

All Classes

Packages

- program
- program.common
- program.connection
- program.connection.listeners
- program.database
- program.ui.controllers
- program.ui.controllers.lecturer
- program.ui.controllers.student

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Package	Description
program	
program.common	
program.connection	
program.connection.listeners	
program.database	
program.ui.controllers	
program.ui.controllers.lecturer	
program.ui.controllers.student	
Skip navigation links	

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For All Packages

Package Hierarchies:

- [program](#),
- [program.common](#),
- [program.connection](#),
- [program.connection.listeners](#),
- [program.database](#),
- [program.ui.controllers](#),
- [program.ui.controllers.lecturer](#),
- [program.ui.controllers.student](#)

Class Hierarchy

- java.lang.Object
 - ◆ javafx.application.Application
 - ◇ program.[ClientMain](#)
 - ◆ program.connection.[ClientConnection](#) (implements java.lang.Runnable)
 - ◆ program.connection.listeners.[ClientListener](#) (implements java.lang.Runnable)
 - ◆ program.connection.[ClientsManager](#)
 - ◆ program.connection.[CommandsManager](#)
 - ◆ program.ui.controllers.lecturer.[CreateLectureController](#) (implements program.ui.controllers.[AppBinder](#))
 - ◆ program.database.[Database](#) (implements java.lang.AutoCloseable)
 - ◆ program.database.[DatabaseKey](#)
 - ◆ program.ui.controllers.lecturer.[LectureBoxController](#)
 - ◆ program.ui.controllers.lecturer.[LectureLoginController](#) (implements program.ui.controllers.[AppBinder](#))

Overview List

- ◆ [program.ui.controllers.lecturer.LectureOverviewController](#) (implements [program.ui.controllers.AppBinder](#), [program.connection.listeners.LectureReciever](#))
- ◆ [program.ui.controllers.lecturer.LectureReviewController](#) (implements [program.ui.controllers.AppBinder](#), [program.connection.listeners.LectureReciever](#))
- ◆ [program.ui.controllers.lecturer.LecturerWindowController](#) (implements [program.ui.controllers.AppBinder](#), [program.connection.listeners.QuestionReciever](#))
- ◆ [program.ui.controllers.student.LectureSelectorController](#) (implements [program.ui.controllers.AppBinder](#))
- ◆ [program.connection.listeners.LectureStatListener](#) (implements [java.lang.Runnable](#))
- ◆ [program.common.Question](#)
- ◆ [program.ui.controllers.QuestionBoxController](#)
- ◆ [program.ui.controllers.RoleSelectorController](#) (implements [program.ui.controllers.AppBinder](#))
- ◆ [program.ui.controllers.RootController](#) (implements [program.ui.controllers.AppBinder](#))
- ◆ [program.ServerMain](#)
- ◆ [program.connection.ServerManager](#)
- ◆ [program.connection.ServerRequests](#)
- ◆ [program.ui.controllers.student.StudentWindowController](#) (implements [program.ui.controllers.AppBinder](#), [program.connection.listeners.QuestionReciever](#))

Interface Hierarchy

- [program.ui.controllers.AppBinder](#)
- [program.connection.CommandsManager.Command](#)
- [program.connection.listeners.LectureReciever](#)
- [program.connection.listeners.QuestionReciever](#)

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)

- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

program

Class ClientMain

- java.lang.Object
- ◆ javafx.application.Application
- ◆ ◇ program.ClientMain

```
public class
extends javafx.application.Application
```

Main class that is instantiated to run the application

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

Nested Class Summary



Nested classes/interfaces inherited from class javafx.application.Application

javafx.application.Application.Parameters

Field Summary

Modifier and Type	Field and Description
-------------------	-----------------------

boolean

DEBUG

Fields inherited from class javafx.application.Application

STYLESHEET_CASPIAN, STYLESHEET_MODENA

Constructor Summary

Constructor and Description

ClientMain()

Method Summary

Modifier and Type	Method and Description
-------------------	------------------------

void

displayAlert(java.lang.String title,
java.lang.String header,
java.lang.String content)

This method is used to alert the user of something that happened, with a popup window.

AppBindergetActiveController()

Method used to get the controller that is currently active in the main window of the root window.

java.lang.String

getClassID()

getter to get the class code the client is connected to.

int

getLectureID()

getter to get the lecture ID the client is connected to

java.lang.String

getLectureName()

getter to get the name of the lecture the client is connected

Overview List

	to
long	<u>getLostMeTimerLenght()</u> Method used to get the length of the timer to controll the cooldown of "you lost me" feature
<u>RootController</u>	<u>getRootController()</u> Getter to get the controller of the root window.
<u>ServerManager</u>	<u>getServerManager()</u> Getter to get the servermanager.
void	<u>loadUI(java.lang.String path)</u> This method loads the view related to the fxml file in parameter, and sets it in the center window of our root window.
static void	<u>main(java.lang.String[] args)</u> Launches the application
void	<u>setClassID(java.lang.String classID)</u> Sets the class code the client is connected to.
void	<u>setLectureID(int lectureID)</u> sets the ID of the lecture the client is connected to.
void	<u>setLectureName(java.lang.String lectureName)</u> sets the name of the lecture the client is connected to.
void	<u>start(javafx.stage.Stage primaryStage)</u> <u>startConnection(java.lang.String serverAdress)</u> This method sets up our ServerManager object that is responsible for managing all requests and incoming info from the server.
void	<u>stop()</u>
javafx.scene.control.Alert	<u>viewAlert()</u>



Methods inherited from class javafx.application.Application

getHostServices, getParameters, getUserAgentStylesheet, init, launch, launch, notifyPreloader, setUserAgentStylesheet



Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



Field Detail

◆ DEBUG

public final boolean DEBUG

See Also:

Constant Field Values



Constructor Detail

◇ ClientMain

```
public ClientMain()
```



Method Detail

◇ startConnection

```
public void startConnection(java.lang.String serverAdress)
```

This method sets up our ServerManager object that is responsible for managing all requests and incoming info from the server.

Parameters:

serverAdress - String that is the IP address of the server

◇ main

```
public static void main(java.lang.String[] args)
```

Launches the application

Parameters:

args -

◇ displayAlert

```
public void displayAlert(java.lang.String title,  
                          java.lang.String header,  
                          java.lang.String content)
```

This method is used to alert the user of something that happened, with a popup window. In our case it's used, for instance, when the lecturer has lost a threshold of students during the lecture.

Parameters:

title - Title of the window that appears

header - Title of the content in the window

content - Text that appears in the window informing the user what happened

◇ loadUI

```
public void loadUI(java.lang.String path)
```

This method loads the view related to the fxml file in parameter, and sets it in the center window of our root window.

Parameters:

path - Relative url to the fxml file

Overview List

◇ start

```
public void start(javafx.stage.Stage primaryStage)
               throws java.lang.Exception
```

Specified by:

start in class `javafx.application.Application`

Throws:

`java.lang.Exception`

◇ stop

```
public void stop()
               throws java.lang.Exception
```

Overrides:

stop in class `javafx.application.Application`

Throws:

`java.lang.Exception`

◇ getServerManager

```
public ServerManager getServerManager()
```

Getter to get the servermanager.

Returns:

manager to handle requests and info from server.

◇ setClassID

```
public void setClassID(java.lang.String classID)
```

Sets the class code the client is connected to.

Parameters:

`classID` - class code for the subject the client is connected to

◇ getClassID

```
public java.lang.String getClassID()
```

getter to get the class code the client is connected to.

Returns:

Class code

◇ setLectureName

```
public void setLectureName(java.lang.String lectureName)
```

sets the name of the lecture the client is connected to.

Parameters:

`lectureName` - name of the lecture.

◇ getLectureName

```
public java.lang.String getLectureName()
```

getter to get the name of the lecture the client is connected to

Returns:

name of the lecture

Overview List

◇ **setLectureID**

```
public void setLectureID(int lectureID)
```

sets the ID of the lecture the client is connected to. This is used to know which lecture in the database info needs to be parsed to, as well as for the server to know which clients info and notifications should be sent to

Parameters:

lectureID - ID of the lecture

◇ **getLectureID**

```
public int getLectureID()
```

getter to get the lecture ID the client is connected to

Returns:

id of the lecture

◇ **viewAlert**

```
public javafx.scene.control.Alert viewAlert()
```

◇ **getRootController**

```
public RootController getRootController()
```

Getter to get the controller of the root window.

Returns:

root controller

◇ **getLostMeTimerLenght**

```
public long getLostMeTimerLenght()
```

Method used to get the length of the timer to controll the cooldown of "you lost me" feature

Returns:

lost-me timer

◇ **getActiveController**

```
public AppBinder getActiveController()
```

Method used to get the controller that is currently active in the main window of the root window. Used to make the back button function properly

Returns:

controller of active window.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)

Overview List

- [Index](#)
- [Help](#)
- [Prev Class](#)
- [Next Class](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)
- Summary:
 - [Nested |](#)
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)
- Detail:
 - [Field |](#)
 - [Constr |](#)
 - [Method](#)

program

Classes

- ClientMain
- ServerMain

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Package program

•	
Class	Description
ClientMain	Main class that is instantiated to run the application
ServerMain	The main program for running the server

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For Package program

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- java.lang.Object
 - ◆ javafx.application.Application
 - ◇ program.[ClientMain](#)
 - ◆ program.[ServerMain](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Package program

- | Package | Description |
|---|-------------|
| program.connection | |
| program.connection.listeners | |
| program.ui.controllers | |
| program.ui.controllers.lecturer | |
| program.ui.controllers.student | |
- | Class and Description |
|---|
| ServerMain
The main program for running the server |
- | Class and Description |
|--|
| ClientMain
Main class that is instantiated to run the application |
- | Class and Description |
|--|
| ClientMain
Main class that is instantiated to run the application |
- | Class and Description |
|-----------------------|
|-----------------------|

Overview List

ClientMain

Main class that is instantiated to run the application

-

Class and Description

ClientMain

Main class that is instantiated to run the application

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev
- Next

- Frames
- No Frames

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

program

Class ServerMain

- java.lang.Object
- ◆ program.ServerMain

```
public class
extends java.lang.Object
```

The main program for running the server

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen

- ◆ **Constructor Summary**

Constructor and Description

[ServerMain\(\)](#)



Method Summary

Modifier and Type	Method and Description
<u>ClientsManager</u>	<u>getClientsManager()</u> Getter for the servers clients-manager
<u>Database</u>	<u>getDatabase()</u> Getter for the database
static void	<u>main</u> (java.lang.String[] args) Launches the server



Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- ◆ **Constructor Detail**

◇ **ServerMain**

```
public ServerMain()
```



Method Detail

◇ **main**

```
public static void main(java.lang.String[] args)
```

Launches the server

Parameters:

args -

◇ **getDatabase**

```
public Database getDatabase()
```

Getter for the database

Returns:

returns the Database class the server is connected to

◇ **getClientsManager**

```
public ClientsManager getClientsManager()
```

Getter for the servers clients-manager

Returns:

Returns the clientManager class the server is using

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class program.ClientMain

- | Package | Description |
|---|-------------|
| program.connection.listeners | |
| program.ui.controllers | |
| program.ui.controllers.lecturer | |
| program.ui.controllers.student | |
- ◆

Uses of [ClientMain](#) in [program.connection.listeners](#)

Constructor and Description

```
ClientListener(ClientMain main, QuestionReciever controller)
LectureStatListener(ClientMain main, LectureReciever controller,
int listenerCounter)
```

- ◆

Uses of [ClientMain](#) in [program.ui.controllers](#)

Modifier and Type

Method and Description

```
void      RootController.setMainApp(ClientMain main)
void      RoleSelectorController.setMainApp(ClientMain main)
```

Overview List

	<code>AppBinder.setMainApp(ClientMain main)</code>
<code>void</code>	Sets up necessary references to client main and gets start info from the database, if any



Uses of [ClientMain](#) in [program.ui.controllers.lecturer](#)

Modifier and Type	Method and Description
<code>void</code>	<code>LecturerWindowController.setMainApp(ClientMain main)</code>
<code>void</code>	<code>LectureReviewController.setMainApp(ClientMain main)</code>
<code>void</code>	<code>LectureOverviewController.setMainApp(ClientMain main)</code>
<code>void</code>	<code>LectureLoginController.setMainApp(ClientMain main)</code>
<code>void</code>	<code>CreateLectureController.setMainApp(ClientMain main)</code>



Uses of [ClientMain](#) in [program.ui.controllers.student](#)

Modifier and Type	Method and Description
<code>void</code>	<code>StudentWindowController.setMainApp(ClientMain main)</code>
<code>void</code>	<code>LectureSelectorController.setMainApp(ClientMain main)</code>

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [Prev](#)
- [Next](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

Uses of Class program.ServerMain

- | Package | Description |
|------------------------------------|-------------|
| program.connection | |
- ♦

Uses of [ServerMain](#) in [program.connection](#)

Modifier and Type Field and Description

ServerMain	ClientsManager. main
----------------------------	--------------------------------------

Constructor and Description

[ClientsManager](#)(int portNumber, [ServerMain](#) main)

This constructor initializes the reference objects, and starts the thread that listens for new connections

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)

Overview List

- [Index](#)
- [Help](#)

- Prev
- Next

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

program.common

Classes

- Question

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Package program.common

- | Class | Description |
|--------------------------|--|
| Question | Question object used as a listener for the question list for both lecturer and student |

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For Package program.common

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- java.lang.Object
 - ◆ program.common.[Question](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Package program.common

-
- | Package | Description |
|--|-------------|
| program.ui.controllers | |
-
- | Class and Description |
|--------------------------|
| Question |

[Question](#)

Question object used as a listener for the question list for both lecturer and student

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.common

Class Question

- java.lang.Object
- ♦ program.common.Question

•

```
public class
extends java.lang.Object
```

Question object used as a listener for the question list for both lecturer and student

Since:

1.0

Version:

"%I%, %G%"

Author:

Anders Hunderi

◆ **Property Summary**

Type	Property and Description
<code>javafx.beans.property.IntegerProperty</code>	<u>rating</u> getter for the window element to update the rating number, whenever rating changes in the Question object.

◆ **Constructor Summary**

Constructor and Description
<u>Question</u> (int id, java.lang.String text, java.lang.String time, int rating)

◆ **Method Summary**

Modifier and Type	Method and Description
int	<u>getId()</u> getter to get ID of the question
java.lang.String	<u>getText()</u> getter to get the question text
int	<u>getRating()</u> getter for the rating of the question
javafx.scene.layout.AnchorPane	<u>getRelatedQuestionPane()</u> getter to get the window element of the question list
java.lang.String	<u>getTime()</u> getter to get the time of the question
javafx.beans.property.IntegerProperty	<u>ratingProperty()</u> getter for the window element to update the rating number in the Question object.
void	<u>setId(int id)</u> sets the id of the question
void	<u>setText(java.lang.String questionText)</u> sets the text of the question
void	<u>setRating(int rating)</u> sets the rating of the question
void	<u>setRelatedQuestionPane(javafx.scene.layout.AnchorPane relatedQuestionPane)</u> sets the window element related to this question
void	<u>setTime(java.lang.String time)</u> sets the time of the question

- ◇ **Methods inherited from class java.lang.Object**

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- ◇ **Property Detail**

- ◇ **rating**

`public javafx.beans.property.IntegerProperty ratingProperty`

getter for the window element to update the rating number, whenever rating changes in the Question object.

