

My Project

Generated by Doxygen 1.8.17

Chapter 1

Namespace Index

1.1 Namespace List

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

com.mycompany.ssproject	This class represents a singleton object that stores user data	??
---	--	----

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

com.mycompany.ssproject.CopyFileController	??
com.mycompany.ssproject.CreateFileController	??
com.mycompany.ssproject.Database	??
com.mycompany.ssproject.DataSingleton	??
com.mycompany.ssproject.DeleteFileController	??
com.mycompany.ssproject.DirOperationsController	??
com.mycompany.ssproject.EditprofileController	??
com.mycompany.ssproject.errorCatcher	??
com.mycompany.ssproject.LoginController	??
com.mycompany.ssproject.MainController	??
com.mycompany.ssproject.MoveFileController	??
com.mycompany.ssproject.RecoverFileController	??
com.mycompany.ssproject.RegisterController	??
com.mycompany.ssproject.RenameFileController	??
com.mycompany.ssproject.ShareFileController	??
com.mycompany.ssproject.UnzipFilesController	??
com.mycompany.ssproject.UploadFileController	??
com.mycompany.ssproject.ZipFilesController	??
Application	
com.mycompany.ssproject.App	??
Initializable	
com.mycompany.ssproject.AdminPanelController	??
com.mycompany.ssproject.psController	??
com.mycompany.ssproject.TreeStructureController	??
com.mycompany.ssproject.whoamiController	??

Chapter 3

Class Index

3.1 Class List

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

com.mycompany.ssproject.AdminPanelController	??
com.mycompany.ssproject.App	??
com.mycompany.ssproject.CopyFileController	??
com.mycompany.ssproject.CreateFileController	
Controller class for the create file window	??
com.mycompany.ssproject.Database	
This class creates the SQLite database and the User and Admin tables	??
com.mycompany.ssproject.DataSingleton	??
com.mycompany.ssproject.DeleteFileController	??
com.mycompany.ssproject.DirOperationsController	??
com.mycompany.ssproject.EditprofileController	??
com.mycompany.ssproject.errorCatcher	
A class for catching and logging errors in the system software	??
com.mycompany.ssproject.LoginController	
This class controls the login process of the application	??
com.mycompany.ssproject.MainController	??
com.mycompany.ssproject.MoveFileController	??
com.mycompany.ssproject.psController	
A controller class for the ps.fxml file, which displays a list of running processes	??
com.mycompany.ssproject.RecoverFileController	??
com.mycompany.ssproject.RegisterController	??
com.mycompany.ssproject.RenameFileController	
Handles file renaming	??
com.mycompany.ssproject.ShareFileController	??
com.mycompany.ssproject.TreeStructureController	??
com.mycompany.ssproject.UnzipFilesController	??
com.mycompany.ssproject.UploadFileController	??
com.mycompany.ssproject.whoamiController	
This class is the controller for the whoami window	??
com.mycompany.ssproject.ZipFilesController	??

Chapter 4

File Index

4.1 File List

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

CreateFileController.java	This file contains the controller for the create file window	??
Database.java	This file contains the class Database, which is responsible for creating the SQLite database and the User and Admin tables	??
DirOperationsController.java	This file contains the DirOperationsController class, responsible for handling events related to directory operations in the application. It contains methods for creating and deleting directories, as well as navigating back to the main menu. The class uses a DataSingleton object to retrieve the directory path	??
errorCatcher.java	Contains the implementation of the errorCatcher class	??
LoginController.java	This file contains the implementation of the LoginController class	??
MoveFileController.java	This file contains the MoveFileController class, which is responsible for handling user interactions with the move file UI. The MoveFileController class is a JavaFX controller class that manages the user interactions with the move file UI. It contains methods for moving files from one directory to another, as well as navigating back to the main UI	??
psController.java	Contains the implementation of the psController class	??
RegisterController.java	Contains the RegisterController class which handles user registration	??
RenameFileController.java	Contains the RenameFileController class which handles file renaming	??
whoamiController.java	This file contains the implementation of the whoamiController class which is the controller for the whoami window	??
ZipFilesController.java	This is the controller class for the ZipFiles.fxml file. It contains methods to zip files and move them to a designated directory	??

Chapter 5

Namespace Documentation

5.1 Package com.mycompany.ssproject

This class represents a singleton object that stores user data.

Classes

- class [AdminPanelController](#)
- class [App](#)
- class [CopyFileController](#)
- class [CreateFileController](#)
Controller class for the create file window.
- class [Database](#)
This class creates the SQLite database and the User and Admin tables.
- class **DataSingletonAdmin**
- class [DataSingleton](#)
- class [DeleteFileController](#)
- class [DirOperationsController](#)
- class [EditprofileController](#)
- class [errorCatcher](#)
A class for catching and logging errors in the system software.
- class [LoginController](#)
This class controls the login process of the application.
- class [MainController](#)
- class [MoveFileController](#)
- class [psController](#)
A controller class for the ps.fxml file, which displays a list of running processes.
- class [RecoverFileController](#)
- class [RegisterController](#)
- class [RenameFileController](#)
Handles file renaming.
- class [ShareFileController](#)
- class [TreeStructureController](#)
- class [UnzipFilesController](#)
- class [UploadFileController](#)
- class [whoamiController](#)
This class is the controller for the whoami window.
- class [ZipFilesController](#)

5.1.1 Detailed Description

This class represents a singleton object that stores user data.

