iLearnRW

0.0.1

Generated by Doxygen 1.8.4

Thu Jun 20 2013 11:57:39

Contents

Chapter 1

Main Page

Source Control

The projects source is managed in a private repository at github.

Project url: `https://github.com/mpakarlsson/ilearnrw`

Chapter 2

Doxygen

A Doxyfile for generating documentation can be found at `docs/Doxygen`.

Installation

Linux

todo: Provide installation documentation.

Windows

To build the documentation a couple of dependencies are required:

Name	Version	Url	Exe	Note
Doxygen	1.8.4	http://doxygen.org	doxygen.exe	-
Mikitex	2.9.3759	http://miktex.org/download	tex.exe	Should work with other tex packages as well.
ghostscript	9.0.7	http://sourceforge.net/projects/ghostscript/	?	-
Graphviz (dot)	2.8	http://www.graphviz.org/	dot.exe	-

The exe files listed must be available in the windows PATH%. variable.

Build documentation

Windows

To build the documentation run the `make_doc.bat` file located at docs/make_doc.bat.

The batchfile will run doxygen and then tex. The output `bin/doxygen/latex/refman.pdf` will be copied to `docs/doxygen.pdf`.

Chapter 3

DataLogger Log format

Following is a quick description about how data is, or will be, stored.

Users

Users are stored in the UserStore. Initially the UserStore just uses Java serialization to write the User objects to disk. This requires that any objects that has a relationship (via class members) to the User object has to implement the java.io.Serializable interface.

This could be changed to a SQL backend in the future.

User Actions

The User actions are envisioned to be stored in a SQL database with a similar setup as described below.

Table UserActions:

Type	Name
PK	actionId
FK	userId d
FK	marker r
FK	applicationId
TimeStamp	timestamp
VARCHAR()	tag
VARCHAR()	value

This would result in SQL output similar to:

Id	Profile	Marker	Application	TimeStamp	Tag	Value
1	child1	Teach1	some_id	2013-06-14	ADD	cat
2	child1	child1	some_id	2013-06-14	REMOVE	cat

Using the getUserActions() function of the IDataLogger interface the database can be queried. To filter the result a UserActionFilter object is provided.

Chapter 4

Todo List

Group DolphinTodo_20130619

Implement IProfileAccessUpdater

DONE: Move interfaces to DataLogger

Implement all usecases provided by IProfileAccessUpdater into console application.

DONE: cleanup the code

DONE: Merge with the main branch.

Class ilearnrw.datalogger.UserActionFilter

Add getters/setters, appropriate functions etc.

Class ilearnrw.datalogger.UserStore

Replace storage backed with SQL database.

Enforce unique usernames.

Member ilearnrw.datalogger.UserStore.update (User user)

Don't allow multiple users with the same username.

Class ilearnrw.user.UserAction

Add some documentation about this class.

Add get/setters

Member ilearnrw.user.UserDetails.username

Something to think about is whether this should be case-sensitive and if unicode should be supported. Unicode can create some headaches, there was a blog post about this recently (of which i only read the headline) see: http://labs.spotify.com/2013/06/18/creative-usernames/

Chapter 5

Module Documentation

5.1 DolphinTodo_20130619

Todo Implement IProfileAccessUpdater

DONE: Move interfaces to DataLogger

Implement all usecases provided by IProfileAccessUpdater into console application.

DONE: cleanup the code

DONE: Merge with the main branch.

Chapter 6

Namespace Documentation

6.1 Package ilearnrw

Packages

- package application
- package datalogger

  DataLogger package.
- package prototype
- package user

6.2 Package ilearnrw.application

Classes

- class ApplicationId

  ApplicationId identifies an application in different parts of the system.

6.3 Package ilearnrw.datalogger

DataLogger package.

Classes

- class DataLogger
- interface IDataLogger

  Provides an interface for applications to log UserActions.
- interface ILoginProvider
- interface IProfileAccessUpdater

  Interface to update profiles in the iLearnRW system.
- interface IUserAdministration

  Provides User administration tasks.
- class UserActionFilter
- class UserStore

  The UserStore is used to create and store user objects persistantly.

6.3.1 Detailed Description

DataLogger package. Four main interfaces has been defined at the moment.

1. IDataLogger
2. ILoginProvider
3. IProfileAccessUpdater
4. IUserAdministration

The DataLogger object implements IDataLogger, ILoginProvider and IProfileAccessUpdater.

See their respective documentation for details.

6.4 Package ilearnrw.prototype

Packages

- package application

6.5 Package ilearnrw.prototype.application

Classes

- class ConsoleMenu
- class ConsoleMenuAction
- class DatabaseManager
- class Program
- class UserLogin

  Main menu of a logged in user.
- class UserProblems

6.6 Package ilearnrw.user

Classes

- class Category
- enum GameAction
- class IlearnException
- enum LanguageCode
- class ProblemDefinition
- class ProblemDefinitionIndex
- class Subcategory
- class Tester
- class User
- class UserAction
- class UserDetails
- class UserPreferences
- class UserProblem
- class UserProblemsList
- class UserProfile
- class UserSession

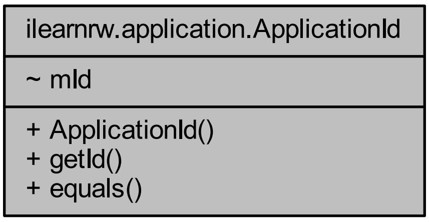
Chapter 7

Class Documentation

7.1 ilearnrw.application.ApplicationId Class Reference

ApplicationId identifies an application in different parts of the system.

Collaboration diagram for ilearnrw.application.ApplicationId:



Public Member Functions

- ApplicationId (String id)
- String getId ()
- boolean equals (ApplicationId id)

Package Attributes

- String mId

7.1.1 Detailed Description

ApplicationId identifies an application in different parts of the system.

It should be a unique identifier

7.1.2 Constructor & Destructor Documentation

7.1.2.1 ilearnrw.application.ApplicationId.ApplicationId (String *id*)

7.1.3 Member Function Documentation

7.1.3.1 boolean ilearnrw.application.ApplicationId.equals (ApplicationId *id*)

7.1.3.2 String ilearnrw.application.ApplicationId.getId ( )

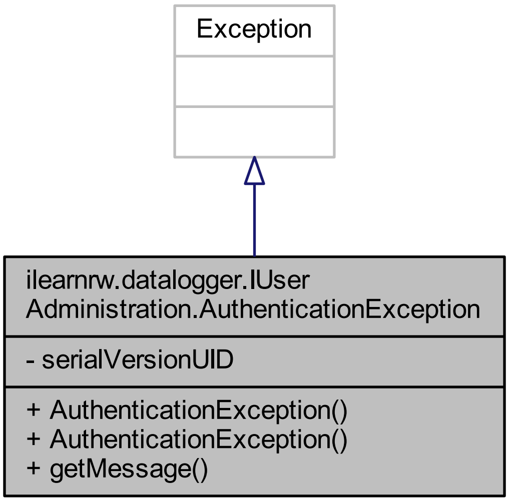
7.1.4 Member Data Documentation

7.1.4.1 String ilearnrw.application.ApplicationId.mId [package]

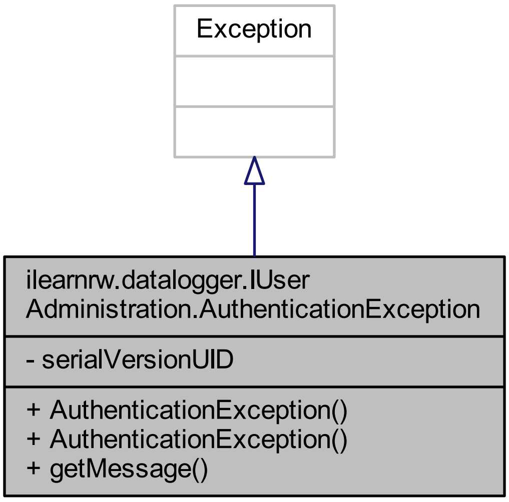
7.2 ilearnrw.datalogger.IUserAdministration.AuthenticationException Class Reference

The marker does not have permission to perform the requested change.

Inheritance diagram for ilearnrw.datalogger.IUserAdministration.AuthenticationException:


Exception
ilearnrw.datalogger.IUser
Administration.AuthenticationException
- serialVersionUID
+ AuthenticationException()
+ AuthenticationException()
+ getMessage()

Collaboration diagram for ilearnrw.datalogger.IUserAdministration.AuthenticationException:


Exception
ilearnrw.datalogger.IUser
Administration.AuthenticationException
- serialVersionUID
+ AuthenticationException()
+ AuthenticationException()
+ getMessage()

Public Member Functions

- AuthenticationException (String message)
- AuthenticationException (String message, Throwable throwable)
- String getMessage ()

Static Private Attributes

- static final long serialVersionUID = 1L

7.2.1 Detailed Description

The marker does not have permission to perform the requested change.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 ilearnrw.datalogger.IUserAdministration.AuthenticationException.AuthenticationException (String *message*)

7.2.2.2 ilearnrw.datalogger.IUserAdministration.AuthenticationException.AuthenticationException (String *message,* Throwable *throwable*)

7.2.3 Member Function Documentation

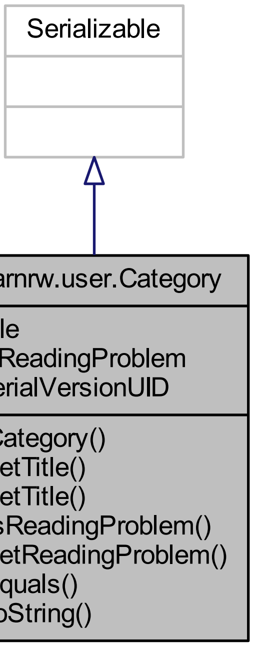
7.2.3.1 String ilearnrw.datalogger.IUserAdministration.AuthenticationException.getMessage ( )

7.2.4 Member Data Documentation

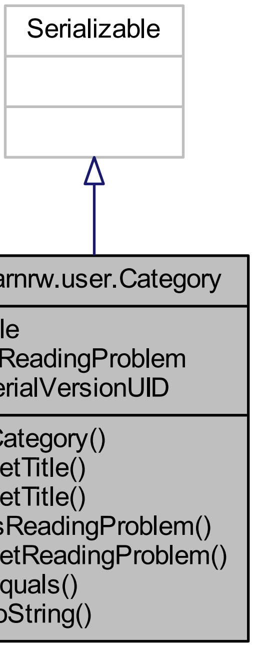
7.2.4.1 final long ilearnrw.datalogger.IUserAdministration.AuthenticationException.serialVersionUID = 1L [static], [private]

7.3 ilearnrw.user.Category Class Reference

Inheritance diagram for ilearnrw.user.Category:



Collaboration diagram for ilearnrw.user.Category:



Public Member Functions

- Category (String title, boolean isReadingProblem)
- String getTitle ()
- void setTitle (String title)
- boolean isReadingProblem ()
- void setReadingProblem (boolean isReadingProblem)
- boolean equals (Object obj)
- String toString ()

Private Attributes

- String title
- boolean isReadingProblem

Static Private Attributes

- static final long serialVersionUID = 1L

7.3.1 Constructor & Destructor Documentation

7.3.1.1 ilearnrw.user.Category.Category (String *title,* boolean *isReadingProblem*)

7.3.2 Member Function Documentation

7.3.2.1 boolean ilearnrw.user.Category.equals (Object *obj*)

7.3.2.2 String ilearnrw.user.Category.getTitle ( )

7.3.2.3 boolean ilearnrw.user.Category.isReadingProblem ( )

7.3.2.4 void ilearnrw.user.Category.setReadingProblem (boolean *isReadingProblem*)

7.3.2.5 void ilearnrw.user.Category.setTitle (String *title*)

7.3.2.6 String ilearnrw.user.Category.toString ( )

7.3.3 Member Data Documentation

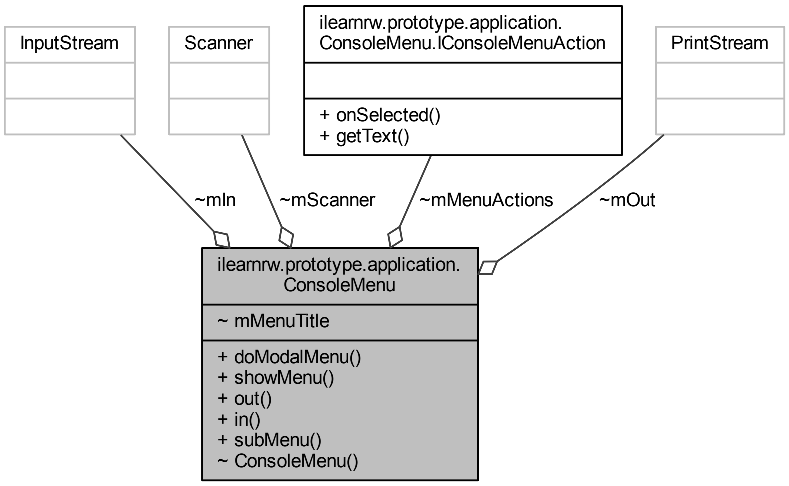
7.3.3.1 boolean ilearnrw.user.Category.isReadingProblem `[private]`

7.3.3.2 final long ilearnrw.user.Category.serialVersionUID = 1L `[static]`, `[private]`

7.3.3.3 String ilearnrw.user.Category.title `[private]`

7.4 ilearnrw.prototype.application.ConsoleMenu Class Reference

Collaboration diagram for ilearnrw.prototype.application.ConsoleMenu:



Classes

- enum EConsoleMenuActionResult
- interface IConsoleMenuAction

Public Member Functions

- void doModalMenu ()

- EConsoleMenuActionResult showMenu ()
- PrintStream out ()
- Scanner in ()
- ConsoleMenu subMenu (String menuTitle, IConsoleMenuAction[] menuActions)

Package Functions

- ConsoleMenu (PrintStream outStream, InputStream inStream, String menuTitle, IConsoleMenuAction[] menuActions)

Package Attributes

- PrintStream mOut
- InputStream mIn
- Scanner mScanner
- String mMenuTitle
- IConsoleMenuAction[] mMenuActions

7.4.1 Constructor & Destructor Documentation

7.4.1.1 ilearnrw.prototype.application.ConsoleMenu.ConsoleMenu (PrintStream *outStream,* InputStream *inStream,* String *menuTitle,* IConsoleMenuAction[] *menuActions*) [package]

7.4.2 Member Function Documentation

7.4.2.1 void ilearnrw.prototype.application.ConsoleMenu.doModalMenu ( )

7.4.2.2 Scanner ilearnrw.prototype.application.ConsoleMenu.in ( )

7.4.2.3 PrintStream ilearnrw.prototype.application.ConsoleMenu.out ( )

7.4.2.4 EConsoleMenuActionResult ilearnrw.prototype.application.ConsoleMenu.showMenu ( )

7.4.2.5 ConsoleMenu ilearnrw.prototype.application.ConsoleMenu.subMenu (String *menuTitle,* IConsoleMenuAction[] *menuActions*)

7.4.3 Member Data Documentation

7.4.3.1 InputStream ilearnrw.prototype.application.ConsoleMenu.mIn [package]

7.4.3.2 IConsoleMenuAction [] ilearnrw.prototype.application.ConsoleMenu.mMenuActions [package]

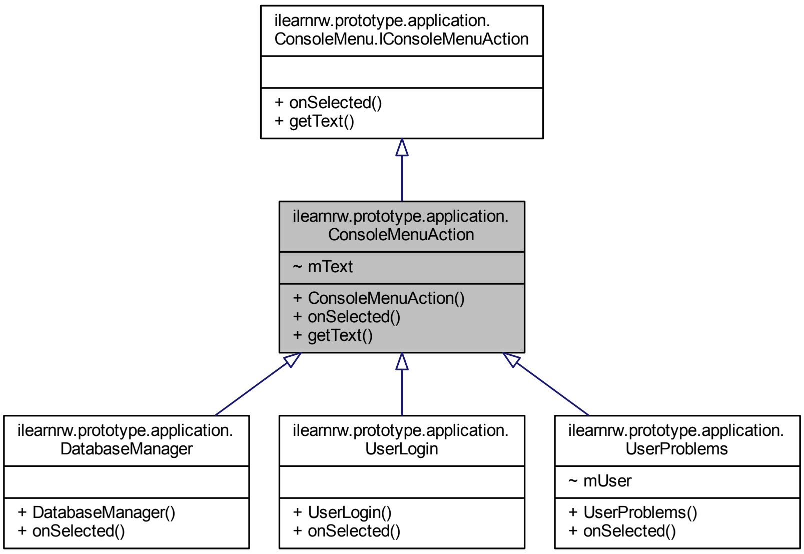
7.4.3.3 String ilearnrw.prototype.application.ConsoleMenu.mMenuTitle [package]

7.4.3.4 PrintStream ilearnrw.prototype.application.ConsoleMenu.mOut [package]

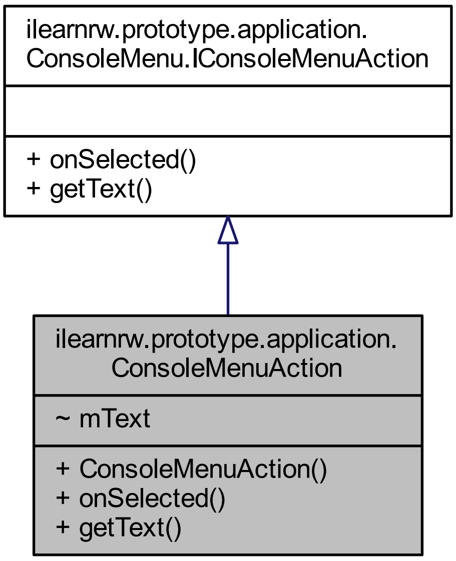
7.4.3.5 Scanner ilearnrw.prototype.application.ConsoleMenu.mScanner [package]

7.5 ilearnrw.prototype.application.ConsoleMenuAction Class Reference

Inheritance diagram for ilearnrw.prototype.application.ConsoleMenuAction:



Collaboration diagram for ilearnrw.prototype.application.ConsoleMenuAction:



Public Member Functions

- ConsoleMenuAction (String text)
- EConsoleMenuActionResult onSelected (ConsoleMenu menu)
- String getText ()

Package Attributes

- String mText

7.5.1 Constructor & Destructor Documentation

7.5.1.1 ilearnrw.prototype.application.ConsoleMenuAction.ConsoleMenuAction (String *text*)

7.5.2 Member Function Documentation

7.5.2.1 String ilearnrw.prototype.application.ConsoleMenuAction.getText ( )

Implements ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.

