

PureHTML

Generated by Doxygen 1.9.1



<b>1 Namespace Index</b>	<b>1</b>
1.1 Packages	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 File Index</b>	<b>7</b>
4.1 File List	7
<b>5 Namespace Documentation</b>	<b>9</b>
5.1 Package it.polimi.tiw.beans	9
5.2 Package it.polimi.tiw.controllers	9
5.3 Package it.polimi.tiw.dao	9
5.4 Package it.polimi.tiw.utils	10
<b>6 Class Documentation</b>	<b>11</b>
6.1 it.polimi.tiw.controllers.CheckLogin Class Reference	11
6.1.1 Detailed Description	12
6.1.2 Constructor & Destructor Documentation	12
6.1.2.1 CheckLogin()	12
6.1.3 Member Function Documentation	12
6.1.3.1 destroy()	13
6.1.3.2 doGet()	13
6.1.3.3 doPost()	13
6.1.3.4 init()	14
6.2 it.polimi.tiw.controllers.CheckRegistrazione Class Reference	15
6.2.1 Detailed Description	16
6.2.2 Constructor & Destructor Documentation	16
6.2.2.1 CheckRegistrazione()	16
6.2.3 Member Function Documentation	16
6.2.3.1 destroy()	16
6.2.3.2 doGet()	17
6.2.3.3 doPost()	17
6.2.3.4 init()	18
6.3 it.polimi.tiw.utils.ConnectionHandler Class Reference	19
6.3.1 Detailed Description	19
6.3.2 Member Function Documentation	19
6.3.2.1 closeConnection()	19
6.3.2.2 getConnection()	20
6.4 it.polimi.tiw.controllers.CreateContenuti Class Reference	22
6.4.1 Detailed Description	23

6.4.2 Constructor & Destructor Documentation	23
6.4.2.1 CreateContenuti()	23
6.4.3 Member Function Documentation	23
6.4.3.1 destroy()	23
6.4.3.2 doGet()	24
6.4.3.3 doPost()	24
6.4.3.4 init()	25
6.5 it.polimi.tiw.controllers.CreateDescrizione Class Reference	26
6.5.1 Detailed Description	27
6.5.2 Constructor & Destructor Documentation	27
6.5.2.1 CreateDescrizione()	27
6.5.3 Member Function Documentation	27
6.5.3.1 destroy()	28
6.5.3.2 doGet()	28
6.5.3.3 doPost()	29
6.5.3.4 init()	30
6.6 it.polimi.tiw.controllers.CreateFile Class Reference	31
6.6.1 Detailed Description	32
6.6.2 Constructor & Destructor Documentation	32
6.6.2.1 CreateFile()	32
6.6.3 Member Function Documentation	32
6.6.3.1 destroy()	33
6.6.3.2 doGet()	33
6.6.3.3 doPost()	34
6.6.3.4 init()	34
6.7 it.polimi.tiw.controllers.CreateGestioneContenuti Class Reference	35
6.7.1 Detailed Description	36
6.7.2 Constructor & Destructor Documentation	36
6.7.2.1 CreateGestioneContenuti()	36
6.7.3 Member Function Documentation	37
6.7.3.1 destroy()	37
6.7.3.2 doGet()	37
6.7.3.3 doPost()	38
6.7.3.4 init()	39
6.8 it.polimi.tiw.controllers.CreateHome Class Reference	40
6.8.1 Detailed Description	41
6.8.2 Constructor & Destructor Documentation	41
6.8.2.1 CreateHome()	41
6.8.3 Member Function Documentation	41
6.8.3.1 destroy()	41
6.8.3.2 doGet()	41
6.8.3.3 doPost()	42

6.8.3.4 init()	43
6.9 it.polimi.tiw.controllers.CreateSposta Class Reference	44
6.9.1 Detailed Description	45
6.9.2 Constructor & Destructor Documentation	45
6.9.2.1 CreateSposta()	45
6.9.3 Member Function Documentation	45
6.9.3.1 destroy()	45
6.9.3.2 doGet()	46
6.9.3.3 doPost()	47
6.9.3.4 init()	48
6.10 it.polimi.tiw.beans.File Class Reference	49
6.10.1 Detailed Description	50
6.10.2 Member Function Documentation	50
6.10.2.1 addFiglio()	50
6.10.2.2 getCreazione()	51
6.10.2.3 getDescrizione()	51
6.10.2.4 getFigli()	51
6.10.2.5 getIdDirPadre()	52
6.10.2.6 getIdDocu()	52
6.10.2.7 getNome()	52
6.10.2.8 getNomePadre()	53
6.10.2.9 getProprietario()	53
6.10.2.10 getTipo()	53
6.10.2.11 setCreazione()	53
6.10.2.12 setDescription()	54
6.10.2.13 setFigli()	54
6.10.2.14 setIdDirPadre()	55
6.10.2.15 setIdDocu()	55
6.10.2.16 setName()	56
6.10.2.17 setNamePadre()	57
6.10.2.18 setProprietario()	57
6.10.2.19 setType()	57
6.11 it.polimi.tiw.dao.FileDAO Class Reference	58
6.11.1 Detailed Description	59
6.11.2 Constructor & Destructor Documentation	59
6.11.2.1 FileDAO()	59
6.11.3 Member Function Documentation	59
6.11.3.1 getAll()	59
6.11.3.2 getDir()	60
6.11.3.3 getFile()	61
6.11.3.4 getFileFrom()	63
6.11.3.5 isDir()	65

6.11.3.6	moveFile()	66
6.11.3.7	newDocu()	67
6.12	it.polimi.tiw.controllers.Logout Class Reference	68
6.12.1	Detailed Description	69
6.12.2	Constructor & Destructor Documentation	69
6.12.2.1	Logout()	69
6.12.3	Member Function Documentation	69
6.12.3.1	doGet()	69
6.12.3.2	doPost()	70
6.13	it.polimi.tiw.controllers.Sposta Class Reference	71
6.13.1	Detailed Description	72
6.13.2	Constructor & Destructor Documentation	72
6.13.2.1	Sposta()	72
6.13.3	Member Function Documentation	72
6.13.3.1	destroy()	72
6.13.3.2	doGet()	72
6.13.3.3	doPost()	73
6.13.3.4	init()	74
6.14	it.polimi.tiw.utils.TreeOrder Class Reference	75
6.14.1	Detailed Description	75
6.14.2	Member Function Documentation	75
6.14.2.1	getOrder()	75
6.15	it.polimi.tiw.beans.User Class Reference	76
6.15.1	Detailed Description	77
6.15.2	Member Function Documentation	77
6.15.2.1	getId()	77
6.15.2.2	getMail()	78
6.15.2.3	getNome()	78
6.15.2.4	setId()	78
6.15.2.5	setMail()	79
6.15.2.6	setNome()	79
6.16	it.polimi.tiw.dao.UserDAO Class Reference	80
6.16.1	Detailed Description	80
6.16.2	Constructor & Destructor Documentation	80
6.16.2.1	UserDAO()	80
6.16.3	Member Function Documentation	80
6.16.3.1	addUser()	81
6.16.3.2	getUser()	82
6.16.3.3	isExistingMail()	83
6.16.3.4	isExistingUsername()	83

## 7 File Documentation

85

7.1 tiw/beans/File.java File Reference . . . . .	85
7.2 tiw/beans/User.java File Reference . . . . .	85
7.3 tiw/controllers/CheckLogin.java File Reference . . . . .	85
7.3.1 Detailed Description . . . . .	86
7.4 tiw/controllers/CheckRegistrazione.java File Reference . . . . .	86
7.4.1 Detailed Description . . . . .	86
7.5 tiw/controllers/CreateContenuti.java File Reference . . . . .	86
7.5.1 Detailed Description . . . . .	87
7.6 tiw/controllers/CreateDescrizione.java File Reference . . . . .	87
7.6.1 Detailed Description . . . . .	87
7.7 tiw/controllers/CreateFile.java File Reference . . . . .	87
7.7.1 Detailed Description . . . . .	87
7.8 tiw/controllers/CreateGestioneContenuti.java File Reference . . . . .	88
7.8.1 Detailed Description . . . . .	88
7.9 tiw/controllers/CreateHome.java File Reference . . . . .	88
7.9.1 Detailed Description . . . . .	88
7.10 tiw/controllers/CreateSposta.java File Reference . . . . .	88
7.10.1 Detailed Description . . . . .	89
7.11 tiw/controllers/Logout.java File Reference . . . . .	89
7.11.1 Detailed Description . . . . .	89
7.12 tiw/controllers/Sposta.java File Reference . . . . .	89
7.12.1 Detailed Description . . . . .	89
7.13 tiw/dao/FileDAO.java File Reference . . . . .	90
7.13.1 Detailed Description . . . . .	90
7.14 tiw/dao/UserDAO.java File Reference . . . . .	90
7.15 tiw/utis/ConnectionHandler.java File Reference . . . . .	90
7.15.1 Detailed Description . . . . .	90
7.16 tiw/utis/TreeOrder.java File Reference . . . . .	91
7.16.1 Detailed Description . . . . .	91

<b>Index</b>	<b>93</b>
--------------	-----------





# Chapter 1

## Namespace Index

### 1.1 Packages

Here are the packages with brief descriptions (if available):