The [AdminPanelController](#) class is a controller for the Admin Panel view of the software system. It initializes and manages the Error Table view of the admin panel, which shows a list of logged errors. It also includes methods for logging out and returning to the main menu.

Main class of the System Software Project. This class extends `javafx.application.Application` and is responsible for initializing the JavaFX application and setting up the scene.

Controller class for the main window of the application.

FXML Controller class for recovering a file. The controller is responsible for handling user input and executing a shell command to move the file from the trash bin to the user's directory. The controller is also responsible for navigating back to the main page. The controller contains the following public methods: `recoverfile(ActionEvent event)`: handles the file recovery action by executing a shell command to move the file from the trash bin to the user's directory. `backtoMain(ActionEvent event)`: handles navigation back to the main page. The controller contains the following private fields: `thefolder`: a `TextField` for user input of the file to be recovered. `data`: an instance of the [DataSingleton](#) class for accessing user data such as the directory and username. `Directory`: a `String` that represents the directory of the user. `userName`: a `String` that represents the username of the user. The controller contains the following private methods: `None`.

Author

ntu-user

This class is the controller for the `ShareFile.fxml` view. It allows the user to share a file or share a file with read-only access to other users on the system. The class contains the methods to handle the user's actions in the view, including checking if the recipient user exists, sharing the file, and going back to the main view.

Author

ntu-user

The [TreeStructureController](#) class represents the controller for the tree structure view. It initializes the view with the root directory and populates the tree with subdirectories and files.

This package contains the controller for the Unzip Files feature of the System Software application. It includes methods for unzipping a file, copying a file to the app container, and navigating back to the main page. It uses a [DataSingleton](#) instance to get the current user's username and create a directory path for the unzipped file.

FXML Controller class for uploading a file to the app container.

Author

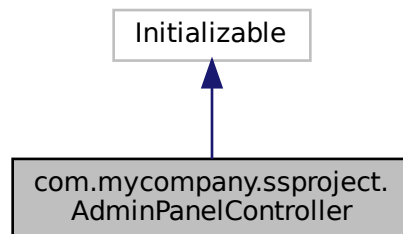
ntu-user

Chapter 6

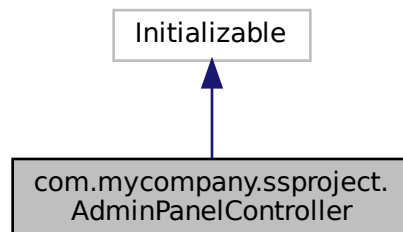
Class Documentation

6.1 com.mycompany.ssproject.AdminPanelController Class Reference

Inheritance diagram for com.mycompany.ssproject.AdminPanelController:



Collaboration diagram for com.mycompany.ssproject.AdminPanelController:



Public Member Functions

- void [initialize](#) (URL location, ResourceBundle resources)

6.1.1 Member Function Documentation

6.1.1.1 initialize()

```
void com.mycompany.ssproject.AdminPanelController.initialize (
    URL location,
    ResourceBundle resources ) [inline]
```

Initializes the Error Table view of the admin panel.

Parameters

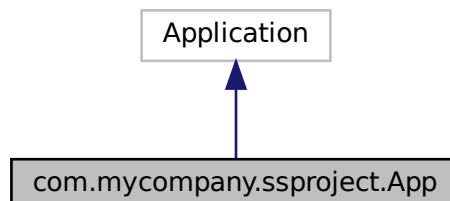
<i>location</i>	The location used to resolve relative paths for the root object.
<i>resources</i>	The resources used to localize the root object.

The documentation for this class was generated from the following file:

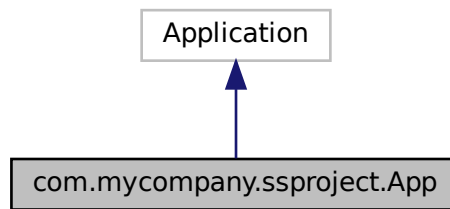
- AdminPanelController.java

6.2 com.mycompany.ssproject.App Class Reference

Inheritance diagram for com.mycompany.ssproject.App:



Collaboration diagram for com.mycompany.ssproject.App:



Public Member Functions

- void `start` (Stage stage) throws IOException

Static Public Member Functions

- static void `main` (String[] args)

6.2.1 Detailed Description

The main class of the application.

6.2.2 Member Function Documentation

6.2.2.1 `main()`

```
static void com.mycompany.ssproject.App.main (  
    String[] args ) [inline], [static]
```

The main method of the application.

Parameters

<code>args</code>	The command line arguments.
-------------------	-----------------------------

6.2.2.2 start()

```
void com.mycompany.ssproject.App.start (
    Stage stage ) throws IOException [inline]
```

The start method of the application. Initializes the scene by loading the login fxml file and setting up the stage.

Parameters

<i>stage</i>	The stage to use for the application.
--------------	---------------------------------------

Exceptions

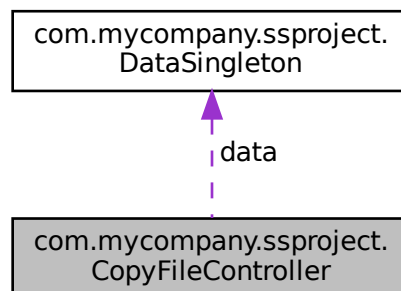
<i>IOException</i>	if there is an error loading the login fxml file.
--------------------	---

The documentation for this class was generated from the following file:

- App.java

6.3 com.mycompany.ssproject.CopyFileController Class Reference

Collaboration diagram for com.mycompany.ssproject.CopyFileController:



Public Attributes

- String `Directory` = `data.getDirectory()`

6.3.1 Detailed Description

FXML Controller class for the copy file screen.

