;
00086     const QString & linkedNotebookGuid() const;
00087     void setLinkedNotebookGuid(const QString & linkedNotebookGuid);
00088
00089     virtual QTextStream & print(QTextStream & strm) const override;
00090
00091 private:
00092     QSharedDataPointer<TagData> d;
00093 };
00094
00095 } // namespace quantier
00096
00097 Q_DECLARE_METATYPE(quantier::Tag)
00098
00099 #endif // LIB_QUENTIER_TYPES_TAG_H

```

## 6.52 User.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_TYPES_USER_H
00020 #define LIB_QUENTIER_TYPES_USER_H
00021
00022 #include <quantier/types/ErrorString.h>
00023 #include <quantier/utility/Printable.h>
00024
00025 #include <qt5qevercloud/QEverCloud.h>
00026
00027 #include <QtGlobal>
00028
00029 namespace quantier {
00030
00031     QT_FORWARD_DECLARE_CLASS(UserData)
00032
00033     class QUENTIER_EXPORT User : public Printable
00034     {
00035     public:
00036         using PrivilegeLevel = qevercloud::PrivilegeLevel;
00037         using ServiceLevel = qevercloud::ServiceLevel;
00038
00039     public:
00040         explicit User();

```

```

00041     explicit User(const qevercloud::User & user);
00042     explicit User(qevercloud::User && user);
00043     User(const User & other);
00044     User(User && other);
00045     User & operator=(const User & other);
00046     User & operator=(User && other);
00047     virtual ~User() override;
00048
00049     bool operator==(const User & other) const;
00050     bool operator!=(const User & other) const;
00051
00052     const qevercloud::User & qevercloudUser() const;
00053     qevercloud::User & qevercloudUser();
00054
00055     void clear();
00056
00057     bool isDirty() const;
00058     void setDirty(const bool dirty);
00059
00060     bool isLocal() const;
00061     void setLocal(const bool local);
00062
00063     bool checkParameters(ErrorString & errorDescription) const;
00064
00065     bool hasId() const;
00066     qint32 id() const;
00067     void setId(const qint32 id);
00068
00069     bool hasUsername() const;
00070     const QString & username() const;
00071     void setUsername(const QString & username);
00072
00073     bool hasEmail() const;
00074     const QString & email() const;
00075     void setEmail(const QString & email);
00076
00077     bool hasName() const;
00078     const QString & name() const;
00079     void setName(const QString & name);
00080
00081     bool hasTimezone() const;
00082     const QString & timezone() const;
00083     void setTimezone(const QString & timezone);
00084
00085     bool hasPrivilegeLevel() const;
00086     PrivilegeLevel privilegeLevel() const;
00087     void setPrivilegeLevel(const qint8 level);
00088
00089     bool hasServiceLevel() const;
00090     ServiceLevel serviceLevel() const;
00091     void setServiceLevel(const qint8 level);
00092
00093     bool hasCreationTimestamp() const;
00094     qint64 creationTimestamp() const;
00095     void setCreationTimestamp(const qint64 timestamp);
00096
00097     bool hasModificationTimestamp() const;
00098     qint64 modificationTimestamp() const;
00099     void setModificationTimestamp(const qint64 timestamp);
00100
00101     bool hasDeletionTimestamp() const;
00102     qint64 deletionTimestamp() const;
00103     void setDeletionTimestamp(const qint64 timestamp);
00104
00105     bool hasActive() const;
00106     bool active() const;
00107     void setActive(const bool active);
00108
00109     bool hasShardId() const;
00110     const QString & shardId() const;
00111     void setShardId(const QString & shardId);
00112
00113     bool hasUserAttributes() const;
00114     const qevercloud::UserAttributes & userAttributes() const;
00115     void setUserAttributes(qevercloud::UserAttributes && attributes);
00116
00117     bool hasAccounting() const;
00118     const qevercloud::Accounting & accounting() const;
00119     void setAccounting(qevercloud::Accounting && accounting);
00120
00121     bool hasBusinessUserInfo() const;
00122     const qevercloud::BusinessUserInfo & businessUserInfo() const;
00123     void setBusinessUserInfo(qevercloud::BusinessUserInfo && info);
00124
00125     bool hasPhotoUrl() const;
00126     QString photoUrl() const;
00127     void setPhotoUrl(const QString & photoUrl);

```

```

00128
00129     bool hasPhotoLastUpdateTimestamp() const;
00130     qint64 photoLastUpdateTimestamp() const;
00131     void setPhotoLastUpdateTimestamp(const qint64 timestamp);
00132
00133     bool hasAccountLimits() const;
00134     const qevercloud::AccountLimits & accountLimits() const;
00135     void setAccountLimits(qevercloud::AccountLimits && limits);
00136
00137     virtual QTextStream & print(QTextStream & strm) const override;
00138
00139     friend class Notebook;
00140
00141 private:
00142     QSharedDataPointer<UserData> d;
00143 };
00144
00145 } // namespace quentier
00146
00147 #endif // LIB_QUENTIER_TYPES_USER_H

```

## 6.53 ApplicationSettings.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_APPLICATION_SETTINGS_H
00020 #define LIB_QUENTIER_UTILITY_APPLICATION_SETTINGS_H
00021
00022 #include <quentier/types/Account.h>
00023
00024 #include <QSettings>
00025
00026 namespace quentier {
00027
00033 class QUINTIER_EXPORT ApplicationSettings : public QSettings, public Printable
00034 {
00035     Q_OBJECT
00036 public:
00046     ApplicationSettings(const QString & settingsName = {});
00047
00059     ApplicationSettings(
00060         const Account & account, const QString & settingsName = {});
00061
00078     ApplicationSettings(
00079         const Account & account, const char * settingsName,
00080         const int settingsNameSize = -1);
00081
00085     virtual ~ApplicationSettings() override;
00086
00087 public:
00092     struct ArrayCloser
00093     {
00094         ArrayCloser(ApplicationSettings & settings) : m_settings(settings) {}
00095
00096         ~ArrayCloser()
00097         {
00098             m_settings.endArray();
00099             m_settings.sync();
00100         }
00101
00102         ApplicationSettings & m_settings;
00103     };
00104
00109     struct GroupCloser
00110     {
00111         GroupCloser(ApplicationSettings & settings) : m_settings(settings) {}
00112

```

```

00113     ~GroupCloser()
00114     {
00115         m_settings.endGroup();
00116         m_settings.sync();
00117     }
00118
00119     ApplicationSettings & m_settings;
00120 };
00121
00122 public:
00123     void beginGroup(const QString & prefix);
00130
00141     void beginGroup(const char * prefix, const int size = -1);
00142
00150     int beginReadArray(const QString & prefix);
00151
00162     int beginReadArray(const char * prefix, const int size = -1);
00163
00174     void beginWriteArray(const QString & prefix, const int arraySize = -1);
00175
00190     void beginWriteArray(
00191         const char * prefix, const int arraySize = -1,
00192         const int prefixSize = -1);
00193
00201     bool contains(const QString & key) const;
00202
00214     bool contains(const char * key, const int size = -1) const;
00215
00222     void remove(const QString & key);
00223
00234     void remove(const char * key, const int size = -1);
00235
00243     void setValue(const QString & key, const QVariant & value);
00244
00256     void setValue(
00257         const char * key, const QVariant & value, const int keySize = -1);
00258
00269     QVariant value(
00270         const QString & key, const QVariant & defaultValue = {}) const;
00271
00286     QVariant value(
00287         const char * key, const QVariant & defaultValue = {},
00288         const int keySize = -1) const;
00289
00290 public:
00291     virtual QTextStream & print(QTextStream & strm) const override;
00292
00293 private:
00294     Q_DISABLE_COPY(ApplicationSettings)
00295 };
00296
00297 } // namespace quentier
00298
00299 #endif // LIB_QUENTIER_UTILITY_APPLICATION_SETTINGS_H

```

## 6.54 Checks.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_CHECKS_H
00020 #define LIB_QUENTIER_UTILITY_CHECKS_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025
00026 namespace quentier {

```

```

00027
00034 bool QUENTIER_EXPORT checkGuid(const QString & guid);
00035
00045 bool QUENTIER_EXPORT
00046 checkUpdateSequenceNumber(const int32_t updateSequenceNumber);
00047
00048 } // namespace quentier
00049
00050 #endif // LIB_QUENTIER_UTILITY_CHECKS_H

```

## 6.55 Compat.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_COMPAT_H
00020 #define LIB_QUENTIER_UTILITY_COMPAT_H
00021
00022 #include <QHash>
00023 #include <QString>
00024 #include <QtGlobal>
00025
00026 #if QT_VERSION < QT_VERSION_CHECK(5, 7, 0)
00027 #include <type_traits>
00028 #endif
00029
00030 // Compatibility with older Qt versions
00031
00032 #if QT_VERSION < QT_VERSION_CHECK(5, 7, 0)
00033
00034 // this adds const to non-const objects (like std::as_const)
00035 template <typename T>
00036 Q_DECL_CONSTEXPR typename std::add_const<T>::type & qAsConst(T & t)
00037     Q_DECL_NOTHROW
00038 {
00039     return t;
00040 }
00041
00042 // prevent rvalue arguments:
00043 template <typename T>
00044 void qAsConst(const T &&) = delete;
00045
00046 #endif
00047
00048 // Compatibility with boost parts which require to take a hash of QString
00049
00050 inline std::size_t hash_value(QString x) noexcept
00051 {
00052     return qHash(x);
00053 }
00054
00055 #endif // LIB_QUENTIER_UTILITY_COMPAT_H

```

## 6.56 DateTime.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *

```

```

00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_DATE_TIME_H
00020 #define LIB_QUENTIER_UTILITY_DATE_TIME_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QFlags>
00025
00026 namespace quentier {
00027
00028 constexpr int secondsToMilliseconds(int seconds) noexcept
00029 {
00030     return seconds * 1000;
00031 }
00032
00033 class QUINTIER_EXPORT DateTimePrint
00034 {
00035 public:
00036     enum Option
00037     {
00038         IncludeNumericTimestamp = 1 « 1,
00039         IncludeMilliseconds = 1 « 2,
00040         IncludeTimezone = 1 « 3
00041     };
00042     Q_DECLARE_FLAGS(Options, Option)
00043 };
00044
00045 Q_DECLARE_OPERATORS_FOR_FLAGS(DateTimePrint::Options)
00046
00047 const QString QUINTIER_EXPORT printableDateTimeFromTimestamp(
00048     const qint64 timestamp,
00049     DateTimePrint::Options options = DateTimePrint::Options(
00050         DateTimePrint::IncludeNumericTimestamp |
00051         DateTimePrint::IncludeMilliseconds | DateTimePrint::IncludeTimezone),
00052     const char * customFormat = nullptr);
00053
00054 } // namespace quentier
00055
00056 #endif // LIB_QUENTIER_UTILITY_DATE_TIME_H

```

## 6.57 EncryptionManager.h

```

00001  /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_ENCRYPTION_MANAGER_H
00020 #define LIB_QUENTIER_UTILITY_ENCRYPTION_MANAGER_H
00021
00022 #include <quentier/types/ErrorMessage.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 #include <QObject>
00026 #include <QString>
00027 #include <QUuid>
00028
00029 namespace quentier {
00030
00031 QT_FORWARD_DECLARE_CLASS(EncryptionManagerPrivate)
00032

```

```

00033
00038 class QUENTIER_EXPORT EncryptionManager : public QObject
00039 {
00040     Q_OBJECT
00041 public:
00042     explicit EncryptionManager(QObject * parent = nullptr);
00043     virtual ~EncryptionManager();
00044
00045     bool decrypt(
00046         const QString & encryptedText, const QString & passphrase,
00047         const QString & cipher, const size_t keyLength, QString & decryptedText,
00048         ErrorString & errorDescription);
00049
00050     bool encrypt(
00051         const QString & textToEncrypt, const QString & passphrase,
00052         QString & cipher, size_t & keyLength, QString & encryptedText,
00053         ErrorString & errorDescription);
00054
00055     Q_SIGNALS:
00056         void decryptedText(
00057             QString text, bool success, ErrorString errorDescription,
00058             QUuid requestId);
00059
00060         void encryptedText(
00061             QString encryptedText, bool success, ErrorString errorDescription,
00062             QUuid requestId);
00063
00064     public Q_SLOTS:
00065         void onDecryptTextRequest(
00066             QString encryptedText, QString passphrase, QString cipher,
00067             size_t keyLength, QUuid requestId);
00068
00069         void onEncryptTextRequest(
00070             QString textToEncrypt, QString passphrase, QString cipher,
00071             size_t keyLength, QUuid requestId);
00072
00073 private:
00074     EncryptionManagerPrivate * const d_ptr;
00075     Q_DECLARE_PRIVATE(EncryptionManager)
00076 };
00077
00078 } // namespace quantier
00079
00080 #endif // LIB_QUENTIER_UTILITY_ENCRYPTION_MANAGER_H