See Also:

[getRating\(\)](#), [setRating\(int\)](#)

- ◇ **Constructor Detail**

- ◇ **Question**

```
public Question(int id,
                java.lang.String text,
                java.lang.String time,
                int rating)
```

Parameters:

`id` - id of the question used in the database

`text` - actual question text

`time` - time a question was asked

`rating` - current rating of the question

- ◇ **Method Detail**

- ◇ **getId**

`public int getId()`

getter to get ID of the question

Returns:

id of the question used in the database

- ◇ **getText**

`public java.lang.String getText()`

getter to get the question text

Returns:

question text

Overview List

◇ **getTime**

```
public java.lang.String getTime()
```

getter to get the time of the question

Returns:

time of the question

◇ **getRelatedQuestionPane**

```
public javafx.scene.layout.AnchorPane getRelatedQuestionPane()
```

getter to get the window element of the question list

Returns:

window element of the question

◇ **setId**

```
public void setId(int id)
```

sets the id of the question

Parameters:

id - question id

◇ **setQuestionText**

```
public void setQuestionText(java.lang.String questionText)
```

sets the text of the question

Parameters:

questionText - question text

◇ **setTime**

```
public void setTime(java.lang.String time)
```

sets the time of the question

Parameters:

time - question time

◇ **setRelatedQuestionPane**

```
public void setRelatedQuestionPane(javafx.scene.layout.AnchorPane relatedQu
```

sets the window element related to this question

Parameters:

relatedQuestionPane - window element

◇ **setRating**

```
public void setRating(int rating)
```

sets the rating of the question

Parameters:

rating - rated score of the question

◇ **getRating**

```
public int getRating()
```

getter for the rating of the question

Returns:

rated score of the question

◇ **ratingProperty**

```
public javafx.beans.property.IntegerProperty ratingProperty()
```

getter for the window element to update the rating number, whenever rating changes in the Question object.

See Also:

[getRating\(\)](#), [setRating\(int\)](#)

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [Prev](#)
- [Next](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

Uses of Class

program.common.Question

- | Package | Description |
|--|---|
| program.ui.controllers | <ul style="list-style-type: none"> ♦ Uses of Question in program.ui.controllers |

Modifier and Type

Method and Description

void

QuestionBoxController.[setQuestion](#)([Question](#) question)
Sets the question object for this window.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [Prev](#)
- [Next](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- [Detail:](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

program.connection

Class ClientConnection

- [java.lang.Object](#)
 - [♦ program.connection.ClientConnection](#)
 - All Implemented Interfaces:
 [java.lang.Runnable](#)
-

```
public class
extends java.lang.Object
implements java.lang.Runnable
```

This class handles the serverside information of a single connected client

Since:

1.0

Version:

"%I%, %G%"

Author:
Erling Ihlen

- ◆ **Constructor Summary**

Constructor and Description

[ClientConnection](#)(java.net.Socket clientSocket, [CommandsManager](#) comManager, [ClientsManager](#) clientsManager)

This constructor initializes the reference objects and and the socket

- ◆ **Method Summary**

Modifier and Type	Method and Description
java.lang.String	<u>getClassID()</u> Getter for which subject the client is looking at
int	<u>getLectureID()</u> Getter for the id for a lecture this client is in
void	<u>run()</u>
void	<u>sendJSON</u> (org.json.JSONObject obj) Sends a given JSONObject to the client
void	<u>setClassID</u> (java.lang.String classID) Setter for which subject this client is looking at
void	<u>setLectureID</u> (int lectureID) Setter for which lecture this client is in
void	<u>setRole</u> (java.lang.String role) Sets a role to the client Possible roles (As of 26-04-2017):



Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- ◆ **Constructor Detail**

◇ **ClientConnection**

```
public ClientConnection(java.net.Socket clientSocket,
                        CommandsManager comManager,
                        ClientsManager clientsManager)
```

This constructor initializes the reference objects and and the socket
Parameters:

Overview List

`clientSocket` - The Socket the client is connected through
`comManager` - The CommandsManager that will handle `JSONObject` received from this client
`clientsManager` - The ClientsManager giving references to other connected clients



Method Detail

◇ **setRole**

```
public void setRole(java.lang.String role)
```

Sets a role to the client

Possible roles (As of 26-04-2017):

`student`

`lecturer`

Parameters:

`role` -

◇ **sendJSON**

```
public void sendJSON(org.json.JSONObject obj)
```

Sends a given `JSONObject` to the client

Parameters:

`obj` - The `JSONObject` to be sent to the client

◇ **setLectureID**

```
public void setLectureID(int lectureID)
```

Setter for which lecture this client is in

Parameters:

`lectureID` - A integer with the given id (Should be the same as in the database)

◇ **getLectureID**

```
public int getLectureID()
```

Getter for the id for a lecture this client is in

Returns:

`integer` thats the id for a lecture

◇ **setClassID**

```
public void setClassID(java.lang.String classID)
```

Setter for which subject this client is looking at

Parameters:

`classID` - A String giving a subject code in the database

Overview List

◇ getClassID

```
public java.lang.String getClassID()
```

Getter for which subject the client is looking at

Returns:

◇ run

```
public void run()
```

Specified by:

run in interface `java.lang.Runnable`

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- [Field](#) |
- [Constr](#) |
- [Method](#)

- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

program.connection

Class ClientsManager

- java.lang.Object
- ◆ program.connection.ClientsManager

•

```
public class
extends java.lang.Object
```

This class decodes all incoming JSONObjects and encodes and sends return-JSONs

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen

- ◆ **Field Summary**

Modifier and Type	Field and Description
<u>ServerMain</u>	<u>main</u>

- ◆ **Constructor Summary**

Constructor and Description
<u>ClientsManager</u> (int portNumber, <u>ServerMain</u> main) This constructor initializes the reference objects, and starts the thread that listens for new connections

- ◆ **Method Summary**

Modifier and Type	Method and Description
void	<u>addClientToCollection</u> (<u>ClientConnection</u> client) Adds a given ClientConnection to the list of connected clients
void	<u>addLecturerToLecture</u> (<u>ClientConnection</u> client, java.lang.String classID) This method adds a lecturer to a given lecture
boolean	<u>doesLectureExist</u> (java.lang.String classID) This method checks if a lecture is being held in a given class
java.util.Collection< <u>ClientConnection</u> >	<u>getClientsConnected</u> () Gets a complete list of all the clients connected to the server
<u>ClientConnection</u>	<u>getLecturer</u> (java.lang.String classID) Gets the connection of the lecturer holding a lecture on a given subject
void	<u>removeConnection</u> (<u>ClientConnection</u> client) Removes a given ClientConnection from the list of connected clients
void	<u>removeLecturerFromLecture</u> (<u>ClientConnection</u> client, java.lang.String classID) This method removes a lecturer from a given lecture
void	<u>sendInfoToLecturer</u> (org.json.JSONObject obj, java.lang.String classID) This method sends a given JSONObject to the lecturer on a given lecture



Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

◆ Field Detail

◆ main

```
public ServerMain main
```



Constructor Detail

◆ ClientsManager

```
public ClientsManager(int portNumber,  
                      ServerMain main)
```

This constructor initializes the reference objects, and starts the thread that listens for new connections

Parameters:

portNumber - An integer with the network port that the clients connect through

main - A ServerMain object giving a reference to the main server program



Method Detail

◆ getClientsConnected

```
public java.util.Collection<ClientConnection> getClientsConnected()
```

Gets a complete list of all the clients connected to the socket

Returns:

returns a Collection<ClientConnection> of all connected clients

◆ addLecturerToLecture

```
public void addLecturerToLecture(ClientConnection client,  
                                java.lang.String classID)
```

This method adds a lecturer to a given lecture

Parameters:

client - The connection of the lecturer to be added

classID - A String giving the subject code the lecture is happening in

◆ removeLecturerFromLecture

```
public void removeLecturerFromLecture(ClientConnection client)
```

This method removes a lecturer from a given lecturer

Parameters:

client - The connection of the lecturer to be removed

Overview List

- ◇ **removeConnection**

```
public void removeConnection(ClientConnection client)
```

Removes a given `ClientConnection` from the list of connected clients

Parameters:

`client` - The connection to be removed

- ◇ **addClientToCollection**

```
public void addClientToCollection(ClientConnection client)
```

Adds a given `ClientConnection` to the list of connected clients

Parameters:

`client` - The connection to be added

- ◇ **sendInfoToLecturer**

```
public void sendInfoToLecturer(org.json.JSONObject obj,  
                               java.lang.String classID)
```

This method sends a given `JSONObject` to the lecturer holding a given lecture

Parameters:

`obj` - The `JSONObject` that should be passed on to the lecturer

`classID` - A `String` giving the id of the subject the lecture is happening in

- ◇ **doesLectureExist**

```
public boolean doesLectureExist(java.lang.String classID)
```

This method checks if a lecture is being held in a given subject

Parameters:

`classID` - A `String` giving the subject that should be checked

Returns:

Returns `true` if a lecture is being held in the given subject

- ◇ **getLecturer**

```
public ClientConnection getLecturer(java.lang.String classID)
```

Gets the connection of the lecturer holding a lecture in a given subject

Parameters:

`classID` - A `String` giving the subject code for that should be checked for lecturers

Returns:

Returns a `ClientConnection` to the found lecturer

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)

Overview List

- [Index](#)
- [Help](#)
- [Prev Class](#)
- [Next Class](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)
- Summary:
 - Nested |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

program.connection

Interface CommandsManager.Command

- Enclosing class:
[CommandsManager](#)
-

```
public static interface
```

This interface is used to allows the stringToFunction map to refer to the different methods of this class

Method Summary

**Modifier and
Type**

Method and Description

Overview List

```
void doFunction(org.json.JSONObject obj,  
ClientConnection client)
```

- ◆

Method Detail

◆ doFunction

```
void doFunction(org.json.JSONObject obj,  
ClientConnection client)
```

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested](#) |
- [Field](#) |
- [Constr](#) |
- [Method](#)

- Detail:
- [Field](#) |
- [Constr](#) |
- [Method](#)

program.connection

Class CommandsManager

- java.lang.Object
- ◆ program.connection.CommandsManager

```
public class
extends java.lang.Object
```

This class decodes all incoming JSONObjects and encodes and sends return-JSONs

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Nested Class Summary**

Modifier and Type	Class and Description
static interface	<u>CommandsManager.Command</u> This interface is used to allows the stringToFunction map to refer to the different methods of this class

- ◆ **Constructor Summary**

Constructor and Description

CommandsManager (ClientsManager clientsManager)
This constructor sets up the reference elements

- ◆ **Method Summary**

Modifier and Type	Method and Description
void	<u>analyzeFunction</u> (org.json.JSONObject obj, <u>ClientConnection</u> client) This method analyzes a given JSONObject and passes it to the correct handler-method
void	<u>voteQuestion</u> (org.json.JSONObject obj, <u>ClientConnection</u> client) This method changes the score of a given question by a given increment, and informs all clients in the lecture The keys used by the JSONObject is as follows:

**Methods inherited from class java.lang.Object**

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ◆ **Constructor Detail**

◇ **CommandsManager**

```
public CommandsManager(ClientsManager clientsManager)
```

This constructor sets up the reference elements

Parameters:

clientsManager - The ClientsManager that allows this class to get references to the clients that are connected to the server

◆ Method Detail

◆ analyzeFunction

```
public void analyzeFunction(org.json.JSONObject obj,  
                           ClientConnection client)
```

This method analyzes a given JSONObject and passes it to the correct handler-method

Parameters:

obj - A JSONObject with data to be passed to the method. Only the key "Function" is used here

client - A ClientConnection referencing the client that sent the object

◆ voteQuestion

```
public void voteQuestion(org.json.JSONObject obj,  
                         ClientConnection client)
```

This method changes the score of a given question by a given increment, and informs all clients in the lecture

The keys used by the JSONObject is as follows:

```
"ScoreChange" = int
```

```
"QuestionID" = int
```

Parameters:

obj - A JSONObject containing data the method needs

client - A ClientConnection referencing the client that sent the command

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:

Overview List

- Nested |
 - Field |
 - Constr |
 - Method
-
- Detail:
 - Field |
 - Constr |
 - Method

program.connection

Interfaces

- CommandsManager.Command

Classes

- ClientConnection
- ClientsManager
- CommandsManager
- ServerManager
- ServerRequests

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Package program.connection

Interface		Description
<u>CommandsManager.Command</u>		This interface is used to allows the stringToFunction map to refer to the different methods of this class
Class		Description
<u>ClientConnection</u>		This class handles the serverside information of a single connected client
<u>ClientsManager</u>		This class decodes all incoming JSONObjects and encodes and sends return-JSONs
<u>CommandsManager</u>		This class decodes all incoming JSONObjects and encodes and sends return-JSONs
<u>ServerManager</u>		The main Class used to handle everything related to the server, from sending info and getting the streams used for listening for info from the server
<u>ServerRequests</u>		Class with static methods to return the most basic needs from the server

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)

Overview List

- [Help](#)
- [Prev Package](#)
- [Next Package](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For Package program.connection

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- java.lang.Object
 - ◆ program.connection.[ClientConnection](#) (implements java.lang.Runnable)
 - ◆ program.connection.[ClientsManager](#)
 - ◆ program.connection.[CommandsManager](#)
 - ◆ program.connection.[ServerManager](#)
 - ◆ program.connection.[ServerRequests](#)

Interface Hierarchy

- program.connection.[CommandsManager.Command](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)

Overview List

- [Help](#)
- [Prev](#)
- [Next](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Package program.connection

- | Package | Description |
|------------------------------------|-------------|
| program | |
| program.connection | |

Class and Description

[ClientsManager](#)

This class decodes all incoming JSONObjectS and encodes and sends return-JSONs

[ServerManager](#)

The main Class used to handle everything related to the server, from sending info and getting the streams used for listening for info from the server

Class and Description

[ClientConnection](#)

This class handles the serverside information of a single connected client

[ClientsManager](#)

This class decodes all incoming JSONObjectS and encodes and sends return-JSONs

[CommandsManager](#)

This class decodes all incoming JSONObjectS and encodes and sends return-JSONs

[ServerManager](#)

The main Class used to handle everything related to the server, from sending info and getting the

Overview List

streams used for listening for info from the server

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

program.connection

Class ServerManager

- java.lang.Object
- ◆ program.connection.ServerManager

•

```
public class
extends java.lang.Object
```

The main Class used to handle everything related to the server, from sending info and getting the streams used for listening for info from the server

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen

- ◆ **Constructor Summary**

Constructor and Description

ServerManager(java.lang.String serverAdress, int portNumber)
Starts the connection to the server

- ◆ **Method Summary**

Modifier and Type	Method and Description
java.io.BufferedReader	<u>getInputStream</u> () Getter to get the input stream of the socket
java.io.PrintWriter	<u>getOutputStream</u> () getter to get the output stream of the socket
java.net.Socket	<u>getSocket</u> () Getter to get the Socket object related to the server.
void	<u>sendJSON</u> (org.json.JSONObject obj) Method used to send a json object to the server.

◇

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait,
wait

- ◆ **Constructor Detail**

◇ **ServerManager**

```
public ServerManager(java.lang.String serverAdress,
                     int portNumber)
```

Starts the connection to the server

Parameters:

serverAdress - String that is the IP address

portNumber - the portnumber the server has open for the client

- ◆ **Method Detail**

Overview List

◇ **sendJSON**

```
public void sendJSON(org.json.JSONObject obj)
```

Method used to send a json object to the server. Makes the jsonobject into a string, which can be sent over socket

Parameters:

obj - The JSON object with all the info the client wants to send

◇ **getSocket**

```
public java.net.Socket getSocket()
```

Getter to get the Socket object related to the server.