6.3.2 Member Data Documentation

6.3.2.1 Directory

```
String com.mycompany.ssproject.CopyFileController.Directory = data.getDirectory()
```

The directory where the file will be copied from.

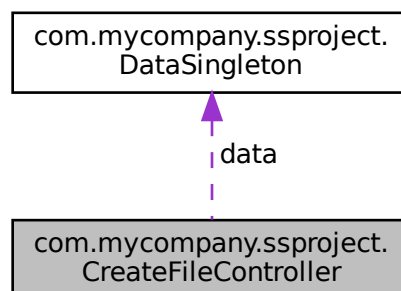
The documentation for this class was generated from the following file:

- CopyFileController.java

6.4 com.mycompany.ssproject.CreateFileController Class Reference

Controller class for the create file window.

Collaboration diagram for com.mycompany.ssproject.CreateFileController:



Public Attributes

- String **Directory** = data.getDirectory()

6.4.1 Detailed Description

Controller class for the create file window.

This class is responsible for handling user input and creating a new file in the selected directory.

The documentation for this class was generated from the following file:

- [CreateFileController.java](#)

6.5 com.mycompany.ssproject.Database Class Reference

This class creates the SQLite database and the User and Admin tables.

Static Public Member Functions

- static void [createDatabase](#) ()
- static void [createUserTable](#) ()
- static void [createAdminTable](#) ()
- static void [main](#) (String args[])

6.5.1 Detailed Description

This class creates the SQLite database and the User and Admin tables.

6.5.2 Member Function Documentation

6.5.2.1 createAdminTable()

```
static void com.mycompany.ssproject.Database.createAdminTable ( ) [inline], [static]
```

This method creates the Admin table in the SQLite database.

6.5.2.2 createDatabase()

```
static void com.mycompany.ssproject.Database.createDatabase ( ) [inline], [static]
```

This method creates the SQLite database.

6.5.2.3 createUserTable()

```
static void com.mycompany.ssproject.Database.createUserTable ( ) [inline], [static]
```

This method creates the User table in the SQLite database.

6.5.2.4 main()

```
static void com.mycompany.ssproject.Database.main (  
    String args[] ) [inline], [static]
```

This is the main method of the [Database](#) class. It creates the SQLite database and the User and Admin tables.

Parameters

<i>args</i>	The command line arguments.
-------------	-----------------------------

The documentation for this class was generated from the following file:

- [Database.java](#)

6.6 com.mycompany.ssproject.DataSingleton Class Reference

Public Member Functions

- int [getUserId](#) ()
The method to get the user ID.
- void [setUserId](#) (int id)
The method to set the user ID.
- String [getUserName](#) ()
The method to get the username.
- String [getPassword](#) ()
The method to get the user password.
- void [setPassword](#) (String password)
The method to set the user password.
- void [setUserName](#) (String username)
The method to set the username.
- String [getDirectory](#) ()
The method to get the directory path.
- void [setDirectory](#) (String Directory)
The method to set the directory path.

Static Public Member Functions

- static [DataSingleton](#) [getInstance](#) ()
The method to get the instance of the singleton object.

6.6.1 Member Function Documentation

6.6.1.1 [getDirectory\(\)](#)

```
String com.mycompany.ssproject.DataSingleton.getDirectory ( ) [inline]
```

The method to get the directory path.

Returns

The directory path.

6.6.1.2 getInstance()

```
static DataSingleton com.mycompany.ssproject.DataSingleton.getInstance ( ) [inline], [static]
```

The method to get the instance of the singleton object.

Returns

The instance of the singleton object.

6.6.1.3 getPassword()

```
String com.mycompany.ssproject.DataSingleton.getPassword ( ) [inline]
```

The method to get the user password.

Returns

The user password.

6.6.1.4 getUserId()

```
int com.mycompany.ssproject.DataSingleton.getUserId ( ) [inline]
```

The method to get the user ID.

Returns

The user ID.

6.6.1.5 getUsername()

```
String com.mycompany.ssproject.DataSingleton.getUserName ( ) [inline]
```

The method to get the username.

Returns

The username.

6.6.1.6 setDirectory()

```
void com.mycompany.ssproject.DataSingleton.setDirectory (
    String Directory ) [inline]
```

The method to set the directory path.

Parameters

<i>Directory</i>	The directory path to set.
------------------	----------------------------

6.6.1.7 setPassword()

```
void com.mycompany.ssproject.DataSingleton.setPassword (
    String password ) [inline]
```

The method to set the user password.

Parameters

<i>password</i>	The user password to set.
-----------------	---------------------------

6.6.1.8 setUserId()

```
void com.mycompany.ssproject.DataSingleton.setUserId (
    int id ) [inline]
```

The method to set the user ID.

Parameters

<i>id</i>	The user ID to set.
-----------	---------------------

6.6.1.9 setUsername()

```
void com.mycompany.ssproject.DataSingleton.setUsername (
    String username ) [inline]
```

The method to set the username.