```

## 6.58 EventLoopWithExitStatus.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquantier
00005  *
00006  * libquantier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquantier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquantier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_EVENT_LOOP_WITH_EXIT_STATUS_H
00020 #define LIB_QUENTIER_UTILITY_EVENT_LOOP_WITH_EXIT_STATUS_H
00021
00022 #include <quantier/types/ErrorString.h>
00023 #include <quantier/utility/Linkage.h>
00024
00025 #include <QEventLoop>
00026
00027 QT_FORWARD_DECLARE_CLASS(QDebug)
00028 QT_FORWARD_DECLARE_CLASS(QTextStream)
00029
00030 namespace quantier {
00031
00032 class QUENTIER_EXPORT EventLoopWithExitStatus : public QEventLoop
00033 {
00034     Q_OBJECT
00035 public:
00036     explicit EventLoopWithExitStatus(QObject * parent = nullptr);

```

```

00037
00038     enum class ExitStatus
00039     {
00040         Success,
00041         Failure,
00042         Timeout
00043     };
00044
00045     friend QDebug & operator«(QDebug & dbg, const ExitStatus status);
00046
00047     friend QDataStream & operator«(
00048         QDataStream & strm, const ExitStatus status);
00049
00050     ExitStatus exitStatus() const;
00051     const QString & errorDescription() const;
00052
00053 public Q_SLOTS:
00054     void exitAsSuccess();
00055     void exitAsFailure();
00056     void exitAsFailureWithError(QString errorDescription);
00057     void exitAsFailureWithError(QString(ErrorString errorDescription);
00058     void exitAsTimeout();
00059
00060 private:
00061     ExitStatus m_exitStatus;
00062     QString m_errorDescription;
00063 };
00064
00065 } // namespace quentier
00066
00067 #endif // LIB_QUENTIER_UTILITY_EVENT_LOOP_WITH_EXIT_STATUS_H

```

## 6.59 FileCopier.h

```

00001 /*
00002  * Copyright 2018-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_FILE_COPIER_H
00020 #define LIB_QUENTIER_UTILITY_FILE_COPIER_H
00021
00022 #include <quentier/types/ErrorMessage.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 #include <QObject>
00026 #include <QString>
00027
00028 QT_FORWARD_DECLARE_CLASS(QDebug)
00029 QT_FORWARD_DECLARE_CLASS(QDataStream)
00030
00031 namespace quentier {
00032
00033 QT_FORWARD_DECLARE_CLASS(FileCopierPrivate)
00034
00035 class QUENTIER_EXPORT FileCopier : public QObject
00036 {
00037     Q_OBJECT
00038 public:
00039     explicit FileCopier(QObject * parent = nullptr);
00040
00041     enum class State
00042     {
00043         Idle = 0,
00044         Copying,
00045         Cancelling
00046     };
00047
00048     friend QDebug & operator«(QDebug & dbg, const State state);
00049     friend QDataStream & operator«(QDataStream & strm, const State state);

```

```

00050
00051     State state() const;
00052
00053     QString sourceFilePath() const;
00054     QString destinationFilePath() const;
00055
00056     double currentProgress() const;
00057
00058     Q_SIGNALS:
00059         void progressUpdate(double progress);
00060         void finished(QString sourcePath, QString destPath);
00061         void cancelled(QString sourcePath, QString destPath);
00062         void notifyError(ErrorString error);
00063
00064     public Q_SLOTS:
00065         void copyFile(QString sourcePath, QString destPath);
00066         void cancel();
00067
00068     private:
00069         Q_DISABLE_COPY(FileCopier)
00070
00071     private:
00072         FileCopierPrivate * d_ptr;
00073         Q_DECLARE_PRIVATE(FileCopier)
00074 };
00075
00076 } // namespace quentier
00077
00078 #endif // LIB_QUENTIER_UTILITY_FILE_COPIER_H

```

## 6.60 FileIOProcessorAsync.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_FILE_IO_PROCESSOR_ASYNC_H
00020 #define LIB_QUENTIER_UTILITY_FILE_IO_PROCESSOR_ASYNC_H
00021
00022 #include <quentier/types/ErrorString.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 #include <QByteArray>
00026 #include <QIODevice>
00027 #include <QObject>
00028 #include <QString>
00029 #include <QUuid>
00030
00031 namespace quentier {
00032
00033     QT_FORWARD_DECLARE_CLASS(FileIOProcessorAsyncPrivate)
00034
00035     class QUENTIER_EXPORT FileIOProcessorAsync : public QObject
00036     {
00037     public:
00038         Q_OBJECT
00039         explicit FileIOProcessorAsync(QObject * parent = nullptr);
00040
00041         void setIdleTimePeriod(qint32 seconds);
00042
00043     Q_SIGNALS:
00044         void readyForIO();
00045
00046         void writeFileRequestProcessed(
00047             bool success, ErrorString errorDescription, QUuid requestId);
00048
00049         void readFileRequestProcessed(
00050             bool success, ErrorString errorDescription, QByteArray data,

```

```

00091         QUuid requestId);
00092
00093 public Q_SLOTS:
00105     void onWriteFileRequest(
00106         QString absoluteFilePath, QByteArray data, QUuid requestId,
00107         bool append);
00108
00116     void onReadFileRequest(QString absoluteFilePath, QUuid requestId);
00117
00118 private:
00119     FileIOProcessorAsyncPrivate * const d_ptr;
00120     Q_DECLARE_PRIVATE(FileIOProcessorAsync)
00121 };
00122
00123 } // namespace quentier
00124
00125 #endif // LIB_QUEENTIER_UTILITY_FILE_IO_PROCESSOR_ASYNC_H

```

## 6.61 FileSystem.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUEENTIER_UTILITY_FILE_SYSTEM_H
00020 #define LIB_QUEENTIER_UTILITY_FILE_SYSTEM_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025
00026 namespace quentier {
00027
00028 QT_FORWARD_DECLARE_CLASS(ErrorString)
00029
00030
00044 const QString QUEENTIER_EXPORT relativePathFromAbsolutePath(
00045     const QString & absolutePath, const QString & relativePathRootFolderPath);
00046
00055 bool QUEENTIER_EXPORT removeFile(const QString & filePath);
00056
00066 bool QUEENTIER_EXPORT removeDir(const QString & dirPath);
00067
00079 QByteArray QUEENTIER_EXPORT
00080 readFileContents(const QString & filePath, ErrorString & errorDescription);
00081
00096 bool QUEENTIER_EXPORT renameFile(
00097     const QString & from, const QString & to, ErrorString & errorDescription);
00098
00099 } // namespace quentier
00100
00101 #endif // LIB_QUEENTIER_UTILITY_FILE_SYSTEM_H

```

## 6.62 FileSystemWatcher.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,

```

```

00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_FILE_SYSTEM_WATCHER_H
00020 #define LIB_QUENTIER_UTILITY_FILE_SYSTEM_WATCHER_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QObject>
00025 #include <QStringList>
00026
00027 #define FILE_SYSTEM_WATCHER_DEFAULT_REMOVAL_TIMEOUT_MSEC (500)
00028
00029 namespace quentier {
00030
00031 QT_FORWARD_DECLARE_CLASS(FileSystemWatcherPrivate)
00032
00033 class QUENTIER_EXPORT FileSystemWatcher : public QObject
00034 {
00035     Q_OBJECT
00036 public:
00037     explicit FileSystemWatcher(
00038         const int removalTimeoutMsec =
00039             FILE_SYSTEM_WATCHER_DEFAULT_REMOVAL_TIMEOUT_MSEC,
00040         QObject * parent = nullptr);
00041
00042     explicit FileSystemWatcher(
00043         const QStringList & paths,
00044         const int removalTimeoutMsec =
00045             FILE_SYSTEM_WATCHER_DEFAULT_REMOVAL_TIMEOUT_MSEC,
00046         QObject * parent = nullptr);
00047
00048     virtual ~FileSystemWatcher() override;
00049
00050     void addPath(const QString & path);
00051     void addPaths(const QStringList & paths);
00052
00053     QStringList directories() const;
00054     QStringList files() const;
00055
00056     void removePath(const QString & path);
00057     void removePaths(const QStringList & paths);
00058
00059 Q_SIGNALS:
00060     void directoryChanged(const QString & path);
00061     void directoryRemoved(const QString & path);
00062
00063     void fileChanged(const QString & path);
00064     void fileRemoved(const QString & path);
00065
00066 private:
00067     Q_DISABLE_COPY(FileSystemWatcher)
00068
00069 private:
00070     FileSystemWatcherPrivate * d_ptr;
00071     Q_DECLARE_PRIVATE(FileSystemWatcher)
00072 };
00073
00074 } // namespace quentier
00075
00076 #endif // LIB_QUENTIER_UTILITY_FILE_SYSTEM_WATCHER_H

```

## 6.63 IKeychainService.h

```

00001  /*
00002  * Copyright 2018-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *

```

```

00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_I_KEYCHAIN_SERVICE_H
00020 #define LIB_QUENTIER_UTILITY_I_KEYCHAIN_SERVICE_H
00021
00022 #include <quentier/types/ErrorMessage.h>
00023 #include <quentier/utility/ForwardDeclarations.h>
00024 #include <quentier/utility/Linkage.h>
00025
00026 #include <QObject>
00027 #include <QUuid>
00028
00029 #include <memory>
00030
00031 QT_FORWARD_DECLARE_CLASS(QDebug)
00032
00033 namespace quentier {
00034
00040 class QUENTIER_EXPORT IKeychainService : public QObject
00041 {
00042     Q_OBJECT
00043 protected:
00044     explicit IKeychainService(QObject * parent = nullptr);
00045
00046 public:
00047     virtual ~IKeychainService() {}
00048
00052     enum class ErrorCode
00053     {
00057         NoError,
00061         EntryNotFound,
00065         CouldNotDeleteEntry,
00069         AccessDeniedByUser,
00073         AccessDenied,
00077         NoBackendAvailable,
00081         NotImplemented,
00085         OtherError
00086     };
00087
00088     friend QTextStream & operator<<(
00089         QTextStream & strm, const ErrorCode errorCode);
00090
00091     friend QDebug & operator<<(QDebug & dbg, const ErrorCode errorCode);
00092
00093 public:
00106     virtual QUuid startWritePasswordJob(
00107         const QString & service, const QString & key,
00108         const QString & password) = 0;
00109
00121     virtual QUuid startReadPasswordJob(
00122         const QString & service, const QString & key) = 0;
00123
00135     virtual QUuid startDeletePasswordJob(
00136         const QString & service, const QString & key) = 0;
00137
00138 Q_SIGNALS:
00151     void writePasswordJobFinished(
00152         QUuid requestId, ErrorCode errorCode, ErrorMessage errorDescription);
00153
00167     void readPasswordJobFinished(
00168         QUuid requestId, ErrorCode errorCode, ErrorMessage errorDescription,
00169         QString password);
00170
00183     void deletePasswordJobFinished(
00184         QUuid requestId, ErrorCode errorCode, ErrorMessage errorDescription);
00185
00186 private:
00187     Q_DISABLE_COPY(IKeychainService);
00188 };
00189
00190 QUENTIER_EXPORT IKeychainServicePtr
00191 newQtKeychainService(QObject * parent = nullptr);
00192
00193 QUENTIER_EXPORT IKeychainServicePtr
00194 newObfuscatingKeychainService(QObject * parent = nullptr);
00195
00196 QUENTIER_EXPORT IKeychainServicePtr newCompositeKeychainService(
00197     QString name, IKeychainServicePtr primaryKeychain,
00198     IKeychainServicePtr secondaryKeychain, QObject * parent = nullptr);
00199
00200 QUENTIER_EXPORT IKeychainServicePtr newMigratingKeychainService(
00201     IKeychainServicePtr sourceKeychain, IKeychainServicePtr sinkKeychain,
00202     QObject * parent = nullptr);
00203
00204 } // namespace quentier