<a href="#">it.polimi.tiw.beans</a>	9
<a href="#">it.polimi.tiw.controllers</a>	9
<a href="#">it.polimi.tiw.dao</a>	9
<a href="#">it.polimi.tiw.utils</a>	10



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

it.polimi.tiw.utils.ConnectionHandler . . . . .	19
it.polimi.tiw.beans.File . . . . .	49
it.polimi.tiw.dao.FileDAO . . . . .	58
it.polimi.tiw.utils.TreeOrder . . . . .	75
it.polimi.tiw.beans.User . . . . .	76
it.polimi.tiw.dao.UserDAO . . . . .	80
HttpServlet	
it.polimi.tiw.controllers.CheckLogin . . . . .	11
it.polimi.tiw.controllers.CheckRegistrazione . . . . .	15
it.polimi.tiw.controllers.CreateContenuti . . . . .	22
it.polimi.tiw.controllers.CreateDescrizione . . . . .	26
it.polimi.tiw.controllers.CreateFile . . . . .	31
it.polimi.tiw.controllers.CreateGestioneContenuti . . . . .	35
it.polimi.tiw.controllers.CreateHome . . . . .	40
it.polimi.tiw.controllers.CreateSpota . . . . .	44
it.polimi.tiw.controllers.Logout . . . . .	68
it.polimi.tiw.controllers.Spota . . . . .	71



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">it.polimi.tiw.controllers.CheckLogin</a>	
A servlet to manage user authentication and session creation . . . . .	11
<a href="#">it.polimi.tiw.controllers.CheckRegistrazione</a>	
Servlet that handles the user registration process . . . . .	15
<a href="#">it.polimi.tiw.utils.ConnectionHandler</a>	19
<a href="#">it.polimi.tiw.controllers.CreateContenuti</a>	22
<a href="#">it.polimi.tiw.controllers.CreateDescrizione</a>	26
<a href="#">it.polimi.tiw.controllers.CreateFile</a>	31
<a href="#">it.polimi.tiw.controllers.CreateGestioneContenuti</a>	35
<a href="#">it.polimi.tiw.controllers.CreateHome</a>	40
<a href="#">it.polimi.tiw.controllers.CreateSposta</a>	44
<a href="#">it.polimi.tiw.beans.File</a>	
Represents a file or directory in the document management system . . . . .	49
<a href="#">it.polimi.tiw.dao.FileDAO</a>	58
<a href="#">it.polimi.tiw.controllers.Logout</a>	68
<a href="#">it.polimi.tiw.controllers.Sposta</a>	71
<a href="#">it.polimi.tiw.utils.TreeOrder</a>	75
<a href="#">it.polimi.tiw.beans.User</a>	
Represents a user in the system . . . . .	76
<a href="#">it.polimi.tiw.dao.UserDAO</a>	
Data Access Object for User entity . . . . .	80



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

tiw/beans/ <a href="#">File.java</a>	85
tiw/beans/ <a href="#">User.java</a>	85
tiw/controllers/ <a href="#">CheckLogin.java</a>	
Servlet that handles user login by checking credentials and managing session creation	85
tiw/controllers/ <a href="#">CheckRegistrazione.java</a>	
Servlet that handles the user registration process by validating input and adding new users to the database	86
tiw/controllers/ <a href="#">CreateContenuti.java</a>	
This servlet handles the creation and display of contents in a user's directory	86
tiw/controllers/ <a href="#">CreateDescrizione.java</a>	
This servlet handles the display of file descriptions	87
tiw/controllers/ <a href="#">CreateFile.java</a>	
This servlet handles the creation of new files and directories	87
tiw/controllers/ <a href="#">CreateGestioneContenuti.java</a>	
This servlet manages the content creation and display for the user	88
tiw/controllers/ <a href="#">CreateHome.java</a>	
This servlet handles the home page for content management, displaying the user's directories	88
tiw/controllers/ <a href="#">CreateSposta.java</a>	
This servlet handles the file moving process, allowing users to move files between directories	88
tiw/controllers/ <a href="#">Logout.java</a>	
This servlet handles user logout functionality by invalidating the session	89
tiw/controllers/ <a href="#">Sposta.java</a>	
This servlet handles the movement of files between directories	89
tiw/dao/ <a href="#">FileDAO.java</a>	
This class provides methods for interacting with the file database	90
tiw/dao/ <a href="#">UserDAO.java</a>	90
tiw/utis/ <a href="#">ConnectionHandler.java</a>	
This class handles the management of database connections	90
tiw/utis/ <a href="#">TreeOrder.java</a>	
This class manages the organization of files into a tree structure	91





## Chapter 5

# Namespace Documentation

### 5.1 Package it.polimi.tiw.beans

#### Classes

- class [File](#)  
*Represents a file or directory in the document management system.*
- class [User](#)  
*Represents a user in the system.*

### 5.2 Package it.polimi.tiw.controllers

#### Classes

- class [CheckLogin](#)  
*A servlet to manage user authentication and session creation.*
- class [CheckRegistrazione](#)  
*Servlet that handles the user registration process.*
- class [CreateContenuti](#)
- class [CreateDescrizione](#)
- class [CreateFile](#)
- class [CreateGestioneContenuti](#)
- class [CreateHome](#)
- class [CreateSposta](#)
- class [Logout](#)
- class [Sposta](#)

### 5.3 Package it.polimi.tiw.dao

#### Classes

- class [FileDAO](#)
- class [UserDAO](#)  
*Data Access Object for User entity.*

## 5.4 Package it.polimi.tiw.utils

### Classes

- class [ConnectionHandler](#)
- class [TreeOrder](#)

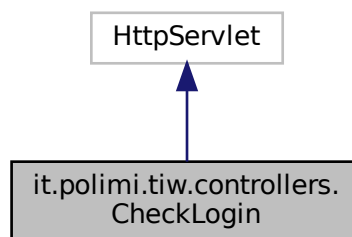
## Chapter 6

# Class Documentation

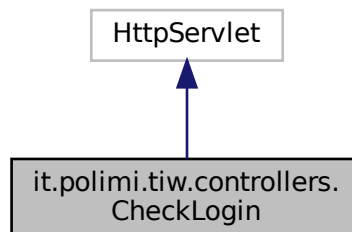
### 6.1 it.polimi.tiw.controllers.CheckLogin Class Reference

A servlet to manage user authentication and session creation.

Inheritance diagram for it.polimi.tiw.controllers.CheckLogin:



Collaboration diagram for it.polimi.tiw.controllers.CheckLogin:



## Public Member Functions

- [CheckLogin](#) ()  
*Constructor for [CheckLogin](#) servlet.*
- void [init](#) () throws ServletException  
*Initializes the servlet and configures the template engine.*
- void [destroy](#) ()  
*Closes the connection when the servlet is destroyed.*

## Protected Member Functions

- void [doGet](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles GET requests by rendering the login page.*
- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles POST requests to authenticate a user.*

### 6.1.1 Detailed Description

A servlet to manage user authentication and session creation.

The servlet retrieves user credentials via a form submission, validates them, and initiates a session for authenticated users. It uses Thymeleaf to display the login page and error messages if authentication fails.

Definition at line 39 of file CheckLogin.java.

### 6.1.2 Constructor & Destructor Documentation

#### 6.1.2.1 CheckLogin()

```
it.polimi.tiw.controllers.CheckLogin.CheckLogin ( )
```

Constructor for [CheckLogin](#) servlet.

Initializes a new instance of the [CheckLogin](#) servlet.

Definition at line 49 of file CheckLogin.java.

### 6.1.3 Member Function Documentation

**6.1.3.1 destroy()**

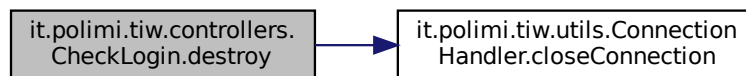
```
void it.polimi.tiw.controllers.CheckLogin.destroy ( )
```

Closes the connection when the servlet is destroyed.

Ensures that the database connection is closed when the servlet is no longer in use.

Definition at line 166 of file CheckLogin.java.