Parameters

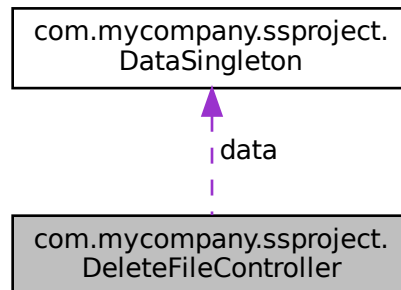
<i>username</i>	The username to set.
-----------------	----------------------

The documentation for this class was generated from the following file:

- DataSingleton.java

6.7 com.mycompany.ssproject.DeleteFileController Class Reference

Collaboration diagram for com.mycompany.ssproject.DeleteFileController:



Public Attributes

- String `Directory` = `data.getDirectory()`

6.7.1 Detailed Description

The `DeleteFileController` class is responsible for handling events related to deleting a file in the application.

It contains methods for creating a "deleted" directory, moving files to the "deleted" directory, checking if the directory exists,

and deleting a file.

Author

ntu-user

6.7.2 Member Data Documentation

6.7.2.1 Directory

```
String com.mycompany.ssproject.DeleteFileController.Directory = data.getDirectory()
```

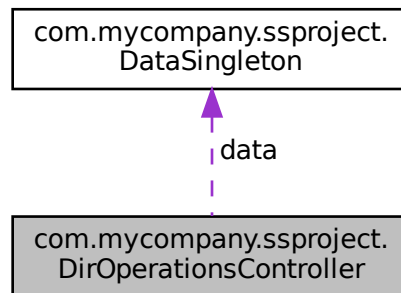
The directory path.

The documentation for this class was generated from the following file:

- `DeleteFileController.java`

6.8 com.mycompany.ssproject.DirOperationsController Class Reference

Collaboration diagram for com.mycompany.ssproject.DirOperationsController:



Public Attributes

- String `Directory` = `data.getDirectory()`

6.8.1 Member Data Documentation

6.8.1.1 Directory

```
String com.mycompany.ssproject.DirOperationsController.Directory = data.getDirectory()
```

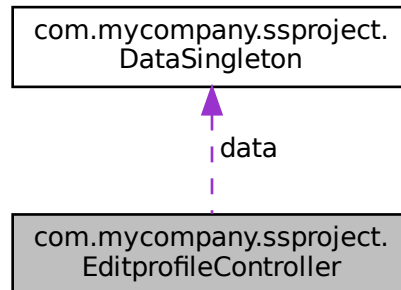
The directory path.

The documentation for this class was generated from the following file:

- [DirOperationsController.java](#)

6.9 com.mycompany.ssproject.EditprofileController Class Reference

Collaboration diagram for com.mycompany.ssproject.EditprofileController:



Public Member Functions

- boolean `editform` (String Username, String Password, int Id)

6.9.1 Detailed Description

FXML Controller class for Edit Profile Handles the updating of user profile information such as username and password

Author

ntu-user

6.9.2 Member Function Documentation

6.9.2.1 `editform()`

```
boolean com.mycompany.ssproject.EditprofileController.editform (
    String Username,
    String Password,
    int Id ) [inline]
```

Updates user's information in the database

Parameters

<i>Username</i>	new username
<i>Password</i>	new password
<i>Id</i>	id of user in database

Returns

boolean value indicating success/failure of update

The documentation for this class was generated from the following file:

- EditprofileController.java

6.10 com.mycompany.ssproject.errorCatcher Class Reference

A class for catching and logging errors in the system software.

Static Public Member Functions

- static void [insertintoError](#) (String Username, String Error, String Location)
Inserts an error message into the Admin table of the systemSoftwareDatabase.
- static void [setError](#) (String Username, String Error, String Location)
Sets an error message to be logged.

6.10.1 Detailed Description

A class for catching and logging errors in the system software.

6.10.2 Member Function Documentation

6.10.2.1 insertintoError()

```
static void com.mycompany.ssproject.errorCatcher.insertintoError (
    String Username,
    String Error,
    String Location ) [inline], [static]
```

Inserts an error message into the Admin table of the systemSoftwareDatabase.

Parameters

<i>Username</i>	The username associated with the error
<i>Error</i>	The error message
<i>Location</i>	The location where the error occurred

6.10.2.2 setError()

```
static void com.mycompany.ssproject.errorCatcher.setError (
    String Username,
    String Error,
    String Location ) [inline], [static]
```

Sets an error message to be logged.

Parameters

<i>Username</i>	The username associated with the error
<i>Error</i>	The error message
<i>Location</i>	The location where the error occurred

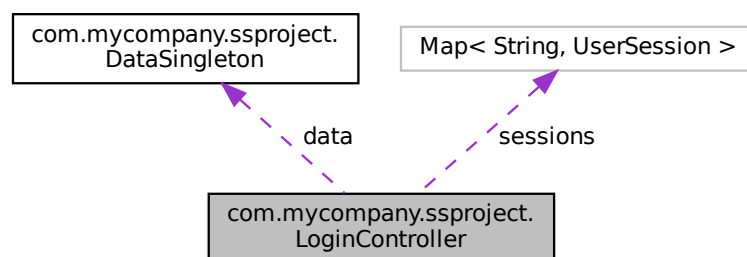
The documentation for this class was generated from the following file:

- [errorCatcher.java](#)

6.11 com.mycompany.ssproject.LoginController Class Reference

This class controls the login process of the application.