7.5.2.2 EConsoleMenuActionResult ilearnrw.prototype.application.ConsoleMenuAction.onSelected (ConsoleMenu *menu*)

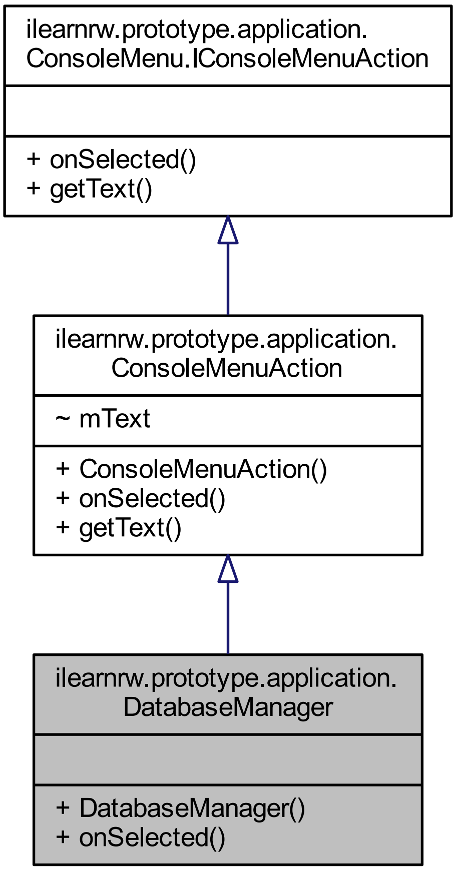
Implements ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.

7.5.3 Member Data Documentation

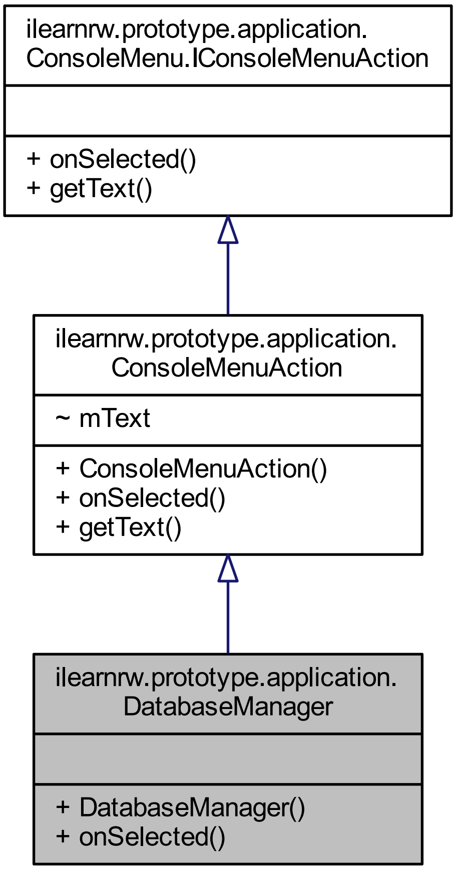
7.5.3.1 String ilearnrw.prototype.application.ConsoleMenuAction.mText `[package]`

7.6 ilearnrw.prototype.application.DatabaseManager Class Reference

Inheritance diagram for ilearnrw.prototype.application.DatabaseManager:



Collaboration diagram for ilearnrw.prototype.application.DatabaseManager:



Classes

- interface ISelectUser

Public Member Functions

- DatabaseManager (String text)
- EConsoleMenuActionResult onSelected (ConsoleMenu menu)

Additional Inherited Members

7.6.1 Constructor & Destructor Documentation

7.6.1.1 ilearnrw.prototype.application.DatabaseManager.DatabaseManager (String *text*)

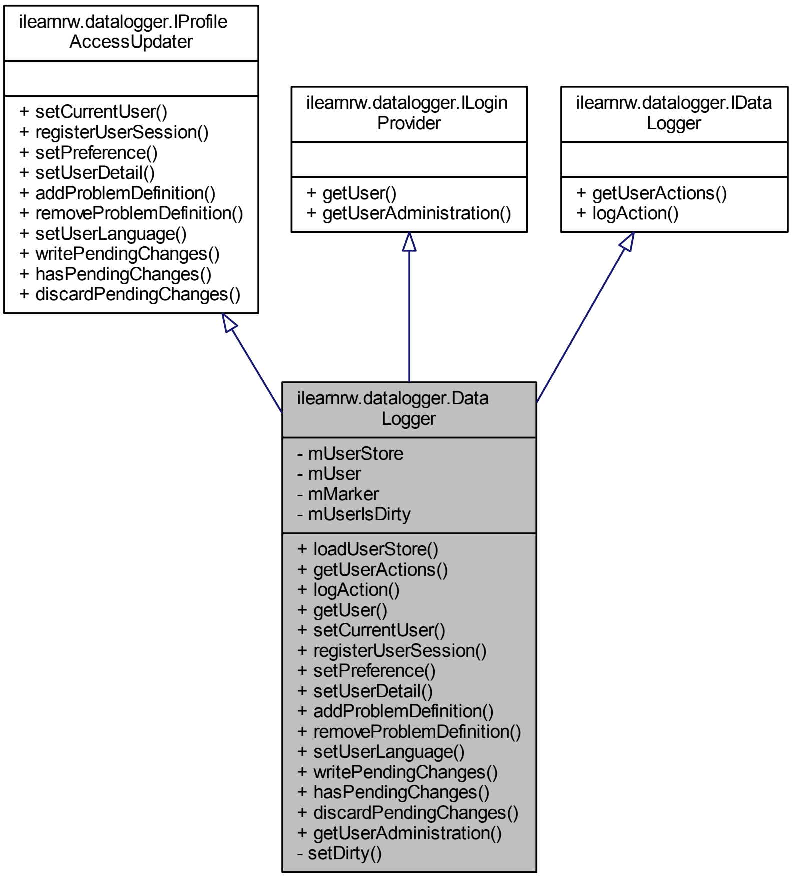
7.6.2 Member Function Documentation

7.6.2.1 EConsoleMenuActionResult ilearnrw.prototype.application.DatabaseManager.onSelected (ConsoleMenu *menu*)

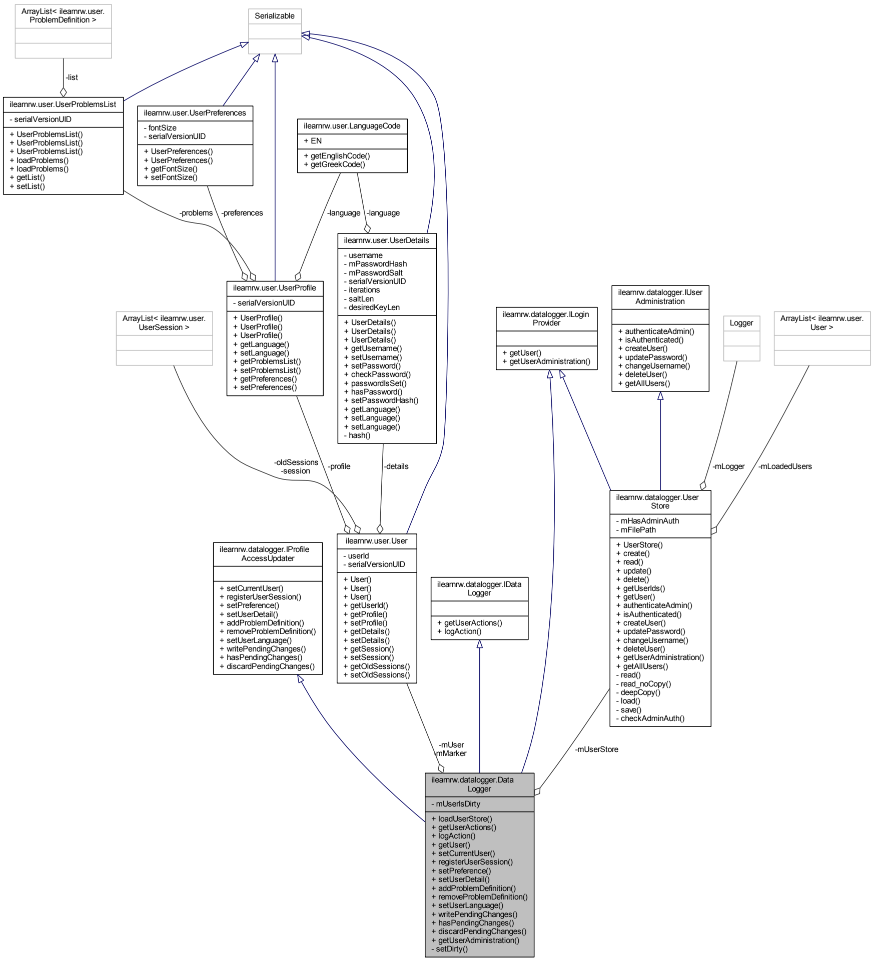
Implements ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.

7.7 ilearnrw.datalogger.DataLogger Class Reference

Inheritance diagram for ilearnrw.datalogger.DataLogger:



Collaboration diagram for ilearnrw.datalogger.DataLogger:



Public Member Functions

- boolean loadUserStore (String filePath)
- List< UserAction > getUserActions (UserActionFilter filter)
- void logAction (UserAction action)

  The Timestamp of the UserAction should be updated before the actions is stored. The timestamp should be local time of the server when this function is called.

- User getUser (String username, String password)
- void setCurrentUser (User user, User marker) throws PermissionException, PendingChangesAvailable

  Sets the userprofile to work with and who will be the `marker` *for the changes.*

- boolean registerUserSession (UserSession session)

  Registers a new User session in the log.

- boolean setPreference (UserPreferences preferences)

  Registers a change to the users preferences.

- boolean setUserDetail (UserDetails userDetails)

  Updates a UserDetail.

- boolean addProblemDefinition (String URI)

  Adds a predefined ProblemDefinition to the user profile.

- boolean removeProblemDefinition (String URI)

Removes a ProblemDefinition.

- boolean setUserLanguage (LanguageCode Language)

    Sets the userlanguage.

- User writePendingChanges ()

    Writes unwritten changes to the back end (persistent storage).

- boolean hasPendingChanges ()

    Checks if there are any changes pending.

- void discardPendingChanges ()
- IUserAdministration getUserAdministration ()

Private Member Functions

- void setDirty ()

Private Attributes

- UserStore mUserStore = null
- User mUser = null
- User mMarker = null
- boolean mUserIsDirty = false

7.7.1 Detailed Description

DataLogger class.

The Datalogger is the main object that Applications will use to query and update the user object and it's properties.

7.7.2 Member Function Documentation

7.7.2.1 boolean ilearnrw.datalogger.DataLogger.addProblemDefinition (String *URI*)

Adds a predefined ProblemDefinition to the user profile.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.2 void ilearnrw.datalogger.DataLogger.discardPendingChanges ( )

Will discard the temporary user object.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.3 User ilearnrw.datalogger.DataLogger.getUser (String *username,* String *password*)

Gets a User object from a username and password.

Returns

A User object if successful. Otherwise null.

Implements ilearnrw.datalogger.ILoginProvider.

7.7.2.4 List<UserAction> ilearnrw.datalogger.DataLogger.getUserActions (UserActionFilter *filter*)

Given a UserActionFilter this function can return a list of UserActions.

Implements ilearnrw.datalogger.IDataLogger.

7.7.2.5 IUserAdministration ilearnrw.datalogger.DataLogger.getUserAdministration ( )

Implements ilearnrw.datalogger.ILoginProvider.

7.7.2.6 boolean ilearnrw.datalogger.DataLogger.hasPendingChanges ( )

Checks if there are any changes pending.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.7 boolean ilearnrw.datalogger.DataLogger.loadUserStore (String *filePath*)

Loads the UserStore from disk.

This has to be called during initialization.

Note

filePath does not have to exist. But the directory it is in has to.

Parameters

filePath	FilePath to a disk copy.

Returns

true if successful. false if already loaded.

7.7.2.8 void ilearnrw.datalogger.DataLogger.logAction (UserAction *action*)

The Timestamp of the UserAction should be updated before the actions is stored. The timestamp should be local time of the server when this function is called.

Implements ilearnrw.datalogger.IDataLogger.

7.7.2.9 boolean ilearnrw.datalogger.DataLogger.registerUserSession (UserSession *session*)

Registers a new User session in the log.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.10 boolean ilearnrw.datalogger.DataLogger.removeProblemDefinition (String *URI*)

Removes a ProblemDefinition.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.11 void ilearnrw.datalogger.DataLogger.setCurrentUser (User *user,* User *marker*) throws PermissionException, PendingChangesAvailable

Sets the userprofile to work with and who will be the `marker` for the changes.

Doing this once saves the application from passing the userprofile to all functions.

The `marker` can be the same as the profile to be edited.

Exceptions

PermissionException	Thrown when the admin profile does not have permissions to update the profile.
PendingChangesAvailable	Thrown when there are pending changes are available.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.12 void ilearnrw.datalogger.DataLogger.setDirty ( ) [private]

7.7.2.13 boolean ilearnrw.datalogger.DataLogger.setPreference (UserPreferences *preferences*)

Registers a change to the users preferences.

Should changes to user preferences be logged so that teachers/experts can be notified/see that a preference has been changed?

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.14 boolean ilearnrw.datalogger.DataLogger.setUserDetail (UserDetails *userDetails*)

Updates a UserDetail.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.15 boolean ilearnrw.datalogger.DataLogger.setUserLanguage (LanguageCode *Language*)

Sets the userlanguage.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.2.16 User ilearnrw.datalogger.DataLogger.writePendingChanges ( )

Writes unwritten changes to the back end (persistent storage).

This will flush all pending operations in the same order as they are added. This is important if for example the same preference is set twice.

Returns

The new user after the changes has been applied.

Implements ilearnrw.datalogger.IProfileAccessUpdater.

7.7.3 Member Data Documentation

7.7.3.1 User ilearnrw.datalogger.DataLogger.mMarker = null [private]

The `marker` to use when logging changes.

7.7.3.2 User ilearnrw.datalogger.DataLogger.mUser = null [private]

The current user to be updated.

7.7.3.3 boolean ilearnrw.datalogger.DataLogger.mUserIsDirty = false [private]

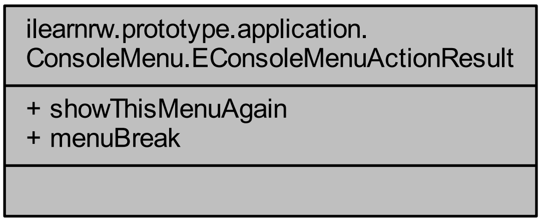
Flag to keep track on whether mUser is dirty;

7.7.3.4 UserStore ilearnrw.datalogger.DataLogger.mUserStore = null [private]

The object that store users.