Here is the call graph for this function:

**6.1.3.2 doGet()**

```
void it.polimi.tiw.controllers.CheckLogin.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests by rendering the login page.

Processes GET requests to display the login page using Thymeleaf templates.

**Parameters**

<i>request</i>	the <code>HttpServletRequest</code> object containing the client request
<i>response</i>	the <code>HttpServletResponse</code> object for sending a response back to the client

**Exceptions**

<i>ServletException</i>	if an error occurs during the request
<i>IOException</i>	if an I/O error occurs during processing

Definition at line 82 of file CheckLogin.java.

**6.1.3.3 doPost()**

```
void it.polimi.tiw.controllers.CheckLogin.doPost (
```

```
HttpServletRequest request,
HttpServletRequest response ) throws ServletException, IOException [protected]
```

Handles POST requests to authenticate a user.

Retrieves user credentials from the request, checks their validity, and creates a session for authenticated users. If the credentials are invalid, an error message is displayed.

#### Parameters

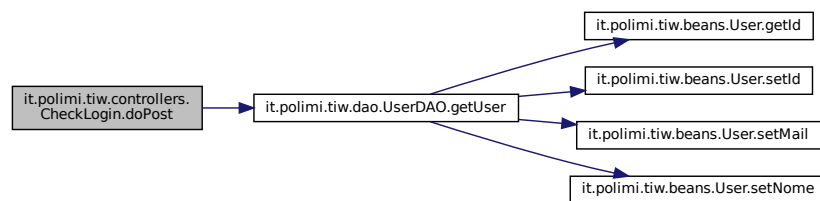
<i>request</i>	the HttpServletRequest object containing the client request
<i>response</i>	the HttpServletResponse object for sending a response back to the client

#### Exceptions

<i>ServletException</i>	if an error occurs during the request
<i>IOException</i>	if an I/O error occurs during processing

Definition at line 101 of file CheckLogin.java.

Here is the call graph for this function:



#### 6.1.3.4 init()

```
void it.polimi.tiw.controllers.CheckLogin.init ( ) throws ServletException
```

Initializes the servlet and configures the template engine.

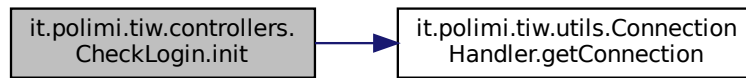
Establishes a database connection and configures Thymeleaf for HTML template rendering.

#### Exceptions

<i>ServletException</i>	if an error occurs during initialization
-------------------------	--

Definition at line 61 of file CheckLogin.java.

Here is the call graph for this function:



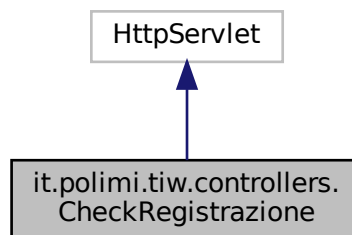
The documentation for this class was generated from the following file:

- [tiw/controllers/CheckLogin.java](#)

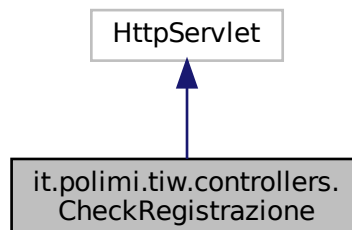
## 6.2 it.polimi.tiw.controllers.CheckRegistrazione Class Reference

Servlet that handles the user registration process.

Inheritance diagram for `it.polimi.tiw.controllers.CheckRegistrazione`:



Collaboration diagram for `it.polimi.tiw.controllers.CheckRegistrazione`:



## Public Member Functions

- [CheckRegistrazione](#) ()
- void [init](#) () throws ServletException  
*Initializes the servlet, setting up the database connection and configuring the Thymeleaf template engine.*
- void [destroy](#) ()  
*Cleans up resources when the servlet is destroyed.*

## Protected Member Functions

- void [doGet](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles GET requests by displaying the registration page.*
- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles POST requests for user registration.*

### 6.2.1 Detailed Description

Servlet that handles the user registration process.

Definition at line 36 of file CheckRegistrazione.java.

### 6.2.2 Constructor & Destructor Documentation

#### 6.2.2.1 CheckRegistrazione()

```
it.polimi.tiw.controllers.CheckRegistrazione.CheckRegistrazione ( )
```

Definition at line 41 of file CheckRegistrazione.java.

### 6.2.3 Member Function Documentation

#### 6.2.3.1 destroy()

```
void it.polimi.tiw.controllers.CheckRegistrazione.destroy ( )
```

Cleans up resources when the servlet is destroyed.

This method closes the database connection to free up resources. It handles any SQL exceptions that may occur during the closing process.

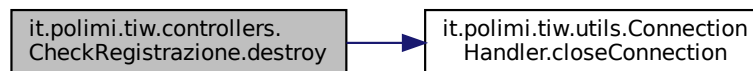


### Exceptions

<i>SQLException</i>	if a database access error occurs while closing the connection
---------------------	--

Definition at line 187 of file CheckRegistrazione.java.

Here is the call graph for this function:



### 6.2.3.2 doGet()

```
void it.polimi.tiw.controllers.CheckRegistrazione.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests by displaying the registration page.

This method processes the GET request and renders the registration page using Thymeleaf.

#### Parameters

<i>request</i>	the HttpServletRequest object containing the client request
<i>response</i>	the HttpServletResponse object used to respond to the client

### Exceptions

<i>ServletException</i>	if an error occurs during the request
<i>IOException</i>	if an I/O error occurs

Definition at line 71 of file CheckRegistrazione.java.

### 6.2.3.3 doPost()

```
void it.polimi.tiw.controllers.CheckRegistrazione.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles POST requests for user registration.

This method validates the input fields, checks if the user already exists, and adds the new user to the database. If there are validation errors, it displays them on the registration page. If registration is successful, it notifies the user.

#### Parameters

<i>request</i>	the <code>HttpServletRequest</code> object containing the client request
<i>response</i>	the <code>HttpServletResponse</code> object used to respond to the client

#### Exceptions

<i>ServletException</i>	if an error occurs during the request
<i>IOException</i>	if an I/O error occurs

Definition at line 89 of file `CheckRegistrazione.java`.

Here is the call graph for this function:



#### 6.2.3.4 `init()`

```
void it.polimi.tiw.controllers.CheckRegistrazione.init ( ) throws ServletException
```

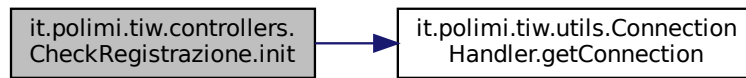
Initializes the servlet, setting up the database connection and configuring the Thymeleaf template engine.

#### Exceptions

<i>ServletException</i>	if an error occurs during servlet initialization
-------------------------	--

Definition at line 51 of file `CheckRegistrazione.java`.

Here is the call graph for this function:



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

- tiw/controllers/[CheckRegistrazione.java](#)

## 6.3 it.polimi.tiw.utils.ConnectionHandler Class Reference

### Static Public Member Functions

- static Connection [getConnection](#) (ServletContext context) throws UnavailableException  
*Establishes a database connection using parameters from the servlet context.*
- static void [closeConnection](#) (Connection connection) throws SQLException  
*Closes the given database connection.*

#### 6.3.1 Detailed Description

This utility class provides methods to establish and close database connections using connection parameters defined in the servlet context.

Definition at line 18 of file ConnectionHandler.java.

#### 6.3.2 Member Function Documentation

##### 6.3.2.1 closeConnection()

```
static void it.polimi.tiw.utils.ConnectionHandler.closeConnection (  
    Connection connection ) throws SQLException [static]
```

Closes the given database connection.

##### Parameters

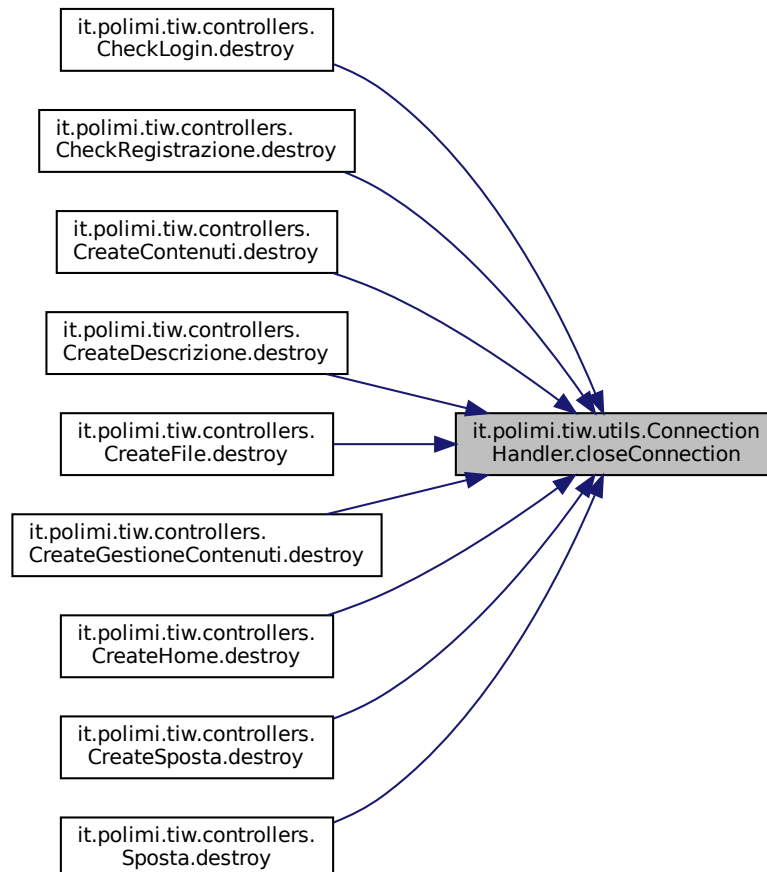
<i>connection</i>	the Connection object to close.
-------------------	---------------------------------

**Exceptions**

<i>SQLException</i>	if an error occurs while closing the connection.
---------------------	--

Definition at line 47 of file ConnectionHandler.java.

Here is the caller graph for this function:

**6.3.2.2 getConnection()**