Collaboration diagram for com.mycompany.ssproject.LoginController:



Classes

- class `UserSession`

Public Member Functions

- void [sessionSave](#) (String Username)
Saves the user session.
- boolean [sessionQuery](#) (String Username) throws IOException, InterruptedException
Queries the user session.
- boolean [validateUser](#) (String user, String pass, String tabName) throws IOException, InterruptedException
Validates the given user credentials and logs the user in if they are correct.

Static Public Member Functions

- static void [listen](#) ()

Static Public Attributes

- static Map< String, UserSession > [sessions](#) = new HashMap<>()
A map that stores the user sessions.

6.11.1 Detailed Description

This class controls the login process of the application.

6.11.2 Member Function Documentation

6.11.2.1 [listen\(\)](#)

```
static void com.mycompany.ssproject.LoginController.listen ( ) [inline], [static]
```

Listens for incoming client connections and verifies whether their session is valid.

6.11.2.2 [sessionQuery\(\)](#)

```
boolean com.mycompany.ssproject.LoginController.sessionQuery (
    String Username ) throws IOException, InterruptedException [inline]
```

Queries the user session.

Parameters

<i>Username</i>	the username of the user.
-----------------	---------------------------

Returns

true if the user session exists, false otherwise.

Exceptions

<i>IOException</i>	if the file for the user session cannot be read.
<i>InterruptedException</i>	if the operation is interrupted.

6.11.2.3 sessionSave()

```
void com.mycompany.ssproject.LoginController.sessionSave (  
    String Username ) [inline]
```

Saves the user session.

Parameters

<i>Username</i>	the username of the user.
-----------------	---------------------------

6.11.2.4 validateUser()

```
boolean com.mycompany.ssproject.LoginController.validateUser (  
    String user,  
    String pass,  
    String tabName ) throws IOException, InterruptedException [inline]
```

Validates the given user credentials and logs the user in if they are correct.

Parameters

<i>user</i>	The username to validate.
<i>pass</i>	The password to validate.
<i>tabName</i>	The name of the table to query the user data from.

Exceptions

<i>IOException</i>	If there is an error with input/output.
<i>InterruptedException</i>	If the thread is interrupted.

Returns

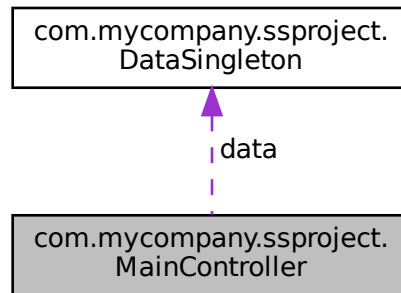
True if the user credentials are valid and the user is logged in, false otherwise.

The documentation for this class was generated from the following file:

- [LoginController.java](#)

6.12 com.mycompany.ssproject.MainController Class Reference

Collaboration diagram for com.mycompany.ssproject.MainController:



Public Member Functions

- void `initialize` () throws `IOException`, `InterruptedException`

Public Attributes

- String **userName** = `data.getUserName().toString()`
- String **Directory** = `"/datastore1/userdir/" + userName + "/"`

6.12.1 Detailed Description

Constructor that calls `deleteAfterThirtyDays()` method upon instantiation of object.

6.12.2 Member Function Documentation

6.12.2.1 `initialize()`

```
void com.mycompany.ssproject.MainController.initialize ( ) throws IOException, InterruptedException [inline]
```

Initializes the controller class.

Exceptions

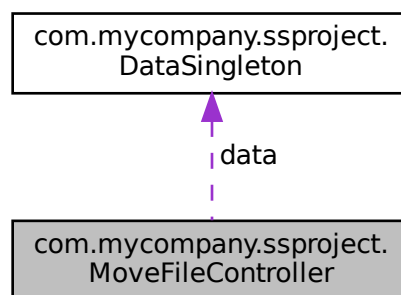
<i>IOException</i>	If an I/O error occurs.
<i>InterruptedException</i>	If a thread is interrupted.

The documentation for this class was generated from the following file:

- MainController.java

6.13 com.mycompany.ssproject.MoveFileController Class Reference

Collaboration diagram for com.mycompany.ssproject.MoveFileController:



Public Attributes

- String `Directory` = `data.getDirectory()`

6.13.1 Detailed Description

The `MoveFileController` class is a JavaFX controller class that manages the user interactions with the move file UI.

6.13.2 Member Data Documentation

6.13.2.1 Directory

```
String com.mycompany.ssproject.MoveFileController.Directory = data.getDirectory()
```

The Directory variable stores the current directory that the user is working with.

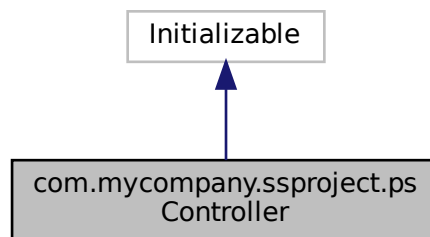
The documentation for this class was generated from the following file:

- [MoveFileController.java](#)

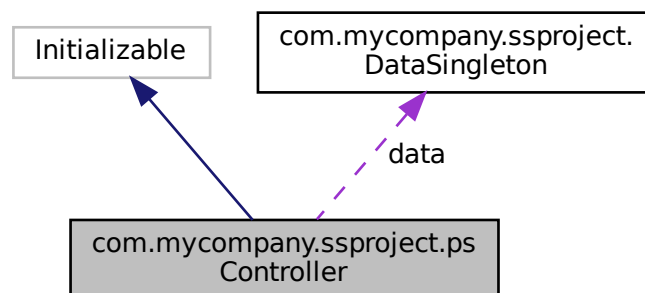
6.14 com.mycompany.ssproject.psController Class Reference

A controller class for the ps.fxml file, which displays a list of running processes.