```

```

00205
00206 #endif // LIB_QUENTIER_UTILITY_I_KEYCHAIN_SERVICE_H

```

## 6.64 Initialize.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_INITIALIZE_H
00020 #define LIB_QUENTIER_UTILITY_INITIALIZE_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 namespace quentier {
00025
00030 void QUENTIER_EXPORT initializeLibquentier();
00031
00032 } // namespace quentier
00033
00034 #endif // LIB_QUENTIER_UTILITY_INITIALIZE_H

```

## 6.65 LimitedStack.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_LIMITED_STACK_H
00020 #define LIB_QUENTIER_UTILITY_LIMITED_STACK_H
00021
00022 #include <QStack>
00023
00024 namespace quentier {
00025
00032 template <class T>
00033 class LimitedStack : public QStack<T>
00034 {
00035 public:
00036     LimitedStack(void (*deleter)(T &) = nullptr) :
00037         m_limit(-1), m_deleter(deleter)
00038     {}
00039
00040     LimitedStack(const LimitedStack<T> & other) :
00041         QStack<T>(other), m_limit(other.m_limit), m_deleter(other.m_deleter)
00042     {}
00043
00044     LimitedStack(LimitedStack<T> && other) :
00045         QStack<T>(std::move(other)), m_limit(std::move(other.m_limit)),
00046         m_deleter(std::move(other.m_deleter))
00047     {}

```

```

00048
00049     LimitedStack<T> & operator=(const LimitedStack<T> & other)
00050     {
00051         if (this != &other) {
00052             QStack<T>::operator=(other);
00053             m_limit = other.m_limit;
00054             m_deleter = other.m_deleter;
00055         }
00056
00057         return *this;
00058     }
00059
00060     LimitedStack<T> & operator=(LimitedStack<T> && other)
00061     {
00062         if (this != &other) {
00063             QStack<T>::operator=(std::move(other));
00064             m_limit = std::move(other.m_limit);
00065             m_deleter = std::move(other.m_deleter);
00066         }
00067
00068         return *this;
00069     }
00070
00071     ~LimitedStack()
00072     {
00073         if (m_deleter) {
00074             while (!QStack<T>::isEmpty()) {
00075                 T t = QStack<T>::pop();
00076                 (*m_deleter)(t);
00077             }
00078         }
00079     }
00080
00081     void swap(const LimitedStack<T> & other)
00082     {
00083         int limit = other.m_limit;
00084         other.m_limit = m_limit;
00085         m_limit = limit;
00086
00087         void (*deleter)(T &) = other.m_deleter;
00088         other.m_deleter = m_deleter;
00089         m_deleter = deleter;
00090
00091         QStack<T>::swap(other);
00092     }
00093
00094     int limit() const
00095     {
00096         return m_limit;
00097     }
00098     void setLimit(const int limit)
00099     {
00100         m_limit = limit;
00101     }
00102
00103     void push(const T & t)
00104     {
00105         if ((m_limit > 0) && (QVector<T>::size() == m_limit - 1)) {
00106             if (m_deleter) {
00107                 (*m_deleter)(QVector<T>::begin());
00108             }
00109             Q_UNUSED(QVector<T>::erase(QVector<T>::begin()));
00110         }
00111
00112         QStack<T>::push(t);
00113     }
00114
00115 private:
00116     int m_limit;
00117     void (*m_deleter)(T &);
00118 };
00119
00120 } // namespace quentier
00121
00122 #endif // LIB_QUENTIER_UTILITY_LIMITED_STACK_H

```

## 6.66 Linkage.h

```

00001 /*
00002  * Copyright 2016-2019 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *

```

```

00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_LINKAGE_H
00020 #define LIB_QUENTIER_UTILITY_LINKAGE_H
00021
00022 #include <QtGlobal>
00023
00024 #if defined(BUILDING_QUENTIER_DLL)
00025 #define QUENTIER_EXPORT Q_DECL_EXPORT
00026 #else
00027 #define QUENTIER_EXPORT Q_DECL_IMPORT
00028 #endif
00029
00030 #endif // LIB_QUENTIER_UTILITY_LINKAGE_H

```

## 6.67 LRUCache.hpp

```

00001  /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_LRU_CACHE_HPP
00020 #define LIB_QUENTIER_UTILITY_LRU_CACHE_HPP
00021
00022 #include <QHash>
00023
00024 #include <cstdint>
00025 #include <list>
00026
00027 namespace quentier {
00028
00029 template <
00030     class Key, class Value,
00031     class Allocator = std::allocator<std::pair<Key, Value>>
00032 class LRUCache
00033 {
00034 public:
00035     LRUCache(const size_t maxSize = 100) :
00036         m_container(), m_currentSize(0), m_maxSize(maxSize), m_mapper()
00037     {}
00038
00039     using key_type = Key;
00040     using mapped_type = Value;
00041     using allocator_type = Allocator;
00042     using value_type = std::pair<key_type, mapped_type>;
00043     using container_type = std::list<value_type, allocator_type>;
00044     using size_type = typename container_type::size_type;
00045     using difference_type = typename container_type::difference_type;
00046     using iterator = typename container_type::iterator;
00047     using const_iterator = typename container_type::const_iterator;
00048     using reverse_iterator = std::reverse_iterator<iterator>;
00049     using const_reverse_iterator = std::reverse_iterator<const_iterator>;
00050
00051     using reference = value_type &;
00052     using const_reference = const value_type &;
00053     using pointer = typename std::allocator_traits<allocator_type>::pointer;
00054
00055     using const_pointer =

```

```

00056         typename std::allocator_traits<allocator_type>::const_pointer;
00057
00058     iterator begin()
00059     {
00060         return m_container.begin();
00061     }
00062     const_iterator begin() const
00063     {
00064         return m_container.begin();
00065     }
00066
00067     reverse_iterator rbegin()
00068     {
00069         return m_container.rbegin();
00070     }
00071     const_reverse_iterator rbegin() const
00072     {
00073         return m_container.rbegin();
00074     }
00075
00076     iterator end()
00077     {
00078         return m_container.end();
00079     }
00080     const_iterator end() const
00081     {
00082         return m_container.end();
00083     }
00084
00085     reverse_iterator rend()
00086     {
00087         return m_container.rend();
00088     }
00089     const_reverse_iterator rend() const
00090     {
00091         return m_container.rend();
00092     }
00093
00094     bool empty() const
00095     {
00096         return m_container.empty();
00097     }
00098     size_t size() const
00099     {
00100         return m_currentSize;
00101     }
00102     size_t max_size() const
00103     {
00104         return m_maxSize;
00105     }
00106
00107     void clear()
00108     {
00109         m_container.clear();
00110         m_mapper.clear();
00111         m_currentSize = 0;
00112     }
00113
00114     void put(const key_type & key, const mapped_type & value)
00115     {
00116         Q_UNUSED(remove(key))
00117
00118         m_container.push_front(value_type(key, value));
00119         m_mapper[key] = m_container.begin();
00120         ++m_currentSize;
00121
00122         fixupSize();
00123     }
00124
00125     const mapped_type * get(const key_type & key) const
00126     {
00127         auto mapperIt = m_mapper.find(key);
00128         if (mapperIt == m_mapper.end()) {
00129             return nullptr;
00130         }
00131
00132         auto it = mapperIt.value();
00133         if (it == m_container.end()) {
00134             return nullptr;
00135         }
00136
00137         m_container.splice(m_container.begin(), m_container, it);
00138         mapperIt.value() = m_container.begin();
00139         return &(mapperIt.value()->second);
00140     }
00141
00142     bool exists(const key_type & key)

```

```

00143     {
00144         auto mapperIt = m_mapper.find(key);
00145         if (mapperIt == m_mapper.end()) {
00146             return false;
00147         }
00148
00149         auto it = mapperIt.value();
00150         return (it != m_container.end());
00151     }
00152
00153     bool remove(const key_type & key)
00154     {
00155         auto mapperIt = m_mapper.find(key);
00156         if (mapperIt == m_mapper.end()) {
00157             return false;
00158         }
00159
00160         auto it = mapperIt.value();
00161         Q_UNUSED(m_container.erase(it))
00162         Q_UNUSED(m_mapper.erase(mapperIt))
00163
00164         if (m_currentSize != 0) {
00165             --m_currentSize;
00166         }
00167
00168         return true;
00169     }
00170
00171     void setMaxSize(const size_t maxSize)
00172     {
00173         if (maxSize >= m_maxSize) {
00174             m_maxSize = maxSize;
00175             return;
00176         }
00177
00178         size_t diff = m_maxSize - maxSize;
00179         for (size_t i = 0; (i < diff) && !m_container.empty(); ++i) {
00180             auto lastIt = m_container.end();
00181             --lastIt;
00182
00183             const key_type & lastElementKey = lastIt->first;
00184             Q_UNUSED(m_mapper.remove(lastElementKey))
00185             Q_UNUSED(m_container.erase(lastIt))
00186
00187             if (m_currentSize != 0) {
00188                 --m_currentSize;
00189             }
00190         }
00191     }
00192
00193 private:
00194     void fixupSize()
00195     {
00196         if (m_currentSize <= m_maxSize) {
00197             return;
00198         }
00199
00200         if (Q_UNLIKELY(m_container.empty())) {
00201             return;
00202         }
00203
00204         auto lastIt = m_container.end();
00205         --lastIt;
00206
00207         const key_type & lastElementKey = lastIt->first;
00208
00209         Q_UNUSED(m_mapper.remove(lastElementKey))
00210         Q_UNUSED(m_container.erase(lastIt))
00211
00212         if (m_currentSize != 0) {
00213             --m_currentSize;
00214         }
00215     }
00216
00217 private:
00218     mutable container_type m_container;
00219     size_t m_currentSize;
00220     size_t m_maxSize;
00221
00222     mutable QHash<Key, iterator> m_mapper;
00223 };
00224
00225 } // namespace quentier
00226
00227 #endif // LIB_QUENTIER_UTILITY_LRU_CACHE_HPP

```

## 6.68 MessageBox.h

```

00001 /*
00002  * Copyright 2017-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_MESSAGE_BOX_H
00020 #define LIB_QUENTIER_UTILITY_MESSAGE_BOX_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QMessageBox>
00025
00026 namespace quentier {
00027
00028 int QUENTIER_EXPORT genericMessageBox(
00029     QWidget * parent, const QString & title, const QString & briefText,
00030     const QString & detailedText = {},
00031     const QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
00032
00033 int QUENTIER_EXPORT informationMessageBox(
00034     QWidget * parent, const QString & title, const QString & briefText,
00035     const QString & detailedText = {},
00036     const QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
00037
00038 int QUENTIER_EXPORT warningMessageBox(
00039     QWidget * parent, const QString & title, const QString & briefText,
00040     const QString & detailedText = {},
00041     const QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
00042
00043 int QUENTIER_EXPORT criticalMessageBox(
00044     QWidget * parent, const QString & title, const QString & briefText,
00045     const QString & detailedText = {},
00046     const QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
00047
00048 int QUENTIER_EXPORT questionMessageBox(
00049     QWidget * parent, const QString & title, const QString & briefText,
00050     const QString & detailedText = {},
00051     const QMessageBox::StandardButtons standardButtons = QMessageBox::Ok |
00052         QMessageBox::Cancel);
00053
00054 void QUENTIER_EXPORT
00055     internalErrorMessageBox(QWidget * parent, QString detailedText = {});
00056
00057 } // namespace quentier
00058
00059 #endif // LIB_QUENTIER_UTILITY_MESSAGE_BOX_H