Returns:

socket object connected to the server

◇ **getInputStream**

```
public java.io.BufferedReader getInputStream()
```

Getter to get the input stream of the socket

Returns:

input stream of the socket

◇ **getOutputStream**

```
public java.io.PrintWriter getOutputStream()
```

getter to get the output stream of the socket

Returns:

output stream of the socket.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:

Overview List

- Nested |
 - Field |
 - Constr |
 - Method
-
- Detail:
 - Field |
 - Constr |
 - Method

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.connection

Class ServerRequests

- [java.lang.Object](#)
- [◆ program.connection.ServerRequests](#)

•

```
public class
extends java.lang.Object
```

Class with static methods to return the most basic needs from the server
Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

Constructor and Description

[ServerRequests\(\)](#)

- ◆ **Method Summary**

Modifier and Type	Method and Description
static javafx.collections.ObservableList<java.lang.String>	<u>getAllSubjectCodes(ServerManager)</u> Static method used to get all the subject codes the client can connect to
static boolean	<u>serverHasLecture(ServerManager, java.lang.String classID)</u> static method to request from the server to see if a lecture is being held in request classID.



Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- ◆ **Constructor Detail**

◇ **ServerRequests**

```
public ServerRequests()
```

- ◆ **Method Detail**

◇ **getAllSubjectCodes**

```
public static javafx.collections.ObservableList<java.lang.String> getAllSubjectCodes()
```

Static method used to get all the subject codes the client can connect to

Parameters:

`sm` - The servermanager object handling the server connection

Returns:

a list of all the subject codes from the database

◇ **serverHasLecture**

```
public static boolean serverHasLecture(ServerManager sm, java.lang.String classID)
```

Overview List

static method to request from the server to see if a lecture is being held in requested classID. This is used as a check to see whether a client can proceed to a new view.

Parameters:

`sm` - the server manager handling the connection to the database

`classID` - the class code

Returns:

a boolean confirming if a lecture is being held with the suggested class code

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- Detail:
- [Field |](#)
- [Constr |](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.connection.ClientConnection

- | Package | Description |
|------------------------------------|-------------|
| program.connection | |
- ♦ **Uses of [ClientConnection](#) in [program.connection](#)**

Modifier and Type	Method and Description
ClientConnection	ClientsManager.getLecturer (java.lang.String classID) Gets the connection of the lecturer holding a lecture in a given subject

Modifier and Type	Method and Description
java.util.Collection< ClientConnection >	ClientsManager.getClientsConnected () Gets a complete list of all the clients connected to the socket

Modifier and Type	Method and Description
void	ClientsManager.addClientToCollection (ClientConnection client) Adds a given ClientConnection to the list of connected clients
void	ClientsManager.addLecturerToLecture (ClientConnection client, java.lang.String classID)

Overview List

This method adds a lecturer to a given lecture

`void CommandsManager.analyzeFunction(org.json.JSONObject obj, ClientConnection client)`

This method analyzes a given JSONObject and passes it to the correct handler-method

`void CommandsManager.Command.doFunction(org.json.JSONObject obj, ClientConnection client)`

`void ClientsManager.removeConnection(ClientConnection client)`

Removes a given ClientConnection from the list of connected clients

`void ClientsManager.removeLecturerFromLecture(ClientConnection client)`

This method removes a lecturer from a given lecturer

`void CommandsManager.voteQuestion(org.json.JSONObject obj, ClientConnection client)`

This method changes the score of a given question by a given increment, and informs all clients in the lecture

The keys used by the JSONObject is as follows:

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)


- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.connection.ClientsManager

- | Package | Description |
|------------------------------------|-------------|
| program | |
| program.connection | |
-  **Uses of [ClientsManager](#) in [program](#)**

Modifier and Type	Method and Description
ClientsManager	ServerMain. getClientsManager() Getter for the servers clients-manager

-  **Uses of [ClientsManager](#) in [program.connection](#)**

Constructor and Description

[ClientConnection](#)(java.net.Socket clientSocket, [CommandsManager](#) comManager, [ClientsManager](#) clientsManager)
This constructor initializes the reference objects and and the socket

[CommandsManager](#)([ClientsManager](#) clientsManager)
This constructor sets up the reference elements

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Interface **program.connection.CommandsManager.Command**

No usage of program.connection.CommandsManager.Command

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.connection.CommandsManager

- | Package | Description |
|------------------------------------|-------------|
| program.connection | |
-  **Uses of [CommandsManager](#) in [program.connection](#)**

Constructor and Description

[ClientConnection](#)([java.net.Socket](#) clientSocket, [CommandsManager](#) comManager, [ClientsManager](#) clientsManager)

This constructor initializes the reference objects and the socket

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)

Overview List

- Next
- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.connection.ServerManager

- | Package | Description |
|------------------------------------|-------------|
| program | |
| program.connection | |

Uses of [ServerManager](#) in [program](#)

Modifier and Type	Method and Description
ServerManager	ClientMain. getServerManager () Getter to get the servermanager.

Uses of [ServerManager](#) in [program.connection](#)

Modifier and Type	Method and Description
static javafx.collections.ObservableList<java.lang.String>	ServerRequests. getAllSubjectConnections () Static method used to get all the subject connections to connect to
static boolean	ServerRequests. serverHasLecture (java.lang.String classID) static method to request from the server if the lecture is being held in requested classID.

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.connection.ServerRequests

No usage of program.connection.ServerRequests

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.connection.listeners

Class ClientListener

- [java.lang.Object](#)
 - [◆ program.connection.listeners.ClientListener](#)

 - All Implemented Interfaces:
[java.lang.Runnable](#)
-

```
public class
extends java.lang.Object
implements java.lang.Runnable
```

A Listener Class that is instantiated in a new thread to work as a listener to update info at the client side, without letting the application hold in a block of code. The listener receives JSONObjects and proceeds to call methods at the UI controllers based on what "function" parameter they get
 Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

Constructor and Description

ClientListener(ClientMain main, QuestionReciever controller)

- ◆ **Method Summary**

Modifier and Type	Method and Description
void	<u>run</u> ()
	<u>stopListening</u> ()
void	A Method made to interrupt the listener and make it escape the code hold from the input stream read



Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ◆ **Constructor Detail**

◇ ClientListener

```
public ClientListener(ClientMain main,
                     QuestionReciever controller)
```

Parameters:

main - Reference to the Main object

controller - the controller of the active window that created this listener

- ◆ **Method Detail**

◇ stopListening

```
public void stopListening()
```

A Method made to interrupt the listener and make it escape the code hold from the input stream read

◇ **run**

```
public void run()
```

Specified by:

run in interface `java.lang.Runnable`

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- Detail:
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.connection.listeners

Interface LectureReciever

- All Known Implementing Classes:
[LectureOverviewController](#), [LectureReviewController](#)
-

```
public interface
```

Interface for controllers that are supposed to recieve information about lectures

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Method Summary**

Modifier and Type	Method and Description
void	<u>fetchLectures()</u> Requests all lectures from the database
void	<u>recieveLectureReview</u> (org.json.JSONObject obj) Receives and displays meta-data about a specific lecture
void	<u>recieveLectures</u> (org.json.JSONObject obj) Receives and displays information about lectures
void	<u>recieveQuestions</u> (org.json.JSONObject obj) Receives and displays questions related to a specific lecture

- ◆ **Method Detail**

◆ **fetchLectures**

```
void fetchLectures()
```

Requests all lectures from the database

◆ **recieveLectures**

```
void recieveLectures(org.json.JSONObject obj)
```

Receives and displays information about lectures

Parameters:

obj - A JSONObject containing a list of all lectures and various meta-data

◆ **recieveQuestions**

```
void recieveQuestions(org.json.JSONObject obj)
```

Receives and displays questions related to a specific lecture

Parameters:

obj - A JSONObject containing a list of questions and various meta-data

◆ **recieveLectureReview**

```
void recieveLectureReview(org.json.JSONObject obj)
```

Receives and displays meta-data about a specific lecture

Parameters:

obj - A JSONObject

Skip navigation links

- [Overview](#)
- [Package](#)

Overview List

- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

program.connection.listeners

Class LectureStatListener

- java.lang.Object
 - ◆ program.connection.listeners.LectureStatListener

 - All Implemented Interfaces:
 - java.lang.Runnable
-

```
public class
extends java.lang.Object
implements java.lang.Runnable
```

Listener that sends info to controllers implementing the LectureReciever interface

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

Constructor and Description

LectureStatListener(ClientMain main, LectureReciever controller, int listenerCounter)

- ◆ **Method Summary**

Modifier and Type Method and Description

void run()



Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ◆ **Constructor Detail**

◆ LectureStatListener

```
public LectureStatListener(ClientMain main,
                           LectureReciever controller,
                           int listenerCounter)
```

Parameters:

main - reference to the Main class

controller - the controller of the active window.

listenerCounter - a counter used to count how many times it will recieve information before the active view doesn't need more info, and closes the listener

- ◆ **Method Detail**

◆ run

```
public void run()
```

Specified by:

run in interface java.lang.Runnable

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

program.connection.listeners

Interfaces

- LectureReciever
- QuestionReciever

Classes

- ClientListener
- LectureStatListener

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Package program.connection.listeners

- | Interface | Description |
|---|---|
| <u>LectureReciever</u> | Interface for controllers that are supposed to recieve information about lectures |
| <u>QuestionReciever</u> | interface controllers can implement, to get methods listeners need, when they have updates and information about questions. |
- | Class | Description |
|--|---|
| <u>ClientListener</u> | A Listener Class that is instantiated in a new thread to work as a listener to update info at the client side, without letting the application hold in a block of code. |
| <u>LectureStatListener</u> | Listener that sends info to controllers implementing the LectureReciever interface |

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

Overview List

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For Package `program.connection.listeners`

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- `java.lang.Object`
 - ◆ `program.connection.listeners.ClientListener` (implements `java.lang.Runnable`)
 - ◆ `program.connection.listeners.LectureStatListener` (implements `java.lang.Runnable`)

Interface Hierarchy

- `program.connection.listeners.LectureReciever`
- `program.connection.listeners.QuestionReciever`

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

Overview List

- [Prev](#)
- [Next](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Package program.connection.listeners

- | Package | Description |
|---|-------------|
| program.connection.listeners | |
| program.ui.controllers.lecturer | |
| program.ui.controllers.student | |
- | Class and Description |
|---|
| LectureReciever
Interface for controllers that are supposed to recieve information about lectures |
| QuestionReciever
interface controllers can implement, to get methods listeners need, when they have updates and information about questions. |
- | Class and Description |
|---|
| LectureReciever
Interface for controllers that are supposed to recieve information about lectures |
| QuestionReciever
interface controllers can implement, to get methods listeners need, when they have updates and information about questions. |
- | Class and Description |
|-----------------------|
|-----------------------|

Overview List

QuestionReceiver

interface controllers can implement, to get methods listeners need, when they have updates and information about questions.

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev
- Next

- Frames
- No Frames

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.connection.listeners

Interface QuestionReciever

- All Known Implementing Classes:
[LecturerWindowController](#), [StudentWindowController](#)
-

```
public interface
```

interface controllers can implement, to get methods listeners need, when they have updates and information about questions.

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Method Summary**

Modifier and Type	Method and Description
void	<u>fetchLiveLectureID()</u> Gets a specific ID for this live lecture so the client receives the correct questions
void	<u>fetchQuestions(int numberOfQuestions)</u> Tell the server to send the most recent questions in the database
void	<u>recieveQuestions(org.json.JSONObject obj)</u> Receives and displays the returned questions
void	<u>setLiveLectureID(int ID)</u> Receives and saves the returned lectureID
void	<u>updateQuestionScore(int questionID, int newScore)</u> Get notified that questions has changed score

- ◆ **Method Detail**

- ◆ **fetchQuestions**

```
void fetchQuestions(int numberOfQuestions)
```

Tell the server to send the most recent questions in the database

Parameters:

`numberOfQuestions` - The amount of recently asked question to receive

- ◆ **recieveQuestions**

```
void recieveQuestions(org.json.JSONObject obj)
```

Receives and displays the returned questions

Parameters:

`obj` - A JSONObject containing a list of questions and various meta-data

- ◆ **fetchLiveLectureID**

```
void fetchLiveLectureID()
```

Gets a specific ID for this live lecture so the client receives the correct questions

- ◆ **setLiveLectureID**

```
void setLiveLectureID(int ID)
```

Receives and saves the returned lectureID

Parameters:

`ID` - The received live lecture ID

◇ **updateQuestionScore**

```
void updateQuestionScore(int questionID,  
                        int newScore)
```

Get notified that questions has changed score

Parameters:

questionID - The question that changed its score

newScore - The new score the question has

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.connection.listeners.ClientListener

No usage of program.connection.listeners.ClientListener

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Interface

program.connection.listeners.LectureReciever

- | Package | Description |
|---|-------------|
| program.connection.listeners | |
| program.ui.controllers.lecturer | |
- ◆

Uses of [LectureReciever](#) in [program.connection.listeners](#)

Constructor and Description

[LectureStatListener](#)([ClientMain](#) main, [LectureReciever](#) controller, int listenerCounter)

- ◆

Uses of [LectureReciever](#) in [program.ui.controllers.lecturer](#)

Modifier and Type	Class and Description
class	LectureOverviewController Controller for the overview page of a subject.
class	LectureReviewController Controller for the view where the lecture gets all the information about a previous lecture

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.connection.listeners.LectureStatListener

No usage of program.connection.listeners.LectureStatListener

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Interface

program.connection.listeners.QuestionReciever

- | Package | Description |
|---|-------------|
| program.connection.listeners | |
| program.ui.controllers.lecturer | |
| program.ui.controllers.student | |
- - ◆ **Uses of [QuestionReciever](#) in [program.connection.listeners](#)**

Constructor and Description

[ClientListener](#)([ClientMain](#) main, [QuestionReciever](#) controller)

- ◆ **Uses of [QuestionReciever](#) in [program.ui.controllers.lecturer](#)**

Modifier and Type

Class and Description

class	LecturerWindowController The controller for the view the lecturer sees when a lecture is happening
-------	---

♦ **Uses of QuestionReciever in program.ui.controllers.student**

Modifier and Type

Class and Description

`class`

[StudentWindowController](#)

Controller for the student view when attending a lecture

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.database

Class Database

- [java.lang.Object](#)
 - [◆ program.database.Database](#)
 - [All Implemented Interfaces:](#)
[java.lang.AutoCloseable](#)
-

```
public class
extends java.lang.Object
implements java.lang.AutoCloseable
```

This class is used to connect and run SQL-queries on the database To use this class, the library mysql-connector-java-5.1.40-bin.jar is needed A URL to gain access is also needed, and is provided in the class DatabaseKey (not checked in to our github)

Since:

1.0

Version:
"%I%, %G%"
Author:
Peder Gjerstad

- ◆ **Constructor Summary**

Constructor and Description

[Database\(\)](#)

- ◆ **Method Summary**

Modifier and Type

int	<u>addStudentC</u> Increases the n
void	<u>close()</u>
boolean	<u>connect()</u> This method s database with
int	<u>createNewLe</u> java.lang.S This method c
boolean	<u>createNewSul</u> java.lang.S This method a
int	<u>createYouLo</u> This method s the database
boolean	<u>executeStat</u> This mehtod e
java.util.ArrayList<java.lang.String>	<u>getAllSubje</u> This method f
java.util.ArrayList<java.util.Map<java.lang.String, java.lang.String>>	<u>getLastestO</u> int numberO This method g and returns the
java.util.ArrayList<java.util.Map<java.lang.String, java.lang.String>>	<u>getLatestLe</u> This method g specified subj
java.util.HashMap<java.lang.String, java.lang.String>	<u>getLecturesS</u> Gets all info a
int	<u>getLiveLect</u> Fetches the id

Overview List

`java.util.ArrayList<java.sql.Timestamp>`

`int`

`boolean`

`int`

`int`



Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

•



Constructor Detail

◇ Database

```
public Database()
```



Method Detail

◇ connect

```
public boolean connect()
```

This method sets up a Connection object, and connects to to the database with the KEY_URL

Returns:

Returns `true` if the connection was successful, otherwise it returns `false`

◇ postNewQuestion

```
public boolean postNewQuestion(java.lang.String question,
                               int lecture_id)
```

This method inserts a new question to the database

Parameters:

`question` - A String containing the question text

`lecture_id` - An integer giving the id of the lecture where the question was asked

Returns:

[`getLostMeTi`](#)

This method f
lecture

[`getScoreQue`](#)

Fetches the sc

[`postNewQues`](#)

int lecture.