```
static Connection it.polimi.tiw.utils.ConnectionHandler.getConnection (
    ServletContext context ) throws UnavailableException [static]
```

Establishes a database connection using parameters from the servlet context.

**Parameters**

<i>context</i>	the ServletContext containing database connection parameters.
----------------	---

**Returns**

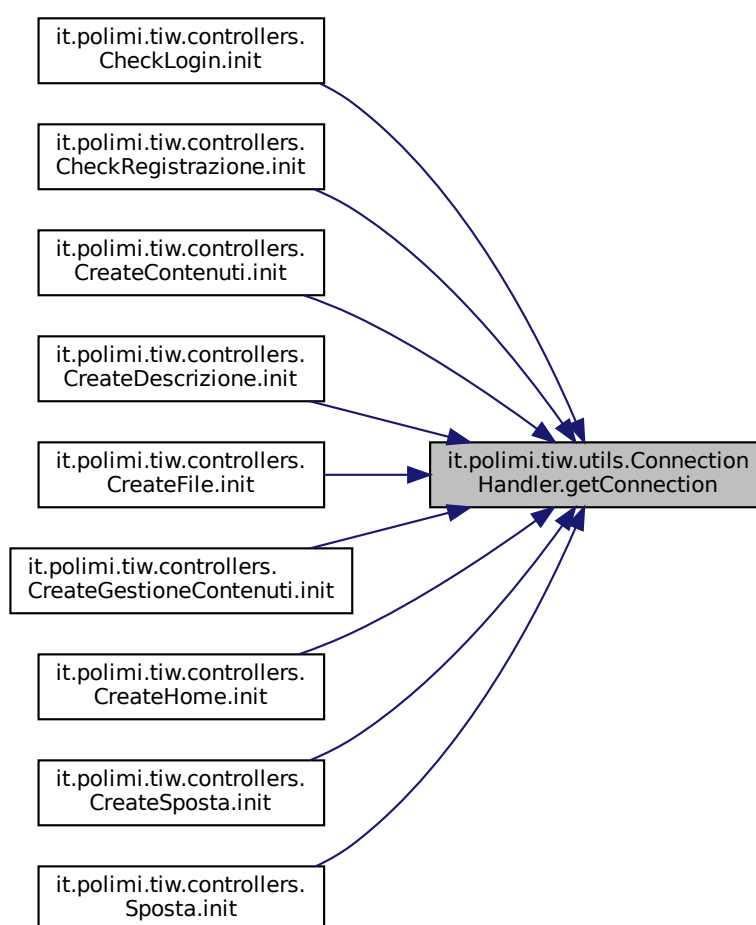
a Connection object to interact with the database.

**Exceptions**

<i>UnavailableException</i>	if the database driver cannot be loaded or if the connection cannot be established.
-----------------------------	---

Definition at line 25 of file ConnectionHandler.java.

Here is the caller graph for this function:

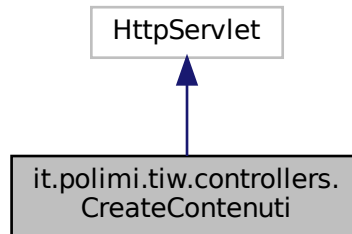


The documentation for this class was generated from the following file:

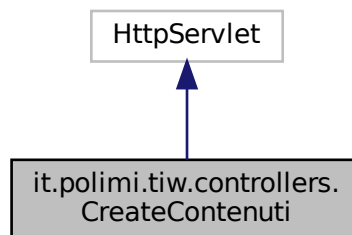
- [tiw/utils/ConnectionHandler.java](#)

## 6.4 it.polimi.tiw.controllers.CreateContenuti Class Reference

Inheritance diagram for it.polimi.tiw.controllers.CreateContenuti:



Collaboration diagram for it.polimi.tiw.controllers.CreateContenuti:



### Public Member Functions

- [CreateContenuti](#) ()
- void [init](#) () throws ServletException  
*Initializes the servlet and establishes a database connection.*
- void [destroy](#) ()  
*Cleans up resources when the servlet is destroyed.*

### Protected Member Functions

- void [doGet](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles GET requests for creating contents.*
- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles POST requests for creating contents.*

### 6.4.1 Detailed Description

Definition at line 26 of file CreateContenuti.java.

### 6.4.2 Constructor & Destructor Documentation

#### 6.4.2.1 CreateContenuti()

```
it.polimi.tiw.controllers.CreateContenuti.CreateContenuti ( )
```

Definition at line 40 of file CreateContenuti.java.

### 6.4.3 Member Function Documentation

#### 6.4.3.1 destroy()

```
void it.polimi.tiw.controllers.CreateContenuti.destroy ( )
```

Cleans up resources when the servlet is destroyed.

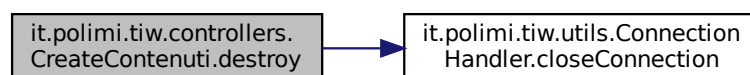
This method closes the database connection to free up resources. It handles any SQL exceptions that may occur during the closing process.

#### Exceptions

<i>SQLException</i>	if a database access error occurs while closing the connection
---------------------	--

Definition at line 182 of file CreateContenuti.java.

Here is the call graph for this function:



### 6.4.3.2 doGet()

```
void it.polimi.tiw.controllers.CreateContenuti.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests for creating contents.

This method retrieves the contents from the specified directory and displays them on the contents page. It checks if the user is logged in and whether the directory ID is valid.

#### Parameters

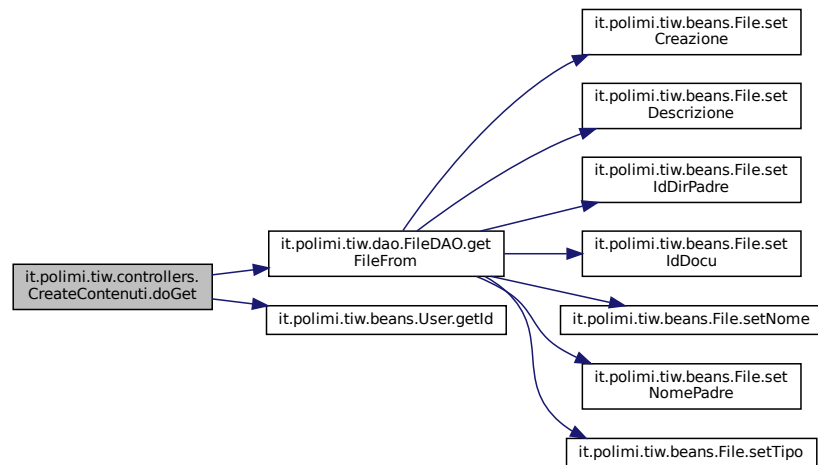
<i>request</i>	the HttpServletRequest object
<i>response</i>	the HttpServletResponse object

#### Exceptions

<i>ServletException</i>	if a servlet error occurs
<i>IOException</i>	if an I/O error occurs during the process

Definition at line 74 of file CreateContenuti.java.

Here is the call graph for this function:



### 6.4.3.3 doPost()

```
void it.polimi.tiw.controllers.CreateContenuti.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```



Handles POST requests for creating contents.

This method processes the request to create contents based on user input and displays the updated contents on the contents page. It checks if the user is logged in and whether the directory ID is valid.

#### Parameters

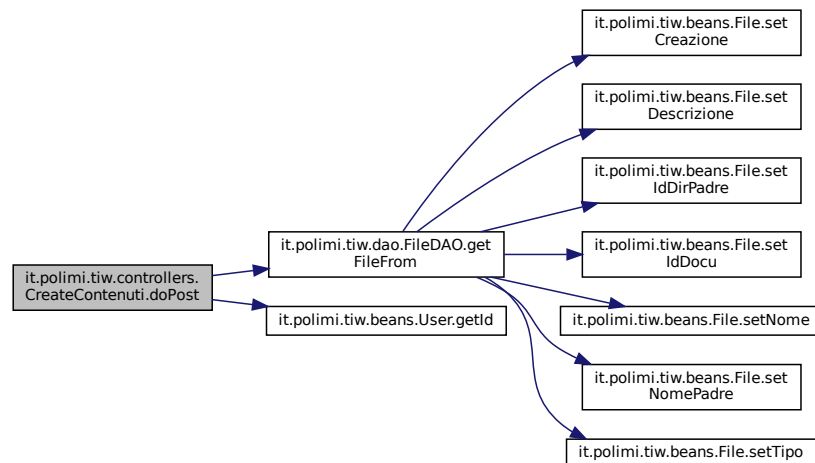
<i>request</i>	the <code>HttpServletRequest</code> object
<i>response</i>	the <code>HttpServletResponse</code> object

#### Exceptions

<i>ServletException</i>	if a servlet error occurs
<i>IOException</i>	if an I/O error occurs during the process

Definition at line 130 of file `CreateContenuti.java`.

Here is the call graph for this function:



#### 6.4.3.4 init()