7.8 ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult Enum Reference

Collaboration diagram for ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult:


ilearnrw.prototype.application.
ConsoleMenu.EConsoleMenuActionResult
+ showThisMenuAgain
+ menuBreak

Public Attributes

- showThisMenuAgain
- menuBreak

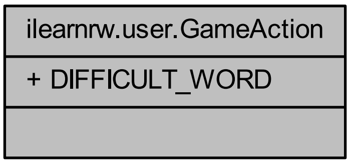
7.8.1 Member Data Documentation

7.8.1.1 ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult.menuBreak

7.8.1.2 ilearnrw.prototype.application.ConsoleMenu.EConsoleMenuActionResult.showThisMenuAgain

7.9 ilearnrw.user.GameAction Enum Reference

Collaboration diagram for ilearnrw.user.GameAction:



Public Attributes

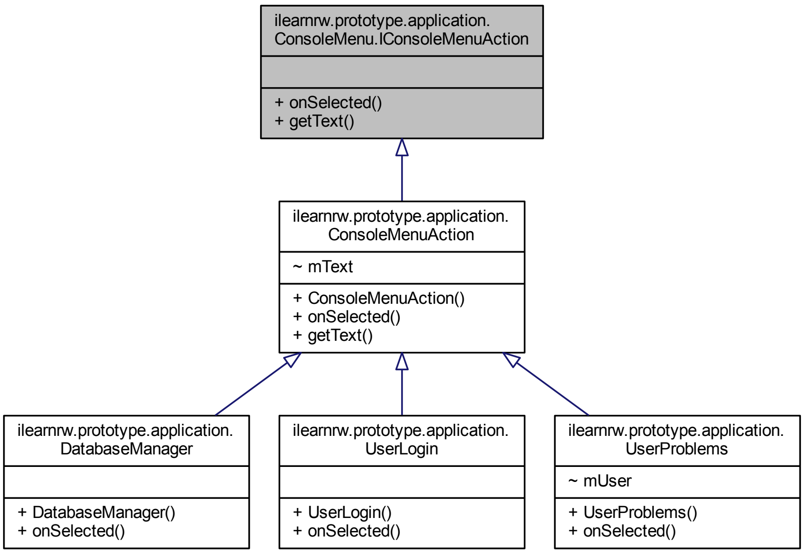
- DIFFICULT_WORD

7.9.1 Member Data Documentation

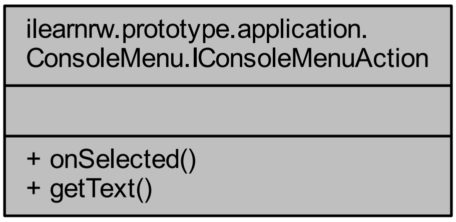
7.9.1.1 ilearnrw.user.GameAction.DIFFICULT_WORD

7.10 ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction Interface Reference

Inheritance diagram for ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction:



Collaboration diagram for ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction:


ilearnrw.prototype.application.
ConsoleMenu.IConsoleMenuAction
+ onSelected()
+ getText()

Public Member Functions

- EConsoleMenuActionResult onSelected (ConsoleMenu menu)
- String getText ()

7.10.1 Member Function Documentation

7.10.1.1 String ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.getText ( )

Implemented in ilearnrw.prototype.application.ConsoleMenuAction.

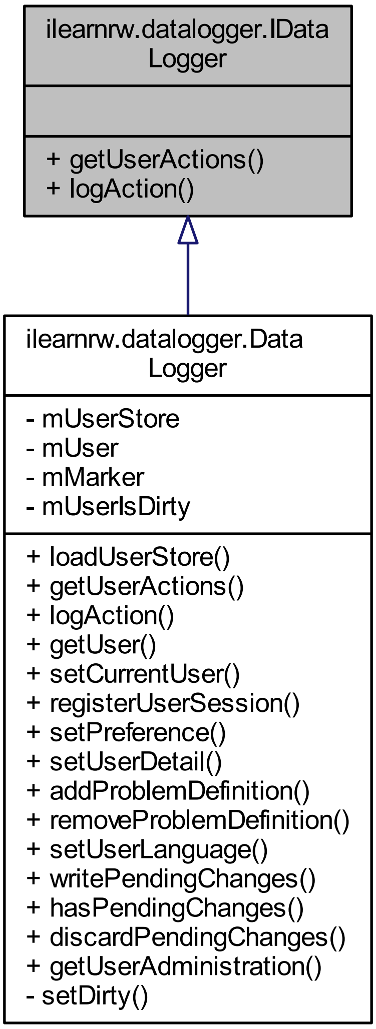
7.10.1.2 EConsoleMenuActionResult ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.onSelected (ConsoleMenu *menu*)

Implemented in ilearnrw.prototype.application.UserProblems, ilearnrw.prototype.application.DatabaseManager, ilearnrw.prototype.application.UserLogin, and ilearnrw.prototype.application.ConsoleMenuAction.

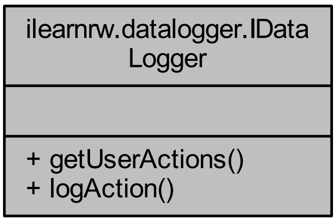
7.11 ilearnrw.datalogger.IDataLogger Interface Reference

Provides an interface for applications to log UserActions.

Inheritance diagram for ilearnrw.datalogger.IDataLogger:



Collaboration diagram for ilearnrw.datalogger.IDataLogger:



Public Member Functions

- List< UserAction > getUserActions (UserActionFilter filter)
- void logAction (UserAction action)

  The Timestamp of the UserAction should be updated before the actions is stored. The timestamp should be local time of the server when this function is called.

7.11.1 Detailed Description

Provides an interface for applications to log UserActions.

7.11.2 Member Function Documentation

7.11.2.1 List<**UserAction**> ilearnrw.datalogger.IDataLogger.getUserActions (**UserActionFilter** *filter*)

Given a UserActionFilter this function can return a list of UserActions.

Implemented in ilearnrw.datalogger.DataLogger.

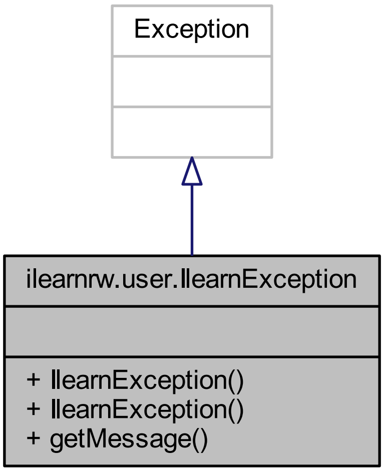
7.11.2.2 void ilearnrw.datalogger.IDataLogger.logAction (**UserAction** *action*)

The Timestamp of the UserAction should be updated before the actions is stored. The timestamp should be local time of the server when this function is called.

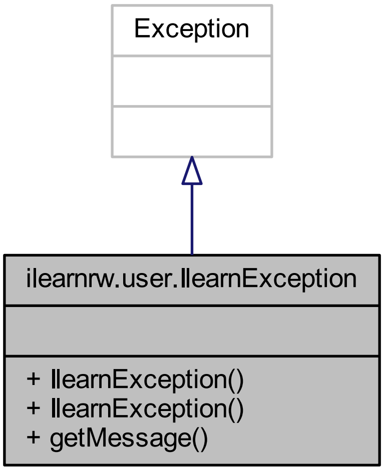
Implemented in ilearnrw.datalogger.DataLogger.

7.12 ilearnrw.user.IlearnException Class Reference

Inheritance diagram for ilearnrw.user.IlearnException:



Collaboration diagram for ilearnrw.user.IlearnException:



Public Member Functions

- IlearnException (String message)
- IlearnException (String message, Throwable throwable)
- String getMessage ()

7.12.1 Constructor & Destructor Documentation

7.12.1.1 ilearnrw.user.IlearnException.IlearnException (String *message*)

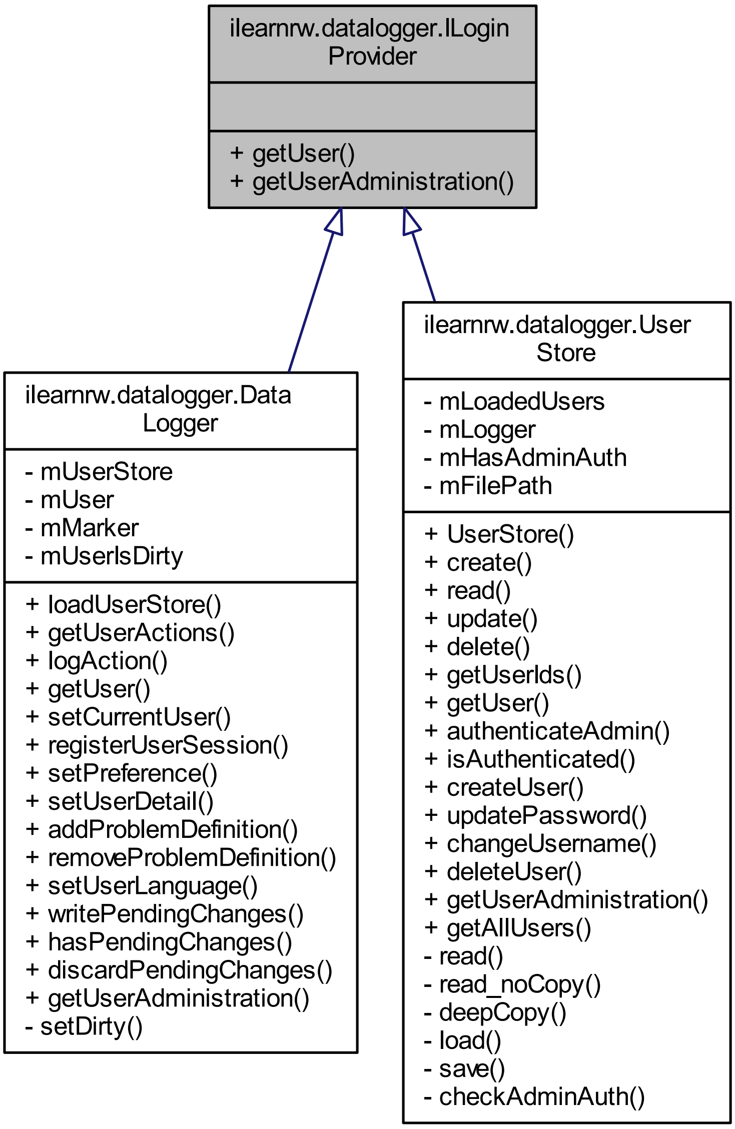
7.12.1.2 ilearnrw.user.IlearnException.IlearnException (String *message,* Throwable *throwable*)

7.12.2 Member Function Documentation

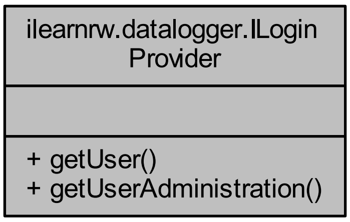
7.12.2.1 String ilearnrw.user.IlearnException.getMessage ( )

7.13 ilearnrw.datalogger.ILoginProvider Interface Reference

Inheritance diagram for ilearnrw.datalogger.ILoginProvider:



Collaboration diagram for ilearnrw.datalogger.ILoginProvider:



Public Member Functions

- User getUser (String username, String password)

  Authenticates a User.

- IUserAdministration getUserAdministration ()

7.13.1 Member Function Documentation

7.13.1.1 User ilearnrw.datalogger.ILoginProvider.getUser (String *username,* String *password*)

Authenticates a User.

Implemented in ilearnrw.datalogger.UserStore, and ilearnrw.datalogger.DataLogger.

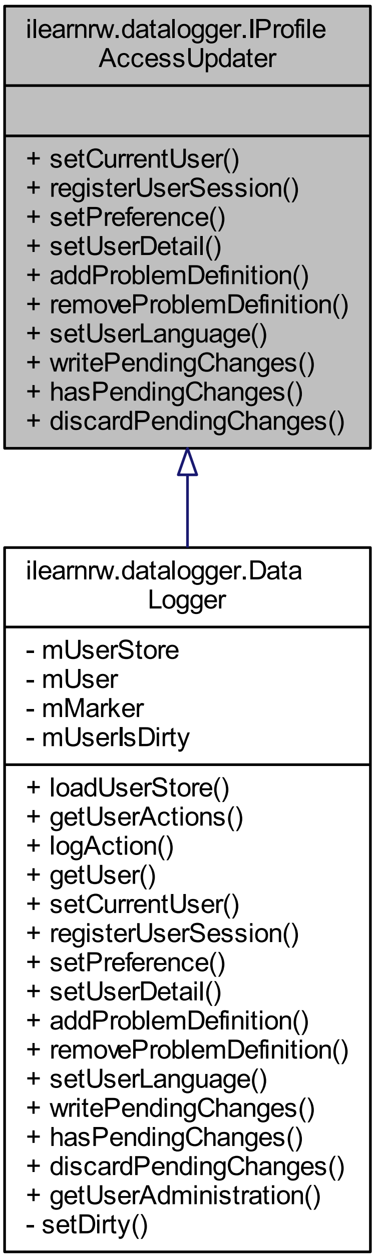
7.13.1.2 IUserAdministration ilearnrw.datalogger.ILoginProvider.getUserAdministration ( )

Implemented in ilearnrw.datalogger.UserStore, and ilearnrw.datalogger.DataLogger.

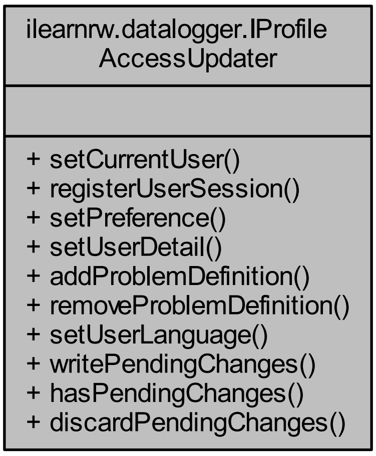
7.14 ilearnrw.datalogger.IProfileAccessUpdater Interface Reference

Interface to update profiles in the iLearnRW system.

Inheritance diagram for ilearnrw.datalogger.IProfileAccessUpdater:



Collaboration diagram for ilearnrw.datalogger.IProfileAccessUpdater:



Classes

- class PendingChangesAvailable

  Indicates that an action cannot be performed due to pending changes.
- class PermissionException

  The marker does not have permission to perform the requested change.

Public Member Functions

- void setCurrentUser (User user, User marker) throws PermissionException, PendingChangesAvailable

  Sets the userprofile to work with and who will be the `marker` *for the changes.*
- boolean registerUserSession (UserSession session)

  Registers a new User session in the log.
- boolean setPreference (UserPreferences preferences)

  Registers a change to the users preferences.
- boolean setUserDetail (UserDetails userDetails)

  Updates a UserDetail.
- boolean addProblemDefinition (String URI)

  Adds a predefined ProblemDefinition to the user profile.
- boolean removeProblemDefinition (String URI)

  Removes a ProblemDefinition.
- boolean setUserLanguage (LanguageCode Language)

  Sets the userlanguage.
- User writePendingChanges ()

  Writes unwritten changes to the back end (persistent storage).
- boolean hasPendingChanges ()

  Checks if there are any changes pending.
- void discardPendingChanges ()

  Discards all pending changes and resets the internal state.

7.14.1 Detailed Description

Interface to update profiles in the iLearnRW system.

This interface assumes that `Apps` would get a read-only copy of the users profile and would then have to use this interface to make changes to the profile. When the profile has been updated a new readonly copy would have to be fetched.

A nice side-effect is that we can make sure to log all changes to the profile.

Updater also manages

Registry - Handles/stores users/profiles.

Note

> This is the interface that the application would use. Not an interface appropriate to a webservice. The implementer of this interface could still use a webservice to store the information in.

UserAdd/delete etc.. Get user (given userid, password)

7.14.2 Member Function Documentation

7.14.2.1 boolean ilearnrw.datalogger.IProfileAccessUpdater.addProblemDefinition (String *URI*)

Adds a predefined ProblemDefinition to the user profile.

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.2 void ilearnrw.datalogger.IProfileAccessUpdater.discardPendingChanges ( )

Discards all pending changes and resets the internal state.

In order to add new changes setCurrentUser has to be called again.

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.3 boolean ilearnrw.datalogger.IProfileAccessUpdater.hasPendingChanges ( )

Checks if there are any changes pending.

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.4 boolean ilearnrw.datalogger.IProfileAccessUpdater.registerUserSession (UserSession *session*)

Registers a new User session in the log.