```

## 6.69 Printable.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_PRINTABLE_H

```

```

00020 #define LIB_QUENTIER_UTILITY_PRINTABLE_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QDebug>
00025 #include <QHash>
00026 #include <QSet>
00027 #include <QString>
00028 #include <QTextStream>
00029
00030 namespace quentier {
00031
00037 class QUENTIER_EXPORT Printable
00038 {
00039 public:
00040     virtual QTextStream & print(QTextStream & strm) const = 0;
00041
00042     virtual const QString toString() const;
00043
00044     friend QUENTIER_EXPORT QTextStream & operator«(
00045         QTextStream & strm, const Printable & printable);
00046
00047     friend QUENTIER_EXPORT QDebug & operator«(
00048         QDebug & debug, const Printable & printable);
00049
00050 protected:
00051     Printable();
00052     Printable(const Printable & other);
00053     Printable & operator=(const Printable & other);
00054     virtual ~Printable();
00055 };
00056
00057 } // namespace quentier
00058
00059 // printing operators for existing classes not inheriting from Printable
00060
00061 template <class T>
00062 const QString ToString(const T & object)
00063 {
00064     QString str;
00065     QTextStream strm(&str, QIODevice::WriteOnly);
00066     strm « object;
00067     return str;
00068 }
00069
00070 template <class TKey, class TValue>
00071 const QString ToString(const QHash<TKey, TValue> & object)
00072 {
00073     QString str;
00074     QTextStream strm(&str, QIODevice::WriteOnly);
00075     strm « QStringLiteral("QHash: \n");
00076
00077     typedef typename QHash<TKey, TValue>::const_iterator CIter;
00078     CIter hashEnd = object.end();
00079     for (CIter it = object.begin(); it != hashEnd; ++it) {
00080         strm « QStringLiteral("[") « it.key() « QStringLiteral("] = ")
00081             « it.value() « QStringLiteral("; \n");
00082     }
00083     return str;
00084 }
00085
00086 template <class T>
00087 const QString ToString(const QSet<T> & object)
00088 {
00089     QString str;
00090     QTextStream strm(&str, QIODevice::WriteOnly);
00091     strm « QStringLiteral("QSet: \n");
00092
00093     typedef typename QSet<T>::const_iterator CIter;
00094     CIter setEnd = object.end();
00095     for (CIter it = object.begin(); it != setEnd; ++it) {
00096         strm « QStringLiteral("[") « *it « QStringLiteral("]; \n");
00097     }
00098     return str;
00099 }
00100
00101 #define QUENTIER_DECLARE_PRINTABLE(type, ...)
00102     QUENTIER_EXPORT QTextStream & operator«(
00103         QTextStream & strm, const type & obj);
00104     inline QDebug & operator«(QDebug & debug, const type & obj)
00105     {
00106         debug « ToString<type, ##__VA_ARGS__>(obj);
00107         return debug;
00108     }
00109 // QUENTIER_DECLARE_PRINTABLE
00110
00111 #endif // LIB_QUENTIER_UTILITY_PRINTABLE_H

```

## 6.70 QuentierApplication.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_QUENTIER_APPLICATION_H
00020 #define LIB_QUENTIER_UTILITY_QUENTIER_APPLICATION_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QApplication>
00025
00026 namespace quentier {
00027
00028 class QUENTIER_EXPORT QuentierApplication : public QApplication
00029 {
00030     Q_OBJECT
00031 public:
00032     QuentierApplication(int & argc, char * argv[]);
00033     virtual ~QuentierApplication() override;
00034
00035     virtual bool notify(QObject * pObject, QEvent * pEvent) override;
00036     virtual bool event(QEvent * pEvent) override;
00037 };
00038
00039 } // namespace quentier
00040
00041 #endif // LIB_QUENTIER_UTILITY_QUENTIER_APPLICATION_H

```

## 6.71 QuentierCheckPtr.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_QUENTIER_CHECK_PTR_H
00020 #define LIB_QUENTIER_UTILITY_QUENTIER_CHECK_PTR_H
00021
00022 #include <quentier/exception/NullPtrException.h>
00023 #include <quentier/logging/QuentierLogger.h>
00024
00025 #ifndef QUENTIER_CHECK_PTR
00026 #define QUENTIER_CHECK_PTR(component, pointer, ...)
00027 {
00028     if (Q_UNLIKELY(!pointer)) {
00029         using quentier::NullPtrException;
00030         QString quentier_null_ptr_error(
00031             QT_TRANSLATE_NOOP("", "Detected unintended null pointer"));
00032         quentier_null_ptr_error.details() = QStringLiteral(__FILE__);
00033         quentier_null_ptr_error.details() += QStringLiteral(" ");
00034         quentier_null_ptr_error.details() += QString::number(__LINE__);
00035         quentier_null_ptr_error.details() += QStringLiteral(" ");
00036         quentier_null_ptr_error.details() +=
00037             QString::fromUtf8(" " #__VA_ARGS__ " ");
00038     }
00039 }
00040
00041 #endif

```

```

00038         QNERROR(component, quentier_null_ptr_error);
00039         throw NullPtrException(quentier_null_ptr_error);
00040     }
00041 }
00042 // QUENTIER_CHECK_PTR
00043 #endif
00044
00045 #endif // LIB_QUENTIER_UTILITY_QUENTIER_CHECK_PTR_H

```

## 6.72 QuentierUndoCommand.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_QUENTIER_UNDO_COMMAND_H
00020 #define LIB_QUENTIER_UTILITY_QUENTIER_UNDO_COMMAND_H
00021
00022 #include <quentier/types/ErrorString.h>
00023
00024 #include <QObject>
00025 #include <QUndoCommand>
00026
00027 namespace quentier {
00028
00029 class QuentierUndoCommand : public QObject, public QUndoCommand
00030 {
00031     Q_OBJECT
00032 public:
00033     QuentierUndoCommand(QUndoCommand * parent = nullptr);
00034     QuentierUndoCommand(const QString & text, QUndoCommand * parent = nullptr);
00035     virtual ~QuentierUndoCommand() override;
00036
00037     virtual void undo() override final;
00038     virtual void redo() override final;
00039
00040     bool onceUndoExecuted() const
00041     {
00042         return m_onceUndoExecuted;
00043     }
00044
00045 Q_SIGNALS:
00046     void notifyError(ErrorString error);
00047
00048 protected:
00049     virtual void undoImpl() = 0;
00050     virtual void redoImpl() = 0;
00051
00052 private:
00053     bool m_onceUndoExecuted;
00054 };
00055
00056 } // namespace quentier
00057
00058 #endif // LIB_QUENTIER_UTILITY_QUENTIER_UNDO_COMMAND_H

```

## 6.73 ShortcutManager.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by

```

```

00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_SHORTCUT_MANAGER_H
00020 #define LIB_QUENTIER_UTILITY_SHORTCUT_MANAGER_H
00021
00022 #include <quentier/types/Account.h>
00023 #include <quentier/utility/Linkage.h>
00024
00025 #include <QKeySequence>
00026 #include <QObject>
00027
00028 namespace quentier {
00029
00030 QT_FORWARD_DECLARE_CLASS(ShortcutManagerPrivate)
00031
00032 class QUENTIER_EXPORT ShortcutManager : public QObject
00033 {
00034     Q_OBJECT
00035 public:
00036     explicit ShortcutManager(QObject * parent = nullptr);
00037
00038     enum QuentierShortcutKey
00039     {
00040         NewNote = 5000,
00041         NewTag,
00042         NewNotebook,
00043         NewSavedSearch,
00044         AddAttachment,
00045         SaveAttachment,
00046         OpenAttachment,
00047         CopyAttachment,
00048         CutAttachment,
00049         RemoveAttachment,
00050         RenameAttachment,
00051         AddAccount,
00052         ExitAccount,
00053         SwitchAccount,
00054         AccountInfo,
00055         NoteSearch,
00056         NewNoteSearch,
00057         ShowNotes,
00058         ShowNotebooks,
00059         ShowTags,
00060         ShowSavedSearches,
00061         ShowDeletedNotes,
00062         ShowStatusBar,
00063         ShowToolBar,
00064         PasteUnformatted,
00065         Font,
00066         UpperIndex,
00067         LowerIndex,
00068         AlignLeft,
00069         AlignCenter,
00070         AlignRight,
00071         AlignFull,
00072         IncreaseIndentation,
00073         DecreaseIndentation,
00074         IncreaseFontSize,
00075         DecreaseFontSize,
00076         InsertNumberedList,
00077         InsertBulletedList,
00078         Strikethrough,
00079         Highlight,
00080         InsertTable,
00081         InsertRow,
00082         InsertColumn,
00083         RemoveRow,
00084         RemoveColumn,
00085         InsertHorizontalLine,
00086         InsertToDoTag,
00087         EditHyperlink,
00088         CopyHyperlink,
00089         RemoveHyperlink,
00090         Encrypt,
00091         Decrypt,
00092         DecryptPermanently,
00093         BackupLocalStorage,
00094         RestoreLocalStorage,

```

```

00095         UpgradeLocalStorage,
00096         LocalStorageStatus,
00097         SpellCheck,
00098         SpellCheckIgnoreWord,
00099         SpellCheckAddWordToUserDictionary,
00100         SaveImage,
00101         AnnotateImage,
00102         ImageRotateClockwise,
00103         ImageRotateCounterClockwise,
00104         Synchronize,
00105         FullSync,
00106         ImportFolders,
00107         Preferences,
00108         ReleaseNotes,
00109         ViewLogs,
00110         About,
00111         UnknownKey = 100000
00112     };
00113
00119     QKeySequence shortcut (
00120         const int key, const Account & account,
00121         const QString & context = {}) const;
00122
00128     QKeySequence shortcut (
00129         const QString & nonStandardKey, const Account & account,
00130         const QString & context = {}) const;
00131
00136     QKeySequence defaultShortcut (
00137         const int key, const Account & account,
00138         const QString & context = {}) const;
00139
00144     QKeySequence defaultShortcut (
00145         const QString & nonStandardKey, const Account & account,
00146         const QString & context = {}) const;
00147
00152     QKeySequence userShortcut (
00153         const int key, const Account & account,
00154         const QString & context = {}) const;
00155
00160     QKeySequence userShortcut (
00161         const QString & nonStandardKey, const Account & account,
00162         const QString & context = {}) const;
00163
00164     Q_SIGNALS:
00165         void shortcutChanged(
00166             int key, QKeySequence shortcut, const Account & account,
00167             QString context);
00168
00169         void nonStandardShortcutChanged(
00170             QString nonStandardKey, QKeySequence shortcut, const Account & account,
00171             QString context);
00172
00173     public Q_SLOTS:
00174         void setUserShortcut (
00175             int key, QKeySequence shortcut, const Account & account,
00176             QString context = {});
00177
00178         void setNonStandardUserShortcut (
00179             QString nonStandardKey, QKeySequence shortcut, const Account & account,
00180             QString context = {});
00181
00182         void setDefaultShortcut (
00183             int key, QKeySequence shortcut, const Account & account,
00184             QString context = {});
00185
00186         void setNonStandardDefaultShortcut (
00187             QString nonStandardKey, QKeySequence shortcut, const Account & account,
00188             QString context = {});
00189
00190     private:
00191         ShortcutManagerPrivate * const d_ptr;
00192         Q_DECLARE_PRIVATE(ShortcutManager)
00193     };
00194
00195 } // namespace quentier
00196
00197 #endif // LIB_QUENTIER_UTILITY_SHORTCUT_MANAGER_H

```

## 6.74 Size.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *

```

```

00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_SIZE_H
00020 #define LIB_QUENTIER_UTILITY_SIZE_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025
00026 namespace quentier {
00027
00037 const QString QUENTIER_EXPORT humanReadableSize(const quint64 bytes);
00038
00039 } // namespace quentier
00040
00041 #endif // LIB_QUENTIER_UTILITY_SIZE_H

```

## 6.75 StandardPaths.h

```

00001 /*
00002  * Copyright 2017-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_STANDARD_PATHS_H
00020 #define LIB_QUENTIER_UTILITY_STANDARD_PATHS_H
00021
00022 #include <quentier/types/Account.h>
00023 #include <quentier/utility/Linkage.h>
00024
00029 #define LIBQUENTIER_PERSISTENCE_STORAGE_PATH \
00030     "LIBQUENTIER_PERSISTENCE_STORAGE_PATH"
00031
00032 namespace quentier {
00033
00042 const QString QUENTIER_EXPORT
00043 applicationPersistentStoragePath(bool * pNonStandardLocation = nullptr);
00044
00054 const QString QUENTIER_EXPORT
00055 accountPersistentStoragePath(const Account & account);
00056
00061 const QString QUENTIER_EXPORT applicationTemporaryStoragePath();
00062
00068 const QString QUENTIER_EXPORT homePath();
00069
00073 const QString QUENTIER_EXPORT documentsPath();
00074
00075 } // namespace quentier
00076
00077 #endif // LIB_QUENTIER_UTILITY_STANDARD_PATHS_H