```
void it.polimi.tiw.controllers.CreateContenuti.init ( ) throws ServletException
```

Initializes the servlet and establishes a database connection.

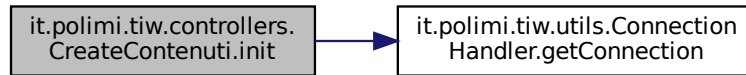
This method sets up the template engine and prepares the connection to the database when the servlet is initialized.

#### Exceptions

<i>ServletException</i>	if a servlet error occurs during initialization
-------------------------	---

Definition at line 52 of file CreateContenuti.java.

Here is the call graph for this function:

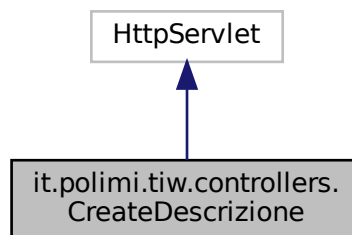


The documentation for this class was generated from the following file:

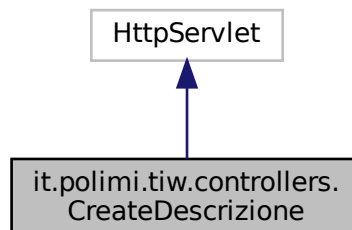
- [tiw/controllers/CreateContenuti.java](#)

## 6.5 `it.polimi.tiw.controllers.CreateDescrizione` Class Reference

Inheritance diagram for `it.polimi.tiw.controllers.CreateDescrizione`:



Collaboration diagram for `it.polimi.tiw.controllers.CreateDescrizione`:



## Public Member Functions

- [CreateDescrizione](#) ()
- void [init](#) () throws ServletException  
*Initializes the servlet and establishes a database connection.*
- void [destroy](#) ()  
*Cleans up resources when the servlet is destroyed.*

## Protected Member Functions

- void [doGet](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles GET requests by redirecting to the doPost method.*
- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles POST requests to retrieve and display file information.*

### 6.5.1 Detailed Description

This class extends HttpServlet and processes GET and POST requests to retrieve and display information about a specific file based on the user's session. It ensures the user is logged in and retrieves the file data from the database.

Definition at line 37 of file CreateDescrizione.java.

### 6.5.2 Constructor & Destructor Documentation

#### 6.5.2.1 CreateDescrizione()

```
it.polimi.tiw.controllers.CreateDescrizione.CreateDescrizione ( )
```

Definition at line 42 of file CreateDescrizione.java.

### 6.5.3 Member Function Documentation

### 6.5.3.1 destroy()

```
void it.polimi.tiw.controllers.CreateDescrizione.destroy ( )
```

Cleans up resources when the servlet is destroyed.

Definition at line 118 of file CreateDescrizione.java.

Here is the call graph for this function:



### 6.5.3.2 doGet()

```
void it.polimi.tiw.controllers.CreateDescrizione.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests by redirecting to the doPost method.

#### Parameters

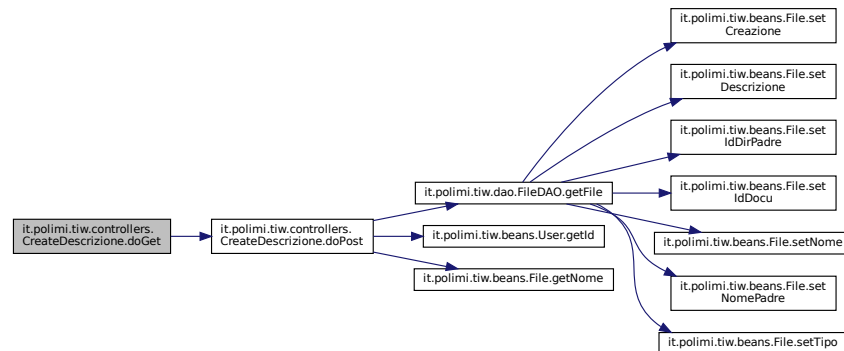
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

#### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 67 of file CreateDescrizione.java.

Here is the call graph for this function:



### 6.5.3.3 doPost()

```
void it.polimi.tiw.controllers.CreateDescrizione.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles POST requests to retrieve and display file information.

#### Parameters

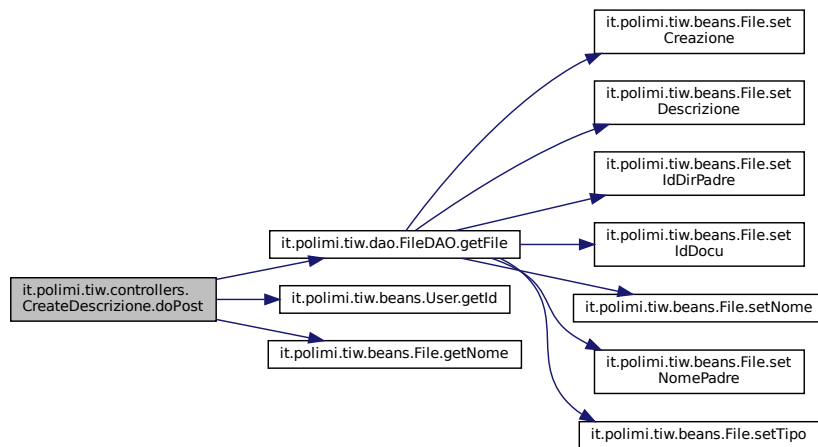
<i>request</i>	the <code>HttpServletRequest</code> object.
<i>response</i>	the <code>HttpServletResponse</code> object.

#### Exceptions

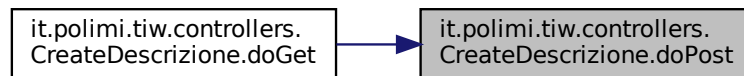
<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 78 of file `CreateDescrizione.java`.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.5.3.4 init()

```
void it.polimi.tiw.controllers.CreateDescrizione.init ( ) throws ServletException
```

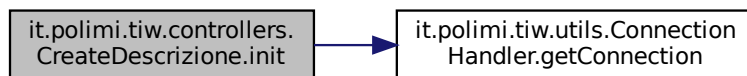
Initializes the servlet and establishes a database connection.

##### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
-------------------------	------------------------------------

Definition at line 50 of file `CreateDescrizione.java`.

Here is the call graph for this function:

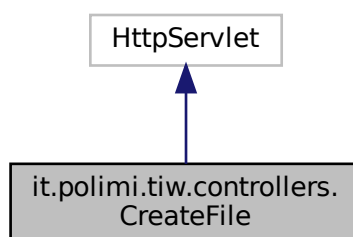


The documentation for this class was generated from the following file:

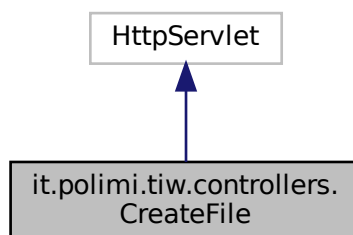
- [tiw/controllers/CreateDescrizione.java](#)

## 6.6 it.polimi.tiw.controllers.CreateFile Class Reference

Inheritance diagram for `it.polimi.tiw.controllers.CreateFile`:



Collaboration diagram for `it.polimi.tiw.controllers.CreateFile`:



## Public Member Functions

- [CreateFile](#) ()
- void [init](#) () throws ServletException  
*Initializes the servlet and establishes a database connection.*
- void [destroy](#) ()  
*Cleans up resources when the servlet is destroyed.*

## Protected Member Functions

- void [doGet](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles GET requests by redirecting to the doPost method.*
- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles POST requests to create a new file or directory.*

### 6.6.1 Detailed Description

This class extends HttpServlet and processes GET and POST requests to create new files or directories in the user's storage based on the provided parameters.

Definition at line 33 of file CreateFile.java.

### 6.6.2 Constructor & Destructor Documentation

#### 6.6.2.1 CreateFile()

```
it.polimi.tiw.controllers.CreateFile.CreateFile ( )
```

Definition at line 38 of file CreateFile.java.

### 6.6.3 Member Function Documentation



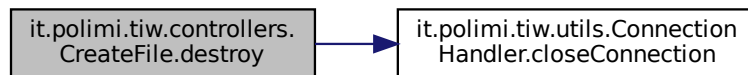
### 6.6.3.1 destroy()

```
void it.polimi.tiw.controllers.CreateFile.destroy ( )
```

Cleans up resources when the servlet is destroyed.

Definition at line 157 of file CreateFile.java.

Here is the call graph for this function:



### 6.6.3.2 doGet()

```
void it.polimi.tiw.controllers.CreateFile.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests by redirecting to the doPost method.

#### Parameters

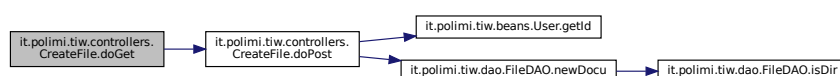
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

#### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 63 of file CreateFile.java.