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.5 boolean ilearnrw.datalogger.IProfileAccessUpdater.removeProblemDefinition (String *URI*)

Removes a ProblemDefinition.

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.6 void ilearnrw.datalogger.IProfileAccessUpdater.setCurrentUser (User *user,* User *marker*) throws PermissionException, PendingChangesAvailable

Sets the userprofile to work with and who will be the `marker` for the changes.

Doing this once saves the application from passing the userprofile to all functions.

The `marker` can be the same as the profile to be edited.

Exceptions

PermissionException	Thrown when the admin profile does not have permissions to update the profile.
PendingChangesAvailable	Thrown when there are pending changes are available.

Parameters

user	Profile of the user to update
marker	Profile of the user who is doing the update

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.7 boolean ilearnrw.datalogger.IProfileAccessUpdater.setPreference (UserPreferences *preferences*)

Registers a change to the users preferences.

Should changes to user preferences be logged so that teachers/experts can be notified/see that a preference has been changed?

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.8 boolean ilearnrw.datalogger.IProfileAccessUpdater.setUserDetail (UserDetails *userDetails*)

Updates a UserDetail.

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.9 boolean ilearnrw.datalogger.IProfileAccessUpdater.setUserLanguage (LanguageCode *Language*)

Sets the userlanguage.

Implemented in ilearnrw.datalogger.DataLogger.

7.14.2.10 User ilearnrw.datalogger.IProfileAccessUpdater.writePendingChanges ( )

Writes unwritten changes to the back end (persistent storage).

This will flush all pending operations in the same order as they are added. This is important if for example the same preference is set twice.

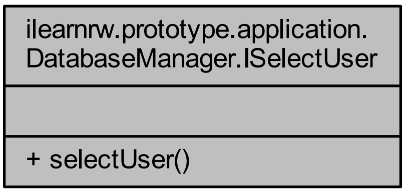
Returns

The new user after the changes has been applied.

Implemented in ilearnrw.datalogger.DataLogger.

7.15 ilearnrw.prototype.application.DatabaseManager.ISelectUser Interface Reference

Collaboration diagram for ilearnrw.prototype.application.DatabaseManager.ISelectUser:



Public Member Functions

- User selectUser (ConsoleMenu menu) throws AuthenticationException

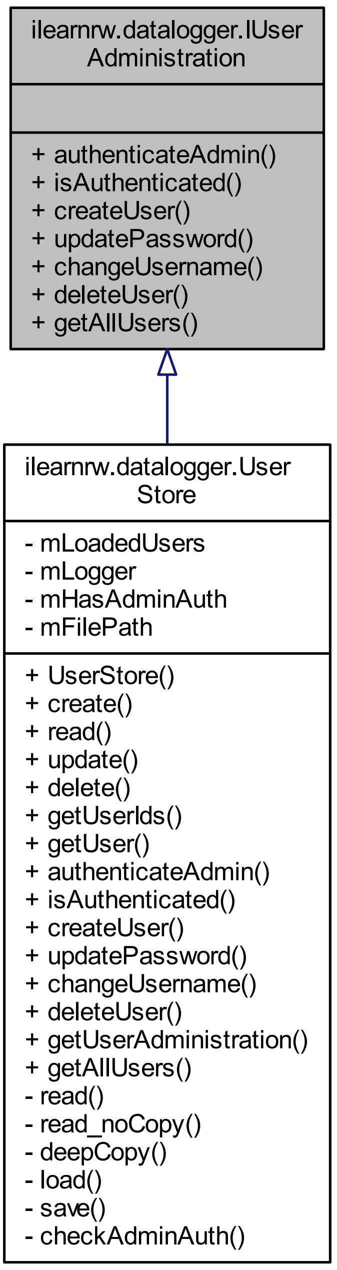
7.15.1 Member Function Documentation

7.15.1.1 User ilearnrw.prototype.application.DatabaseManager.ISelectUser.selectUser (ConsoleMenu *menu*) throws AuthenticationException

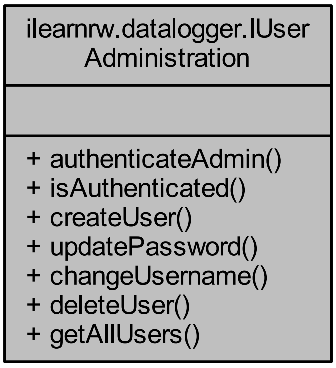
7.16 ilearnrw.datalogger.IUserAdministration Interface Reference

Provides User administration tasks.

Inheritance diagram for ilearnrw.datalogger.IUserAdministration:



Collaboration diagram for ilearnrw.datalogger.IUserAdministration:



Classes

- class AuthenticationException

  The marker does not have permission to perform the requested change.

Public Member Functions

- void authenticateAdmin (String adminPassword) throws AuthenticationException
- boolean isAuthenticated ()
- User createUser (String username, String password) throws AuthenticationException
- User updatePassword (String username, String newPassword) throws AuthenticationException
- User changeUsername (String oldUsername, String newUsername) throws AuthenticationException
- boolean deleteUser (String username) throws AuthenticationException
- List< User > getAllUsers () throws AuthenticationException

7.16.1 Detailed Description

Provides User administration tasks.

7.16.2 Member Function Documentation

7.16.2.1 void ilearnrw.datalogger.IUserAdministration.authenticateAdmin (String *adminPassword*) throws AuthenticationException

Authenticates an Administrator by requiring a password.

Parameters

adminPassword	The administrator password. This could be set in a configuration file.

Implemented in ilearnrw.datalogger.UserStore.

7.16.2.2 User ilearnrw.datalogger.IUserAdministration.changeUsername (String *oldUsername,* String *newUsername*) throws AuthenticationException

Implemented in ilearnrw.datalogger.UserStore.

7.16.2.3 User ilearnrw.datalogger.IUserAdministration.createUser (String *username,* String *password*) throws AuthenticationException

Creates a new User.

Parameters

username	
password	

Returns

The new User.

Implemented in ilearnrw.datalogger.UserStore.

7.16.2.4 boolean ilearnrw.datalogger.IUserAdministration.deleteUser (String *username*) throws AuthenticationException

Implemented in ilearnrw.datalogger.UserStore.

7.16.2.5 List<User> ilearnrw.datalogger.IUserAdministration.getAllUsers ( ) throws AuthenticationException

Implemented in ilearnrw.datalogger.UserStore.

7.16.2.6 boolean ilearnrw.datalogger.IUserAdministration.isAuthenticated ( )

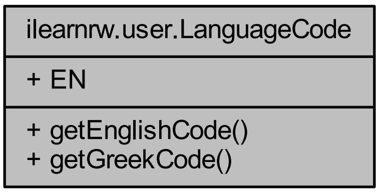
Implemented in ilearnrw.datalogger.UserStore.

7.16.2.7 User ilearnrw.datalogger.IUserAdministration.updatePassword (String *username,* String *newPassword*) throws AuthenticationException

Implemented in ilearnrw.datalogger.UserStore.

7.17 ilearnrw.user.LanguageCode Enum Reference

Collaboration diagram for ilearnrw.user.LanguageCode:



Static Public Member Functions

- static byte getEnglishCode ()
- static byte getGreekCode ()

Public Attributes

- EN

7.17.1 Member Function Documentation

7.17.1.1 static byte ilearnrw.user.LanguageCode.getEnglishCode ( ) [static]

7.17.1.2 static byte ilearnrw.user.LanguageCode.getGreekCode ( ) [static]

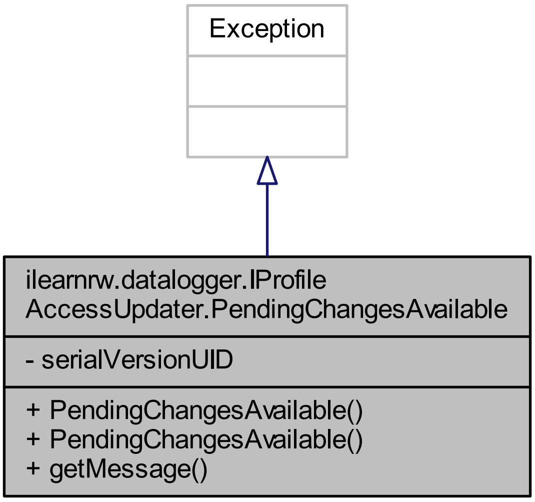
7.17.2 Member Data Documentation

7.17.2.1 ilearnrw.user.LanguageCode.EN

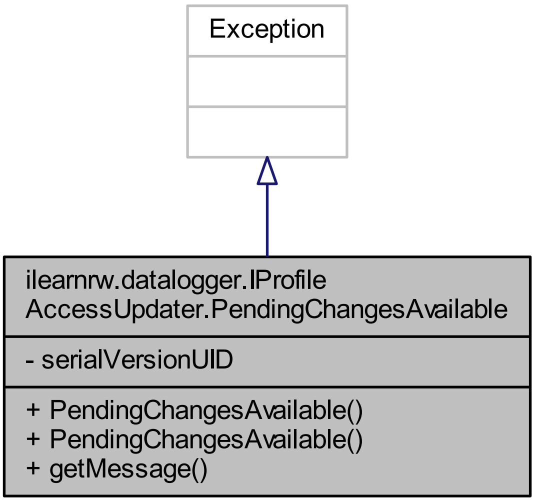
7.18 ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable Class Reference

Indicates that an action cannot be performed due to pending changes.

Inheritance diagram for ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable:


Exception
ilearnrw.datalogger.IProfile
AccessUpdater.PendingChangesAvailable
- serialVersionUID
+ PendingChangesAvailable()
+ PendingChangesAvailable()
+ getMessage()

Collaboration diagram for ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable:


Exception
ilearnrw.datalogger.IProfile
AccessUpdater.PendingChangesAvailable
- serialVersionUID
+ PendingChangesAvailable()
+ PendingChangesAvailable()
+ getMessage()

Public Member Functions

- PendingChangesAvailable (String message)
- PendingChangesAvailable (String message, Throwable throwable)
- String getMessage ()

Static Private Attributes

- static final long serialVersionUID = 1L

7.18.1 Detailed Description

Indicates that an action cannot be performed due to pending changes.

If the action was to be performed the pending changes would be lost.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable.PendingChangesAvailable (String *message*)

7.18.2.2 ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable.PendingChangesAvailable (String *message,* Throwable *throwable*)

7.18.3 Member Function Documentation

7.18.3.1 String ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable.getMessage ( )

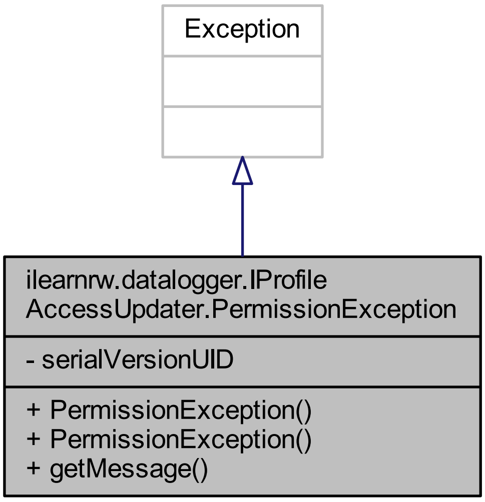
7.18.4 Member Data Documentation

7.18.4.1 final long ilearnrw.datalogger.IProfileAccessUpdater.PendingChangesAvailable.serialVersionUID = 1L `[static]`, `[private]`

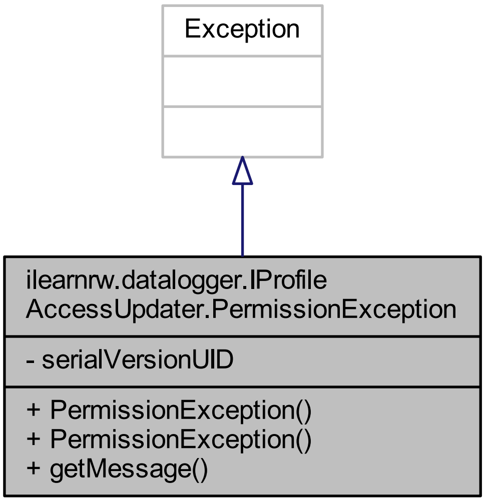
7.19 ilearnrw.datalogger.IProfileAccessUpdater.PermissionException Class Reference

The marker does not have permission to perform the requested change.

Inheritance diagram for ilearnrw.datalogger.IProfileAccessUpdater.PermissionException:



Collaboration diagram for ilearnrw.datalogger.IProfileAccessUpdater.PermissionException:



Public Member Functions

- PermissionException (String message)
- PermissionException (String message, Throwable throwable)
- String getMessage ()

Static Private Attributes

- static final long serialVersionUID = 1L

7.19.1 Detailed Description

The marker does not have permission to perform the requested change.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 ilearnrw.datalogger.IProfileAccessUpdater.PermissionException.PermissionException (String *message*)

7.19.2.2 ilearnrw.datalogger.IProfileAccessUpdater.PermissionException.PermissionException (String *message,* Throwable *throwable*)

7.19.3 Member Function Documentation

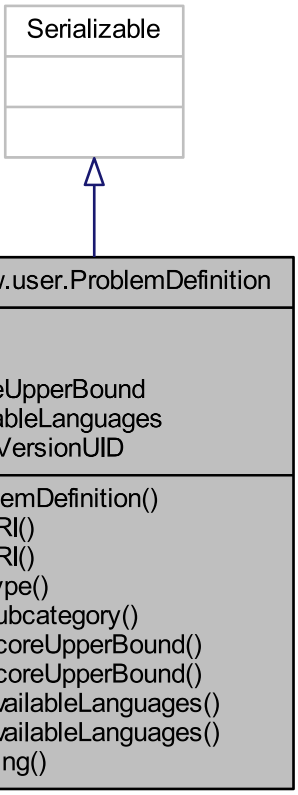
7.19.3.1 String ilearnrw.datalogger.IProfileAccessUpdater.PermissionException.getMessage ( )

7.19.4 Member Data Documentation

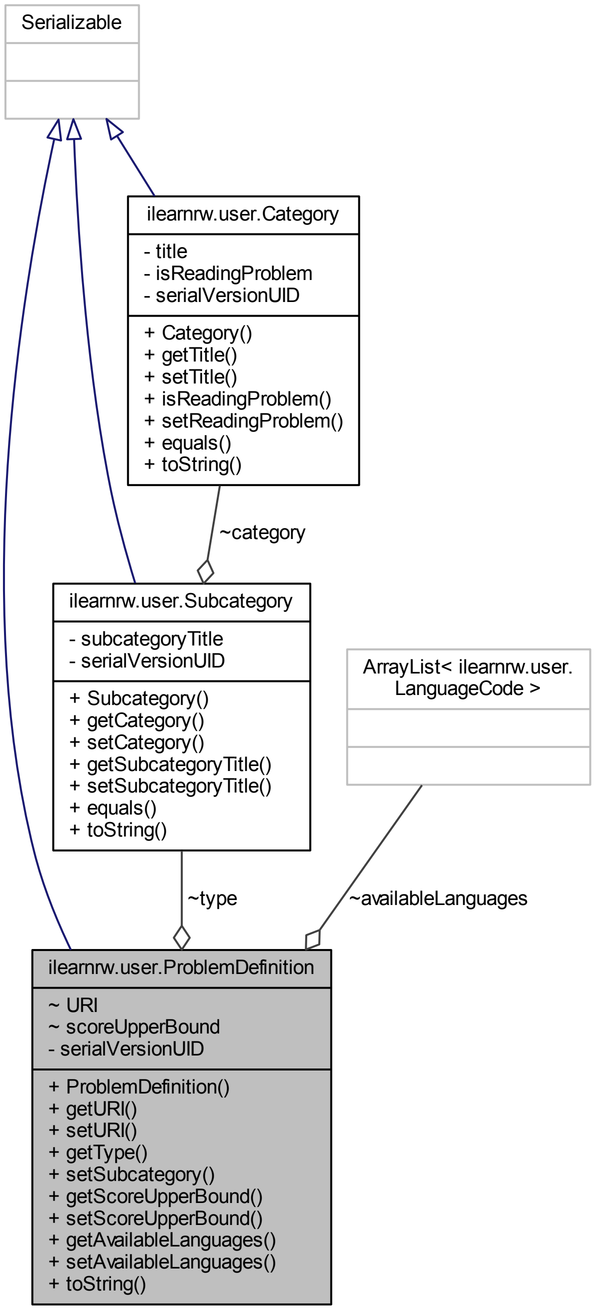
7.19.4.1 final long ilearnrw.datalogger.IProfileAccessUpdater.PermissionException.serialVersionUID = 1L [static], [private]

7.20 ilearnrw.user.ProblemDefinition Class Reference

Inheritance diagram for ilearnrw.user.ProblemDefinition:



Collaboration diagram for ilearnrw.user.ProblemDefinition:



Public Member Functions

- ProblemDefinition (String URI, Subcategory type, int scoreUpperBound, ArrayList< LanguageCode > availableLanguages)
- String getURI ()
- void setURI (String uRI)
- Subcategory getType ()
- void setSubcategory (Subcategory type)

- int getScoreUpperBound ()
- void setScoreUpperBound (int scoreUpperBound)
- ArrayList< LanguageCode > getAvailableLanguages ()
- void setAvailableLanguages (ArrayList< LanguageCode > availableLanguages)
- String toString ()

Package Attributes

- String URI
- Subcategory type
- int scoreUpperBound
- ArrayList< LanguageCode > availableLanguages

Static Private Attributes

- static final long serialVersionUID = 1L

7.20.1 Constructor & Destructor Documentation

7.20.1.1 ilearnrw.user.ProblemDefinition.ProblemDefinition (String *URI,* Subcategory *type,* int *scoreUpperBound,* ArrayList< LanguageCode > *availableLanguages*)

7.20.2 Member Function Documentation

7.20.2.1 ArrayList<LanguageCode> ilearnrw.user.ProblemDefinition.getAvailableLanguages ( )

7.20.2.2 int ilearnrw.user.ProblemDefinition.getScoreUpperBound ( )

7.20.2.3 Subcategory ilearnrw.user.ProblemDefinition.getType ( )

7.20.2.4 String ilearnrw.user.ProblemDefinition.getURI ( )

7.20.2.5 void ilearnrw.user.ProblemDefinition.setAvailableLanguages (ArrayList< LanguageCode > *availableLanguages*)

7.20.2.6 void ilearnrw.user.ProblemDefinition.setScoreUpperBound (int *scoreUpperBound*)

7.20.2.7 void ilearnrw.user.ProblemDefinition.setSubcategory (Subcategory *type*)

7.20.2.8 void ilearnrw.user.ProblemDefinition.setURI (String *uRI*)

7.20.2.9 String ilearnrw.user.ProblemDefinition.toString ( )

7.20.3 Member Data Documentation

7.20.3.1 ArrayList<LanguageCode> ilearnrw.user.ProblemDefinition.availableLanguages [package]

7.20.3.2 int ilearnrw.user.ProblemDefinition.scoreUpperBound [package]

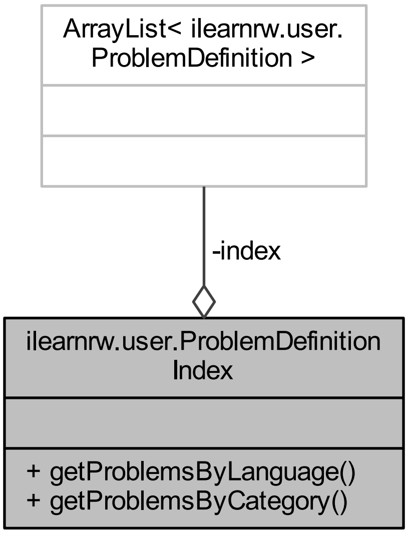
7.20.3.3 final long ilearnrw.user.ProblemDefinition.serialVersionUID = 1L [static], [private]

7.20.3.4 Subcategory ilearnrw.user.ProblemDefinition.type [package]

7.20.3.5 String ilearnrw.user.ProblemDefinition.URI [package]

7.21 ilearnrw.user.ProblemDefinitionIndex Class Reference

Collaboration diagram for ilearnrw.user.ProblemDefinitionIndex:


ArrayList< ilearnrw.user.
ProblemDefinition >
-index
ilearnrw.user.ProblemDefinition
Index
+ getProblemsByLanguage()
+ getProblemsByCategory()

Public Member Functions

- ArrayList< ProblemDefinition > getProblemsByLanguage (LanguageCode x)
- ArrayList< ProblemDefinition > getProblemsByCategory (Category x)

Private Attributes

- ArrayList< ProblemDefinition > index

7.21.1 Member Function Documentation

7.21.1.1 ArrayList<**ProblemDefinition**> ilearnrw.user.ProblemDefinitionIndex.getProblemsByCategory (**Category** *x*)

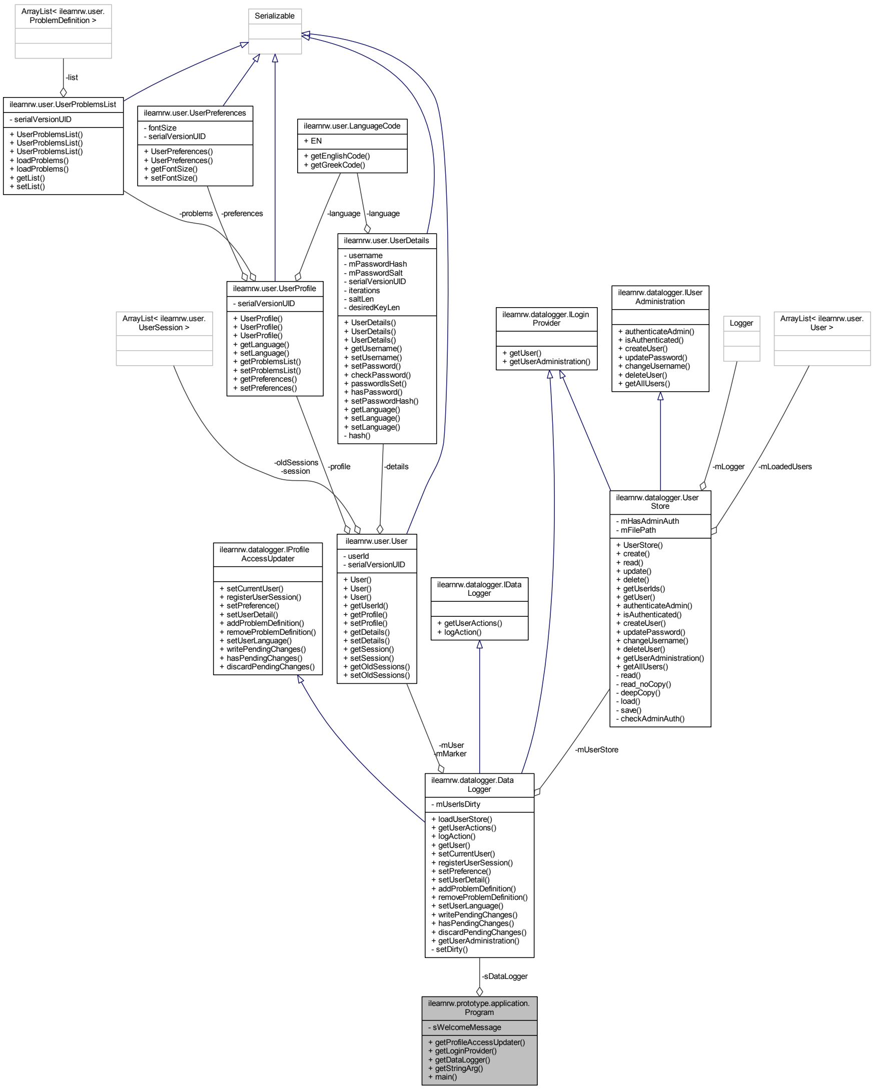
7.21.1.2 ArrayList<**ProblemDefinition**> ilearnrw.user.ProblemDefinitionIndex.getProblemsByLanguage (**LanguageCode** *x*)

7.21.2 Member Data Documentation

7.21.2.1 ArrayList<**ProblemDefinition**> ilearnrw.user.ProblemDefinitionIndex.index `[private]`

7.22 ilearnrw.prototype.application.Program Class Reference

Collaboration diagram for ilearnrw.prototype.application.Program:



Static Public Member Functions

- static IProfileAccessUpdater getProfileAccessUpdater ()
- static ILoginProvider getLoginProvider ()
- static IDataLogger getDataLogger ()
- static String getStringArg (String name, String[ ] args) throws Exception
- static void main (String[ ] args)

Static Private Attributes

- static DataLogger sDataLogger = new DataLogger()
- static String sWelcomeMessage

7.22.1 Member Function Documentation

7.22.1.1 static **IDataLogger** ilearnrw.prototype.application.Program.getDataLogger ( ) [static]

7.22.1.2 static **ILoginProvider** ilearnrw.prototype.application.Program.getLoginProvider ( ) [static]

7.22.1.3 static **IProfileAccessUpdater** ilearnrw.prototype.application.Program.getProfileAccessUpdater ( ) [static]

7.22.1.4 static String ilearnrw.prototype.application.Program.getStringArg (String *name,* String[ ] *args*) throws Exception [static]

7.22.1.5 static void ilearnrw.prototype.application.Program.main (String[ ] *args*) [static]

Parameters

args	Commandline arguments.

Commandline arguments:

–db <filePath> Database file to use (required)

7.22.2 Member Data Documentation

7.22.2.1 **DataLogger** ilearnrw.prototype.application.Program.sDataLogger = new **DataLogger()** [static], [private]

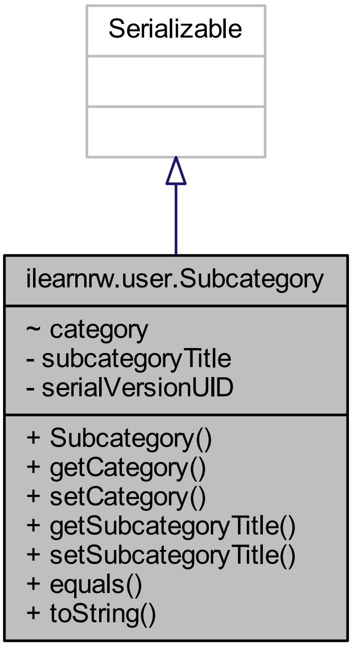
7.22.2.2 String ilearnrw.prototype.application.Program.sWelcomeMessage [static], [private]

Initial value:

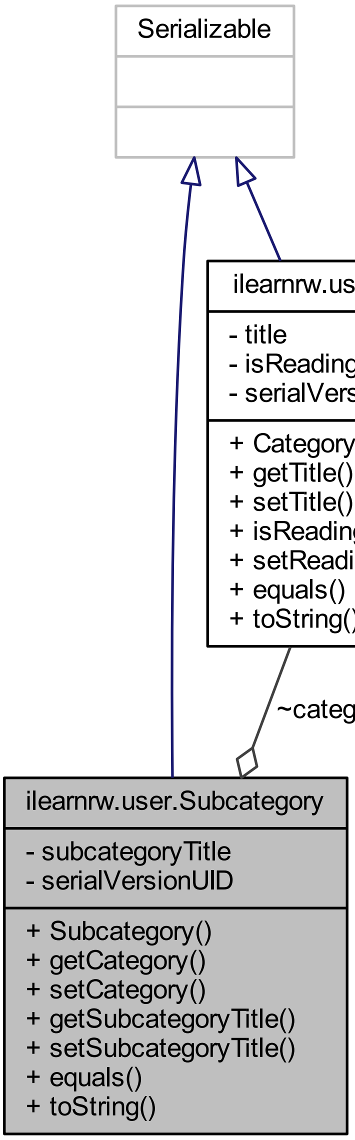
```
= "Welcome to the iLearnRW prototype application\n"
                                            + "\n"
                                            + "Developer notes:\n"
                                            + "\tUsernames are not unique.\n"
                                            + "\tUsernames cannot contain spaces.\n"
                                            + "\n"
```

7.23 ilearnrw.user.Subcategory Class Reference

Inheritance diagram for ilearnrw.user.Subcategory:



Collaboration diagram for ilearnrw.user.Subcategory:



Public Member Functions

- Subcategory (String subcategoryTitle, Category category)
- Category getCategory ()
- void setCategory (Category category)
- String getSubcategoryTitle ()
- void setSubcategoryTitle (String subcategoryTitle)
- boolean equals (Object obj)
- String toString ()

Package Attributes

- Category category

Private Attributes

- String subcategoryTitle

Static Private Attributes

- static final long serialVersionUID = 1L

7.23.1 Constructor & Destructor Documentation

7.23.1.1 ilearnrw.user.Subcategory.Subcategory (String *subcategoryTitle,* Category *category*)

7.23.2 Member Function Documentation

7.23.2.1 boolean ilearnrw.user.Subcategory.equals (Object *obj*)

7.23.2.2 Category ilearnrw.user.Subcategory.getCategory ( )

7.23.2.3 String ilearnrw.user.Subcategory.getSubcategoryTitle ( )

7.23.2.4 void ilearnrw.user.Subcategory.setCategory (Category *category*)

7.23.2.5 void ilearnrw.user.Subcategory.setSubcategoryTitle (String *subcategoryTitle*)

7.23.2.6 String ilearnrw.user.Subcategory.toString ( )

7.23.3 Member Data Documentation

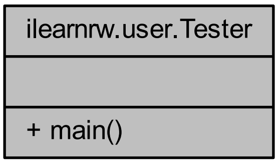
7.23.3.1 Category ilearnrw.user.Subcategory.category `[package]`

7.23.3.2 final long ilearnrw.user.Subcategory.serialVersionUID = 1L `[static], [private]`

7.23.3.3 String ilearnrw.user.Subcategory.subcategoryTitle `[private]`

7.24 ilearnrw.user.Tester Class Reference

Collaboration diagram for ilearnrw.user.Tester:



Static Public Member Functions

- static void main (String[] args)

7.24.1 Member Function Documentation

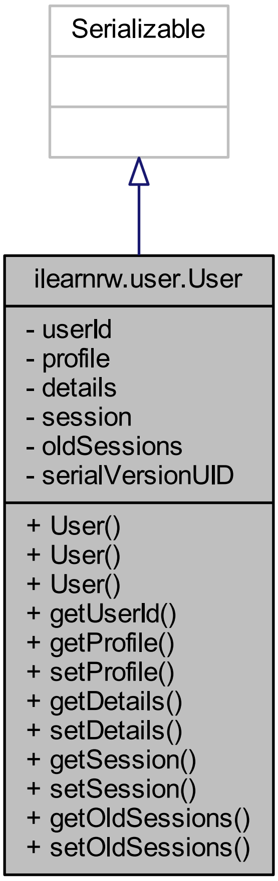
7.24.1.1 static void ilearnrw.user.Tester.main (String[] *args*) [static]

Parameters

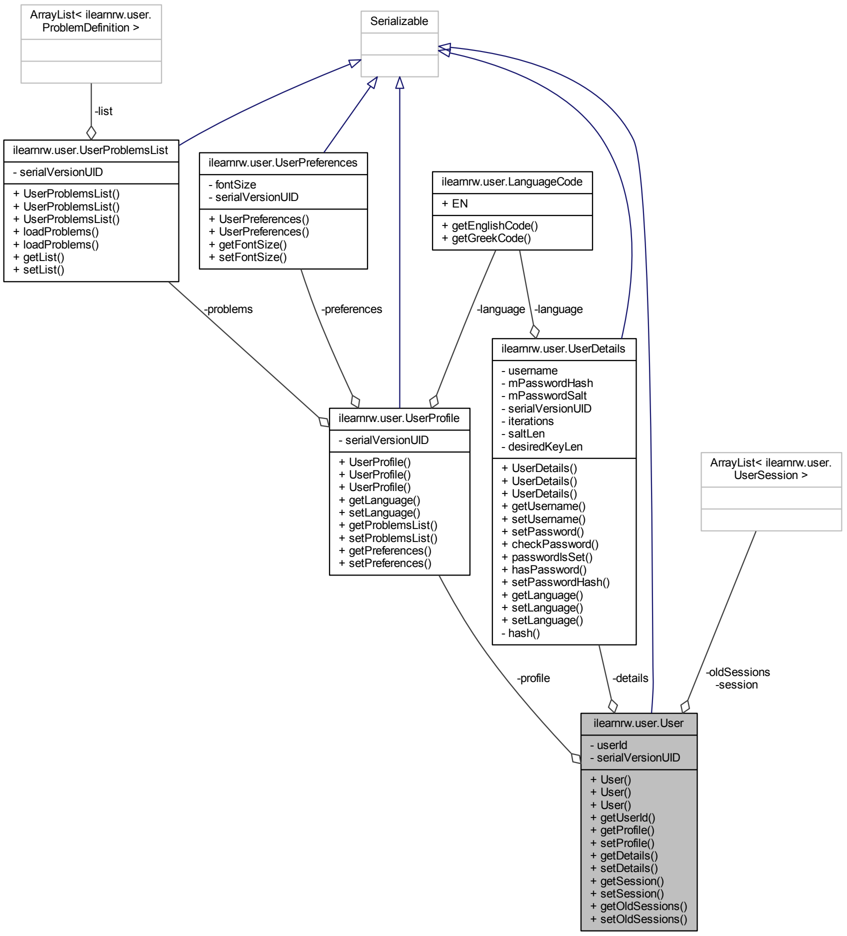
args	

7.25 ilearnrw.user.User Class Reference

Inheritance diagram for ilearnrw.user.User:



Collaboration diagram for ilearnrw.user.User:



Public Member Functions

- User (int userId, UserProfile profile, UserDetails details, ArrayList< UserSession > oldSessions)
- User (LanguageCode lc)
- User (int userId)
- int getUserId ()
- UserProfile getProfile ()
- void setProfile (UserProfile profile)
- UserDetails getDetails ()
- void setDetails (UserDetails details)
- ArrayList< UserSession > getSession ()
- void setSession (ArrayList< UserSession > session)
- ArrayList< UserSession > getOldSessions ()
- void setOldSessions (ArrayList< UserSession > oldSessions)

Private Attributes

- int userId
- UserProfile profile
- UserDetails details

- ArrayList< UserSession > session
- ArrayList< UserSession > oldSessions

Static Private Attributes

- static final long serialVersionUID = 1L

7.25.1 Constructor & Destructor Documentation

7.25.1.1 ilearnrw.user.User.User (int *userId,* UserProfile *profile,* UserDetails *details,* ArrayList< UserSession > *oldSessions*)

7.25.1.2 ilearnrw.user.User.User (LanguageCode *lc*)

7.25.1.3 ilearnrw.user.User.User (int *userId*)

7.25.2 Member Function Documentation

7.25.2.1 UserDetails ilearnrw.user.User.getDetails ( )

7.25.2.2 ArrayList<UserSession> ilearnrw.user.User.getOldSessions ( )

7.25.2.3 UserProfile ilearnrw.user.User.getProfile ( )

7.25.2.4 ArrayList<UserSession> ilearnrw.user.User.getSession ( )

7.25.2.5 int ilearnrw.user.User.getUserId ( )

7.25.2.6 void ilearnrw.user.User.setDetails (UserDetails *details*)

7.25.2.7 void ilearnrw.user.User.setOldSessions (ArrayList< UserSession > *oldSessions*)

7.25.2.8 void ilearnrw.user.User.setProfile (UserProfile *profile*)

7.25.2.9 void ilearnrw.user.User.setSession (ArrayList< UserSession > *session*)

7.25.3 Member Data Documentation

7.25.3.1 UserDetails ilearnrw.user.User.details [private]

7.25.3.2 ArrayList<UserSession> ilearnrw.user.User.oldSessions [private]

7.25.3.3 UserProfile ilearnrw.user.User.profile [private]

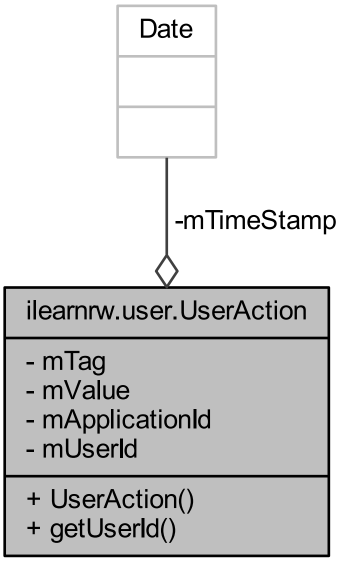
7.25.3.4 final long ilearnrw.user.User.serialVersionUID = 1L [static], [private]

7.25.3.5 ArrayList<UserSession> ilearnrw.user.User.session [private]

7.25.3.6 int ilearnrw.user.User.userId [private]

7.26 ilearnrw.user.UserAction Class Reference

Collaboration diagram for ilearnrw.user.UserAction:



Public Member Functions

- UserAction (String tag, String value, String applicationId, int userId)
- int getUserId ()

Private Attributes

- String mTag
- String mValue
- String mApplicationId
- int mUserId
- Date mTimeStamp

7.26.1 Detailed Description

Todo Add some documentation about this class.

Add get/setters

7.26.2 Constructor & Destructor Documentation

7.26.2.1 ilearnrw.user.UserAction.UserAction (String *tag*, String *value*, String *applicationId*, int *userId*)

Default constructor used by applications when logging a user action.

Parameters

tag	
value	
applicationId	
userId	

7.26.3 Member Function Documentation

7.26.3.1 int ilearnrw.user.UserAction.getUserId ( )

7.26.4 Member Data Documentation

7.26.4.1 String ilearnrw.user.UserAction.mApplicationId [private]

7.26.4.2 String ilearnrw.user.UserAction.mTag [private]

7.26.4.3 Date ilearnrw.user.UserAction.mTimeStamp [private]

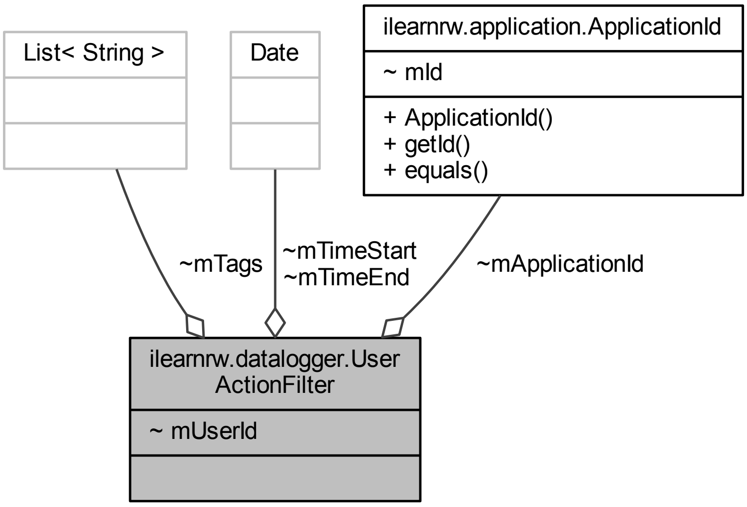
Not sure about the datatype Date. Does it contain the time as well? Might have to change.

7.26.4.4 int ilearnrw.user.UserAction.mUserId [private]

7.26.4.5 String ilearnrw.user.UserAction.mValue [private]

7.27 ilearnrw.datalogger.UserActionFilter Class Reference

Collaboration diagram for ilearnrw.datalogger.UserActionFilter:



Package Attributes

- int mUserId
- ApplicationId mApplicationId
- List< String > mTags
- Date mTimeStart
- Date mTimeEnd

7.27.1 Detailed Description

The UserActionFilter is used to query the datalogger for information

Todo Add getters/setters, appropriate functions etc.

7.27.2 Member Data Documentation

7.27.2.1 ApplicationId ilearnrw.datalogger.UserActionFilter.mApplicationId [package]

7.27.2.2 List<String> ilearnrw.datalogger.UserActionFilter.mTags [package]

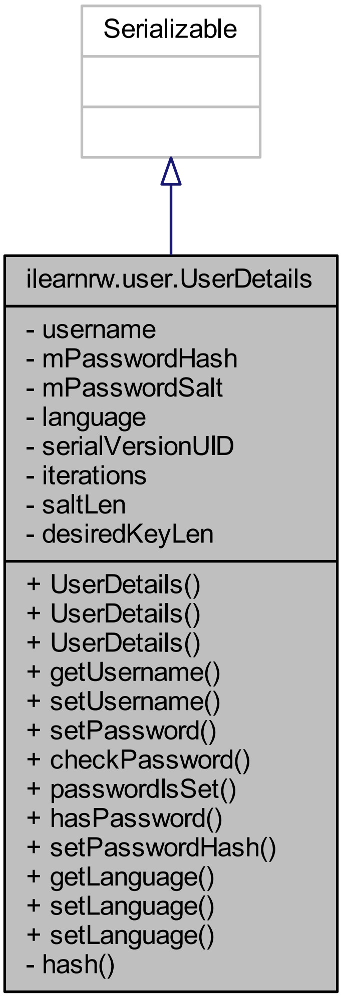
7.27.2.3 Date ilearnrw.datalogger.UserActionFilter.mTimeEnd [package]

7.27.2.4 Date ilearnrw.datalogger.UserActionFilter.mTimeStart [package]

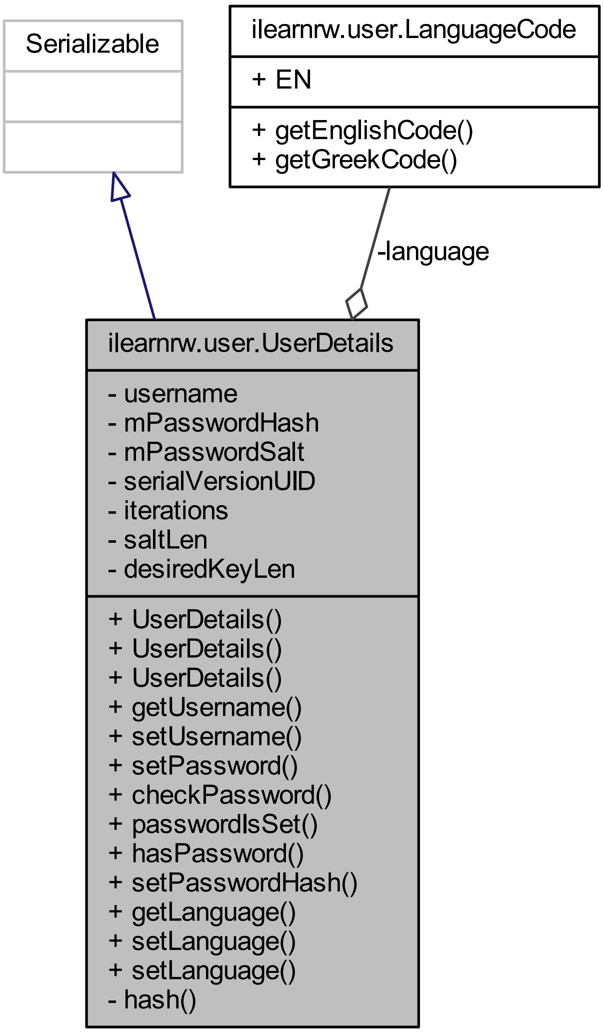
7.27.2.5 int ilearnrw.datalogger.UserActionFilter.mUserId [package]

7.28 ilearnrw.user.UserDetails Class Reference

Inheritance diagram for ilearnrw.user.UserDetails:



Collaboration diagram for ilearnrw.user.UserDetails:



Public Member Functions

- UserDetails (String username, int password, LanguageCode language)
- UserDetails (LanguageCode language)
- UserDetails (int userId)
- String getUsername ()
- void setUsername (String username)
- final void setPassword (String plaintextPassword)
- final boolean checkPassword (String plaintextPassword)
- boolean passwordIsSet ()
- boolean hasPassword ()
- final void setPasswordHash (byte[] hash, byte[] salt)
- LanguageCode getLanguage ()
- void setLanguage (LanguageCode language)
- void setLanguage (byte languageCode)

Static Private Member Functions

- static byte[] hash (String plaintextPassword, byte[] salt) throws NoSuchAlgorithmException, InvalidKeySpecException

Private Attributes

- String username
- byte[] mPasswordHash

  Stores hash(plainTextPassword + mPasswordSalt)
- byte[] mPasswordSalt

  Stores the salt used to hash the password.
- LanguageCode language

Static Private Attributes

- static final long serialVersionUID = 1L
- static final int iterations = 10∗1024
- static final int saltLen = 32
- static final int desiredKeyLen = 256

7.28.1 Constructor & Destructor Documentation

7.28.1.1 ilearnrw.user.UserDetails.UserDetails (String *username,* int *password,* LanguageCode *language*)

7.28.1.2 ilearnrw.user.UserDetails.UserDetails (LanguageCode *language*)

7.28.1.3 ilearnrw.user.UserDetails.UserDetails (int *userId*)

7.28.2 Member Function Documentation

7.28.2.1 final boolean ilearnrw.user.UserDetails.checkPassword (String *plaintextPassword*)

7.28.2.2 LanguageCode ilearnrw.user.UserDetails.getLanguage ( )

7.28.2.3 String ilearnrw.user.UserDetails.getUsername ( )

7.28.2.4 static byte [] ilearnrw.user.UserDetails.hash (String *plaintextPassword,* byte[] *salt*) throws NoSuchAlgorithmException, InvalidKeySpecException `[static], [private]`

7.28.2.5 boolean ilearnrw.user.UserDetails.hasPassword ( )

7.28.2.6 boolean ilearnrw.user.UserDetails.passwordIsSet ( )

7.28.2.7 void ilearnrw.user.UserDetails.setLanguage (LanguageCode *language*)

Parameters

the	language code as a enum item

7.28.2.8 void ilearnrw.user.UserDetails.setLanguage (byte *languageCode*)

Parameters

the	language code as a enum item

7.28.2.9 final void ilearnrw.user.UserDetails.setPassword (String *plaintextPassword*)

7.28.2.10 final void ilearnrw.user.UserDetails.setPasswordHash (byte[] *hash,* byte[] *salt*)

7.28.2.11 void ilearnrw.user.UserDetails.setUsername (String *username*)

7.28.3 Member Data Documentation

7.28.3.1 final int ilearnrw.user.UserDetails.desiredKeyLen = 256 `[static], [private]`

7.28.3.2 final int ilearnrw.user.UserDetails.iterations = 10∗1024 `[static], [private]`

7.28.3.3 LanguageCode ilearnrw.user.UserDetails.language `[private]`

7.28.3.4 byte [] ilearnrw.user.UserDetails.mPasswordHash `[private]`

Stores hash(plainTextPassword + mPasswordSalt)

Note

In the future we might want to change this to store the value in a base64 encoded String;

7.28.3.5 byte [] ilearnrw.user.UserDetails.mPasswordSalt `[private]`

Stores the salt used to hash the password.

Note

In the future we might want to change this to store the value in a base64 encoded String;

7.28.3.6 final int ilearnrw.user.UserDetails.saltLen = 32 `[static], [private]`

7.28.3.7 final long ilearnrw.user.UserDetails.serialVersionUID = 1L `[static], [private]`

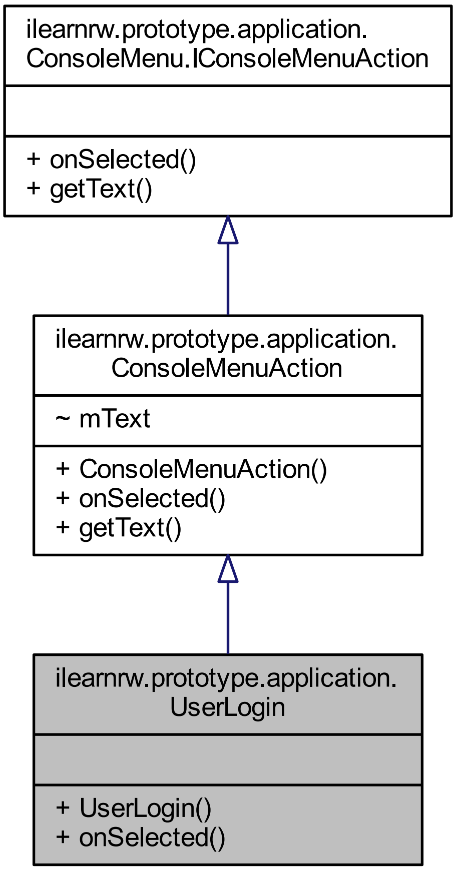
7.28.3.8 String ilearnrw.user.UserDetails.username `[private]`

Todo Something to think about is whether this should be case-sensitive and if unicode should be supported. Unicode can create some headaches, there was a blog post about this recently (of which i only read the headline) see: `http://labs.spotify.com/2013/06/18/creative-usernames/`

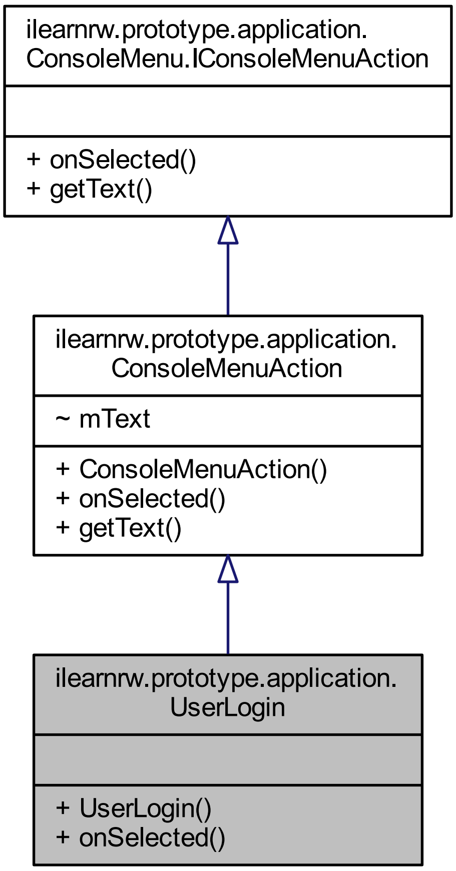
7.29 ilearnrw.prototype.application.UserLogin Class Reference

Main menu of a logged in user.