```

## 6.76 StringUtils.h

```

00001 /*

```

```

00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_STRING_UTILS_H
00020 #define LIB_QUENTIER_UTILITY_STRING_UTILS_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QSet>
00025 #include <QString>
00026 #include <QVector>
00027
00028 namespace quentier {
00029
00030 QT_FORWARD_DECLARE_CLASS(StringUtilsPrivate)
00031
00032 class QUINTIER_EXPORT StringUtils
00033 {
00034 public:
00035     StringUtils();
00036     virtual ~StringUtils();
00037
00038     void removePunctuation(
00039         QString & str, const QVector<QChar> & charactersToPreserve = {}) const;
00040
00041     void removeDiacritics(QString & str) const;
00042     void removeNewlines(QString & str) const;
00043
00044     struct StringFilterPredicate
00045     {
00046         StringFilterPredicate(QSet<QString> & filteredStrings) :
00047             m_filteredStrings(filteredStrings)
00048         {}
00049
00050         bool operator()(const QString & str) const
00051         {
00052             return m_filteredStrings.contains(str);
00053         }
00054
00055         QSet<QString> & m_filteredStrings;
00056     };
00057
00058 private:
00059     StringUtilsPrivate * const d_ptr;
00060     Q_DECLARE_PRIVATE(StringUtils);
00061 };
00062
00063 } // namespace quentier
00064
00065 #endif // LIB_QUENTIER_UTILITY_STRING_UTILS_H

```

## 6.77 SuppressWarnings.h

```

00001 /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.

```

```

00017  */
00018
00019 #ifndef LIB_QUEANTIER_UTILITY_SUPPRESS_WARNINGS_H
00020 #define LIB_QUEANTIER_UTILITY_SUPPRESS_WARNINGS_H
00021
00023 // Common macros
00025
00026 #define STRINGIFY(a) #a
00027
00028 // Define empty macros doing nothing for supported compilers, they would be used
00029 // as fallback when any of these compilers are not actually used
00030
00031 #define SAVE_WARNINGS
00032
00033 #define CLANG_SUPPRESS_WARNING(warning)
00034 #define GCC_SUPPRESS_WARNING(warning)
00035 #define MSVC_SUPPRESS_WARNING(warning)
00036
00037 #define RESTORE_WARNINGS
00038
00040 // Clang implementation
00042
00043 #if defined(__clang__)
00044
00045 #undef CLANG_SUPPRESS_WARNING
00046
00047 #define CLANG_SUPPRESS_WARNING(warning) \
00048     _Pragma( \
00049         STRINGIFY(clang diagnostic ignored #warning)) // CLANG_IGNORE_WARNING
00050
00051 #undef SAVE_WARNINGS
00052
00053 #define SAVE_WARNINGS _Pragma("clang diagnostic push") // SAVE_WARNINGS
00054
00055 #undef RESTORE_WARNINGS
00056
00057 #define RESTORE_WARNINGS _Pragma("clang diagnostic pop") // RESTORE_WARNINGS
00058
00059 #endif // clang
00060
00062 // GCC implementation
00064
00065 // Clang can mimic gcc so need to ensure it's indeed gcc
00066 #if defined(__GNUC__) && !defined(__clang__)
00067
00068 #undef GCC_SUPPRESS_WARNING
00069
00070 #define GCC_SUPPRESS_WARNING(warning) \
00071     _Pragma(STRINGIFY(GCC diagnostic ignored #warning)) // GCC_SUPPRESS_WARNING
00072
00073 #undef SAVE_WARNINGS
00074
00075 #define SAVE_WARNINGS _Pragma("GCC diagnostic push") // SAVE_WARNINGS
00076
00077 #undef RESTORE_WARNINGS
00078
00079 #define RESTORE_WARNINGS _Pragma("GCC diagnostic pop") // RESTORE_WARNINGS
00080
00081 #endif // GCC
00082
00084 // MSVC implementation
00086
00087 #if defined(_MSC_VER)
00088
00089 #undef MSVC_SUPPRESS_WARNING
00090
00091 #define MSVC_SUPPRESS_WARNING(number) \
00092     __pragma(warning(disable : number)) // MSVC_SUPPRESS_WARNING
00093
00094 #undef SAVE_WARNINGS
00095
00096 #define SAVE_WARNINGS __pragma(warning(push)) // SAVE_WARNINGS
00097
00098 #undef RESTORE_WARNINGS
00099
00100 #define RESTORE_WARNINGS __pragma(warning(pop)) // RESTORE_WARNINGS
00101
00102 #endif // MSVC
00103
00104 #endif // LIB_QUEANTIER_UTILITY_SUPPRESS_WARNINGS_H

```

## 6.78 SysInfo.h

```
00001 /*
```

```

00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_SYS_INFO_H
00020 #define LIB_QUENTIER_UTILITY_SYS_INFO_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025
00026 namespace quentier {
00027
00028 QT_FORWARD_DECLARE_CLASS(SysInfoPrivate)
00029
00030 class QUENTIER_EXPORT SysInfo
00031 {
00032 public:
00033     SysInfo();
00034     ~SysInfo();
00035
00036     qint64 pageSize();
00037     qint64 totalMemory();
00038     qint64 freeMemory();
00039
00040     QString stackTrace();
00041
00042     QString platformName();
00043
00044 private:
00045     Q_DISABLE_COPY(SysInfo)
00046
00047 private:
00048     SysInfoPrivate * const d_ptr;
00049     Q_DECLARE_PRIVATE(SysInfo)
00050 };
00051
00052 } // namespace quentier
00053
00054 #endif // LIB_QUENTIER_UTILITY_SYS_INFO_H

```

## 6.79 System.h

```

00001  /*
00002  * Copyright 2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUENTIER_UTILITY_SYSTEM_H
00020 #define LIB_QUENTIER_UTILITY_SYSTEM_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>
00025 #include <QUrl>
00026
00027 namespace quentier {

```

```

00028
00032 QString QUINTIER_EXPORT getCurrentUserName();
00033
00037 QString QUINTIER_EXPORT getCurrentUserFullName();
00038
00042 void QUINTIER_EXPORT openUrl(const QUrl & url);
00043
00044 } // namespace quentier
00045
00046 #endif // LIB_QUINTIER_UTILITY_SYSTEM_H

```

## 6.80 TagSortByParentChildRelations.h

```

00001 /*
00002  * Copyright 2017-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUINTIER_UTILITY_TAG_SORT_BY_PARENT_CHILD_RELATIONS_H
00020 #define LIB_QUINTIER_UTILITY_TAG_SORT_BY_PARENT_CHILD_RELATIONS_H
00021
00022 #include <quentier/types/ErrorMessage.h>
00023 #include <quentier/types/Tag.h>
00024
00025 #include <qt5qevercloud/QEverCloud.h>
00026
00027 #include <QList>
00028
00029 namespace quentier {
00030
00042 bool QUINTIER_EXPORT sortTagsByParentChildRelations(
00043     QList<qevercloud::Tag> & tagList, ErrorMessage & errorMessage);
00044
00057 bool QUINTIER_EXPORT sortTagsByParentChildRelations(
00058     QList<Tag> & tagList, ErrorMessage errorMessage);
00059
00060 } // namespace quentier
00061
00062 #endif // LIB_QUINTIER_UTILITY_TAG_SORT_BY_PARENT_CHILD_RELATIONS_H

```

## 6.81 UidGenerator.h

```

00001 /*
00002  * Copyright 2016-2020 Dmitry Ivanov
00003  *
00004  * This file is part of libquentier
00005  *
00006  * libquentier is free software; you can redistribute it and/or modify
00007  * it under the terms of the GNU Lesser General Public License as published by
00008  * the Free Software Foundation, version 3 of the License.
00009  *
00010  * libquentier is distributed in the hope that it will be useful,
00011  * but WITHOUT ANY WARRANTY; without even the implied warranty of
00012  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
00013  * GNU Lesser General Public License for more details.
00014  *
00015  * You should have received a copy of the GNU Lesser General Public License
00016  * along with libquentier. If not, see <http://www.gnu.org/licenses/>.
00017  */
00018
00019 #ifndef LIB_QUINTIER_UTILITY_UID_GENERATOR_H
00020 #define LIB_QUINTIER_UTILITY_UID_GENERATOR_H
00021
00022 #include <quentier/utility/Linkage.h>
00023
00024 #include <QString>

```

```
00025 #include <QUuid>
00026
00027 namespace quantier {
00028
00029 class QUENTIER_EXPORT UidGenerator
00030 {
00031 public:
00032     static QString Generate();
00033     static QString UidToString(const QUuid & uid);
00034 };
00035
00036 } // namespace quantier
00037
00038 #endif // LIB_QUENTIER_UTILITY_UID_GENERATOR_H
```



# Index

- ~ApplicationSettings
  - quentier::ApplicationSettings, [17](#)
- AccessDenied
  - quentier::IKeychainService, [59](#)
- AccessDeniedByUser
  - quentier::IKeychainService, [59](#)
- Account.h, [277](#)
- accountHighUsn
  - quentier::LocalStorageManager, [111](#)
- active
  - quentier::SynchronizationManager, [212](#)
- addEnResource
  - quentier::LocalStorageManager, [112](#)
- addLinkedNotebook
  - quentier::LocalStorageManager, [112](#)
- addNote
  - quentier::LocalStorageManager, [112](#)
- addNotebook
  - quentier::LocalStorageManager, [113](#)
- addSavedSearch
  - quentier::LocalStorageManager, [113](#)
- addTag
  - quentier::LocalStorageManager, [114](#)
- addUser
  - quentier::LocalStorageManager, [114](#)
- ApplicationSettings
  - quentier::ApplicationSettings, [16](#), [17](#)
- ApplicationSettings.h, [303](#)
- ApplicationSettingsInitializationException.h, [232](#)
- apply
  - quentier::ILocalStoragePatch, [67](#)
- authenticate
  - quentier::SynchronizationManager, [212](#)
- authenticateCurrentAccount
  - quentier::SynchronizationManager, [212](#)
- authenticateToSharedNotebook
  - quentier::INoteStore, [75](#)
- authenticationFinished
  - quentier::SynchronizationManager, [212](#)
- AuthenticationManager.h, [269](#)
- authenticationRevoked
  - quentier::SynchronizationManager, [213](#)
- backend
  - quentier::NoteEditor, [168](#)
- backupLocalStorage
  - quentier::ILocalStoragePatch, [68](#)
- backupProgress
  - quentier::ILocalStoragePatch, [68](#)
- beginGroup
  - quentier::ApplicationSettings, [17](#), [18](#)
- beginReadArray
  - quentier::ApplicationSettings, [18](#)
- beginWriteArray
  - quentier::ApplicationSettings, [19](#)
- checkLinkedNotebooks
  - quentier::DefaultLocalStorageCacheExpiryChecker, [37](#)
  - quentier::ILocalStorageCacheExpiryChecker, [64](#)
- checkNotebooks
  - quentier::DefaultLocalStorageCacheExpiryChecker, [37](#)
  - quentier::ILocalStorageCacheExpiryChecker, [64](#)
- checkNotes
  - quentier::DefaultLocalStorageCacheExpiryChecker, [38](#)
  - quentier::ILocalStorageCacheExpiryChecker, [64](#)
- checkParameters
  - quentier::LinkedNotebook, [98](#)
  - quentier::Note, [157](#)
  - quentier::Notebook, [162](#)
  - quentier::Resource, [188](#)
  - quentier::SavedSearch, [196](#)
  - quentier::Tag, [222](#)
- checkResources
  - quentier::DefaultLocalStorageCacheExpiryChecker, [38](#)
  - quentier::ILocalStorageCacheExpiryChecker, [64](#)
- Checks.h, [304](#)
- checkSavedSearches
  - quentier::DefaultLocalStorageCacheExpiryChecker, [38](#)
  - quentier::ILocalStorageCacheExpiryChecker, [64](#)
- checkTags
  - quentier::DefaultLocalStorageCacheExpiryChecker, [38](#)
  - quentier::ILocalStorageCacheExpiryChecker, [65](#)
- checkVersion
  - quentier::IUserStore, [93](#)
- cleanupExternalHtml
  - quentier::ENMLConverter, [43](#)
- clear
  - quentier::LinkedNotebook, [98](#)
  - quentier::Note, [157](#)
  - quentier::Notebook, [162](#)
  - quentier::NoteEditor, [168](#)
  - quentier::Resource, [188](#)
  - quentier::SavedSearch, [196](#)