Here is the call graph for this function:



### 6.6.3.3 doPost()

```
void it.polimi.tiw.controllers.CreateFile.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles POST requests to create a new file or directory.

#### Parameters

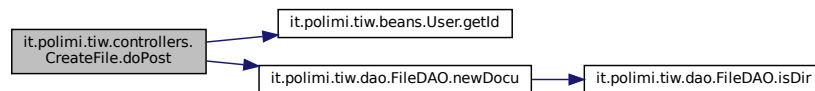
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

#### Exceptions

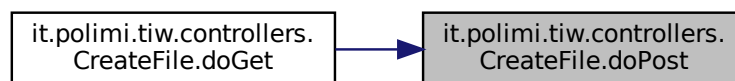
<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 74 of file CreateFile.java.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.6.3.4 init()

```
void it.polimi.tiw.controllers.CreateFile.init ( ) throws ServletException
```

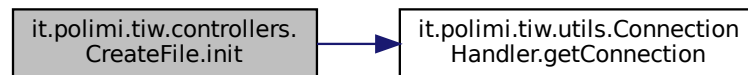
Initializes the servlet and establishes a database connection.

### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
-------------------------	------------------------------------

Definition at line 46 of file CreateFile.java.

Here is the call graph for this function:

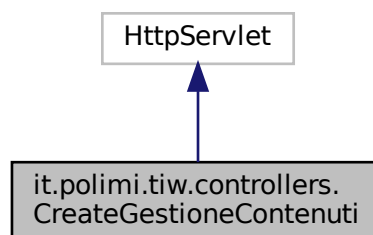


The documentation for this class was generated from the following file:

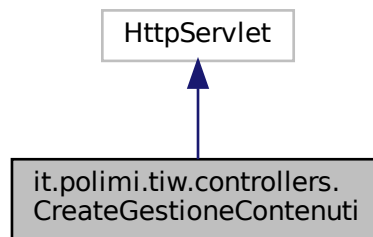
- tiw/controllers/[CreateFile.java](#)

## 6.7 it.polimi.tiw.controllers.CreateGestioneContenuti Class Reference

Inheritance diagram for `it.polimi.tiw.controllers.CreateGestioneContenuti`:



Collaboration diagram for `it.polimi.tiw.controllers.CreateGestioneContenuti`:



## Public Member Functions

- `CreateGestioneContenuti ()`
- `void init ()` throws `ServletException`  
*Initializes the servlet and establishes a database connection.*
- `void destroy ()`  
*Cleans up resources when the servlet is destroyed.*

## Protected Member Functions

- `void doGet (HttpServletRequest request, HttpServletResponse response)` throws `ServletException`, `IOException`  
*Handles GET requests to display the content management interface.*
- `void doPost (HttpServletRequest request, HttpServletResponse response)` throws `ServletException`, `IOException`  
*Handles POST requests by redirecting to the doGet method.*

### 6.7.1 Detailed Description

This class extends `HttpServlet` and processes GET and POST requests to manage the user's content, displaying directories and handling file creation errors.

Definition at line 33 of file `CreateGestioneContenuti.java`.

### 6.7.2 Constructor & Destructor Documentation

#### 6.7.2.1 CreateGestioneContenuti()

```
it.polimi.tiw.controllers.CreateGestioneContenuti.CreateGestioneContenuti ( )
```

Definition at line 39 of file `CreateGestioneContenuti.java`.

### 6.7.3 Member Function Documentation

#### 6.7.3.1 destroy()

```
void it.polimi.tiw.controllers.CreateGestioneContenuti.destroy ( )
```

Cleans up resources when the servlet is destroyed.

Definition at line 107 of file CreateGestioneContenuti.java.

Here is the call graph for this function:



#### 6.7.3.2 doGet()

```
void it.polimi.tiw.controllers.CreateGestioneContenuti.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests to display the content management interface.

##### Parameters

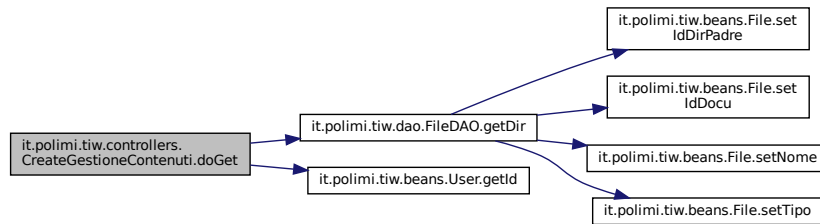
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

##### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 64 of file CreateGestioneContenuti.java.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.7.3.3 doPost()

```

void it.polimi.tiw.controllers.CreateGestioneContenuti.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
  
```

Handles POST requests by redirecting to the doGet method.

#### Parameters

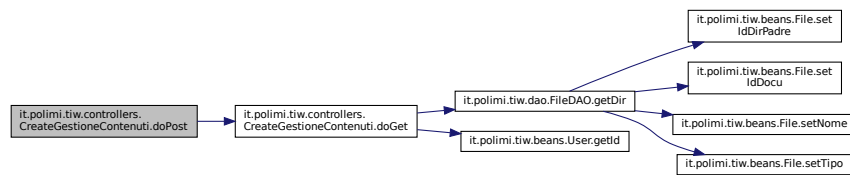
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

#### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 100 of file CreateGestioneContenuti.java.

Here is the call graph for this function:



#### 6.7.3.4 init()

```
void it.polimi.tiw.controllers.CreateGestioneContenuti.init ( ) throws ServletException
```

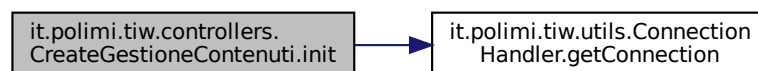
Initializes the servlet and establishes a database connection.

##### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
-------------------------	------------------------------------

Definition at line 47 of file CreateGestioneContenuti.java.

Here is the call graph for this function:

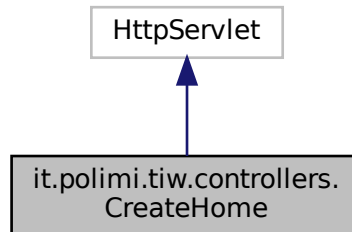


The documentation for this class was generated from the following file:

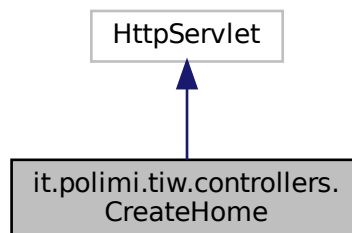
- tiw/controllers/[CreateGestioneContenuti.java](#)

## 6.8 it.polimi.tiw.controllers.CreateHome Class Reference

Inheritance diagram for it.polimi.tiw.controllers.CreateHome:



Collaboration diagram for it.polimi.tiw.controllers.CreateHome:



### Public Member Functions

- `CreateHome ()`
- void `init ()` throws `ServletException`  
*Initializes the servlet and establishes a database connection.*
- void `destroy ()`  
*Cleans up resources when the servlet is destroyed.*

### Protected Member Functions

- void `doGet` (`HttpServletRequest request`, `HttpServletResponse response`) throws `ServletException`, `IOException`  
*Handles GET requests to display the user's home page.*
- void `doPost` (`HttpServletRequest request`, `HttpServletResponse response`) throws `ServletException`, `IOException`  
*Handles POST requests by redirecting to the doGet method.*



### 6.8.1 Detailed Description

This class extends `HttpServlet` and processes GET and POST requests to display the user's home page with their directories, handling session management and errors.

Definition at line 34 of file `CreateHome.java`.

### 6.8.2 Constructor & Destructor Documentation

#### 6.8.2.1 `CreateHome()`

```
it.polimi.tiw.controllers.CreateHome.CreateHome ( )
```

Definition at line 40 of file `CreateHome.java`.

### 6.8.3 Member Function Documentation

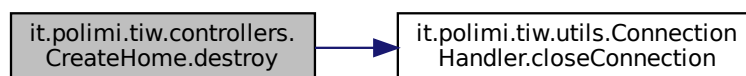
#### 6.8.3.1 `destroy()`

```
void it.polimi.tiw.controllers.CreateHome.destroy ( )
```

Cleans up resources when the servlet is destroyed.

Definition at line 114 of file `CreateHome.java`.

Here is the call graph for this function:



#### 6.8.3.2 `doGet()`

```
void it.polimi.tiw.controllers.CreateHome.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests to display the user's home page.

## Parameters

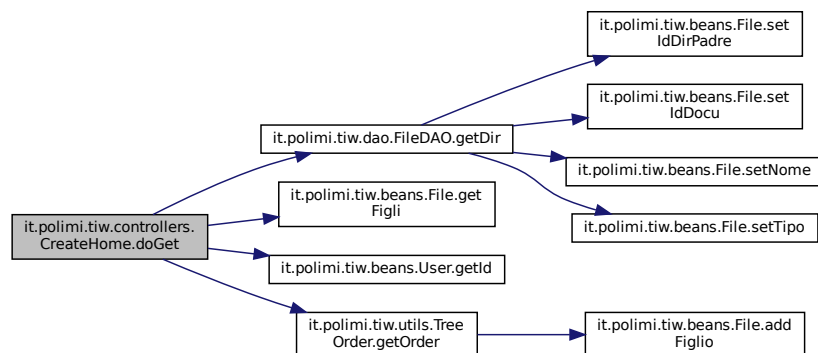
<i>request</i>	the <code>HttpServletRequest</code> object.
<i>response</i>	the <code>HttpServletResponse</code> object.

## Exceptions

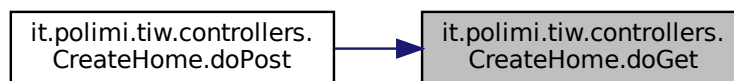
<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 65 of file `CreateHome.java`.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.8.3.3 doPost()

```

void it.polimi.tiw.controllers.CreateHome.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
  
```

Handles POST requests by redirecting to the `doGet` method.

## Parameters

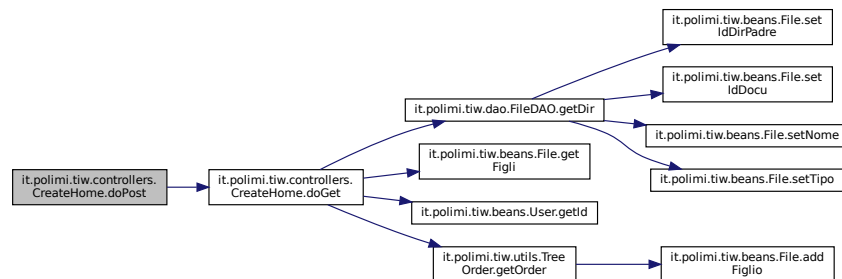
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

## Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 106 of file CreateHome.java.

Here is the call graph for this function:



## 6.8.3.4 init()