This method i

[`setEndLectu`](#)

This method u

stop-column

[`voteQuestion`](#)

Updates and i

Overview List

Returns `true` if the query was was executed successfully, otherwise it returns `false`

◇ **executeStatement**

```
public boolean executeStatement(java.lang.String statement_as_String)
```

This mehtod executes a query given as a parameter

Parameters:

`statement_as_String` - A `String` that can be executed as a SQL query

Returns:

Returns `true` if the query was was executed successfully, otherwise it returns `false`

◇ **getLastestQuestions**

```
public java.util.ArrayList<java.util.Map<java.lang.String, java.lang.String>>
```

This method gets a specified amount of questions from the database and returns the result

Parameters:

`lecture_id` - An `integer` with the id of the lecture the questions should be selected from

`numberOfQuestions` - An `integer` with the amount of questions that should be returned

Returns:

Returns an `ArrayList<Map<String, String>>` consisting of the `numberOfQuestions` newest questions. The list will be empty if the query was unsuccessful

◇ **getLatestLectures**

```
public java.util.ArrayList<java.util.Map<java.lang.String, java.lang.String>>
```

This method gets all info from all lectures in the database, for a specified subject-code

Parameters:

`classID` - A `String` with the subject-code that the lectures should be retrieved from

Returns:

Returns an `ArrayList<Map<String, String>>` consisting of the lectures in the subject. The list will be empty if the query was unsuccessful

◇ **getLiveLectureID**

```
public int getLiveLectureID(java.lang.String classID)
```

Fetches the id of the latest lecture in a given subject-code

Parameters:

`classID` - A `String` giving the subject-code

Returns:

Returns an `integer`, which equals the ID of the lecture if the query executed successfully, and 0 otherwise

Overview List

◇ **createNewLecture**

```
public int createNewLecture(java.lang.String lectureName,  
                           java.lang.String classID)
```

This method creates a new lecture in a given subject

Parameters:

lectureName - A String with the name of the lecture

classID - A String with the subject code the lecture should be tied to

Returns:

Returns an integer with the lecture id of the new lecture, if the query was successful. Otherwise it returns 0

◇ **setEndLecture**

```
public int setEndLecture(int lectureID)
```

This method updates a given lecture with a time-stamp in the stop-coloumn

Parameters:

lectureID - A String giving the id of the lecture

Returns:

Returns an integer, which equals 1 if the query executed successfully, and 0 otherwise

◇ **createNewSubject**

```
public boolean createNewSubject(java.lang.String classID,  
                               java.lang.String name)
```

This method adds a new subject to the database

Parameters:

classID - A String with an UNIQUE subject code (ex: TMT4100)

name - A String giving a describing name for the subject (ex: "Program Development")

Returns:

Returns true if the query was was executed successfully, otherwise it returns false

◇ **voteQuestion**

```
public int voteQuestion(int questionID,  
                       int val)
```

Updates and increases a given question's score by a value

Parameters:

questionID - An integer giving the id of the question that should be updated

val - An integer giving the amount the score should be increased by

Returns:

Returns an integer, which equals 1 if the query executed successfully, and 0 otherwise

Overview List

◇ **getScoreQuestion**

```
public int getScoreQuestion(int questionID)
```

Fetches the score of a given question from the database

Parameters:

`questionID` - An integer giving the id of the question that should be checked

Returns:

Returns an integer, which equals the score if the query executed successfully, and 0 otherwise

◇ **getAllSubjectCodes**

```
public java.util.ArrayList<java.lang.String> getAllSubjectCodes()
```

This method fetches all subject codes from the database

Returns:

An `ArrayList<String>` consisting of all the subject codes in the database. If the query was unsuccessful, the array will be empty

◇ **addStudentCountToLecture**

```
public int addStudentCountToLecture(int lectureID)
```

Increases the number of connected students in a given lecture

Parameters:

`lectureID` - A String giving the id of the lecture where the amount should be updated

Returns:

Returns an integer, which equals 1 if the query executed successfully, and 0 otherwise

◇ **createYouLostMe**

```
public int createYouLostMe(int lectureID)
```

This method saves a time-stamp for the lost student notifications in the database

Parameters:

`lectureID` - An integer giving the id of the lecture the time-stamp is tied to

Returns:

Returns an integer, which equals 1 if the query executed successfully, and 0 otherwise

◇ **getLostMeTimestamps**

```
public java.util.ArrayList<java.sql.Timestamp> getLostMeTimestamps(int lect
```

This method fetches all time-stamps for lost students, for a given lecture

Parameters:

`lectureID` - An integer giving the id of the lecture to get the time-stamps from

Returns:

Returns an `ArrayList<Timestamp>` of all the time-stamps that the given lecture had. Returns an empty array if the query was unsuccessful

Overview List

◇ **getLectureStats**

```
public java.util.HashMap<java.lang.String, java.lang.String> getLectureStats
```

Gets all info about a lecture

Parameters:

lectureID - An integer giving the id of the lecture the statistics should be gotten from

Returns:

Returns `HashMap<String, String>` with the following keys: "name", "studentsJoined", "start", "stop", "date"

◇ **close**

```
public void close()
```

Specified by:

close in interface `java.lang.AutoCloseable`

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.database

Class DatabaseKey

- [java.lang.Object](#)
- [◆ program.database.DatabaseKey](#)

```
public class
extends java.lang.Object
```

This class is only a holder for the URL needed to connect to the database OBS: This class must not be uploaded onto github.

Since:

1.0

Version:

"%I%, %G%"

Author:

Peder Gjerstad

- **Field Summary**

Modifier and Type	Field and Description
<code>static java.lang.String</code>	<u>KEY_URL</u>

- **Constructor Summary**

Constructor and Description
<u>DatabaseKey()</u>

- **Method Summary**

- ◇ **Methods inherited from class java.lang.Object**

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- **Field Detail**

- ◇ **KEY_URL**

`public static java.lang.String KEY_URL`

- **Constructor Detail**

- ◇ **DatabaseKey**

`public DatabaseKey()`

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)

Overview List

- [Index](#)
- [Help](#)
- [Prev Class](#)
- [Next Class](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)
- Summary:
 - Nested |
 - [Field](#) |
 - [Constr](#) |
 - [Method](#)
- Detail:
 - [Field](#) |
 - [Constr](#) |
 - Method

program.database

Classes

- Database
- DatabaseKey

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Package program.database

Class	Description
Database	This class is used to connect and run SQL-queries on the database To use this class, the library mysql-connector-java-5.1.40-bin.jar is needed A URL to gain access is also needed, and is provided in the class DatabaseKey (not checked in to our github)
DatabaseKey	This class is only a holder for the URL needed to connect to the database OBS: This class must not be uploaded onto github.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For Package program.database

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- java.lang.Object
 - ◆ program.database.[Database](#) (implements java.lang.AutoCloseable)
 - ◆ program.database.[DatabaseKey](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Package program.database

- **Package Description**
[program](#)

Class and Description

[Database](#)

This class is used to connect and run SQL-queries on the database To use this class, the library mysql-connector-java-5.1.40-bin.jar is needed A URL to gain access is also needed, and is provided in the class DatabaseKey (not checked in to our github)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)


- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class program.database.Database

- **Package Description**
[program](#)
-  **Uses of [Database](#) in [program](#)**

Modifier and Type	Method and Description
Database	ServerMain. getDatabase () Getter for the database

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class program.database.DatabaseKey

No usage of program.database.DatabaseKey

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.ui.controllers

Interface AppBinder

- All Known Implementing Classes:
[CreateLectureController](#), [LectureLoginController](#), [LectureOverviewController](#),
[LectureReviewController](#), [LecturerWindowController](#), [LectureSelectorController](#),
[RoleSelectorController](#), [RootController](#), [StudentWindowController](#)
-

```
public interface
```

interface the controllers can implement, so we can have 1 method in MainClass to load UI, since the controllers are different classes.

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Method Summary**

Modifier and Type	Method and Description
void	<code>closeController()</code> Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
void	<code>localBackChanges()</code> Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
void	<code>setMainApp(ClientMain main)</code> Sets up necessary references to client main and gets start info from the database, if any

- ◆ **Method Detail**

◆ **setMainApp**

```
void setMainApp(ClientMain main)
```

Sets up necessary references to client main and gets start info from the database, if any

Parameters:

`main` - The ClientMain class that runs the application

◆ **closeController**

```
void closeController()
```

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

◆ **localBackChanges**

```
void localBackChanges()
```

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Skip navigation links

- [Overview](#)
- [Package](#)
- Class
- [Use](#)

Overview List

- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
 - [Nested I](#)
 - [Field I](#)
 - [Constr I](#)
 - [Method](#)

- Detail:
 - [Field I](#)
 - [Constr I](#)
 - [Method](#)

program.ui.controllers

Interfaces

- AppBinder

Classes

- QuestionBoxController
- RoleSelectorController
- RootController

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [Prev Package](#)
- [Next Package](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

Package program.ui.controllers

- | Interface | Description |
|---------------------------|--|
| AppBinder | interface the controllers can implement, so we can have 1 method in MainClass to load UI, since the controllers are different classes. |
- | Class | Description |
|--|--|
| QuestionBoxController | |
| RoleSelectorController | View that lets the user select what role to have when using the program |
| RootController | The controller for the root view of the app, has a back button in the top, and a window in the middle where we load views based on what state the user is in |

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [Prev Package](#)

Overview List

- [Next Package](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For Package `program.ui.controllers`

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- `java.lang.Object`
 - ◆ `program.ui.controllers.QuestionBoxController`
 - ◆ `program.ui.controllers.RoleSelectorController` (implements `program.ui.controllers.AppBinder`)
 - ◆ `program.ui.controllers.RootController` (implements `program.ui.controllers.AppBinder`)

Interface Hierarchy

- `program.ui.controllers.AppBinder`

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

Overview List

- [Prev](#)
- [Next](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Package program.ui.controllers

- | Package | Description |
|---|-------------|
| program | |
| program.ui.controllers | |
| program.ui.controllers.lecturer | |
| program.ui.controllers.student | |

Class and Description

[AppBinder](#)

interface the controllers can implement, so we can have 1 method in MainClass to load UI, since the controllers are different classes.

[RootController](#)

The controller for the root view of the app, has a back button in the top, and a window in the middle where we load views based on what state the user is in

- | Class and Description |
|-----------------------|
|-----------------------|

[AppBinder](#)

interface the controllers can implement, so we can have 1 method in MainClass to load UI, since the controllers are different classes.

- | Class and Description |
|-----------------------|
|-----------------------|

Overview List

AppBinder

interface the controllers can implement, so we can have 1 method in MainClass to load UI, since the controllers are different classes.

-

Class and Description

AppBinder

interface the controllers can implement, so we can have 1 method in MainClass to load UI, since the controllers are different classes.

QuestionBoxController

Skip navigation links

- Overview
- Package
- Class
- Use
- Tree
- Deprecated
- Index
- Help

- Prev
- Next

- Frames
- No Frames

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

program.ui.controllers

Class QuestionBoxController

- java.lang.Object
- ◆ program.ui.controllers.QuestionBoxController

```
public class
extends java.lang.Object
```

Since:

1.0

Version:

"%I%, %G%"

Author:

Anders Hunderi

- ◆ **Property Summary**

Type	Property and Description
<code>javafx.beans.property.StringProperty</code>	<u><a>good</u> get the property of the string of vote.

- ◆ **Constructor Summary**

Constructor and DescriptionQuestionBoxController()

- ◆ **Method Summary**

Modifier and Type	Method and Description
<code>int</code>	<u><a>getQuestionId()</u> getter to get the id of the question
<code>int</code>	<u><a>getQuestionScore()</u>
<code>java.lang.String</code>	<u><a>getQuestionText()</u> getter to get the text of the question object.
<code>java.lang.String</code>	<u><a>getVote()</u> getter to get the vote state of the question
<code>javafx.beans.property.StringProperty</code>	<u><a>goodProperty()</u> get the property of the string of vote.
<code>void</code>	<u><a>initialize()</u> inits the view.
<code>void</code>	<u><a>setQuestion(Question question)</u> Sets the question object for this window.
<code>void</code>	<u><a>setScoreVisible(boolean value)</u> Method used to determine whether the score of the question should be visible
<code>void</code>	<u><a>voteQuestion(java.lang.String vote)</u> Method used to vote on a the question in this window

**Methods inherited from class java.lang.Object**

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

◆ Property Detail

◇ good

```
public javafx.beans.property.StringProperty goodProperty
```

get the property of the string of vote.



Constructor Detail

◇ QuestionBoxController

```
public QuestionBoxController()
```



Method Detail

◇ initialize

```
public void initialize()
```

inits the view.

◇ getText

```
public java.lang.String getText()
```

getter to get the text of the question object.

Returns:

question text

◇ setScoreVisible

```
public void setScoreVisible(boolean value)
```

Method used to determine whether the score of the question should be visible

Parameters:

bool - to determine if we should show score or not

◇ setQuestion

```
public void setQuestion(Question question)
```

Sets the question object for this window.

Parameters:

question - question object the window should be related to.

◇ getId

```
public int getId()
```

getter to get the id of the question

Returns:

Overview List

- ID of the question used in the database
 - ◊ **voteQuestion**

```
public void voteQuestion(java.lang.String vote)
```

Method used to vote on a the question in this window

Parameters:

 - `vote` - string to determine if a question has been rated good or bad
 - ◊ **getVote**

```
public java.lang.String getVote()
```

getter to get the vote state of the question

Returns:

 - vote state, good or bad
 - ◊ **goodProperty**

```
public javafx.beans.property.StringProperty goodProperty()
```

get the property of the string of vote.
- ◊ **getQuestionScore**

```
public int getQuestionScore()
```

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [Prev Class](#)
- [Next Class](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)
- Summary:
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)
- Detail:

Overview List

- Field |
- Constr |
- Method

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

program.ui.controllers

Class RoleSelectorController

- java.lang.Object
 - ◆ program.ui.controllers.RoleSelectorController

 - All Implemented Interfaces:
 - [AppBinder](#)
-

```
public class
extends java.lang.Object
implements AppBinder
```

View that lets the user select what role to have when using the program

Since:

1.0

Version:

"%I%, %G%"

Author:
Erling Ihlen

- ◆ **Constructor Summary**

Constructor and Description

[RoleSelectorController\(\)](#)

- ◆ **Method Summary**

**Modifier and
Type**

Method and Description

	<u>closeController()</u>
void	Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
	<u>hostAvailabilityCheck()</u>
boolean	A check to see if the server with proposed IP address is running.
	<u>localBackChanges()</u>
void	Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
	<u>setMainApp(ClientMain main)</u>
void	Sets up necessary references to client main and gets start info from the database, if any



Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- ◆ **Constructor Detail**

◇ **RoleSelectorController**

```
public RoleSelectorController()
```

- ◆ **Method Detail**

◇ **hostAvailabilityCheck**

```
public boolean hostAvailabilityCheck()
```

A check to see if the server with proposed IP address is running.