Inheritance diagram for ilearnrw.prototype.application.UserLogin:



Collaboration diagram for ilearnrw.prototype.application.UserLogin:



Public Member Functions

- UserLogin (String text)
- EConsoleMenuActionResult onSelected (ConsoleMenu menu)

Additional Inherited Members

7.29.1 Detailed Description

Main menu of a logged in user.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 ilearnrw.prototype.application.UserLogin.UserLogin (String *text*)

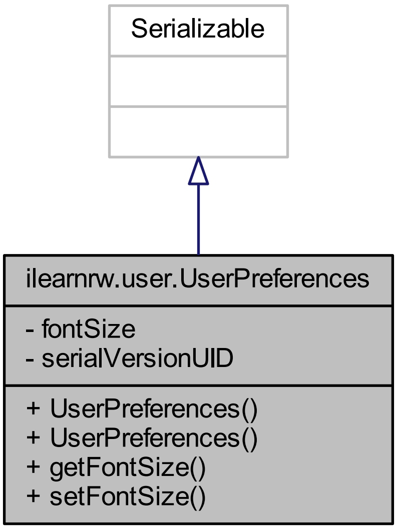
7.29.3 Member Function Documentation

7.29.3.1 EConsoleMenuActionResult ilearnrw.prototype.application.UserLogin.onSelected (ConsoleMenu *menu*)

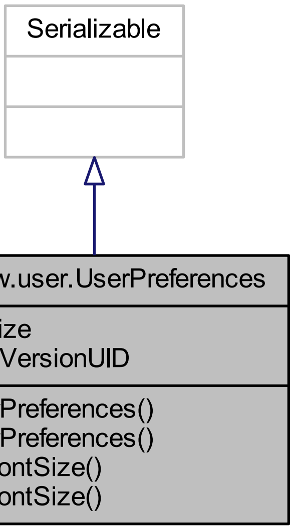
Implements ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.

7.30 ilearnrw.user.UserPreferences Class Reference

Inheritance diagram for ilearnrw.user.UserPreferences:



Collaboration diagram for ilearnrw.user.UserPreferences:



Public Member Functions

- UserPreferences ()
- UserPreferences (int fontSize)
- int getFontSize ()
- void setFontSize (int fontSize)

Private Attributes

- int fontSize

Static Private Attributes

- static final long serialVersionUID = 1L

7.30.1 Constructor & Destructor Documentation

7.30.1.1 ilearnrw.user.UserPreferences.UserPreferences ( )

7.30.1.2 ilearnrw.user.UserPreferences.UserPreferences (int *fontSize*)

7.30.2 Member Function Documentation

7.30.2.1 int ilearnrw.user.UserPreferences.getFontSize ( )

7.30.2.2 void ilearnrw.user.UserPreferences.setFontSize (int *fontSize*)

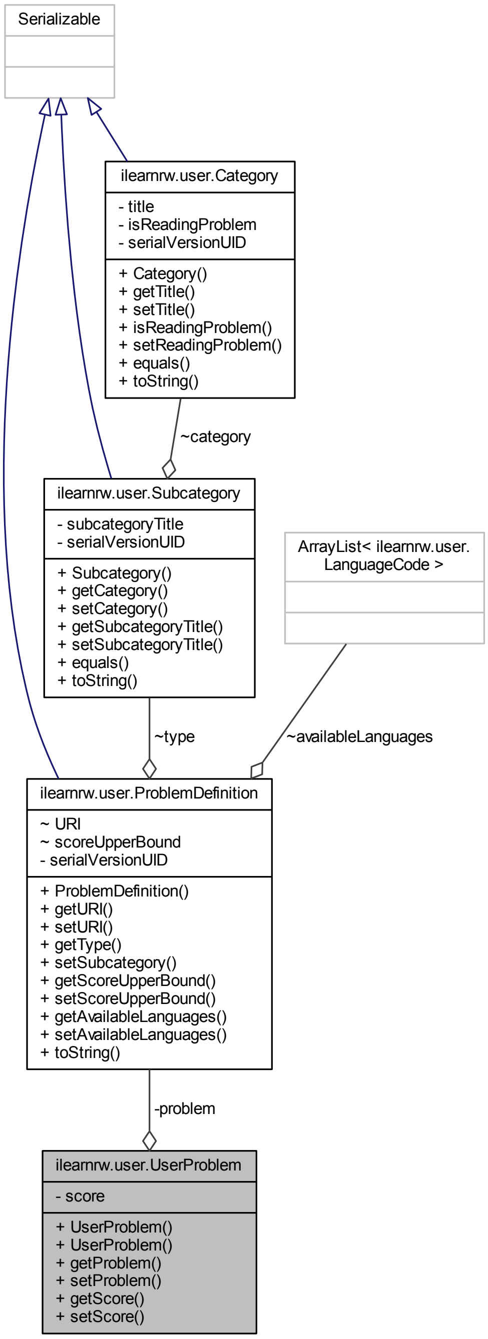
7.30.3 Member Data Documentation

7.30.3.1 int ilearnrw.user.UserPreferences.fontSize [private]

7.30.3.2 final long ilearnrw.user.UserPreferences.serialVersionUID = 1L [static], [private]

7.31 ilearnrw.user.UserProblem Class Reference

Collaboration diagram for ilearnrw.user.UserProblem:



Public Member Functions

- UserProblem (ProblemDefinition problem, int score)
- UserProblem (ProblemDefinition problem)
- ProblemDefinition getProblem ()
- void setProblem (ProblemDefinition problem)
- int getScore ()
- void setScore (int score)

Private Attributes

- ProblemDefinition problem
- int score

7.31.1 Constructor & Destructor Documentation

7.31.1.1 ilearnrw.user.UserProblem.UserProblem (**ProblemDefinition** *problem,* int *score*)

7.31.1.2 ilearnrw.user.UserProblem.UserProblem (**ProblemDefinition** *problem*)

7.31.2 Member Function Documentation

7.31.2.1 **ProblemDefinition** ilearnrw.user.UserProblem.getProblem ( )

Returns

the ProblemDefintion of the user's problem

7.31.2.2 int ilearnrw.user.UserProblem.getScore ( )

Returns

the frequency that the user does errors related to this problem

7.31.2.3 void ilearnrw.user.UserProblem.setProblem (**ProblemDefinition** *problem*)

7.31.2.4 void ilearnrw.user.UserProblem.setScore (int *score*)

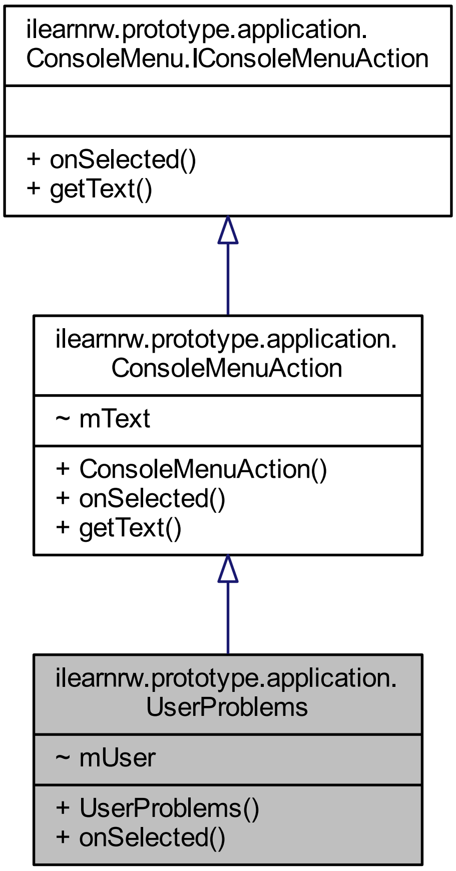
7.31.3 Member Data Documentation

7.31.3.1 **ProblemDefinition** ilearnrw.user.UserProblem.problem `[private]`

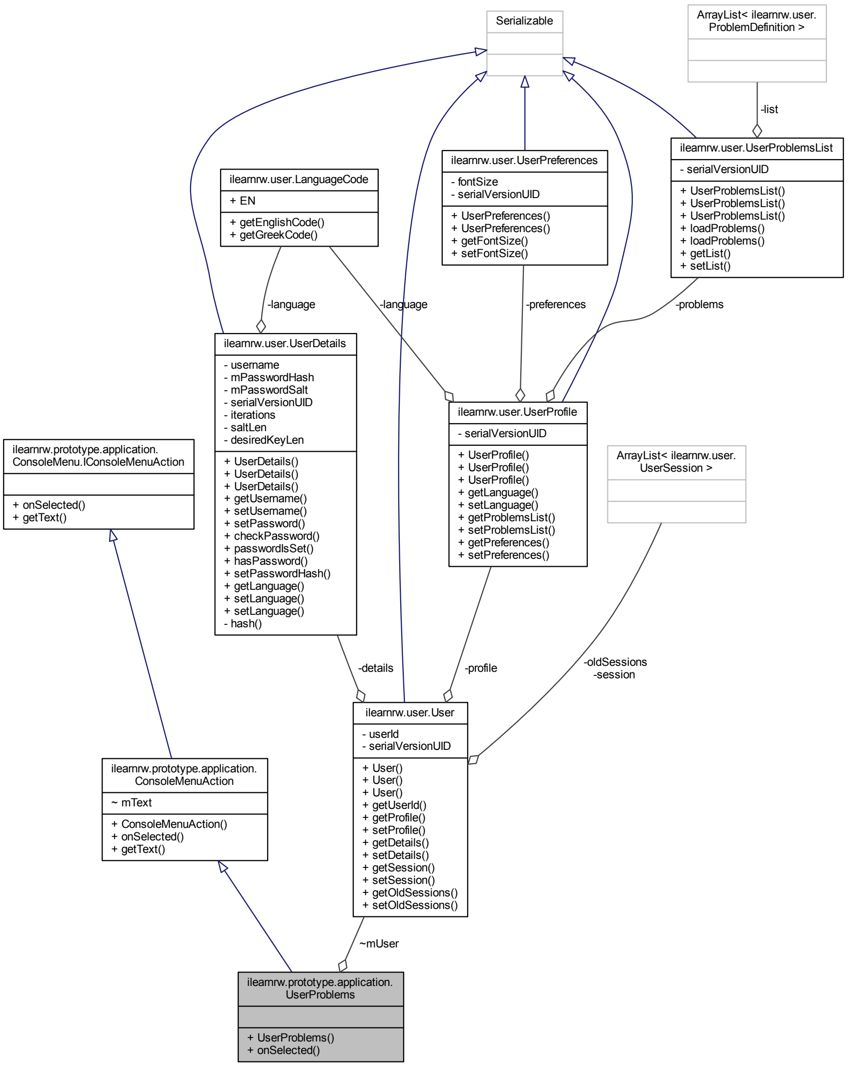
7.31.3.2 int ilearnrw.user.UserProblem.score `[private]`

7.32 ilearnrw.prototype.application.UserProblems Class Reference

Inheritance diagram for ilearnrw.prototype.application.UserProblems:



Collaboration diagram for ilearnrw.prototype.application.UserProblems:



Public Member Functions

- UserProblems (String text, User user)
- EConsoleMenuActionResult onSelected (ConsoleMenu menu)

Package Attributes

- User mUser = null

7.32.1 Constructor & Destructor Documentation

7.32.1.1 ilearnrw.prototype.application.UserProblems.UserProblems (String *text,* User *user*)

7.32.2 Member Function Documentation

7.32.2.1 EConsoleMenuActionResult ilearnrw.prototype.application.UserProblems.onSelected (ConsoleMenu *menu*)

Should display a menu where you can:

- List problem definitions
- Add problem definitions to a profile from the ProblemDefinitionIndex.
- Remove problem definitions

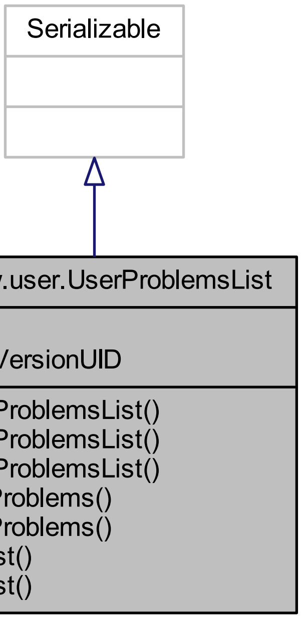
Implements ilearnrw.prototype.application.ConsoleMenu.IConsoleMenuAction.

7.32.3 Member Data Documentation

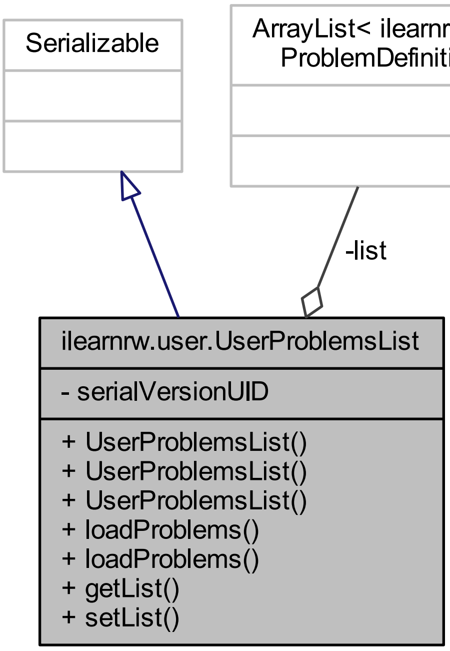
7.32.3.1 User ilearnrw.prototype.application.UserProblems.mUser = null `[package]`

7.33 ilearnrw.user.UserProblemsList Class Reference

Inheritance diagram for ilearnrw.user.UserProblemsList:



Collaboration diagram for ilearnrw.user.UserProblemsList:



Public Member Functions

- UserProblemsList (int userId)
- UserProblemsList (LanguageCode lc)
- UserProblemsList (ArrayList< ProblemDefinition > list)
- void loadProblems (LanguageCode lc)
- void loadProblems (int userId)
- ArrayList< ProblemDefinition > getList ()
- void setList (ArrayList< ProblemDefinition > list)

Private Attributes

- ArrayList< ProblemDefinition > list

Static Private Attributes

- static final long serialVersionUID = 1L

7.33.1 Constructor & Destructor Documentation

7.33.1.1 ilearnrw.user.UserProblemsList.UserProblemsList (int *userId*)

7.33.1.2 ilearnrw.user.UserProblemsList.UserProblemsList (LanguageCode *lc*)

7.33.1.3 ilearnrw.user.UserProblemsList.UserProblemsList (ArrayList< ProblemDefinition > *list*)

7.33.2 Member Function Documentation

7.33.2.1 ArrayList<ProblemDefinition> ilearnrw.user.UserProblemsList.getList ( )

7.33.2.2 void ilearnrw.user.UserProblemsList.loadProblems (LanguageCode *lc*)

7.33.2.3 void ilearnrw.user.UserProblemsList.loadProblems (int *userId*)

7.33.2.4 void ilearnrw.user.UserProblemsList.setList (ArrayList< ProblemDefinition > *list*)