```
void it.polimi.tiw.controllers.CreateHome.init ( ) throws ServletException
```

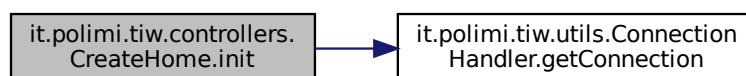
Initializes the servlet and establishes a database connection.

## Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
-------------------------	------------------------------------

Definition at line 48 of file CreateHome.java.

Here is the call graph for this function:

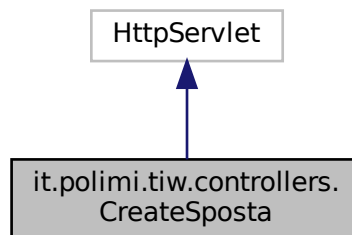


The documentation for this class was generated from the following file:

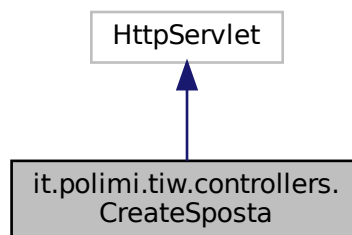
- tiw/controllers/[CreateHome.java](#)

## 6.9 it.polimi.tiw.controllers.CreateSposta Class Reference

Inheritance diagram for it.polimi.tiw.controllers.CreateSposta:



Collaboration diagram for it.polimi.tiw.controllers.CreateSposta:



### Public Member Functions

- [CreateSposta](#) ()
- void [init](#) () throws ServletException  
*Initializes the servlet and establishes a database connection.*
- void [destroy](#) ()  
*Cleans up resources when the servlet is destroyed.*

## Protected Member Functions

- void `doGet` (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles GET requests by redirecting to the doPost method.*
- void `doPost` (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles POST requests to move a file to a specified directory.*

### 6.9.1 Detailed Description

This class extends HttpServlet and processes GET and POST requests to manage file movement, handling session management and error responses.

Definition at line 34 of file CreateSposta.java.

### 6.9.2 Constructor & Destructor Documentation

#### 6.9.2.1 CreateSposta()

```
it.polimi.tiw.controllers.CreateSposta.CreateSposta ( )
```

Definition at line 40 of file CreateSposta.java.

### 6.9.3 Member Function Documentation

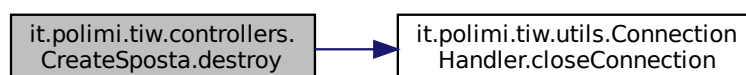
#### 6.9.3.1 destroy()

```
void it.polimi.tiw.controllers.CreateSposta.destroy ( )
```

Cleans up resources when the servlet is destroyed.

Definition at line 126 of file CreateSposta.java.

Here is the call graph for this function:



### 6.9.3.2 doGet()

```
void it.polimi.tiw.controllers.CreateSposta.doGet (  
    HttpServletRequest request,  
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests by redirecting to the doPost method.

## Parameters

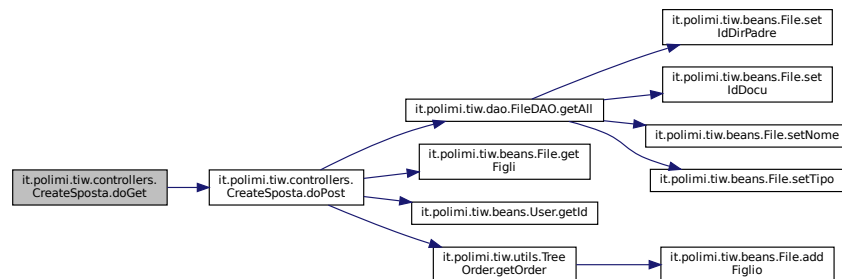
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

## Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 65 of file CreateSposta.java.

Here is the call graph for this function:



## 6.9.3.3 doPost()

```

void it.polimi.tiw.controllers.CreateSposta.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
  
```

Handles POST requests to move a file to a specified directory.

## Parameters

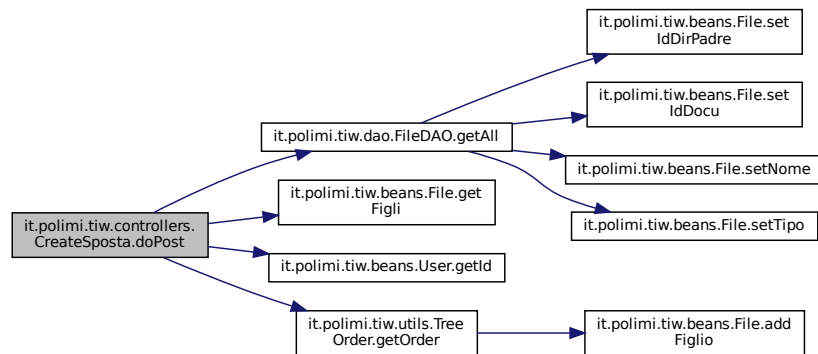
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

## Exceptions

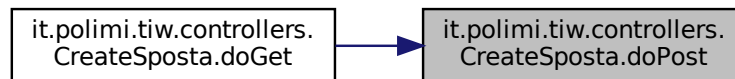
<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 76 of file CreateSposta.java.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.9.3.4 init()

```
void it.polimi.tiw.controllers.CreateSposta.init ( ) throws ServletException
```

Initializes the servlet and establishes a database connection.

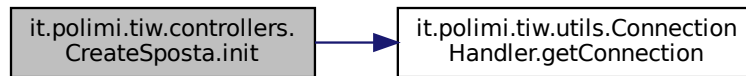
##### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
-------------------------	------------------------------------

Definition at line 48 of file `CreateSposta.java`.



Here is the call graph for this function:



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

- tiw/controllers/[CreateSposta.java](#)

## 6.10 it.polimi.tiw.beans.File Class Reference

Represents a file or directory in the document management system.

### Public Member Functions

- int [getIdDocu](#) ()  
*Gets the document ID.*
- int [getIdDirPadre](#) ()  
*Gets the parent directory ID.*
- int [getProprietario](#) ()  
*Gets the owner ID.*
- String [getNome](#) ()  
*Gets the name of the file or directory.*
- String [getCreazione](#) ()  
*Gets the creation date of the file or directory.*
- String [getTipo](#) ()  
*Gets the type of the file.*
- String [getDescrizione](#) ()  
*Gets the description of the file or directory.*
- ArrayList< [File](#) > [getFigli](#) ()  
*Gets the list of child files or directories.*
- void [setIdDocu](#) (int idDocu)  
*Sets the document ID.*
- void [setIdDirPadre](#) (int idDirPadre)  
*Sets the parent directory ID.*
- void [setProprietario](#) (int proprietario)  
*Sets the owner ID.*
- void [setNome](#) (String nome)  
*Sets the name of the file or directory.*
- void [setCreazione](#) (String creazione)  
*Sets the creation date of the file or directory.*
- void [setTipo](#) (String tipo)

- Sets the type of the file.*
  - void [setDescription](#) (String descrizione)
- Sets the description of the file or directory.*
  - void [setFigli](#) (ArrayList< [File](#) > figli)
- Sets the list of child files or directories.*
  - void [addFiglio](#) ([File](#) figlio)
- Adds a child file or directory.*
  - String [getNomePadre](#) ()
- Gets the name of the parent directory.*
  - void [setNomePadre](#) (String nomePadre)
- Sets the name of the parent directory.*

### 6.10.1 Detailed Description

Represents a file or directory in the document management system.

This class encapsulates the properties of a file or directory, including its ID, parent directory ID, owner, name, creation date, type, description, and child files.

Definition at line 15 of file File.java.

### 6.10.2 Member Function Documentation

#### 6.10.2.1 addFiglio()

```
void it.polimi.tiw.beans.File.addFiglio (
    File figlio )
```

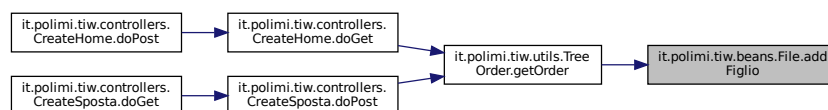
Adds a child file or directory.

##### Parameters

<i>figlio</i>	The child file to add.
---------------	------------------------

Definition at line 194 of file File.java.

Here is the caller graph for this function:



### 6.10.2.2 getCreazione()

```
String it.polimi.tiw.beans.File.getCreazione ( )
```

Gets the creation date of the file or directory.

#### Returns

The creation date as a string.

Definition at line 86 of file File.java.

### 6.10.2.3 getDescrizione()

```
String it.polimi.tiw.beans.File.getDescrizione ( )
```

Gets the description of the file or directory.

#### Returns

The description.

Definition at line 104 of file File.java.

### 6.10.2.4 getFigli()

```
ArrayList<File> it.polimi.tiw.beans.File.getFigli ( )
```

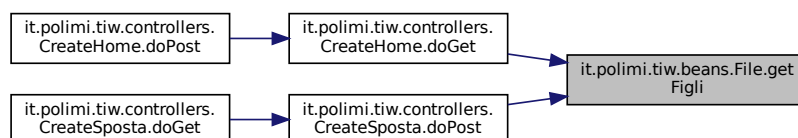
Gets the list of child files or directories.