Overview List

Returns:

true or false if the server is available

◇ **setMainApp**

```
public void setMainApp(ClientMain main)
```

Description copied from interface: [AppBinder](#)

Sets up necessary references to client main and gets start info from the database, if any

Specified by:

[setMainApp](#) in interface [AppBinder](#)

Parameters:

main - The ClientMain class that runs the application

◇ **closeController**

```
public void closeController()
```

Description copied from interface: [AppBinder](#)

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

[closeController](#) in interface [AppBinder](#)

◇ **localBackChanges**

```
public void localBackChanges()
```

Description copied from interface: [AppBinder](#)

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Specified by:

[localBackChanges](#) in interface [AppBinder](#)

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Overview List

- Summary:
- Nested |
- Field |
- Constr |
- Method

- Detail:
- Field |
- Constr |
- Method

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

program.ui.controllers

Class RootController

- java.lang.Object
- ◆ program.ui.controllers.RootController

- All Implemented Interfaces:
- [AppBinder](#)

```
public class
extends java.lang.Object
implements AppBinder
```

The controller for the root view of the app, has a back button in the top, and a window in the middle where we load views based on what state the user is in

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

• ◆ Constructor Summary

Constructor and Description

RootController()

◆ Method Summary

Modifier and Type	Method and Description
void	<u>addNewPath</u> (java.lang.String path) Adds the view the user was on, when entering a new view, to the stack, so the back button has something to refer to.
void	<u>closeController</u> () Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
void	<u>initialize</u> () inits view
void	<u>localBackChanges</u> () Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
void	<u>setBackOnlyLocal</u> (boolean backOnlyLocal) Used so you can press the back button when the question box appear, without exiting the student window
void	<u>setMainApp</u> (ClientMain main) Sets up necessary references to client main and gets start info from the database, if any
void	<u>setTitle</u> (java.lang.String title) sets the title in the top of the root controller, used to give the user info about where he or she is.
boolean	<u>skipAddPath</u> () Used if you don't want to skip a view to add to the stack of backs, used on our initial loadui when starting the app.
javafx.scene.control.Button	<u>viewBackButton</u> ()

◇ **Methods inherited from class java.lang.Object**

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

◆ **Constructor Detail**

◇ **RootController**

```
public RootController()
```

◆ **Method Detail**

◇ **initialize**

```
public void initialize()
```

inits view

◇ **addNewPath**

```
public void addNewPath(java.lang.String path)
```

Adds the view the user was on, when entering a new view, to the stack, so the back button has something to refer to.

Parameters:

`path` - fxml path of previous view.

◇ **skipAddPath**

```
public boolean skipAddPath()
```

Used if you don't want to skip a view to add to the stack of backs, used on our initial loadui when starting the app. Also used to let the stack know not to add the view you're on, when you go back, so you're not ping ponging between views.

Returns:

bool to skip or not skip a stack addition

◇ **setTitle**

```
public void setTitle(java.lang.String title)
```

sets the title in the top of the root controller, used to give the user info about where he or she is.

Parameters:

`title` - the title of the page the user is in.

◇ **setMainApp**

```
public void setMainApp(ClientMain main)
```

Description copied from interface: AppBinder

Overview List

Sets up necessary references to client main and gets start info from the database, if any

Specified by:

[setMainApp](#) in interface [AppBinder](#)

Parameters:

main - The ClientMain class that runs the application

◇ **setBackOnlyLocal**

```
public void setBackOnlyLocal(boolean backOnlyLocal)
```

Used so you can press the back button when the question box appear, without exiting the student window

Parameters:

backOnlyLocal - if you have a popup that you want the back button to close.

◇ **viewBackButton**

```
public javafx.scene.control.Button viewBackButton()
```

◇ **closeController**

```
public void closeController()
```

Description copied from interface: [AppBinder](#)

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

[closeController](#) in interface [AppBinder](#)

◇ **localBackChanges**

```
public void localBackChanges()
```

Description copied from interface: [AppBinder](#)

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Specified by:

[localBackChanges](#) in interface [AppBinder](#)

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

Overview List

- Frames
- No Frames
- All Classes
- Summary:
 - Nested |
 - Field |
 - Constr |
 - Method
- Detail:
 - Field |
 - Constr |
 - Method

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Interface program.ui.controllers.AppBinder

- | Package | Description |
|---|-------------|
| program | |
| program.ui.controllers | |
| program.ui.controllers.lecturer | |
| program.ui.controllers.student | |
- ◆

Uses of [AppBinder](#) in [program](#)

Modifier and Type	Method and Description
AppBinder	ClientMain. getActiveController() Method used to get the controller that is currently active in the main window of the root window.

- ◆

Uses of [AppBinder](#) in [program.ui.controllers](#)

Modifier and Type	Class and Description
class	RoleSelectorController View that lets the user select what role to have when using the program

Overview List

[RootController](#)
class The controller for the root view of the app, has a back button in the top, and a window in the middle where we load views based on what state the user is in



Uses of [AppBinder](#) in [program.ui.controllers.lecturer](#)

Modifier and Type	Class and Description
class	<u>CreateLectureController</u>
class	<u>LectureLoginController</u>
class	<u>LectureOverviewController</u> Controller for the overview page of a subject.
class	<u>LectureReviewController</u> Controller for the view where the lecture gets all the information about a previous lecture
class	<u>LecturerWindowController</u> The controller for the view the lecturer sees when a lecture is happening



Uses of [AppBinder](#) in [program.ui.controllers.student](#)

Modifier and Type	Class and Description
class	<u>LectureSelectorController</u> Controller for the student where he or she can select what subject to join a lecture in
class	<u>StudentWindowController</u> Controller for the student view when attending a lecture

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.QuestionBoxController

- | Package | Description |
|--|-------------|
| program.ui.controllers.student | |
- ♦ **Uses of [QuestionBoxController](#) in [program.ui.controllers.student](#)**

Modifier and Type	Method and Description
QuestionBoxController	StudentWindowController. getQuestionBoxController (int questionId) Used for unitTesting and debugging ONLY This method requires debug to be true, else it will always return null

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)

Overview List

- Next
- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.RoleSelectorController

No usage of program.ui.controllers.RoleSelectorController

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.RootController

- **Package Description**
- [program](#)

-  **Uses of [RootController](#) in [program](#)**

Modifier and Type

Method and Description

[RootController](#)

ClientMain.[getRootController\(\)](#)
 Getter to get the controller of the root window.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.ui.controllers.lecturer

Class CreateLectureController

- java.lang.Object
 - ♦ program.ui.controllers.lecturer.CreateLectureController
 - All Implemented Interfaces:
[AppBinder](#)
-

```
public class  
extends java.lang.Object  
implements AppBinder
```

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

- ◆ **Constructor and Description**

- ◆ CreateLectureController()

- ◆ **Method Summary**

- ◆ **Modifier and Type**

- ◆ **Method and Description**

	<u>closeController()</u>
void	Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
	<u>localBackChanges()</u>
void	Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
	<u>setMainApp(ClientMain main)</u>
void	Sets up necessary references to client main and gets start info from the database, if any



- ◆ **Methods inherited from class java.lang.Object**

- ◆ equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ◆ **Constructor Detail**

- ◆ ◇ **CreateLectureController**

- ◆ public CreateLectureController()

- ◆ **Method Detail**

- ◆ ◇ **setMainApp**

- ◆ public void setMainApp(ClientMain main)

- ◆ Description copied from interface: AppBinder

- ◆ Sets up necessary references to client main and gets start info from the database, if any

- ◆ Specified by:

Overview List

[setMainApp](#) in interface [AppBinder](#)

Parameters:

`main` - The ClientMain class that runs the application

◇ **closeController**

```
public void closeController()
```

Description copied from interface: [AppBinder](#)

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

[closeController](#) in interface [AppBinder](#)

◇ **localBackChanges**

```
public void localBackChanges()
```

Description copied from interface: [AppBinder](#)

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Specified by:

[localBackChanges](#) in interface [AppBinder](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)

- Method

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

program.ui.controllers.lecturer

Class LectureBoxController

- java.lang.Object
- ♦ program.ui.controllers.lecturer.LectureBoxController

•

```
public class  
extends java.lang.Object
```

this controller is used for each box you see when you get a list of all the lectures in a subject
Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen

- ◆ **Constructor Summary**

Constructor and Description

[LectureBoxController\(\)](#)



Method Summary

Modifier and Type	Method and Description
int	<u>getLectureID()</u> getter to get the id of the lecture
void	<u>initialize()</u> inits view
void	<u>setDate(java.lang.String date)</u> sets the date of this lecture
void	<u>setLectureID(int lectureID)</u> sets the ID of the lecture this box is related to
void	<u>setLectureName(java.lang.String lectureName)</u> sets the name of the lecture to be accessible from this box
javafx.scene.control.Button	<u>viewButton()</u> Getter to get the button that allows the lecturer to open the lecture.



Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`



Constructor Detail

◇ **LectureBoxController**

`public LectureBoxController()`



Method Detail

◇ **initialize**

`public void initialize()`

inits view

Overview List

◇ **setLectureID**

```
public void setLectureID(int lectureID)
```

sets the ID of the lecture this box is related to

Parameters:

lectureID - ID of lecture

◇ **setLectureName**

```
public void setLectureName(java.lang.String lectureName)
```

sets the name of the lecture to be accessible from this box

Parameters:

lectureName - name of the lecture

◇ **setDate**

```
public void setDate(java.lang.String date)
```

sets the date of this lecture

Parameters:

date - date of the lecture

◇ **getLectureID**

```
public int getLectureID()
```

getter to get the id of the lecture

Returns:

id of the lecture

◇ **viewButton**

```
public javafx.scene.control.Button viewButton()
```

Getter to get the button that allows the lecturer to open the lecture. used as a listener to open a view

Returns:

view lecture button

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

Overview List

- Frames
- No Frames
- All Classes
- Summary:
 - Nested |
 - Field |
 - Constr |
 - Method
- Detail:
 - Field |
 - Constr |
 - Method

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

program.ui.controllers.lecturer

Class LectureLoginController

- java.lang.Object
 - ◆ program.ui.controllers.lecturer.LectureLoginController

 - All Implemented Interfaces:
 - [AppBinder](#)
-

```
public class
extends java.lang.Object
implements AppBinder
```

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

Constructor Summary

Constructor and Description

LectureLoginController()

Method Summary

Modifier and Type

Method and Description

	<u>closeController</u> ()
void	Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
	<u>localBackChanges</u> ()
void	Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
	<u>setMainApp</u> (<u>ClientMain</u> main)
void	Sets up necessary references to client main and gets start info from the database, if any



Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

◆ LectureLoginController

```
public LectureLoginController()
```

Method Detail

◆ setMainApp

```
public void setMainApp(ClientMain main)
```

Description copied from interface: AppBinder

Sets up necessary references to client main and gets start info from the database, if any

Specified by:

Overview List

[setMainApp](#) in interface [AppBinder](#)

Parameters:

`main` - The ClientMain class that runs the application

◇ **closeController**

```
public void closeController()
```

Description copied from interface: [AppBinder](#)

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

[closeController](#) in interface [AppBinder](#)

◇ **localBackChanges**

```
public void localBackChanges()
```

Description copied from interface: [AppBinder](#)

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Specified by:

[localBackChanges](#) in interface [AppBinder](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)

- Method

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

program.ui.controllers.lecturer

Class LectureOverviewController

- java.lang.Object
 - ◆ program.ui.controllers.lecturer.LectureOverviewController

 - All Implemented Interfaces:
[LectureReciever](#), [AppBinder](#)
-

```
public class
extends java.lang.Object
implements AppBinder, LectureReciever
```

Controller for the overview page of a subject. This page lets the user see a list of all previous lectures of a subject, and open them to see stats and other information

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

Constructor and Description

LectureOverviewController()

- ◆ **Method Summary**

Modifier and Type	Method and Description
void	<u>addLecture</u> (java.lang.String lectureName, java.lang.String date, int lectureID, int studentsJoined) Adds a lecture to the list of previous lectures
void	<u>closeController</u> () Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
void	<u>fetchLectures</u> () Requests all lectures from the database
<u>LectureBoxController</u>	<u>getLectureBoxController</u> (int lectureBoxIndex)
void	<u>localBackChanges</u> () Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
void	<u>recieveLectureReview</u> (org.json.JSONObject obj) Receives and displays meta-data about a specific lecture
void	<u>recieveLectures</u> (org.json.JSONObject obj) Receives and displays information about lectures
void	<u>recieveQuestions</u> (org.json.JSONObject obj) Receives and displays questions related to a specific lecture
void	<u>setMainApp</u> (ClientMain main) Sets up necessary references to client main and gets start info from the database, if any

◇

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

◆ Constructor Detail

◇ LectureOverviewController

```
public LectureOverviewController()
```



Method Detail

◇ addLecture

```
public void addLecture(java.lang.String lectureName,
                       java.lang.String date,
                       int lectureID,
                       int studentsJoined)
```

Adds a lecture to the list of previous lectures

Parameters:

lectureName - name of the lecture

date - date of the lecture

lectureID - ID of the lecture

studentsJoined - how many students joined the lecture

◇ getLectureBoxController

```
public LectureBoxController getLectureBoxController(int lectureBoxIndex)
```

◇ setMainApp

```
public void setMainApp(ClientMain main)
```

Description copied from interface: [AppBinder](#)

Sets up necessary references to client main and gets start info from the database, if any

Specified by:

[setMainApp](#) in interface [AppBinder](#)

Parameters:

main - The ClientMain class that runs the application

◇ closeController

```
public void closeController()
```

Description copied from interface: [AppBinder](#)

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

[closeController](#) in interface [AppBinder](#)

◇ localBackChanges

```
public void localBackChanges()
```

Description copied from interface: [AppBinder](#)

Overview List

- Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
Specified by:
[localBackChanges](#) in interface [AppBinder](#)
◇ **recieveLectures**

```
public void recieveLectures(org.json.JSONObject obj)
```


Description copied from interface: [LectureReciever](#)
Receives and displays information about lectures
Specified by:
[recieveLectures](#) in interface [LectureReciever](#)
Parameters:
obj - A JSONObject containing a list of all lectures and various meta-data
◇ **fetchLectures**

```
public void fetchLectures()
```


Description copied from interface: [LectureReciever](#)
Requests all lectures from the database
Specified by:
[fetchLectures](#) in interface [LectureReciever](#)
◇ **recieveQuestions**

```
public void recieveQuestions(org.json.JSONObject obj)
```


Description copied from interface: [LectureReciever](#)
Receives and displays questions related to a specific lecture
Specified by:
[recieveQuestions](#) in interface [LectureReciever](#)
Parameters:
obj - A JSONObject containing a list of questions and various meta-data
◇ **recieveLectureReview**

```
public void recieveLectureReview(org.json.JSONObject obj)
```


Description copied from interface: [LectureReciever](#)
Receives and displays meta-data about a specific lecture
Specified by:
[recieveLectureReview](#) in interface [LectureReciever](#)
Parameters:
obj - A JSONObject

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)

Overview List

- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
 - Nested |
 - Field |
 - [Constr](#) |
 - [Method](#)

- Detail:
 - Field |
 - [Constr](#) |
 - [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

program.ui.controllers.lecturer

Class LectureReviewController

- java.lang.Object
 - ◆ program.ui.controllers.lecturer.LectureReviewController

 - All Implemented Interfaces:
[LectureReciever](#), [AppBinder](#)
-

```
public class
extends java.lang.Object
implements AppBinder, LectureReciever
```