- quentier::Tag, 222
- ClearDatabase
  - quentier::LocalStorageManager, 110
- clone
  - quentier::DefaultLocalStorageCacheExpiryChecker, 39
  - quentier::ILocalStorageCacheExpiryChecker, 65
- Compat.h, 305
- contains
  - quentier::ApplicationSettings, 19, 20
- convertToNote
  - quentier::NoteEditor, 168
- CouldNotDeleteEntry
  - quentier::IKeychainService, 59
- createNote
  - quentier::INoteStore, 75
- createNotebook
  - quentier::INoteStore, 76
- createSavedSearch
  - quentier::INoteStore, 76
- createTag
  - quentier::INoteStore, 77
- currentNoteLocalUid
  - quentier::NoteEditor, 168
- DatabaseLockedException.h, 233
- DatabaseLockFailedException.h, 233
- DatabaseOpeningException.h, 234
- DatabaseRequestException.h, 234
- DateTime.h, 305
- DecryptedTextManager.h, 229
- defaultFont
  - quentier::NoteEditor, 168
- DefaultLocalStorageCacheExpiryChecker.h, 239
- defaultPalette
  - quentier::NoteEditor, 168
- defaultShortcut
  - quentier::ShortcutManager, 204
- deletePasswordJobFinished
  - quentier::IKeychainService, 60
- deleteUser
  - quentier::LocalStorageManager, 114
- detectedConflictDuringLocalChangesSending
  - quentier::SynchronizationManager, 213
- displayName
  - quentier::Account, 13
- downloadNoteThumbnailsOption
  - quentier::SynchronizationManager, 213
- EmptyDataElementException.h, 235
- EncryptionManager.h, 306
- ENMLConverter.h, 230
- enResourceCount
  - quentier::LocalStorageManager, 115
- EntryNotFound
  - quentier::IKeychainService, 59
- ErrorCode
  - quentier::IKeychainService, 59
- ErrorString.h, 279
- EventLoopWithExitStatus.h, 307
- evernoteAccountType
  - quentier::Account, 13
- evernoteHost
  - quentier::Account, 13
- exceptionDisplayName
  - quentier::ApplicationSettingsInitializationException, 25
  - quentier::DatabaseLockedException, 29
  - quentier::DatabaseLockFailedException, 31
  - quentier::DatabaseOpeningException, 32
  - quentier::DatabaseRequestException, 34
  - quentier::EmptyDataElementException, 41
  - quentier::LocalStorageCacheManagerException, 103
  - quentier::LoggerInitializationException, 152
  - quentier::NoteEditorInitializationException, 175
  - quentier::NoteEditorPluginInitializationException, 176
  - quentier::NullPtrException, 181
- exportNotesToEnex
  - quentier::ENMLConverter, 44
- expungeEnResource
  - quentier::LocalStorageManager, 115
- expungeLinkedNotebook
  - quentier::LocalStorageManager, 116
- expungeNote
  - quentier::LocalStorageManager, 116
- expungeNotebook
  - quentier::LocalStorageManager, 116
- expungeNotelessTagsFromLinkedNotebooks
  - quentier::LocalStorageManager, 117
- expungeSavedSearch
  - quentier::LocalStorageManager, 117
- expungeTag
  - quentier::LocalStorageManager, 118
- expungeUser
  - quentier::LocalStorageManager, 118
- failed
  - quentier::SynchronizationManager, 213
- FileCopier.h, 308
- FileIOProcessorAsync.h, 309
- FileSystem.h, 310
- FileSystemWatcher.h, 310
- findDefaultNotebook
  - quentier::LocalStorageManager, 118
- findDefaultOrLastUsedNotebook
  - quentier::LocalStorageManager, 119
- findEnResource
  - quentier::LocalStorageManager, 119
- findLastUsedNotebook
  - quentier::LocalStorageManager, 120
- findLinkedNotebook
  - quentier::LocalStorageManager, 120
- findNote
  - quentier::LocalStorageManager, 120
- findNotebook
  - quentier::LocalStorageManager, 121

- findNoteLocalUidsWithSearchQuery
  - quentier::LocalStorageManager, 121
- findNotesWithSearchQuery
  - quentier::LocalStorageManager, 122
- findSavedSearch
  - quentier::LocalStorageManager, 122
- findTag
  - quentier::LocalStorageManager, 123
- findUser
  - quentier::LocalStorageManager, 123
- finished
  - quentier::SynchronizationManager, 213
- ForwardDeclarations.h, 270
- fromVersion
  - quentier::ILocalStoragePatch, 68
- getAccountLimits
  - quentier::UserStore, 93
- getLinkedNotebookSyncChunk
  - quentier::INoteStore, 77
- getLinkedNotebookSyncState
  - quentier::INoteStore, 79
- getNote
  - quentier::INoteStore, 79
- getNoteAsync
  - quentier::INoteStore, 80
- GetNoteOption
  - quentier::LocalStorageManager, 109
- getResource
  - quentier::INoteStore, 81
- getResourceAsync
  - quentier::INoteStore, 82
- GetResourceOption
  - quentier::LocalStorageManager, 110
- getSyncChunk
  - quentier::INoteStore, 82
- getSyncState
  - quentier::INoteStore, 83
- getUser
  - quentier::IUserStore, 94
- guid
  - quentier::LinkedNotebook, 98
  - quentier::Note, 157
  - quentier::Notebook, 162
  - quentier::Resource, 188
  - quentier::SavedSearch, 196
  - quentier::Tag, 223
- hasGuid
  - quentier::LinkedNotebook, 98
  - quentier::Note, 157
  - quentier::Notebook, 163
  - quentier::Resource, 188
  - quentier::SavedSearch, 196
  - quentier::Tag, 223
- hasUpdateSequenceNumber
  - quentier::LinkedNotebook, 98
  - quentier::Note, 157
  - quentier::Notebook, 163
- quentier::Resource, 188
- quentier::SavedSearch, 196
- quentier::Tag, 223
- highestSupportedLocalStorageVersion
  - quentier::LocalStorageManager, 123
- HTMLCleaner.h, 232
- htmlToQTextDocument
  - quentier::ENMLConverter, 44
- IAuthenticationManager.h, 271
- id
  - quentier::Account, 13
- idleTime
  - quentier::NoteEditor, 168
- IFavoritableDataElement.h, 279
- IKeychainService.h, 311
- ILocalStorageCacheExpiryChecker.h, 239
- ILocalStorageDataElement.h, 280
- ILocalStoragePatch.h, 240
- importEnex
  - quentier::ENMLConverter, 45
- inAppNoteLinkPasteRequested
  - quentier::NoteEditor, 169
- IncludeMilliseconds
  - quentier::DateTimePrint, 35
- IncludeNumericTimestamp
  - quentier::DateTimePrint, 35
- IncludeTimezone
  - quentier::DateTimePrint, 35
- initialize
  - quentier::NoteEditor, 169
- Initialize.h, 313
- INoteEditorBackend.h, 262
- INoteStore.h, 272
- INoteStoreDataElement.h, 281
- IQuentierException.h, 235
- isEditorPageModified
  - quentier::NoteEditor, 169
- isEmpty
  - quentier::Account, 13
- isLocalStorageVersionTooHigh
  - quentier::LocalStorageManager, 124
- isModified
  - quentier::NoteEditor, 170
- isNoteLoaded
  - quentier::NoteEditor, 170
- ISyncStateStorage.h, 274
- IUserStore.h, 274
- libquentier, 1
- LimitedStack.h, 313
- Linkage.h, 314
- LinkedNotebook.h, 283
- linkedNotebookCount
  - quentier::LocalStorageManager, 124
- linkedNotebooksNotesDownloadProgress
  - quentier::SynchronizationManager, 214
- linkedNotebooksResourcesDownloadProgress
  - quentier::SynchronizationManager, 214

- linkedNotebooksSyncChunksDownloaded
  - quentier::SynchronizationManager, [214](#)
- linkedNotebookSyncChunksDownloadProgress
  - quentier::SynchronizationManager, [215](#)
- listAllLinkedNotebooks
  - quentier::LocalStorageManager, [124](#)
- listAllNotebooks
  - quentier::LocalStorageManager, [125](#)
- listAllSavedSearches
  - quentier::LocalStorageManager, [126](#)
- listAllSharedNotebooks
  - quentier::LocalStorageManager, [126](#)
- listAllTags
  - quentier::LocalStorageManager, [126](#)
- listAllTagsPerNote
  - quentier::LocalStorageManager, [127](#)
- listLinkedNotebooks
  - quentier::LocalStorageManager, [128](#)
- listNotebooks
  - quentier::LocalStorageManager, [128](#)
- listNotes
  - quentier::LocalStorageManager, [129](#)
- listNotesByLocalUids
  - quentier::LocalStorageManager, [130](#)
- listNotesPerNotebook
  - quentier::LocalStorageManager, [130](#)
- listNotesPerNotebooksAndTags
  - quentier::LocalStorageManager, [131](#)
- listNotesPerTag
  - quentier::LocalStorageManager, [132](#)
- ListObjectsOption
  - quentier::LocalStorageManager, [110](#)
- Lists.h, [241](#)
- listSavedSearches
  - quentier::LocalStorageManager, [132](#)
- listSharedNotebooksPerNotebookGuid
  - quentier::LocalStorageManager, [133](#)
- listTags
  - quentier::LocalStorageManager, [133](#)
- listTagsWithNoteLocalUids
  - quentier::LocalStorageManager, [134](#)
- LocalStorageCacheManager.h, [242](#)
- LocalStorageCacheManagerException.h, [236](#)
- LocalStorageManager
  - quentier::LocalStorageManager, [111](#)
- LocalStorageManager.h, [243](#)
- LocalStorageManagerAsync.h, [250](#)
- localStorageRequiresUpgrade
  - quentier::LocalStorageManager, [135](#)
- localStorageVersion
  - quentier::LocalStorageManager, [135](#)
- LoggerInitializationException.h, [237](#)
- LRUCache.hpp, [315](#)
- MessageBox.h, [318](#)
- name
  - quentier::Account, [14](#)
- NoBackendAvailable
  - quentier::IKeychainService, [59](#)
- NoError
  - quentier::IKeychainService, [59](#)
- Note.h, [284](#)
- Notebook.h, [287](#)
- notebookCount
  - quentier::LocalStorageManager, [135](#)
- notebookModifier
  - quentier::NoteSearchQuery, [179](#)
- noteCount
  - quentier::LocalStorageManager, [136](#)
- noteCountPerNotebook
  - quentier::LocalStorageManager, [136](#)
- noteCountPerNotebooksAndTags
  - quentier::LocalStorageManager, [137](#)
- noteCountPerTag
  - quentier::LocalStorageManager, [137](#)
- noteCountsPerAllTags
  - quentier::LocalStorageManager, [137](#)
- NoteEditor.h, [265](#)
- NoteEditorInitializationException.h, [237](#)
- NoteEditorPluginInitializationException.h, [238](#)
- notesDownloadProgress
  - quentier::SynchronizationManager, [215](#)
- NoteSearchQuery.h, [259](#)
- noteStoreUrl
  - quentier::INoteStore, [83](#)
- notifySyncStateUpdated
  - quentier::ISyncStateStorage, [92](#)
- NotImplemented
  - quentier::IKeychainService, [59](#)
- NullPtrException.h, [238](#)
- onReadFileRequest
  - quentier::FileIOProcessorAsync, [51](#)
- onWriteFileRequest
  - quentier::FileIOProcessorAsync, [51](#)
- Option
  - quentier::DateTimePrint, [35](#)
- OtherError
  - quentier::IKeychainService, [59](#)
- OverrideLock
  - quentier::LocalStorageManager, [110](#)
- patchLongDescription
  - quentier::ILocalStoragePatch, [68](#)
- patchShortDescription
  - quentier::ILocalStoragePatch, [69](#)
- preparedDirtyObjectsForSending
  - quentier::SynchronizationManager, [215](#)
- preparedLinkedNotebooksDirtyObjectsForSending
  - quentier::SynchronizationManager, [215](#)
- print
  - quentier::Account, [14](#)
  - quentier::ApplicationSettings, [20](#)
  - quentier::DefaultLocalStorageCacheExpiryChecker, [39](#)
  - quentier::ENMLConverter::SkipHtmlElementRule, [207](#)