Inheritance diagram for com.mycompany.ssproject.psController:



Collaboration diagram for com.mycompany.ssproject.psController:



Public Member Functions

- void [initialize](#) (URL url, ResourceBundle rb)
Initializes the controller class.

Public Attributes

- String **Directory** = data.getDirectory()

6.14.1 Detailed Description

A controller class for the ps.fxml file, which displays a list of running processes.

FXML Controller class

Author

ntu-user

6.14.2 Member Function Documentation

6.14.2.1 initialize()

```
void com.mycompany.ssproject.psController.initialize (  
    URL url,  
    ResourceBundle rb ) [inline]
```

Initializes the controller class.

< The TreeView that displays the list of running processes

Parameters

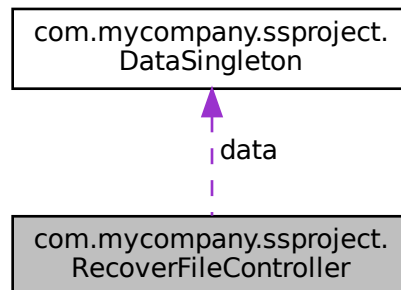
<i>url</i>	The location used to resolve relative paths for the root object
<i>rb</i>	The resources used to localize the root object

The documentation for this class was generated from the following file:

- [psController.java](#)

6.15 com.mycompany.ssproject.RecoverFileController Class Reference

Collaboration diagram for com.mycompany.ssproject.RecoverFileController:



Public Attributes

- String **Directory** = data.getDirectory()
- String **userName** = data.getUserName().toString()

The documentation for this class was generated from the following file:

- RecoverFileController.java

6.16 com.mycompany.ssproject.RegisterController Class Reference

Public Member Functions

- boolean [insert](#) (String Username, String Password)
Inserts the username and password into the database.

6.16.1 Detailed Description

user registration

6.16.2 Member Function Documentation

6.16.2.1 insert()

```
boolean com.mycompany.ssproject.RegisterController.insert (
    String Username,
    String Password ) [inline]
```

Inserts the username and password into the database.

Parameters

<i>Username</i>	the username to insert
<i>Password</i>	the password to insert

Returns

true if the insertion was successful, false otherwise

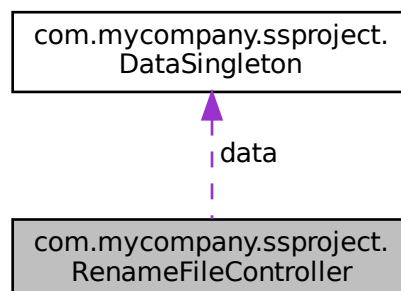
The documentation for this class was generated from the following file:

- [RegisterController.java](#)

6.17 com.mycompany.ssproject.RenameFileController Class Reference

Handles file renaming.

Collaboration diagram for com.mycompany.ssproject.RenameFileController:

**Public Attributes**

- String `Directory` = `data.getDirectory()`
Directory path to navigate to.

6.17.1 Detailed Description

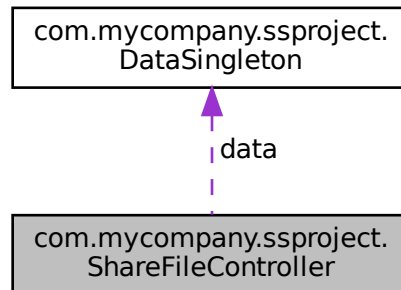
Handles file renaming.

The documentation for this class was generated from the following file:

- [RenameFileController.java](#)

6.18 com.mycompany.ssproject.ShareFileController Class Reference

Collaboration diagram for com.mycompany.ssproject.ShareFileController:



Public Attributes

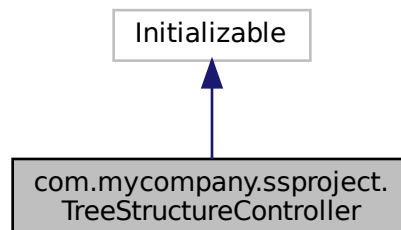
- String **Directory** = data.getDirectory()
- String **userName** = data.getUserName().toString()

The documentation for this class was generated from the following file:

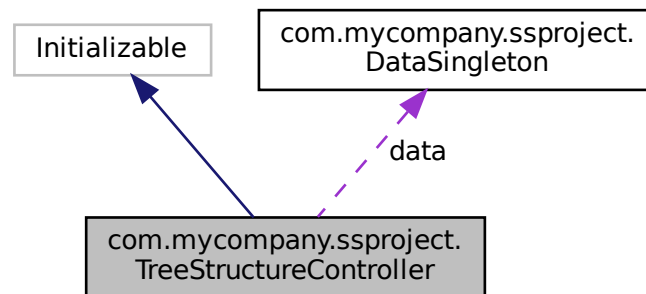
- ShareFileController.java

6.19 com.mycompany.ssproject.TreeStructureController Class Reference

Inheritance diagram for com.mycompany.ssproject.TreeStructureController:



Collaboration diagram for `com.mycompany.ssproject.TreeStructureController`:



Public Member Functions

- void [initialize](#) (URL *url*, ResourceBundle *rb*)

Public Attributes

- String **Directory** = `data.getDirectory()`

6.19.1 Detailed Description

The [TreeStructureController](#) class represents the controller for the tree structure view.