Controller for the view where the lecture gets all the information about a previous lecture

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

- ◆ **Constructor and Description**

- ◆ [LectureReviewController\(\)](#)

- ◆ **Method Summary**

Modifier and Type	Method and Description
	<u>closeController()</u>
void	Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
	<u>fetchLectures()</u>
void	Requests all lectures from the database
	<u>localBackChanges()</u>
void	Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
	<u>recieveLectureReview(org.json.JSONObject obj)</u>
void	Receives and displays meta-data about a specific lecture
	<u>recieveLectures(org.json.JSONObject obj)</u>
void	Receives and displays information about lectures
	<u>recieveQuestions(org.json.JSONObject obj)</u>
void	Receives and displays questions related to a specific lecture
	<u>setMainApp(ClientMain main)</u>
void	Sets up necessary references to client main and gets start info from the database, if any

◇

Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

- ◆ **Constructor Detail**

- ◆ **LectureReviewController**

```
public LectureReviewController()
```

Method Detail

◇ **setMainApp**

```
public void setMainApp(ClientMain main)
```

Description copied from interface: AppBinder

Sets up necessary references to client main and gets start info from the database, if any

Specified by:

setMainApp in interface AppBinder

Parameters:

main - The ClientMain class that runs the application

◇ **closeController**

```
public void closeController()
```

Description copied from interface: AppBinder

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

closeController in interface AppBinder

◇ **localBackChanges**

```
public void localBackChanges()
```

Description copied from interface: AppBinder

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Specified by:

localBackChanges in interface AppBinder

◇ **recieveQuestions**

```
public void recieveQuestions(org.json.JSONObject obj)
```

Description copied from interface: LectureReciever

Receives and displays questions related to a specific lecture

Specified by:

recieveQuestions in interface LectureReciever

Parameters:

obj - A JSONObject containing a list of questions and various meta-data

◇ **recieveLectureReview**

```
public void recieveLectureReview(org.json.JSONObject obj)
```

Description copied from interface: LectureReciever

Receives and displays meta-data about a specific lecture

Specified by:

recieveLectureReview in interface LectureReciever

Parameters:



obj - A JSONObject

◇ **fetchLectures**

```
public void fetchLectures()
```

Description copied from interface: [LectureReciever](#)

Requests all lectures from the database

Specified by:

[fetchLectures](#) in interface [LectureReciever](#)

◇ **recieveLectures**

```
public void recieveLectures(org.json.JSONObject obj)
```

Description copied from interface: [LectureReciever](#)

Receives and displays information about lectures

Specified by:

[recieveLectures](#) in interface [LectureReciever](#)

Parameters:

obj - A JSONObject containing a list of all lectures and various meta-data

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- Nested |
- Field |
- [Constr](#) |
- [Method](#)

- Detail:
- Field |
- [Constr](#) |
- [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

program.ui.controllers.lecturer

Class LecturerWindowController

- java.lang.Object
 - ♦ program.ui.controllers.lecturer.LecturerWindowController

 - All Implemented Interfaces:
[QuestionReciever](#), [AppBinder](#)
-

```
public class
extends java.lang.Object
implements AppBinder, QuestionReciever
```

The controller for the view the lecturer sees when a lecture is happening

Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

Constructor and Description

LecturerWindowController()

- ◆ **Method Summary**

Modifier and Type	Method and Description
	<u>closeController()</u>
void	Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
	<u>fetchLiveLectureID()</u>
void	Gets a specific ID for this live lecture so the client receives the correct questions
	<u>fetchQuestions</u> (int numberOfQuestions)
void	Tell the server to send the most recent questions in the database
	<u>initialize()</u>
void	inits the view
	<u>localBackChanges()</u>
void	Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
	<u>recieveQuestions</u> (org.json.JSONObject obj)
void	Receives and displays the returned questions
	<u>setLiveLectureID</u> (int ID)
void	Receives and saves the returned lectureID
	<u>setMainApp</u> (ClientMain main)
void	Sets up necessary references to client main and gets start info from the database, if any
	<u>setTitleAndNameText</u> (java.lang.String newTitle, java.lang.String newName)
void	This method changes the title text and lecture name text
	<u>studentJoined()</u>
void	Increments the number of students, and update ui elements
	<u>studentLost()</u>
void	Increments the number of LOST students, and updates the pieChart.
	<u>updatePieChartValues()</u>
void	This method checks how many students are lost, and updates the pie chart accordingly
	<u>updateQuestionScore</u> (int questionID, int newScore)
void	

Overview List

Get notified that questions has changed score

```
void updateStudentsConnectedAmount()
```

This method checks connectedStudents and updates the text



Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait,`
`wait`



Constructor Detail

◇ LecturerWindowController

```
public LecturerWindowController()
```



Method Detail

◇ initialize

```
public void initialize()
```

inits the view

◇ setTitleAndNameText

```
public void setTitleAndNameText(java.lang.String newTitle,  
                                java.lang.String newName)
```

This method changes the title text and lecture name text

Parameters:

`newTitle` -
`newName` -

◇ updateStudentsConnectedAmount

```
public void updateStudentsConnectedAmount()
```

This method checks connectedStudents and updates the text

◇ updatePieChartValues

```
public void updatePieChartValues()
```

This method checks how many students are lost, and updates the pie chart accordingly

◇ studentJoined

```
public void studentJoined()
```

Increments the number of students, and update ui elements

Overview List

◇ **studentLost**

```
public void studentLost()
```

Increments the number of LOST students, and updates the pieChart. Also checks and warns the teacher if the threshold is reached

◇ **setMainApp**

```
public void setMainApp(ClientMain main)
```

Description copied from interface: AppBinder

Sets up necessary references to client main and gets start info from the database, if any

Specified by:

setMainApp in interface AppBinder

Parameters:

main - The ClientMain class that runs the application

◇ **closeController**

```
public void closeController()
```

Description copied from interface: AppBinder

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

closeController in interface AppBinder

◇ **localBackChanges**

```
public void localBackChanges()
```

Description copied from interface: AppBinder

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Specified by:

localBackChanges in interface AppBinder

◇ **fetchQuestions**

```
public void fetchQuestions(int numberOfQuestions)
```

Description copied from interface: QuestionReciever

Tell the server to send the most recent questions in the database

Specified by:

fetchQuestions in interface QuestionReciever

Parameters:

numberOfQuestions - The amount of recently asked question to receive

◇ **recieveQuestions**

```
public void recieveQuestions(org.json.JSONObject obj)
```

Description copied from interface: QuestionReciever

Receives and displays the returned questions

Specified by:

Overview List

[recieveQuestions](#) in interface [QuestionReciever](#)

Parameters:

obj - A JSONObject containing a list of questions and various meta-data

◇ **fetchLiveLectureID**

```
public void fetchLiveLectureID()
```

Description copied from interface: [QuestionReciever](#)

Gets a specific ID for this live lecture so the client receives the correct questions

Specified by:

[fetchLiveLectureID](#) in interface [QuestionReciever](#)

◇ **setLiveLectureID**

```
public void setLiveLectureID(int ID)
```

Description copied from interface: [QuestionReciever](#)

Receives and saves the returned lectureID

Specified by:

[setLiveLectureID](#) in interface [QuestionReciever](#)

Parameters:

ID - The received live lecture ID

◇ **updateQuestionScore**

```
public void updateQuestionScore(int questionID,  
                                int newScore)
```

Description copied from interface: [QuestionReciever](#)

Get notified that questions has changed score

Specified by:

[updateQuestionScore](#) in interface [QuestionReciever](#)

Parameters:

questionID - The question that changed its score

newScore - The new score the question has

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

Overview List

- All Classes
- Summary:
 - Nested |
 - Field |
 - Constr |
 - Method
- Detail:
 - Field |
 - Constr |
 - Method

program.ui.controllers.lecturer

Classes

- CreateLectureController
- LectureBoxController
- LectureLoginController
- LectureOverviewController
- LectureReviewController
- LecturerWindowController

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Package program.ui.controllers.lecturer

Class	Description
CreateLectureController	
LectureBoxController	this controller is used for each box you see when you get a list of all the lectures in a subject
LectureLoginController	
LectureOverviewController	Controller for the overview page of a subject.
LectureReviewController	Controller for the view where the lecture gets all the information about a previous lecture
LecturerWindowController	The controller for the view the lecturer sees when a lecture is happening

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)

Overview List

- Next Package
- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For Package program.ui.controllers.lecturer

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- java.lang.Object
 - ◆ program.ui.controllers.lecturer.[CreateLectureController](#) (implements program.ui.controllers.[AppBinder](#))
 - ◆ program.ui.controllers.lecturer.[LectureBoxController](#)
 - ◆ program.ui.controllers.lecturer.[LectureLoginController](#) (implements program.ui.controllers.[AppBinder](#))
 - ◆ program.ui.controllers.lecturer.[LectureOverviewController](#) (implements program.ui.controllers.[AppBinder](#), program.connection.listeners.[LectureReciever](#))
 - ◆ program.ui.controllers.lecturer.[LectureReviewController](#) (implements program.ui.controllers.[AppBinder](#), program.connection.listeners.[LectureReciever](#))
 - ◆ program.ui.controllers.lecturer.[LecturerWindowController](#) (implements program.ui.controllers.[AppBinder](#), program.connection.listeners.[QuestionReciever](#))

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)

Overview List

- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Package program.ui.controllers.lecturer

- | Package | Description |
|---|-------------|
| program.ui.controllers.lecturer | |
- | Class and Description |
|---|
| LectureBoxController
this controller is used for each box you see when you get a list of all the lectures in a subject |

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.lecturer.CreateLectureController

No usage of program.ui.controllers.lecturer.CreateLectureController

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.lecturer.LectureBoxController

- | Package | Description |
|---|--|
| program.ui.controllers.lecturer | <ul style="list-style-type: none"> ♦ Uses of LectureBoxController in program.ui.controllers.lecturer |

Modifier and Type	Method and Description
LectureBoxController	LectureOverviewController. getLectureBoxController (int lectureBox)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.lecturer.LectureLoginController

No usage of program.ui.controllers.lecturer.LectureLoginController

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- Use
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- Prev
- Next

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.lecturer.LectureOverviewController

No usage of program.ui.controllers.lecturer.LectureOverviewController

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- Use
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- Prev
- Next

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.lecturer.LectureReviewController

No usage of program.ui.controllers.lecturer.LectureReviewController

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.lecturer.LecturerWindowController

No usage of program.ui.controllers.lecturer.LecturerWindowController

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

program.ui.controllers.student

Class LectureSelectorController

- java.lang.Object
 - ◆ program.ui.controllers.student.LectureSelectorController
 - All Implemented Interfaces:
 [AppBinder](#)
-

```
public class
extends java.lang.Object
implements AppBinder
```

Controller for the student where he or she can select what subject to join a lecture in
Since:

1.0

Version:

"%I%, %G%"

Author:

Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

- ◆ **Constructor and Description**

- [LectureSelectorController\(\)](#)

- ◆ **Method Summary**

- ◆ **Modifier and Type**

- ◆ **Method and Description**

	<u>closeController()</u>
void	Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
	<u>localBackChanges()</u>
void	Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
	<u>setMainApp(ClientMain main)</u>
void	Sets up necessary references to client main and gets start info from the database, if any

◇

- ◆ **Methods inherited from class java.lang.Object**

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

- ◆ **Constructor Detail**

- ◆ **LectureSelectorController**

- public LectureSelectorController()

- ◆ **Method Detail**

- ◆ **setMainApp**

- public void setMainApp([ClientMain](#) main)

Description copied from interface: [AppBinder](#)

Sets up necessary references to client main and gets start info from the database, if any

Overview List

Specified by:

[setMainApp](#) in interface [AppBinder](#)

Parameters:

main - The ClientMain class that runs the application

◇ **closeController**

```
public void closeController()
```

Description copied from interface: [AppBinder](#)

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

[closeController](#) in interface [AppBinder](#)

◇ **localBackChanges**

```
public void localBackChanges()
```

Description copied from interface: [AppBinder](#)

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Specified by:

[localBackChanges](#) in interface [AppBinder](#)

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- [Summary:](#)
- [Nested |](#)
- [Field |](#)
- [Constr |](#)
- [Method](#)

- [Detail:](#)
- [Field |](#)

Overview List

- Constr l
- Method

program.ui.controllers.student

Classes

- LectureSelectorController
- StudentWindowController

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Package program.ui.controllers.student

Class	Description
LectureSelectorController	Controller for the student where he or she can select what subject to join a lecture in
StudentWindowController	Controller for the student view when attending a lecture

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Package](#)
- [Next Package](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Hierarchy For Package program.ui.controllers.student

Package Hierarchies:

- [All Packages](#)

Class Hierarchy

- java.lang.Object
 - ◆ program.ui.controllers.student.[LectureSelectorController](#) (implements program.ui.controllers.[AppBinder](#))
 - ◆ program.ui.controllers.student.[StudentWindowController](#) (implements program.ui.controllers.[AppBinder](#), program.connection.listeners.[QuestionReciever](#))

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Package program.ui.controllers.student

No usage of program.ui.controllers.student

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
- [Nested I](#)
- [Field I](#)
- [Constr I](#)
- [Method](#)

- Detail:
- [Field I](#)
- [Constr I](#)
- [Method](#)

program.ui.controllers.student

Class StudentWindowController

- java.lang.Object
 - ◆ program.ui.controllers.student.StudentWindowController
 - All Implemented Interfaces:
[QuestionReciever](#), [AppBinder](#)
-

```
public class
extends java.lang.Object
implements AppBinder, QuestionReciever
```

Controller for the student view when attending a lecture

Since:

1.0

Version:

"%I%, %G%"

Author:
Erling Ihlen, Anders Hunderi

- ◆ **Constructor Summary**

Constructor and Description

[StudentWindowController\(\)](#)

- ◆ **Method Summary**

Modifier and Type	Method and Description
void	<u>closeController()</u> Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)
void	<u>fetchLiveLectureID()</u> Gets a specific ID for this live lecture so the client receives the correct questions
void	<u>fetchQuestions(int numberOfQuestions)</u> Tell the server to send the most recent questions in the database
<u>QuestionBoxController</u>	<u>getQuestionBoxController(int questionIndex)</u> Used for unitTesting and debugging ONLY This method requires debug to be true, else it will always return null
void	<u>initialize()</u> inits view
void	<u>localBackChanges()</u> Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton
void	<u>recieveQuestions(org.json.JSONObject obj)</u> Receives and displays the returned questions
void	<u>setLiveLectureID(int ID)</u> Receives and saves the returned lectureID
void	<u>setMainApp(ClientMain main)</u> Sets up necessary references to client main and gets start info from the database, if any
void	<u>updateQuestionScore(int questionID, int newScore)</u> Get notified that questions has changed score



Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

◆ Constructor Detail

◇ StudentWindowController

```
public StudentWindowController()
```



Method Detail

◇ initialize

```
public void initialize()
```

inits view

◇ getQuestionBoxController

```
public QuestionBoxController getQuestionBoxController(int questionIndex)
```

Used for unitTesting and debugging ONLY This method requires debug to be true, else it will always return null

Parameters:

questionIndex - The index of the question in the vBox container

Returns:

If main.DEBUG is true, this will return the controller for a specified question, else null

◇ setMainApp

```
public void setMainApp(ClientMain main)
```

Description copied from interface: AppBinder

Sets up necessary references to client main and gets start info from the database, if any

Specified by:

setMainApp in interface AppBinder

Parameters:

main - The ClientMain class that runs the application

◇ closeController

```
public void closeController()
```

Description copied from interface: AppBinder

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

Specified by:

closeController in interface AppBinder

◇ localBackChanges

```
public void localBackChanges()
```

Description copied from interface: AppBinder

Overview List

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

Specified by:

localBackChanges in interface AppBinder

◇ **fetchQuestions**

```
public void fetchQuestions(int numberOfQuestions)
```

Description copied from interface: QuestionReciever

Tell the server to send the most recent questions in the database

Specified by:

fetchQuestions in interface QuestionReciever

Parameters:

numberOfQuestions - The amount of recently asked question to receive

◇ **recieveQuestions**

```
public void recieveQuestions(org.json.JSONObject obj)
```

Description copied from interface: QuestionReciever

Receives and displays the returned questions

Specified by:

recieveQuestions in interface QuestionReciever

Parameters:

obj - A JSONObject containing a list of questions and various meta-data

◇ **fetchLiveLectureID**

```
public void fetchLiveLectureID()
```

Description copied from interface: QuestionReciever

Gets a specific ID for this live lecture so the client receives the correct questions

Specified by:

fetchLiveLectureID in interface QuestionReciever

◇ **setLiveLectureID**

```
public void setLiveLectureID(int ID)
```

Description copied from interface: QuestionReciever

Receives and saves the returned lectureID

Specified by:

setLiveLectureID in interface QuestionReciever

Parameters:

ID - The received live lecture ID

◇ **updateQuestionScore**

```
public void updateQuestionScore(int questionID,  
                                int newScore)
```

Description copied from interface: QuestionReciever

Get notified that questions has changed score

Specified by:

updateQuestionScore in interface QuestionReciever

Parameters:

Overview List

- `questionID` - The question that changed its score
`newScore` - The new score the question has

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Class](#)
- [Next Class](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

- Summary:
 - [Nested I](#)
 - [Field I](#)
 - [Constr I](#)
 - [Method](#)

- Detail:
 - [Field I](#)
 - [Constr I](#)
 - [Method](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.student.LectureSelectorController

No usage of program.ui.controllers.student.LectureSelectorController

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Uses of Class

program.ui.controllers.student.StudentWindowController

No usage of program.ui.controllers.student.StudentWindowController

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

Frame Alert

This document is designed to be viewed using the frames feature. If you see this message, you are using a non-frame-capable web client. Link to [Non-frame version](#).