- quantier::ErrorString, 47
- quantier::ILocalStorageCacheExpiryChecker, 65
- quantier::IQuantierException, 89
- quantier::ISyncStateStorage::ISyncState, 91
- quantier::LinkedNotebook, 98
- quantier::LocalStorageCacheManager, 101
- quantier::Note, 157
- quantier::Notebook, 163
- quantier::NoteSearchQuery, 179
- quantier::Printable, 183
- quantier::Resource, 189
- quantier::ResourceRecognitionIndexItem, 191
- quantier::ResourceRecognitionIndices, 193
- quantier::SavedSearch, 196
- quantier::SharedNote, 200
- quantier::SharedNotebook, 202
- quantier::Tag, 223
- quantier::User, 227
- Printable.h, 318
- progress
  - quantier::ILocalStoragePatch, 69
- quantier::Account, 11
  - displayName, 13
  - evernoteAccountType, 13
  - evernoteHost, 13
  - id, 13
  - isEmpty, 13
  - name, 14
  - print, 14
  - setDisplayName, 14
  - shardId, 14
  - type, 14
- quantier::ApplicationSettings, 15
  - ~ApplicationSettings, 17
  - ApplicationSettings, 16, 17
  - beginGroup, 17, 18
  - beginReadArray, 18
  - beginWriteArray, 19
  - contains, 19, 20
  - print, 20
  - remove, 20
  - setValue, 22
  - value, 22, 23
- quantier::ApplicationSettings::ArrayCloser, 25
- quantier::ApplicationSettings::GroupCloser, 54
- quantier::ApplicationSettingsInitializationException, 23
  - exceptionDisplayName, 25
- quantier::AuthenticationManager, 26
- quantier::DatabaseLockedException, 28
  - exceptionDisplayName, 29
- quantier::DatabaseLockFailedException, 29
  - exceptionDisplayName, 31
- quantier::DatabaseOpeningException, 31
  - exceptionDisplayName, 32
- quantier::DatabaseRequestException, 33
  - exceptionDisplayName, 34
- quantier::DateTimePrint, 34
  - IncludeMilliseconds, 35
  - IncludeNumericTimestamp, 35
  - IncludeTimezone, 35
  - Option, 35
- quantier::DecryptedTextManager, 35
- quantier::DefaultLocalStorageCacheExpiryChecker, 35
  - checkLinkedNotebooks, 37
  - checkNotebooks, 37
  - checkNotes, 38
  - checkResources, 38
  - checkSavedSearches, 38
  - checkTags, 38
  - clone, 39
  - print, 39
- quantier::EmptyDataElementException, 39
  - exceptionDisplayName, 41
- quantier::EncryptionManager, 41
- quantier::ENMLConverter, 42
  - cleanupExternalHtml, 43
  - exportNotesToEnex, 44
  - htmlToQTextDocument, 44
  - importEnex, 45
- quantier::ENMLConverter::NoteContentToHtmlExtraData, 164
- quantier::ENMLConverter::SkipHtmlElementRule, 206
  - print, 207
- quantier::ErrorString, 45
  - print, 47
- quantier::EventLoopWithExitStatus, 47
- quantier::FileCopier, 49
- quantier::FileIOProcessorAsync, 50
  - onReadFileRequest, 51
  - onWriteFileRequest, 51
  - readFileRequestProcessed, 52
  - setIdleTimePeriod, 52
  - writeFileRequestProcessed, 52
- quantier::FileSystemWatcher, 53
- quantier::HTMLCleaner, 55
- quantier::IAuthenticationManager, 55
- quantier::IFavoritableDataElement, 56
- quantier::IKeychainService, 58
  - AccessDenied, 59
  - AccessDeniedByUser, 59
  - CouldNotDeleteEntry, 59
  - deletePasswordJobFinished, 60
  - EntryNotFound, 59
  - ErrorCode, 59
  - NoBackendAvailable, 59
  - NoError, 59
  - NotImplemented, 59
  - OtherError, 59
  - readPasswordJobFinished, 60
  - startDeletePasswordJob, 60
  - startReadPasswordJob, 61
  - startWritePasswordJob, 61
  - writePasswordJobFinished, 61
- quantier::ILocalStorageCacheExpiryChecker, 62
  - checkLinkedNotebooks, 64
  - checkNotebooks, 64

- checkNotes, 64
- checkResources, 64
- checkSavedSearches, 64
- checkTags, 65
- clone, 65
- print, 65
- quentier::ILocalStorageDataElement, 66
- quentier::ILocalStoragePatch, 66
  - apply, 67
  - backupLocalStorage, 68
  - backupProgress, 68
  - fromVersion, 68
  - patchLongDescription, 68
  - patchShortDescription, 69
  - progress, 69
  - removeLocalStorageBackup, 69
  - restoreBackupProgress, 69
  - restoreLocalStorageFromBackup, 70
  - toVersion, 70
- quentier::INoteEditorBackend, 70
- quentier::INoteStore, 73
  - authenticateToSharedNotebook, 75
  - createNote, 75
  - createNotebook, 76
  - createSavedSearch, 76
  - createTag, 77
  - getLinkedNotebookSyncChunk, 77
  - getLinkedNotebookSyncState, 79
  - getNote, 79
  - getNoteAsync, 80
  - getResource, 81
  - getResourceAsync, 82
  - getSyncChunk, 82
  - getSyncState, 83
  - noteStoreUrl, 83
  - setAuthData, 83
  - setNoteStoreUrl, 83
  - stop, 84
  - updateNote, 84
  - updateNotebook, 84
  - updateSavedSearch, 85
  - updateTag, 85
- quentier::INoteStoreDataElement, 86
- quentier::IQuentierException, 87
  - print, 89
- quentier::ISyncStateStorage, 91
  - notifySyncStateUpdated, 92
- quentier::ISyncStateStorage::ISyncState, 90
  - print, 91
- quentier::IUserStore, 93
  - checkVersion, 93
  - getAccountLimits, 93
  - getUser, 94
  - setAuthData, 94
- quentier::LimitedStack< T >, 95
- quentier::LinkedNotebook, 96
  - checkParameters, 98
  - clear, 98
  - guid, 98
  - hasGuid, 98
  - hasUpdateSequenceNumber, 98
  - print, 98
  - setGuid, 99
  - setUpdateSequenceNumber, 99
  - updateSequenceNumber, 99
- quentier::LocalStorageCacheManager, 99
  - print, 101
- quentier::LocalStorageCacheManagerException, 101
  - exceptionDisplayName, 103
- quentier::LocalStorageManager, 103
  - accountHighUsn, 111
  - addEnResource, 112
  - addLinkedNotebook, 112
  - addNote, 112
  - addNotebook, 113
  - addSavedSearch, 113
  - addTag, 114
  - addUser, 114
  - ClearDatabase, 110
  - deleteUser, 114
  - enResourceCount, 115
  - expungeEnResource, 115
  - expungeLinkedNotebook, 116
  - expungeNote, 116
  - expungeNotebook, 116
  - expungeNotelessTagsFromLinkedNotebooks, 117
  - expungeSavedSearch, 117
  - expungeTag, 118
  - expungeUser, 118
  - findDefaultNotebook, 118
  - findDefaultOrLastUsedNotebook, 119
  - findEnResource, 119
  - findLastUsedNotebook, 120
  - findLinkedNotebook, 120
  - findNote, 120
  - findNotebook, 121
  - findNoteLocalUidsWithSearchQuery, 121
  - findNotesWithSearchQuery, 122
  - findSavedSearch, 122
  - findTag, 123
  - findUser, 123
  - GetNoteOption, 109
  - GetResourceOption, 110
  - highestSupportedLocalStorageVersion, 123
  - isLocalStorageVersionTooHigh, 124
  - linkedNotebookCount, 124
  - listAllLinkedNotebooks, 124
  - listAllNotebooks, 125
  - listAllSavedSearches, 126
  - listAllSharedNotebooks, 126
  - listAllTags, 126
  - listAllTagsPerNote, 127
  - listLinkedNotebooks, 128
  - listNotebooks, 128
  - listNotes, 129
  - listNotesByLocalUids, 130

- listNotesPerNotebook, [130](#)
- listNotesPerNotebooksAndTags, [131](#)
- listNotesPerTag, [132](#)
- ListObjectsOption, [110](#)
- listSavedSearches, [132](#)
- listSharedNotebooksPerNotebookGuid, [133](#)
- listTags, [133](#)
- listTagsWithNoteLocalUids, [134](#)
- LocalStorageManager, [111](#)
- localStorageRequiresUpgrade, [135](#)
- localStorageVersion, [135](#)
- notebookCount, [135](#)
- noteCount, [136](#)
- noteCountPerNotebook, [136](#)
- noteCountPerNotebooksAndTags, [137](#)
- noteCountPerTag, [137](#)
- noteCountsPerAllTags, [137](#)
- OverrideLock, [110](#)
- requiredLocalStoragePatches, [138](#)
- savedSearchCount, [138](#)
- StartupOption, [110](#)
- switchUser, [138](#)
- tagCount, [139](#)
- updateEnResource, [139](#)
- updateLinkedNotebook, [140](#)
- updateNote, [140](#)
- updateNotebook, [141](#)
- UpdateNoteOption, [111](#)
- UpdateResourceBinaryData, [111](#)
- UpdateResourceMetadata, [111](#)
- updateSavedSearch, [141](#)
- updateTag, [142](#)
- UpdateTags, [111](#)
- updateUser, [142](#)
- upgradeProgress, [143](#)
- userCount, [143](#)
- WithBinaryData, [110](#)
- WithResourceBinaryData, [110](#)
- WithResourceMetadata, [110](#)
- quentier::LocalStorageManagerAsync, [143](#)
- quentier::LoggerInitializationException, [150](#)
  - exceptionDisplayName, [152](#)
- quentier::LRUCache< Key, Value, Allocator >, [152](#)
- quentier::Note, [153](#)
  - checkParameters, [157](#)
  - clear, [157](#)
  - guid, [157](#)
  - hasGuid, [157](#)
  - hasUpdateSequenceNumber, [157](#)
  - print, [157](#)
  - setGuid, [157](#)
  - setUpdateSequenceNumber, [157](#)
  - updateSequenceNumber, [158](#)
- quentier::Notebook, [158](#)
  - checkParameters, [162](#)
  - clear, [162](#)
  - guid, [162](#)
  - hasGuid, [163](#)
  - hasUpdateSequenceNumber, [163](#)
  - print, [163](#)
  - setGuid, [163](#)
  - setUpdateSequenceNumber, [163](#)
  - updateSequenceNumber, [163](#)
- quentier::NoteEditor, [164](#)
  - backend, [168](#)
  - clear, [168](#)
  - convertToNote, [168](#)
  - currentNoteLocalUid, [168](#)
  - defaultFont, [168](#)
  - defaultPalette, [168](#)
  - idleTime, [168](#)
  - inAppNoteLinkPasteRequested, [169](#)
  - initialize, [169](#)
  - isEditorPageModified, [169](#)
  - isModified, [170](#)
  - isNoteLoaded, [170](#)
  - saveNoteToLocalStorage, [170](#)
  - setAccount, [170](#)
  - setBackend, [170](#)
  - setCurrentNoteLocalUid, [170](#)
  - setDefaultFont, [171](#)
  - setDefaultPalette, [171](#)
  - setFocus, [171](#)
  - setInitialPageHtml, [171](#)
  - setNoteDeletedPageHtml, [171](#)
  - setNoteLoadingPageHtml, [172](#)
  - setNoteNotFoundPageHtml, [172](#)
  - setNoteTitle, [172](#)
  - setTagIds, [172](#)
  - setUndoStack, [173](#)
  - undoStack, [173](#)
- quentier::NoteEditorInitializationException, [173](#)
  - exceptionDisplayName, [175](#)
- quentier::NoteEditorPluginInitializationException, [175](#)
  - exceptionDisplayName, [176](#)
- quentier::NoteSearchQuery, [177](#)
  - notebookModifier, [179](#)
  - print, [179](#)
  - queryString, [179](#)
- quentier::NullPtrException, [180](#)
  - exceptionDisplayName, [181](#)
- quentier::Printable, [182](#)
  - print, [183](#)
- quentier::QuentierApplication, [183](#)
- quentier::QuentierUndoCommand, [184](#)
- quentier::Resource, [185](#)
  - checkParameters, [188](#)
  - clear, [188](#)
  - guid, [188](#)
  - hasGuid, [188](#)
  - hasUpdateSequenceNumber, [188](#)
  - print, [189](#)
  - setGuid, [189](#)
  - setUpdateSequenceNumber, [189](#)
  - updateSequenceNumber, [189](#)
- quentier::ResourceRecognitionIndexItem, [189](#)