It initializes the view with the root directory and populates the tree with subdirectories and files.

6.19.2 Member Function Documentation

6.19.2.1 initialize()

```
void com.mycompany.ssproject.TreeStructureController.initialize (
    URL url,
    ResourceBundle rb ) [inline]
```

Initializes the controller class and starts the process to populate the tree.

Parameters

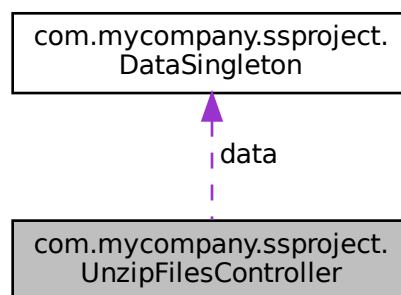
<i>url</i>	The URL of the FXML file.
<i>rb</i>	The ResourceBundle object.

The documentation for this class was generated from the following file:

- TreeStructureController.java

6.20 com.mycompany.ssproject.UnzipFilesController Class Reference

Collaboration diagram for com.mycompany.ssproject.UnzipFilesController:



Public Attributes

- String **userName** = data.getUserName().toString()
- String **Directory** = "/datastore1/userdir/" + userName + "/"

6.20.1 Detailed Description

FXML Controller class

Author

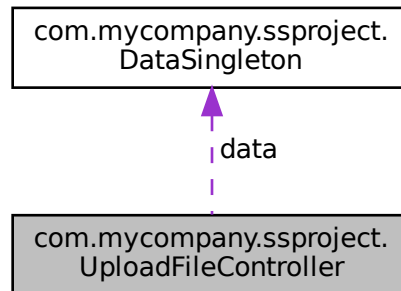
ntu-user

The documentation for this class was generated from the following file:

- UnzipFilesController.java

6.21 com.mycompany.ssproject.UploadFileController Class Reference

Collaboration diagram for com.mycompany.ssproject.UploadFileController:



Public Attributes

- String **Directory** = data.getDirectory()

6.21.1 Detailed Description

FXML Controller class

Author

ntu-user

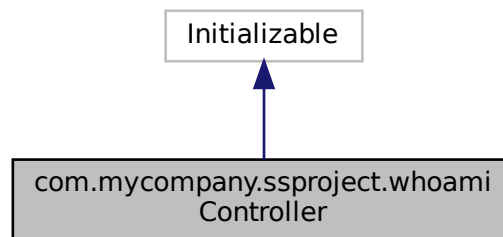
The documentation for this class was generated from the following file:

- UploadFileController.java

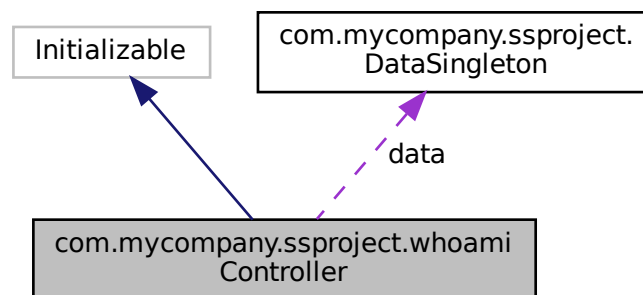
6.22 com.mycompany.ssproject.whoamiController Class Reference

This class is the controller for the whoami window.

Inheritance diagram for com.mycompany.ssproject.whoamiController:



Collaboration diagram for com.mycompany.ssproject.whoamiController:



Public Member Functions

- void `initialize` (URL url, ResourceBundle rb)
This function initializes the controller class.

Public Attributes

- String `userName` = `data.getUserName().toString()`

6.22.1 Detailed Description

This class is the controller for the whoami window.

Author

ntu-user

6.22.2 Member Function Documentation

6.22.2.1 initialize()

```
void com.mycompany.ssproject.whoamiController.initialize (
    URL url,
    ResourceBundle rb ) [inline]
```

This function initializes the controller class.

Parameters

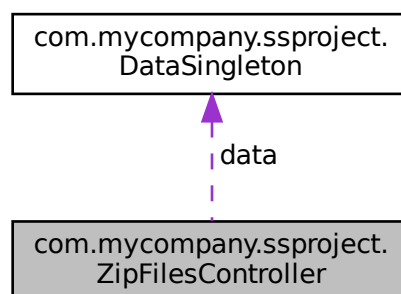
<i>url</i>	The URL to initialize the class.
<i>rb</i>	The ResourceBundle to initialize the class.