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

A

[addClientToCollection\(ClientConnection\)](#) - Method in class `program.connection.ClientsManager`

Adds a given `ClientConnection` to the list of connected clients

[addLecture\(String, String, int, int\)](#) - Method in class `program.ui.controllers.lecturer.LectureOverviewController`

Adds a lecture to the list of previous lectures

[addLecturerToLecture\(ClientConnection, String\)](#) - Method in class `program.connection.ClientsManager`

This method adds a lecturer to a given lecture

[addNewPath\(String\)](#) - Method in class `program.ui.controllers.RootController`

Adds the view the user was on, when entering a new view, to the stack, so the back button has something to refer to.

[addStudentCountToLecture\(int\)](#) - Method in class `program.database.Database`

Increases the number of connected students in a given lecture

[analyzeFunction\(JSONObject, ClientConnection\)](#) - Method in class `program.connection.CommandsManager`

This method analyzes a given `JSONObject` and passes it to the correct handler-method

[AppBinder](#) - Interface in `program.ui.controllers`

interface the controllers can implement, so we can have 1 method in `MainClass` to load UI, since the controllers are different classes.

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)

Overview List

- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

L

[LectureBoxController](#) - Class in [program.ui.controllers.lecturer](#)

this controller is used for each box you see when you get a list of all the lectures in a subject

[LectureBoxController\(\)](#) - Constructor for class [program.ui.controllers.lecturer.LectureBoxController](#)

[LectureLoginController](#) - Class in [program.ui.controllers.lecturer](#)

[LectureLoginController\(\)](#) - Constructor for class [program.ui.controllers.lecturer.LectureLoginController](#)

[LectureOverviewController](#) - Class in [program.ui.controllers.lecturer](#)

Controller for the overview page of a subject.

[LectureOverviewController\(\)](#) - Constructor for class [program.ui.controllers.lecturer.LectureOverviewController](#)

[LectureReciever](#) - Interface in [program.connection.listeners](#)

Interface for controllers that are supposed to recieve information about lectures

[LectureReviewController](#) - Class in [program.ui.controllers.lecturer](#)

Controller for the view where the lecture gets all the information about a previous lecture

[LectureReviewController\(\)](#) - Constructor for class [program.ui.controllers.lecturer.LectureReviewController](#)

[LecturerWindowController](#) - Class in [program.ui.controllers.lecturer](#)

The controller for the view the lecturer sees when a lecture is happening

[LecturerWindowController\(\)](#) - Constructor for class [program.ui.controllers.lecturer.LecturerWindowController](#)

[LectureSelectorController](#) - Class in [program.ui.controllers.student](#)

Controller for the student where he or she can select what subject to join a lecture in

Overview List

[LectureSelectorController\(\)](#) - Constructor for class `program.ui.controllers.student.LectureSelectorController`

[LectureStatListener](#) - Class in `program.connection.listeners`

Listener that sends info to controllers implementing the `LectureReceiver` interface

[LectureStatListener\(ClientMain, LectureReceiver, int\)](#) - Constructor for class `program.connection.listeners.LectureStatListener`

[loadUI\(String\)](#) - Method in class `program.ClientMain`

This method loads the view related to the fxml file in parameter, and sets it in the center window of our root window.

[localBackChanges\(\)](#) - Method in interface `program.ui.controllers.AppBinder`

Used when the rootController doesn't go back, but the center controller wants to change something locally using the backButton

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.lecturer.CreateLectureController`

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.lecturer.LectureLoginController`

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.lecturer.LectureOverviewController`

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.lecturer.LectureReviewController`

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.RoleSelectorController`

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.RootController`

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.student.LectureSelectorController`

[localBackChanges\(\)](#) - Method in class `program.ui.controllers.student.StudentWindowController`

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

M

[main\(String\[\]\)](#) - Static method in class [program.ClientMain](#)

Launches the application

[main](#) - Variable in class [program.connection.ClientsManager](#)

[main\(String\[\]\)](#) - Static method in class [program.ServerMain](#)

Launches the server

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

P

[postNewQuestion\(String, int\)](#) - Method in class [program.database.Database](#)

This method inserts a new question to the database

[program](#) - package [program](#)

[program.common](#) - package [program.common](#)

[program.connection](#) - package [program.connection](#)

[program.connection.listeners](#) - package [program.connection.listeners](#)

[program.database](#) - package [program.database](#)

[program.ui.controllers](#) - package [program.ui.controllers](#)

[program.ui.controllers.lecturer](#) - package [program.ui.controllers.lecturer](#)

[program.ui.controllers.student](#) - package [program.ui.controllers.student](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)

Overview List

- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

Q

[Question](#) - Class in [program.common](#)

Question object used as a listener for the question list for both lecturer and student

[Question\(int, String, String, int\)](#) - Constructor for class [program.common.Question](#)

[QuestionBoxController](#) - Class in [program.ui.controllers](#)

[QuestionBoxController\(\)](#) - Constructor for class [program.ui.controllers.QuestionBoxController](#)

[QuestionReciever](#) - Interface in [program.connection.listeners](#)

interface controllers can implement, to get methods listeners need, when they have updates and information about questions.

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

Overview List

- Frames
- No Frames
- All Classes

Overview List

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

R

[ratingProperty\(\)](#) - Method in class `program.common.Question`

getter for the window element to update the rating number, whenever rating changes in the Question object.

[recieveLectureReview\(JSONObject\)](#) - Method in interface `program.connection.listeners.LectureReciever`

Receives and displays meta-data about a specific lecture

[recieveLectureReview\(JSONObject\)](#) - Method in class

`program.ui.controllers.lecturer.LectureOverviewController`

[recieveLectureReview\(JSONObject\)](#) - Method in class

`program.ui.controllers.lecturer.LectureReviewController`

[recieveLectures\(JSONObject\)](#) - Method in interface `program.connection.listeners.LectureReciever`

Receives and displays information about lectures

[recieveLectures\(JSONObject\)](#) - Method in class `program.ui.controllers.lecturer.LectureOverviewController`

[recieveLectures\(JSONObject\)](#) - Method in class `program.ui.controllers.lecturer.LectureReviewController`

[recieveQuestions\(JSONObject\)](#) - Method in interface `program.connection.listeners.LectureReciever`

Receives and displays questions related to a specific lecture

[recieveQuestions\(JSONObject\)](#) - Method in interface `program.connection.listeners.QuestionReciever`

Receives and displays the returned questions

[recieveQuestions\(JSONObject\)](#) - Method in class `program.ui.controllers.lecturer.LectureOverviewController`

[recieveQuestions\(JSONObject\)](#) - Method in class `program.ui.controllers.lecturer.LectureReviewController`

[recieveQuestions\(JSONObject\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

Overview List

[recieveQuestions\(JSONObject\)](#) - Method in class `program.ui.controllers.student.StudentWindowController`

[removeConnection\(ClientConnection\)](#) - Method in class `program.connection.ClientsManager`

Removes a given `ClientConnection` from the list of connected clients

[removeLecturerFromLecture\(ClientConnection\)](#) - Method in class `program.connection.ClientsManager`

This method removes a lecturer from a given lecturer

[RoleSelectorController](#) - Class in `program.ui.controllers`

View that lets the user select what role to have when using the program

[RoleSelectorController\(\)](#) - Constructor for class `program.ui.controllers.RoleSelectorController`

[RootController](#) - Class in `program.ui.controllers`

The controller for the root view of the app, has a back button in the top, and a window in the middle where we load views based on what state the user is in

[RootController\(\)](#) - Constructor for class `program.ui.controllers.RootController`

[run\(\)](#) - Method in class `program.connection.ClientConnection`

[run\(\)](#) - Method in class `program.connection.listeners.ClientListener`

[run\(\)](#) - Method in class `program.connection.listeners.LectureStatListener`

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

S

[sendInfoToLecturer\(JSONObject, String\)](#) - Method in class [program.connection.ClientsManager](#)

This method sends a given `JSONObject` to the lecturer holding a given lecture

[sendJSON\(JSONObject\)](#) - Method in class [program.connection.ClientConnection](#)

Sends a given `JSONObject` to the client

[sendJSON\(JSONObject\)](#) - Method in class [program.connection.ServerManager](#)

Method used to send a json object to the server.

[serverHasLecture\(ServerManager, String\)](#) - Static method in class [program.connection.ServerRequests](#)

static method to request from the server to see if a lecture is being held in requested classID.

[ServerMain](#) - Class in [program](#)

The main program for running the server

[ServerMain\(\)](#) - Constructor for class [program.ServerMain](#)

[ServerManager](#) - Class in [program.connection](#)

The main Class used to handle everything related to the server, from sending info and getting the streams used for listening for info from the server

[ServerManager\(String, int\)](#) - Constructor for class [program.connection.ServerManager](#)

Starts the connection to the server

[ServerRequests](#) - Class in [program.connection](#)

Class with static methods to return the most basic needs from the server

[ServerRequests\(\)](#) - Constructor for class [program.connection.ServerRequests](#)

[setBackOnlyLocal\(boolean\)](#) - Method in class [program.ui.controllers.RootController](#)

Used so you can press the back button when the question box appear, without exiting the student window

[setClassID\(String\)](#) - Method in class [program.ClientMain](#)

Sets the class code the client is connected to.

Overview List

setClassID(String) - Method in class `program.connection.ClientConnection`

Setter for which subject this client is looking at

setDate(String) - Method in class `program.ui.controllers.lecturer.LectureBoxController`

sets the date of this lecture

setEndLecture(int) - Method in class `program.database.Database`

This method updates a given lecture with a time-stamp in the stop-coloumn

setId(int) - Method in class `program.common.Question`

sets the id of the question

setLectureID(int) - Method in class `program.ClientMain`

sets the ID of the lecture the client is connected to.

setLectureID(int) - Method in class `program.connection.ClientConnection`

Setter for which lecture this client is in

setLectureID(int) - Method in class `program.ui.controllers.lecturer.LectureBoxController`

sets the ID of the lecture this box is related to

setLectureName(String) - Method in class `program.ClientMain`

sets the name of the lecture the client is connected to.

setLectureName(String) - Method in class `program.ui.controllers.lecturer.LectureBoxController`

sets the name of the lecture to be accessible from this box

setLiveLectureID(int) - Method in interface `program.connection.listeners.QuestionReciever`

Receives and saves the returned lectureID

setLiveLectureID(int) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

setLiveLectureID(int) - Method in class `program.ui.controllers.student.StudentWindowController`

setMainApp(ClientMain) - Method in interface `program.ui.controllers.AppBinder`

Sets up necessary references to client main and gets start info from the database, if any

setMainApp(ClientMain) - Method in class `program.ui.controllers.lecturer.CreateLectureController`

setMainApp(ClientMain) - Method in class `program.ui.controllers.lecturer.LectureLoginController`

setMainApp(ClientMain) - Method in class `program.ui.controllers.lecturer.LectureOverviewController`

setMainApp(ClientMain) - Method in class `program.ui.controllers.lecturer.LectureReviewController`

setMainApp(ClientMain) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

setMainApp(ClientMain) - Method in class `program.ui.controllers.RoleSelectorController`

setMainApp(ClientMain) - Method in class `program.ui.controllers.RootController`

setMainApp(ClientMain) - Method in class `program.ui.controllers.student.LectureSelectorController`

setMainApp(ClientMain) - Method in class `program.ui.controllers.student.StudentWindowController`

setQuestion(Question) - Method in class `program.ui.controllers.QuestionBoxController`

Sets the question object for this window.

setQuestionText(String) - Method in class `program.common.Question`

sets the text of the question

setRating(int) - Method in class `program.common.Question`

sets the rating of the question

setRelatedQuestionPane(AnchorPane) - Method in class `program.common.Question`

Overview List

sets the window element related to this question

[setRole\(String\)](#) - Method in class `program.connection.ClientConnection`

Sets a role to the client

Possible roles (As of 26-04-2017):

[setScoreVisible\(boolean\)](#) - Method in class `program.ui.controllers.QuestionBoxController`

Method used to determine whether the score of the question should be visible

[setTime\(String\)](#) - Method in class `program.common.Question`

sets the time of the question

[setTitle\(String\)](#) - Method in class `program.ui.controllers.RootController`

sets the title in the top of the root controller, used to give the user info about where he or she is.

[setTitleAndNameText\(String, String\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

This method changes the title text and lecture name text

[skipAddPath\(\)](#) - Method in class `program.ui.controllers.RootController`

Used if you don't want to skip a view to add to the stack of backs, used on our initial loadui when starting the app.

[start\(Stage\)](#) - Method in class `program.ClientMain`

[startConnection\(String\)](#) - Method in class `program.ClientMain`

This method sets up our ServerManager object that is responsible for managing all requests and incoming info from the server.

[stop\(\)](#) - Method in class `program.ClientMain`

[stopListening\(\)](#) - Method in class `program.connection.listeners.ClientListener`

A Method made to interrupt the listener and make it escape the code hold from the input stream read

[studentJoined\(\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

Increments the number of students, and update ui elements

[studentLost\(\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

Increments the number of LOST students, and updates the pieChart.