- print, 191
- quentier::ResourceRecognitionIndexItem::BarcodeItem, 27
- quentier::ResourceRecognitionIndexItem::ObjectItem, 181
- quentier::ResourceRecognitionIndexItem::ShapeItem, 197
- quentier::ResourceRecognitionIndexItem::TextItem, 224
- quentier::ResourceRecognitionIndices, 192
  - print, 193
- quentier::SavedSearch, 193
  - checkParameters, 196
  - clear, 196
  - guid, 196
  - hasGuid, 196
  - hasUpdateSequenceNumber, 196
  - print, 196
  - setGuid, 196
  - setUpdateSequenceNumber, 196
  - updateSequenceNumber, 197
- quentier::SharedNote, 197
  - print, 200
- quentier::SharedNotebook, 200
  - print, 202
- quentier::ShortcutManager, 203
  - defaultShortcut, 204
  - shortcut, 204, 205
  - userShortcut, 205
- quentier::SpellChecker, 208
- quentier::StringUtils, 209
- quentier::StringUtils::StringFilterPredicate, 209
- quentier::SynchronizationManager, 209
  - active, 212
  - authenticate, 212
  - authenticateCurrentAccount, 212
  - authenticationFinished, 212
  - authenticationRevoked, 213
  - detectedConflictDuringLocalChangesSending, 213
  - downloadNoteThumbnailsOption, 213
  - failed, 213
  - finished, 213
  - linkedNotebooksNotesDownloadProgress, 214
  - linkedNotebooksResourcesDownloadProgress, 214
  - linkedNotebooksSyncChunksDownloaded, 214
  - linkedNotebookSyncChunksDownloadProgress, 215
  - notesDownloadProgress, 215
  - preparedDirtyObjectsForSending, 215
  - preparedLinkedNotebooksDirtyObjectsForSending, 215
  - rateLimitExceeded, 216
  - remoteToLocalSyncDone, 216
  - remoteToLocalSyncStopped, 216
  - resourcesDownloadProgress, 216
  - revokeAuthentication, 217
  - sendLocalChangesStopped, 217
  - setAccount, 217
  - setAccountDone, 217
  - setDownloadInkNoteImages, 217
  - setDownloadInkNoteImagesDone, 217
  - setDownloadNoteThumbnails, 218
  - setDownloadNoteThumbnailsDone, 218
  - setInkNoteImagesStoragePath, 218
  - setInkNoteImagesStoragePathDone, 218
  - started, 218
  - stop, 219
  - stopped, 219
  - syncChunksDownloaded, 219
  - syncChunksDownloadProgress, 219
  - SynchronizationManager, 211
  - synchronize, 219
  - willRepeatRemoteToLocalSyncAfterSendingChanges, 220
- quentier::SysInfo, 220
- quentier::Tag, 220
  - checkParameters, 222
  - clear, 222
  - guid, 223
  - hasGuid, 223
  - hasUpdateSequenceNumber, 223
  - print, 223
  - setGuid, 223
  - setUpdateSequenceNumber, 223
  - updateSequenceNumber, 223
- quentier::UidGenerator, 224
- quentier::User, 224
  - print, 227
- QuentierApplication.h, 320
- QuentierCheckPtr.h, 320
- QuentierLogger.h, 261
- QuentierUndoCommand.h, 321
- queryString
  - quentier::NoteSearchQuery, 179
- rateLimitExceeded
  - quentier::SynchronizationManager, 216
- readFileRequestProcessed
  - quentier::FileIOProcessorAsync, 52
- readPasswordJobFinished
  - quentier::IKeychainService, 60
- RegisterMetatypes.h, 290
- remoteToLocalSyncDone
  - quentier::SynchronizationManager, 216
- remoteToLocalSyncStopped
  - quentier::SynchronizationManager, 216
- remove
  - quentier::ApplicationSettings, 20
- removeLocalStorageBackup
  - quentier::ILocalStoragePatch, 69
- requiredLocalStoragePatches
  - quentier::LocalStorageManager, 138
- Resource.h, 290
- ResourceRecognitionIndexItem.h, 292
- ResourceRecognitionIndices.h, 294
- resourcesDownloadProgress
  - quentier::SynchronizationManager, 216

- restoreBackupProgress
  - quentier::ILocalStoragePatch, [69](#)
- restoreLocalStorageFromBackup
  - quentier::ILocalStoragePatch, [70](#)
- revokeAuthentication
  - quentier::SynchronizationManager, [217](#)
- SavedSearch.h, [295](#)
- savedSearchCount
  - quentier::LocalStorageManager, [138](#)
- saveNoteToLocalStorage
  - quentier::NoteEditor, [170](#)
- sendLocalChangesStopped
  - quentier::SynchronizationManager, [217](#)
- setAccount
  - quentier::NoteEditor, [170](#)
  - quentier::SynchronizationManager, [217](#)
- setAccountDone
  - quentier::SynchronizationManager, [217](#)
- setAuthData
  - quentier::INoteStore, [83](#)
  - quentier::IUserStore, [94](#)
- setBackend
  - quentier::NoteEditor, [170](#)
- setCurrentNoteLocalUid
  - quentier::NoteEditor, [170](#)
- setDefaultFont
  - quentier::NoteEditor, [171](#)
- setDefaultPalette
  - quentier::NoteEditor, [171](#)
- setDisplayName
  - quentier::Account, [14](#)
- setDownloadInkNoteImages
  - quentier::SynchronizationManager, [217](#)
- setDownloadInkNoteImagesDone
  - quentier::SynchronizationManager, [217](#)
- setDownloadNoteThumbnails
  - quentier::SynchronizationManager, [218](#)
- setDownloadNoteThumbnailsDone
  - quentier::SynchronizationManager, [218](#)
- setFocus
  - quentier::NoteEditor, [171](#)
- setGuid
  - quentier::LinkedNotebook, [99](#)
  - quentier::Note, [157](#)
  - quentier::Notebook, [163](#)
  - quentier::Resource, [189](#)
  - quentier::SavedSearch, [196](#)
  - quentier::Tag, [223](#)
- setIdleTimePeriod
  - quentier::FileIOProcessorAsync, [52](#)
- setInitialPageHtml
  - quentier::NoteEditor, [171](#)
- setInkNoteImagesStoragePath
  - quentier::SynchronizationManager, [218](#)
- setInkNoteImagesStoragePathDone
  - quentier::SynchronizationManager, [218](#)
- setNoteDeletedPageHtml
  - quentier::NoteEditor, [171](#)
- setNoteLoadingPageHtml
  - quentier::NoteEditor, [172](#)
- setNoteNotFoundPageHtml
  - quentier::NoteEditor, [172](#)
- setNoteStoreUrl
  - quentier::INoteStore, [83](#)
- setNoteTitle
  - quentier::NoteEditor, [172](#)
- setTagIds
  - quentier::NoteEditor, [172](#)
- setUndoStack
  - quentier::NoteEditor, [173](#)
- setUpdateSequenceNumber
  - quentier::LinkedNotebook, [99](#)
  - quentier::Note, [157](#)
  - quentier::Notebook, [163](#)
  - quentier::Resource, [189](#)
  - quentier::SavedSearch, [196](#)
  - quentier::Tag, [223](#)
- setValue
  - quentier::ApplicationSettings, [22](#)
- shardId
  - quentier::Account, [14](#)
- SharedNote.h, [296](#)
- SharedNotebook.h, [298](#)
- shortcut
  - quentier::ShortcutManager, [204](#), [205](#)
- ShortcutManager.h, [321](#)
- Size.h, [323](#)
- SpellChecker.h, [268](#)
- StandardPaths.h, [324](#)
- startDeletePasswordJob
  - quentier::IKeychainService, [60](#)
- started
  - quentier::SynchronizationManager, [218](#)
- startReadPasswordJob
  - quentier::IKeychainService, [61](#)
- StartupOption
  - quentier::LocalStorageManager, [110](#)
- startWritePasswordJob
  - quentier::IKeychainService, [61](#)
- stop
  - quentier::INoteStore, [84](#)
  - quentier::SynchronizationManager, [219](#)
- stopped
  - quentier::SynchronizationManager, [219](#)
- StringUtils.h, [324](#)
- SuppressWarnings.h, [325](#)
- switchUser
  - quentier::LocalStorageManager, [138](#)
- syncChunksDownloaded
  - quentier::SynchronizationManager, [219](#)
- syncChunksDownloadProgress
  - quentier::SynchronizationManager, [219](#)
- SynchronizationManager
  - quentier::SynchronizationManager, [211](#)
- SynchronizationManager.h, [275](#)
- synchronize

- quentier::SynchronizationManager, 219
- SysInfo.h, 326
  - System.h, 327
- Tag.h, 300
- tagCount
  - quentier::LocalStorageManager, 139
- TagSortByParentChildRelations.h, 328
- toVersion
  - quentier::ILocalStoragePatch, 70
- type
  - quentier::Account, 14
- UidGenerator.h, 328
- undoStack
  - quentier::NoteEditor, 173
- updateEnResource
  - quentier::LocalStorageManager, 139
- updateLinkedNotebook
  - quentier::LocalStorageManager, 140
- updateNote
  - quentier::INoteStore, 84
  - quentier::LocalStorageManager, 140
- updateNotebook
  - quentier::INoteStore, 84
  - quentier::LocalStorageManager, 141
- UpdateNoteOption
  - quentier::LocalStorageManager, 111
- UpdateResourceBinaryData
  - quentier::LocalStorageManager, 111
- UpdateResourceMetadata
  - quentier::LocalStorageManager, 111
- updateSavedSearch
  - quentier::INoteStore, 85
  - quentier::LocalStorageManager, 141
- updateSequenceNumber
  - quentier::LinkedNotebook, 99
  - quentier::Note, 158
  - quentier::Notebook, 163
  - quentier::Resource, 189
  - quentier::SavedSearch, 197
  - quentier::Tag, 223
- updateTag
  - quentier::INoteStore, 85
  - quentier::LocalStorageManager, 142
- UpdateTags
  - quentier::LocalStorageManager, 111
- updateUser
  - quentier::LocalStorageManager, 142
- upgradeProgress
  - quentier::LocalStorageManager, 143
- User.h, 301
- userCount
  - quentier::LocalStorageManager, 143
- userShortcut
  - quentier::ShortcutManager, 205
- value
  - quentier::ApplicationSettings, 22, 23
- willRepeatRemoteToLocalSyncAfterSendingChanges
  - quentier::SynchronizationManager, 220
- WithBinaryData
  - quentier::LocalStorageManager, 110
- WithResourceBinaryData
  - quentier::LocalStorageManager, 110
- WithResourceMetadata
  - quentier::LocalStorageManager, 110
- writeFileRequestProcessed
  - quentier::FileIOProcessorAsync, 52
- writePasswordJobFinished
  - quentier::IKeychainService, 61