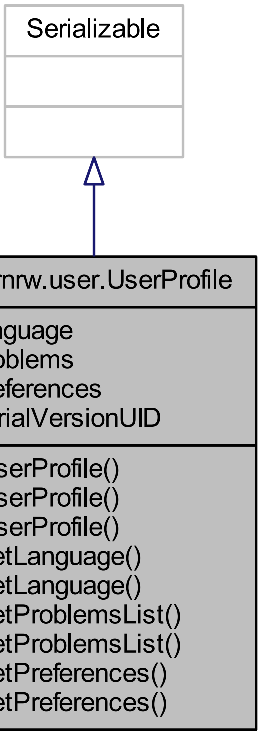
7.33.3 Member Data Documentation

7.33.3.1 ArrayList<ProblemDefinition> ilearnrw.user.UserProblemsList.list [private]

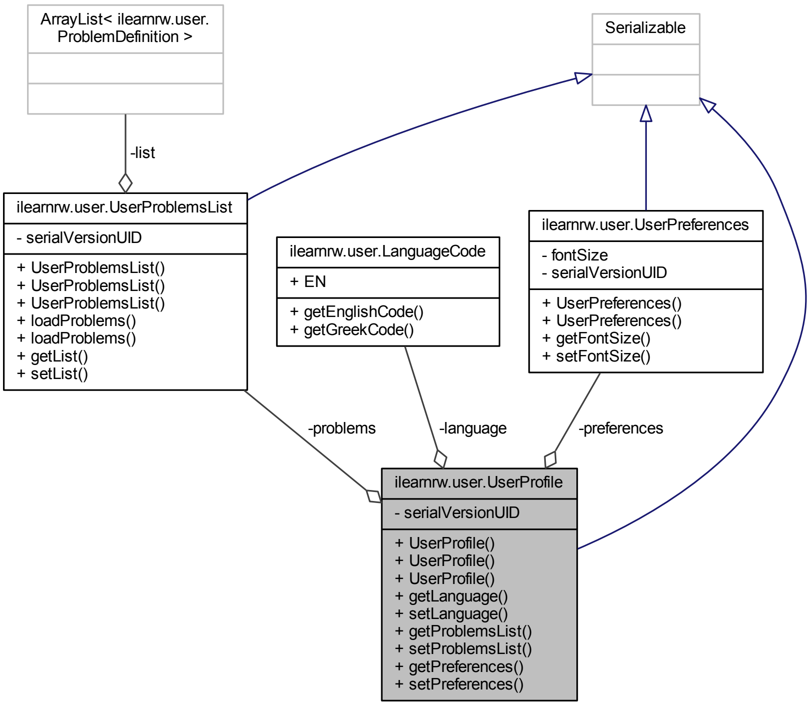
7.33.3.2 final long ilearnrw.user.UserProblemsList.serialVersionUID = 1L [static], [private]

7.34 ilearnrw.user.UserProfile Class Reference

Inheritance diagram for ilearnrw.user.UserProfile:



Collaboration diagram for ilearnrw.user.UserProfile:



Public Member Functions

- UserProfile (LanguageCode language, UserProblemsList problems, UserPreferences preferences)
- UserProfile (LanguageCode language)
- UserProfile (int userId)
- LanguageCode getLanguage ()
- void setLanguage (LanguageCode language)
- UserProblemsList getProblemsList ()
- void setProblemsList (UserProblemsList problems)
- UserPreferences getPreferences ()
- void setPreferences (UserPreferences preferences)

Private Attributes

- LanguageCode language
- UserProblemsList problems
- UserPreferences preferences

Static Private Attributes

- static final long serialVersionUID = 1L

7.34.1 Constructor & Destructor Documentation

7.34.1.1 ilearnrw.user.UserProfile.UserProfile (**LanguageCode** *language,* **UserProblemsList** *problems,* **UserPreferences** *preferences*)

7.34.1.2 ilearnrw.user.UserProfile.UserProfile (**LanguageCode** *language*)

7.34.1.3 ilearnrw.user.UserProfile.UserProfile (int *userId*)

7.34.2 Member Function Documentation

7.34.2.1 **LanguageCode** ilearnrw.user.UserProfile.getLanguage ( )

7.34.2.2 **UserPreferences** ilearnrw.user.UserProfile.getPreferences ( )

7.34.2.3 **UserProblemsList** ilearnrw.user.UserProfile.getProblemsList ( )

7.34.2.4 void ilearnrw.user.UserProfile.setLanguage (**LanguageCode** *language*)

7.34.2.5 void ilearnrw.user.UserProfile.setPreferences (**UserPreferences** *preferences*)

7.34.2.6 void ilearnrw.user.UserProfile.setProblemsList (**UserProblemsList** *problems*)

7.34.3 Member Data Documentation

7.34.3.1 **LanguageCode** ilearnrw.user.UserProfile.language `[private]`

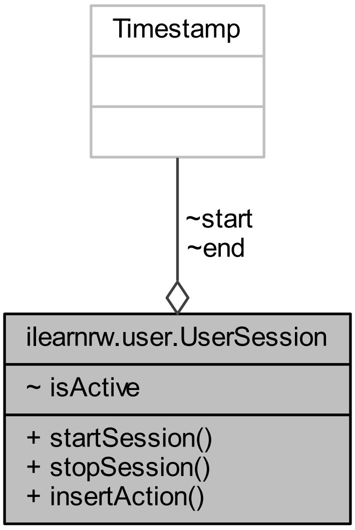
7.34.3.2 **UserPreferences** ilearnrw.user.UserProfile.preferences `[private]`

7.34.3.3 **UserProblemsList** ilearnrw.user.UserProfile.problems `[private]`

7.34.3.4 final long ilearnrw.user.UserProfile.serialVersionUID = 1L `[static]`, `[private]`

7.35 ilearnrw.user.UserSession Class Reference

Collaboration diagram for ilearnrw.user.UserSession:



Public Member Functions

- void startSession ()
- void stopSession ()
- void insertAction (UserAction x) throws IlearnException

Package Attributes

- boolean isActive
- Timestamp start
- Timestamp end

7.35.1 Detailed Description

Note

At the moment there will be issues if a user has 2 concurrent sessions. The reason for this is that the UserActions are not tied directly to a session. To fix: Add deviceid to session and actions.

7.35.2 Member Function Documentation

7.35.2.1 void ilearnrw.user.UserSession.insertAction (UserAction *x*) throws IlearnException

7.35.2.2 void ilearnrw.user.UserSession.startSession ( )

sets the timestamp and turns the boolean variable 'isActive' to true!

7.35.2.3 void ilearnrw.user.UserSession.stopSession ( )

sets the timestamp and turns the boolean variable ’isActive’ to true!

7.35.3 Member Data Documentation

7.35.3.1 Timestamp ilearnrw.user.UserSession.end [package]

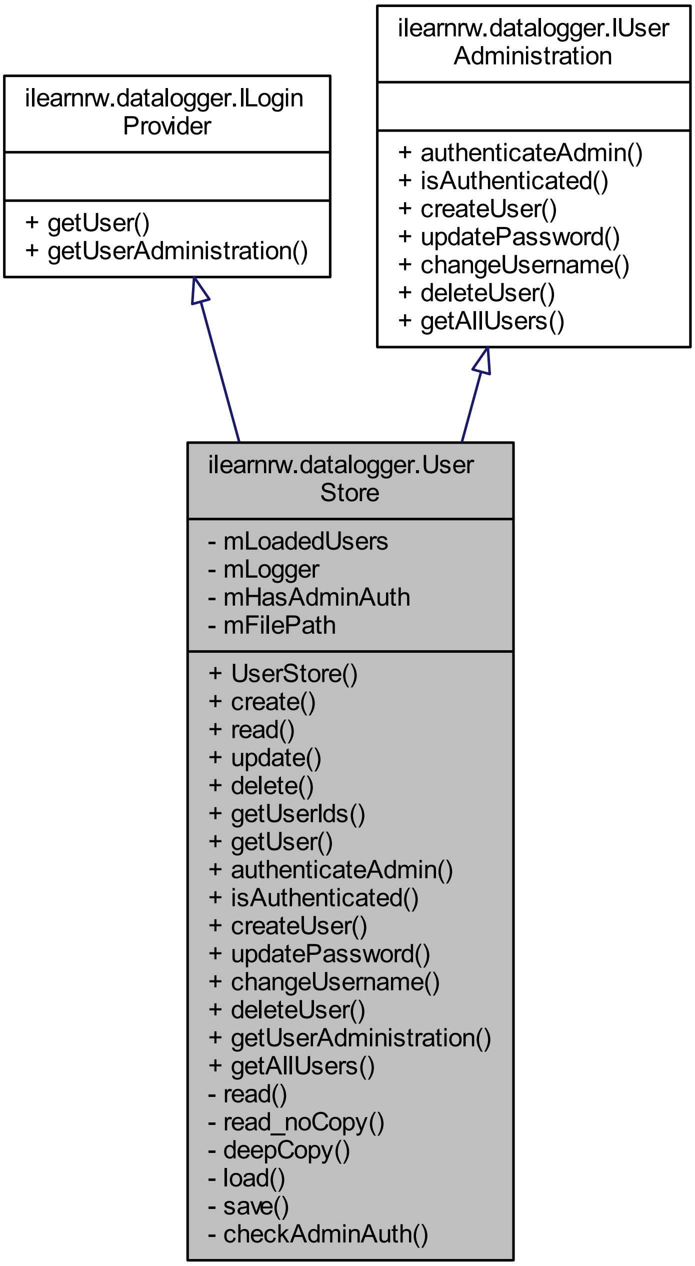
7.35.3.2 boolean ilearnrw.user.UserSession.isActive [package]

7.35.3.3 Timestamp ilearnrw.user.UserSession.start [package]

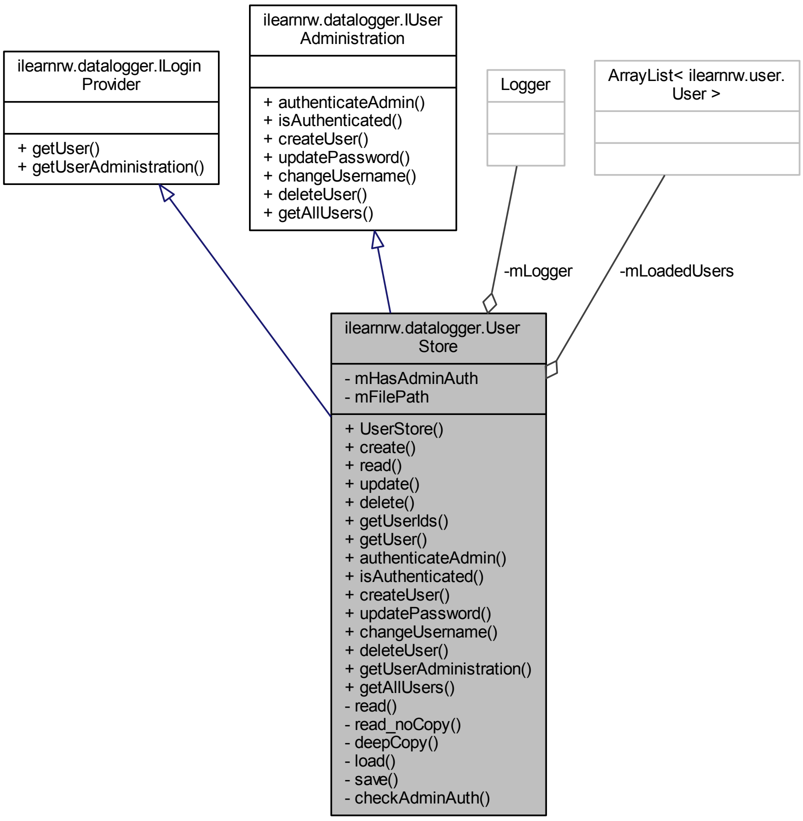
7.36 ilearnrw.datalogger.UserStore Class Reference

The UserStore is used to create and store user objects persistantly.

Inheritance diagram for ilearnrw.datalogger.UserStore:



Collaboration diagram for ilearnrw.datalogger.UserStore:



Public Member Functions

- UserStore (String filePath)
- User create ()
- User read (int userId)
- User update (User user)
- boolean delete (int userId)
- LinkedList< Integer > getUserIds ()
- User getUser (String username, String password)

  Authenticates a User.
- void authenticateAdmin (String adminPassword) throws AuthenticationException
- boolean isAuthenticated ()
- User createUser (String username, String password) throws AuthenticationException
- User updatePassword (String username, String newPassword) throws AuthenticationException
- User changeUsername (String oldUsername, String newUsername) throws AuthenticationException
- boolean deleteUser (String username) throws AuthenticationException
- IUserAdministration getUserAdministration ()
- List< User > getAllUsers () throws AuthenticationException

Private Member Functions

- User read (String username)

- User read_noCopy (int userId)
- User deepCopy (User user)
- void load ()

  Loads the serialized objects in `mFilePath` into mLoadedUsers.
- void save ()

  Serializes mLoadedUsers into `mFilePath`
- void checkAdminAuth () throws AuthenticationException

Private Attributes

- ArrayList< User > mLoadedUsers
- Logger mLogger
- boolean mHasAdminAuth = false

Static Private Attributes

- static String mFilePath

  The complete path to the file to read/write the user information from/to.

7.36.1 Detailed Description

The UserStore is used to create and store user objects persistantly.

Todo Replace storage backed with SQL database.

Enforce unique usernames.

Note

At the moment Java's built in serialization is used to store users to disk. This requires all objects to be stored to implement the Serializable interface.

Implements a CRUD interface for users and save/load functions to store the data persistently.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 ilearnrw.datalogger.UserStore.UserStore (String *filePath*)

Parameters

filePath	The full path to the file to use to store the user profiles etc.

7.36.3 Member Function Documentation

7.36.3.1 void ilearnrw.datalogger.UserStore.authenticateAdmin (String *adminPassword*) throws AuthenticationException

Authenticates an Administrator by requiring a password.

Parameters

adminPassword	The administrator password. This could be set in a configuration file.

Implements ilearnrw.datalogger.IUserAdministration.

7.36.3.2 User ilearnrw.datalogger.UserStore.changeUsername (String *oldUsername*, String *newUsername*) throws AuthenticationException

Implements ilearnrw.datalogger.IUserAdministration.

7.36.3.3 void ilearnrw.datalogger.UserStore.checkAdminAuth ( ) throws AuthenticationException `[private]`

7.36.3.4 User ilearnrw.datalogger.UserStore.create ( )

Create a new user.

Will generate a new UserId for the user.

Returns

A new User object.

7.36.3.5 User ilearnrw.datalogger.UserStore.createUser (String *username*, String *password*) throws AuthenticationException

Creates a new User.

Parameters

username	
password	

Returns

The new User.

Implements ilearnrw.datalogger.IUserAdministration.

7.36.3.6 User ilearnrw.datalogger.UserStore.deepCopy (User *user*) `[private]`

Returns a copy of a User object

This is called from functions returning a user in this class since we do not want to give out the actual reference stored in the _loadedUsers list.

7.36.3.7 boolean ilearnrw.datalogger.UserStore.delete (int *userId*)

Deletes a user from the store.

Parameters

userId	userId of the user to delete

Returns

true if successful.

7.36.3.8 boolean ilearnrw.datalogger.UserStore.deleteUser (String *username*) throws AuthenticationException

Implements ilearnrw.datalogger.IUserAdministration.

7.36.3.9 List<User> ilearnrw.datalogger.UserStore.getAllUsers ( ) throws AuthenticationException

Implements ilearnrw.datalogger.IUserAdministration.

7.36.3.10 User ilearnrw.datalogger.UserStore.getUser (String *username,* String *password*)

Authenticates a User.

Implements ilearnrw.datalogger.ILoginProvider.

7.36.3.11 IUserAdministration ilearnrw.datalogger.UserStore.getUserAdministration ( )

Implements ilearnrw.datalogger.ILoginProvider.

7.36.3.12 LinkedList<Integer> ilearnrw.datalogger.UserStore.getUserIds ( )

Returns

A list of all stored userid's.

7.36.3.13 boolean ilearnrw.datalogger.UserStore.isAuthenticated ( )

Implements ilearnrw.datalogger.IUserAdministration.

7.36.3.14 void ilearnrw.datalogger.UserStore.load ( ) `[private]`

Loads the serialized objects in `mFilePath` into mLoadedUsers.

7.36.3.15 User ilearnrw.datalogger.UserStore.read (int *userId*)

Reads a User from the UserStore

Parameters

userId	The userId of the user to load

Returns

The user with userId == userId

7.36.3.16 User ilearnrw.datalogger.UserStore.read (String *username*) `[private]`

7.36.3.17 User ilearnrw.datalogger.UserStore.read_noCopy (int *userId*) `[private]`

7.36.3.18 void ilearnrw.datalogger.UserStore.save ( ) `[private]`

Serializes mLoadedUsers into `mFilePath`

7.36.3.19 User ilearnrw.datalogger.UserStore.update (User *user*)

Updates the user currently stored with the one supplied.

Todo Don't allow multiple users with the same username.

Parameters

user	The new user copy to store.

Returns

A new copy of the user object.

7.36.3.20 User ilearnrw.datalogger.UserStore.updatePassword (String *username,* String *newPassword*) throws AuthenticationException

Implements ilearnrw.datalogger.IUserAdministration.

7.36.4 Member Data Documentation

7.36.4.1 String ilearnrw.datalogger.UserStore.mFilePath [static], [private]

The complete path to the file to read/write the user information from/to.

7.36.4.2 boolean ilearnrw.datalogger.UserStore.mHasAdminAuth = false [private]

7.36.4.3 ArrayList<User> ilearnrw.datalogger.UserStore.mLoadedUsers [private]

7.36.4.4 Logger ilearnrw.datalogger.UserStore.mLogger [private]

Index