[StudentWindowController](#) - Class in `program.ui.controllers.student`

Controller for the student view when attending a lecture

[StudentWindowController\(\)](#) - Constructor for class `program.ui.controllers.student.StudentWindowController`

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

U

[updatePieChartValues\(\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

This method checks how many students are lost, and updates the pie chart accordingly

[updateQuestionScore\(int, int\)](#) - Method in interface `program.connection.listeners.QuestionReciever`

Get notified that questions has changed score

[updateQuestionScore\(int, int\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

[updateQuestionScore\(int, int\)](#) - Method in class `program.ui.controllers.student.StudentWindowController`

[updateStudentsConnectedAmount\(\)](#) - Method in class
`program.ui.controllers.lecturer.LecturerWindowController`

This method checks connectedStudents and updates the text

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

V

[viewAlert\(\)](#) - Method in class `program.ClientMain`

[viewBackButton\(\)](#) - Method in class `program.ui.controllers.RootController`

[viewButton\(\)](#) - Method in class `program.ui.controllers.lecturer.LectureBoxController`

Getter to get the button that allows the lecturer to open the lecture.

[voteQuestion\(JSONObject, ClientConnection\)](#) - Method in class `program.connection.CommandsManager`

This method changes the score of a given question by a given increment, and informs all clients in the lecture

The keys used by the `JSONObject` is as follows:

[voteQuestion\(int, int\)](#) - Method in class `program.database.Database`

Updates and increases a given question's score by a value

[voteQuestion\(String\)](#) - Method in class `program.ui.controllers.QuestionBoxController`

Method used to vote on a the question in this window

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

Overview List

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

C

[ClientConnection](#) - Class in [program.connection](#)

This class handles the serverside information of a single connected client

[ClientConnection\(Socket, CommandsManager, ClientsManager\)](#) - Constructor for class [program.connection.ClientConnection](#)

This constructor initializes the reference objects and the socket

[ClientListener](#) - Class in [program.connection.listeners](#)

A Listener Class that is instantiated in a new thread to work as a listener to update info at the client side, without letting the application hold in a block of code.

[ClientListener\(ClientMain, QuestionReciever\)](#) - Constructor for class [program.connection.listeners.ClientListener](#)

[ClientMain](#) - Class in [program](#)

Main class that is instantiated to run the application

[ClientMain\(\)](#) - Constructor for class [program.ClientMain](#)

[ClientsManager](#) - Class in [program.connection](#)

This class decodes all incoming JSONObject and encodes and sends return-JSONs

[ClientsManager\(int, ServerMain\)](#) - Constructor for class [program.connection.ClientsManager](#)

This constructor initializes the reference objects, and starts the thread that listens for new connections

[close\(\)](#) - Method in class [program.database.Database](#)

[closeController\(\)](#) - Method in interface [program.ui.controllers.AppBinder](#)

Runs all necessary functions to close threads, if any Also notifies the server, if necessary(For example if a live lecture ends)

[closeController\(\)](#) - Method in class [program.ui.controllers.lecturer.CreateLectureController](#)

Overview List

[closeController\(\)](#) - Method in class `program.ui.controllers.lecturer`.[LectureLoginController](#)

[closeController\(\)](#) - Method in class `program.ui.controllers.lecturer`.[LectureOverviewController](#)

[closeController\(\)](#) - Method in class `program.ui.controllers.lecturer`.[LectureReviewController](#)

[closeController\(\)](#) - Method in class `program.ui.controllers.lecturer`.[LecturerWindowController](#)

[closeController\(\)](#) - Method in class `program.ui.controllers`.[RoleSelectorController](#)

[closeController\(\)](#) - Method in class `program.ui.controllers`.[RootController](#)

[closeController\(\)](#) - Method in class `program.ui.controllers.student`.[LectureSelectorController](#)

[closeController\(\)](#) - Method in class `program.ui.controllers.student`.[StudentWindowController](#)

[CommandsManager](#) - Class in `program.connection`

This class decodes all incoming JSONObjects and encodes and sends return-JSONs

[CommandsManager\(ClientsManager\)](#) - Constructor for class `program.connection.CommandsManager`

This constructor sets up the reference elements

[CommandsManager.Command](#) - Interface in `program.connection`

This interface is used to allows the stringToFunction map to refer to the different methods of this class

[connect\(\)](#) - Method in class `program.database.Database`

This method sets up a Connection object, and connects to to the database with the `KEY_URL`

[CreateLectureController](#) - Class in `program.ui.controllers.lecturer`

[CreateLectureController\(\)](#) - Constructor for class `program.ui.controllers.lecturer.CreateLectureController`

[createNewLecture\(String, String\)](#) - Method in class `program.database.Database`

This method creates a new lecture in a given subject

[createNewSubject\(String, String\)](#) - Method in class `program.database.Database`

This method adds a new subject to the database

[createYouLostMe\(int\)](#) - Method in class `program.database.Database`

This method saves a time-stamp for the lost student notifications in the database

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

D

[Database](#) - Class in [program.database](#)

This class is used to connect and run SQL-queries on the database To use this class, the library mysql-connector-java-5.1.40-bin.jar is needed A URL to gain access is also needed, and is provided in the class DatabaseKey (not checked in to our github)

[Database\(\)](#) - Constructor for class [program.database.Database](#)

[DatabaseKey](#) - Class in [program.database](#)

This class is only a holder for the URL needed to connect to the database OBS: This class must not be uploaded onto github.

[DatabaseKey\(\)](#) - Constructor for class [program.database.DatabaseKey](#)

[DEBUG](#) - Variable in class [program.ClientMain](#)

[displayAlert\(String, String, String\)](#) - Method in class [program.ClientMain](#)

This method is used to alert the user of something that happened, with a popup window.

[doesLectureExist\(String\)](#) - Method in class [program.connection.ClientsManager](#)

This method checks if a lecture is being held in a given subject

[doFunction\(JSONObject, ClientConnection\)](#) - Method in interface [program.connection.CommandsManager.Command](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)

Overview List

- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

E

[executeStatement\(String\)](#) - Method in class `program.database.Database`
This method executes a query given as a parameter

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

F

[fetchLectures\(\)](#) - Method in interface `program.connection.listeners.LectureReciever`

Requests all lectures from the database

[fetchLectures\(\)](#) - Method in class `program.ui.controllers.lecturer.LectureOverviewController`

[fetchLectures\(\)](#) - Method in class `program.ui.controllers.lecturer.LectureReviewController`

[fetchLiveLectureID\(\)](#) - Method in interface `program.connection.listeners.QuestionReciever`

Gets a specific ID for this live lecture so the client receives the correct questions

[fetchLiveLectureID\(\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

[fetchLiveLectureID\(\)](#) - Method in class `program.ui.controllers.student.StudentWindowController`

[fetchQuestions\(int\)](#) - Method in interface `program.connection.listeners.QuestionReciever`

Tell the server to send the most recent questions in the database

[fetchQuestions\(int\)](#) - Method in class `program.ui.controllers.lecturer.LecturerWindowController`

[fetchQuestions\(int\)](#) - Method in class `program.ui.controllers.student.StudentWindowController`

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)

Overview List

- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

G

[getActiveController\(\)](#) - Method in class [program.ClientMain](#)

Method used to get the controller that is currently active in the main window of the root window.

[getAllSubjectCodes\(ServerManager\)](#) - Static method in class [program.connection.ServerRequests](#)

Static method used to get all the subject codes the client can connect to

[getAllSubjectCodes\(\)](#) - Method in class [program.database.Database](#)

This method fetches all subject codes from the database

[getClassID\(\)](#) - Method in class [program.ClientMain](#)

getter to get the class code the client is connected to.

[getClassID\(\)](#) - Method in class [program.connection.ClientConnection](#)

Getter for which subject the client is looking at

[getClientsConnected\(\)](#) - Method in class [program.connection.ClientsManager](#)

Gets a complete list of all the clients connected to the socket

[getClientsManager\(\)](#) - Method in class [program.ServerMain](#)

Getter for the servers clients-manager

[getDatabase\(\)](#) - Method in class [program.ServerMain](#)

Getter for the database

[getId\(\)](#) - Method in class [program.common.Question](#)

getter to get ID of the question

[getInputStream\(\)](#) - Method in class [program.connection.ServerManager](#)

Getter to get the input stream of the socket

[getLastestQuestions\(int, int\)](#) - Method in class [program.database.Database](#)

This method gets a specified amount of questions from the database and returns the result

[getLatestLectures\(String\)](#) - Method in class [program.database.Database](#)

This method gets all info from all lectures in the database, for a specified subject-code

[getLectureBoxController\(int\)](#) - Method in class [program.ui.controllers.lecturer.LectureOverviewController](#)

Overview List

getLectureID() - Method in class program.ClientMain
getter to get the lecture ID the client is connected to

getLectureID() - Method in class program.connection.ClientConnection
Getter for the id for a lecture this client is in

getLectureID() - Method in class program.ui.controllers.lecturer.LectureBoxController
getter to get the id of the lecture

getLectureName() - Method in class program.ClientMain
getter to get the name of the lecture the client is connected to

getLecturer(String) - Method in class program.connection.ClientsManager
Gets the connection of the lecturer holding a lecture in a given subject

getLectureStats(int) - Method in class program.database.Database
Gets all info about a lecture

getLiveLectureID(String) - Method in class program.database.Database
Fetches the id of the latest lecture in a given subject-code

getLostMeTimerLenght() - Method in class program.ClientMain
Method used to get the length of the timer to controll the cooldown of "you lost me" feature

getLostMeTimestamps(int) - Method in class program.database.Database
This method fetches all time-stamps for lost students, for a given lecture

getOutputStream() - Method in class program.connection.ServerManager
getter to get the output stream of the socket

getQuestionBoxController(int) - Method in class program.ui.controllers.student.StudentWindowController
Used for unitTesting and debugging ONLY This method requires debug to be true, else it will always return null

getQuestionId() - Method in class program.ui.controllers.QuestionBoxController
getter to get the id of the question

getQuestionScore() - Method in class program.ui.controllers.QuestionBoxController

getQuestionText() - Method in class program.common.Question
getter to get the question text

getQuestionText() - Method in class program.ui.controllers.QuestionBoxController
getter to get the text of the question object.

getRating() - Method in class program.common.Question
getter for the rating of the question

getRelatedQuestionPane() - Method in class program.common.Question
getter to get the window element of the question list

getRootController() - Method in class program.ClientMain
Getter to get the controller of the root window.

getScoreQuestion(int) - Method in class program.database.Database
Fetches the score of a given question from the database

getServerManager() - Method in class program.ClientMain
Getter to get the servermanager.

getSocket() - Method in class program.connection.ServerManager
Getter to get the Socket object related to the server.

getTime() - Method in class program.common.Question
getter to get the time of the question

getVote() - Method in class program.ui.controllers.QuestionBoxController
getter to get the vote state of the question

goodProperty() - Method in class program.ui.controllers.QuestionBoxController
get the property of the string of vote.

A C D E F G H I K L M P Q R S U V

Skip navigation links

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

H

[hostAvailabilityCheck\(\)](#) - Method in class `program.ui.controllers`.[RoleSelectorController](#)
A check to see if the server with proposed IP address is running.

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

I

[initialize\(\)](#) - Method in class program.ui.controllers.lecturer.[LectureBoxController](#)
 inits view

[initialize\(\)](#) - Method in class program.ui.controllers.lecturer.[LecturerWindowController](#)
 inits the view

[initialize\(\)](#) - Method in class program.ui.controllers.[QuestionBoxController](#)
 inits the view.

[initialize\(\)](#) - Method in class program.ui.controllers.[RootController](#)
 inits view

[initialize\(\)](#) - Method in class program.ui.controllers.student.[StudentWindowController](#)
 inits view

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) **[I](#)** [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

K

[KEY_URL](#) - Static variable in class `program.database`. [DatabaseKey](#)

[A](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) **[I](#)** [K](#) [L](#) [M](#) [P](#) [Q](#) [R](#) [S](#) [U](#) [V](#)

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev Letter](#)
- [Next Letter](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

All Classes

- AppBinder
- ClientConnection
- ClientListener
- ClientMain
- ClientsManager
- CommandsManager
- CommandsManager.Command
- CreateLectureController
- Database
- DatabaseKey
- LectureBoxController
- LectureLoginController
- LectureOverviewController
- LectureReciever
- LectureReviewController
- LecturerWindowController
- LectureSelectorController
- LectureStatListener
- Question
- QuestionBoxController
- QuestionReciever
- RoleSelectorController
- RootController
- ServerMain
- ServerManager
- ServerRequests
- StudentWindowController

All Classes

- AppBinder
- ClientConnection
- ClientListener
- ClientMain
- ClientsManager
- CommandsManager
- CommandsManager.Command
- CreateLectureController
- Database
- DatabaseKey
- LectureBoxController
- LectureLoginController
- LectureOverviewController
- LectureReciever
- LectureReviewController
- LecturerWindowController
- LectureSelectorController
- LectureStatListener
- Question
- QuestionBoxController
- QuestionReciever
- RoleSelectorController
- RootController
- ServerMain
- ServerManager
- ServerRequests
- StudentWindowController

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [Prev](#)
- [Next](#)
- [Frames](#)
- [No Frames](#)
- [All Classes](#)

Constant Field Values

Contents

- [program.*](#)

program.*

•			
	Modifier and Type	Constant Field	Value
	public final boolean	DEBUG	false

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)
- [Prev](#)
- [Next](#)

Overview List

- Frames
- No Frames
- All Classes

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

Deprecated API

Contents

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

JavaScript is disabled on your browser.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)

How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

• Overview

The [Overview](#) page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

• Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain six categories:

- ◆ Interfaces (*italic*)
- ◆ Classes
- ◆ Enums
- ◆ Exceptions
- ◆ Errors
- ◆ Annotation Types

• Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- ◆ Class inheritance diagram
- ◆ Direct Subclasses
- ◆ All Known Subinterfaces

- ◆ All Known Implementing Classes
- ◆ Class/interface declaration
- ◆ Class/interface description
- ◆ Nested Class Summary
- ◆ Field Summary
- ◆ Constructor Summary
- ◆ Method Summary
- ◆ Field Detail
- ◆ Constructor Detail
- ◆ Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

• **Annotation Type**

Each annotation type has its own separate page with the following sections:

- ◆ Annotation Type declaration
- ◆ Annotation Type description
- ◆ Required Element Summary
- ◆ Optional Element Summary
- ◆ Element Detail

• **Enum**

Each enum has its own separate page with the following sections:

- ◆ Enum declaration
- ◆ Enum description
- ◆ Enum Constant Summary
- ◆ Enum Constant Detail

• **Use**

Each documented package, class and interface has its own Use page. This page describes what packages, classes, methods, constructors and fields use any part of the given class or package. Given a class or interface A, its Use page includes subclasses of A, fields declared as A, methods that return A, and methods and constructors with parameters of type A. You can access this page by first going to the package, class or interface, then clicking on the "Use" link in the navigation bar.

• **Tree (Class Hierarchy)**

There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- ◆ When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- ◆ When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

• **Deprecated API**

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

• **Index**

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

• **Prev/Next**

These links take you to the next or previous class, interface, package, or related page.

• **Frames/No Frames**

These links show and hide the HTML frames. All pages are available with or without frames.

• **All Classes**

The [All Classes](#) link shows all classes and interfaces except non-static nested types.

• **Serialized Form**

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

• **Constant Field Values**

The [Constant Field Values](#) page lists the static final fields and their values.

This help file applies to API documentation generated using the standard doclet.

[Skip navigation links](#)

- [Overview](#)
- [Package](#)
- [Class](#)
- [Use](#)
- [Tree](#)
- [Deprecated](#)
- [Index](#)
- [Help](#)

- [Prev](#)
- [Next](#)

- [Frames](#)
- [No Frames](#)

- [All Classes](#)