The documentation for this class was generated from the following file:

- [whoamiController.java](#)

6.23 com.mycompany.ssproject.ZipFilesController Class Reference

Collaboration diagram for com.mycompany.ssproject.ZipFilesController:



Public Attributes

- String **userName** = `data.getUserName().toString()`
- String **Directory** = `"/datastore1/userdir/" + userName + "/"`

6.23.1 Detailed Description

This class is the controller for the ZipFiles.fxml file. It contains methods to zip files and move them to a designated directory. FXML Controller class

Author

ntu-user

The documentation for this class was generated from the following file:

- [ZipFilesController.java](#)

Chapter 7

File Documentation

7.1 CreateFileController.java File Reference

This file contains the controller for the create file window.

Classes

- class [com.mycompany.ssproject.CreateFileController](#)
Controller class for the create file window.

Packages

- package [com.mycompany.ssproject](#)
This class represents a singleton object that stores user data.

7.1.1 Detailed Description

This file contains the controller for the create file window.

7.2 Database.java File Reference

This file contains the class Database, which is responsible for creating the SQLite database and the User and Admin tables.

Classes

- class [com.mycompany.ssproject.Database](#)
This class creates the SQLite database and the User and Admin tables.

Packages

- package [com.mycompany.ssproject](#)

This class represents a singleton object that stores user data.

7.2.1 Detailed Description

This file contains the class Database, which is responsible for creating the SQLite database and the User and Admin tables.

7.3 DirOperationsController.java File Reference

This file contains the DirOperationsController class, responsible for handling events related to directory operations in the application. It contains methods for creating and deleting directories, as well as navigating back to the main menu. The class uses a DataSingleton object to retrieve the directory path.

Classes

- class [com.mycompany.ssproject.DirOperationsController](#)

Packages

- package [com.mycompany.ssproject](#)

This class represents a singleton object that stores user data.

7.3.1 Detailed Description

This file contains the DirOperationsController class, responsible for handling events related to directory operations in the application. It contains methods for creating and deleting directories, as well as navigating back to the main menu. The class uses a DataSingleton object to retrieve the directory path.

Author

ntu-user

7.4 errorCatcher.java File Reference

Contains the implementation of the errorCatcher class.

Classes

- class [com.mycompany.ssproject.errorCatcher](#)

A class for catching and logging errors in the system software.

Packages

- package [com.mycompany.ssproject](#)

This class represents a singleton object that stores user data.

7.4.1 Detailed Description

Contains the implementation of the errorCatcher class.

7.5 LoginController.java File Reference

This file contains the implementation of the LoginController class.

Classes

- class [com.mycompany.ssproject.LoginController](#)
This class controls the login process of the application.
- class [com.mycompany.ssproject.LoginController.UserSession](#)

Packages

- package [com.mycompany.ssproject](#)

This class represents a singleton object that stores user data.

7.5.1 Detailed Description

This file contains the implementation of the LoginController class.

7.6 MoveFileController.java File Reference

This file contains the MoveFileController class, which is responsible for handling user interactions with the move file UI. The MoveFileController class is a JavaFX controller class that manages the user interactions with the move file UI. It contains methods for moving files from one directory to another, as well as navigating back to the main UI.

Classes

- class [com.mycompany.ssproject.MoveFileController](#)

Packages

- package [com.mycompany.ssproject](#)

This class represents a singleton object that stores user data.

7.6.1 Detailed Description

This file contains the MoveFileController class, which is responsible for handling user interactions with the move file UI. The MoveFileController class is a JavaFX controller class that manages the user interactions with the move file UI. It contains methods for moving files from one directory to another, as well as navigating back to the main UI.

Author

ntu-user

Date

2023-04-23

7.7 psController.java File Reference

Contains the implementation of the psController class.

Classes

- class [com.mycompany.ssproject.psController](#)
A controller class for the ps.fxml file, which displays a list of running processes.

Packages

- package [com.mycompany.ssproject](#)
This class represents a singleton object that stores user data.

7.7.1 Detailed Description

Contains the implementation of the psController class.

7.8 RegisterController.java File Reference

Contains the RegisterController class which handles user registration.

Classes

- class [com.mycompany.ssproject.RegisterController](#)

Packages

- package [com.mycompany.ssproject](#)
This class represents a singleton object that stores user data.

7.8.1 Detailed Description

Contains the RegisterController class which handles user registration.

7.9 RenameFileController.java File Reference

Contains the RenameFileController class which handles file renaming.

Classes

- class [com.mycompany.ssproject.RenameFileController](#)
Handles file renaming.

Packages

- package [com.mycompany.ssproject](#)
This class represents a singleton object that stores user data.

7.9.1 Detailed Description

Contains the RenameFileController class which handles file renaming.

7.10 whoamiController.java File Reference

This file contains the implementation of the whoamiController class which is the controller for the whoami window.

Classes

- class [com.mycompany.ssproject.whoamiController](#)
This class is the controller for the whoami window.

Packages

- package [com.mycompany.ssproject](#)
This class represents a singleton object that stores user data.

7.10.1 Detailed Description

This file contains the implementation of the whoamiController class which is the controller for the whoami window.

Author

ntu-user

Date

2023-04-23

7.11 ZipFilesController.java File Reference

This is the controller class for the ZipFiles.fxml file. It contains methods to zip files and move them to a designated directory.

Classes

- class [com.mycompany.ssproject.ZipFilesController](#)

Packages

- package [com.mycompany.ssproject](#)

This class represents a singleton object that stores user data.

7.11.1 Detailed Description

This is the controller class for the ZipFiles.fxml file. It contains methods to zip files and move them to a designated directory.

Author

ntu-user