#### Returns

An ArrayList of child [File](#) objects.

Definition at line 113 of file File.java.

Here is the caller graph for this function:



#### 6.10.2.5 getIdDirPadre()

```
int it.polimi.tiw.beans.File.getIdDirPadre ( )
```

Gets the parent directory ID.

##### Returns

The parent directory ID.

Definition at line 59 of file File.java.

#### 6.10.2.6 getIdDocu()

```
int it.polimi.tiw.beans.File.getIdDocu ( )
```

Gets the document ID.

##### Returns

The document ID.

Definition at line 50 of file File.java.

#### 6.10.2.7 getNome()

```
String it.polimi.tiw.beans.File.getNome ( )
```

Gets the name of the file or directory.

##### Returns

The name of the file or directory.

Definition at line 77 of file File.java.

Here is the caller graph for this function:



### 6.10.2.8 getNomePadre()

```
String it.polimi.tiw.beans.File.getNomePadre ( )
```

Gets the name of the parent directory.

#### Returns

The name of the parent directory.

Definition at line 203 of file File.java.

### 6.10.2.9 getProprietario()

```
int it.polimi.tiw.beans.File.getProprietario ( )
```

Gets the owner ID.

#### Returns

The owner ID.

Definition at line 68 of file File.java.

### 6.10.2.10 getTipo()

```
String it.polimi.tiw.beans.File.getTipo ( )
```

Gets the type of the file.

#### Returns

The type of the file.

Definition at line 95 of file File.java.

### 6.10.2.11 setCreazione()

```
void it.polimi.tiw.beans.File.setCreazione (
    String creazione )
```

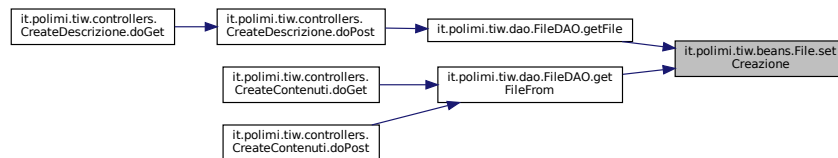
Sets the creation date of the file or directory.

## Parameters

<i>creazione</i>	The creation date to set.
------------------	---------------------------

Definition at line 158 of file File.java.

Here is the caller graph for this function:



### 6.10.2.12 setDescrizione()

```
void it.polimi.tiw.beans.File.setDescrizione (
    String descrizione )
```

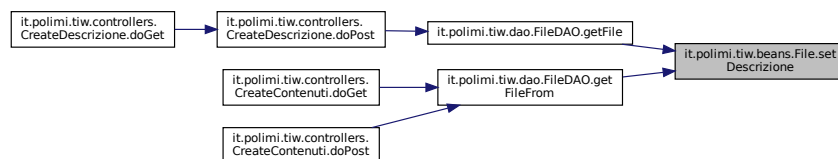
Sets the description of the file or directory.

## Parameters

<i>descrizione</i>	The description to set.
--------------------	-------------------------

Definition at line 176 of file File.java.

Here is the caller graph for this function:



### 6.10.2.13 setFigli()

```
void it.polimi.tiw.beans.File.setFigli (
    ArrayList< File > figli )
```

Sets the list of child files or directories.

## Parameters

<i>figli</i>	The list of child files to set.
--------------	---------------------------------

Definition at line 185 of file File.java.

**6.10.2.14 setIdDirPadre()**

```
void it.polimi.tiw.beans.File.setIdDirPadre (
    int idDirPadre )
```

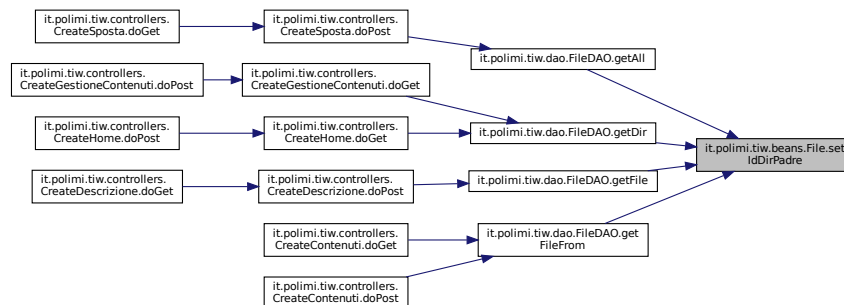
Sets the parent directory ID.

## Parameters

<i>idDirPadre</i>	The parent directory ID to set.
-------------------	---------------------------------

Definition at line 131 of file File.java.

Here is the caller graph for this function:

**6.10.2.15 setIdDocu()**

```
void it.polimi.tiw.beans.File.setIdDocu (
    int idDocu )
```

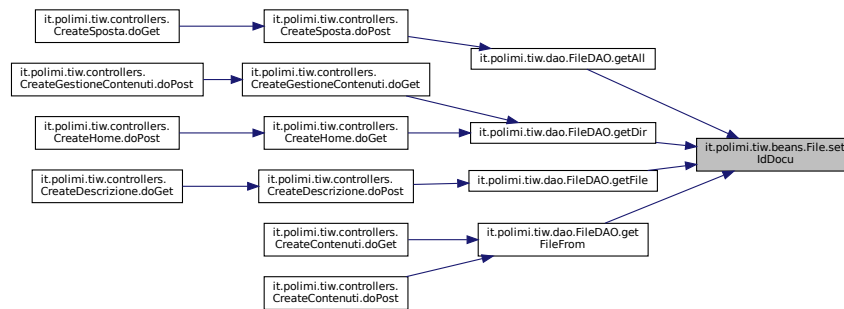
Sets the document ID.

## Parameters

<i>idDocu</i>	The document ID to set.
---------------	-------------------------

Definition at line 122 of file File.java.

Here is the caller graph for this function:



#### 6.10.2.16 setName()

```
void it.polimi.tiw.beans.File.setName (
    String nome )
```

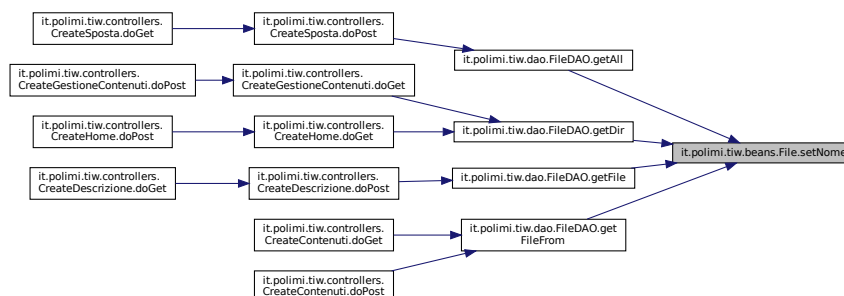
Sets the name of the file or directory.

##### Parameters

<i>nome</i>	The name to set.
-------------	------------------

Definition at line 149 of file File.java.

Here is the caller graph for this function:





**6.10.2.17 setNamePadre()**

```
void it.polimi.tiw.beans.File.setNamePadre (
    String nomePadre )
```

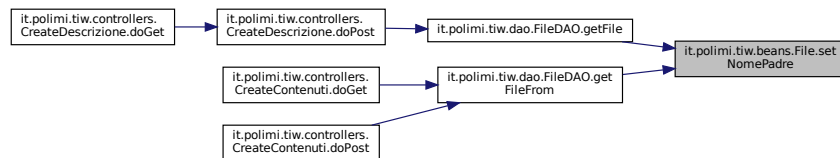
Sets the name of the parent directory.

**Parameters**

<i>nomePadre</i>	The name of the parent directory to set.
------------------	--

Definition at line 212 of file File.java.

Here is the caller graph for this function:

**6.10.2.18 setProprietario()**

```
void it.polimi.tiw.beans.File.setProprietario (
    int proprietario )
```

Sets the owner ID.

**Parameters**

<i>proprietario</i>	The owner ID to set.
---------------------	----------------------

Definition at line 140 of file File.java.

**6.10.2.19 setTipo()**

```
void it.polimi.tiw.beans.File.setTipo (
    String tipo )
```

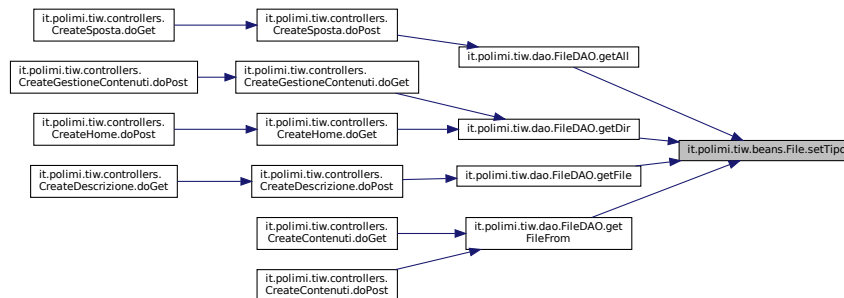
Sets the type of the file.

## Parameters

<i>tipo</i>	The type to set.
-------------	------------------

Definition at line 167 of file File.java.

Here is the caller graph for this function:



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

- [tiw/beans/File.java](#)

## 6.11 it.polimi.tiw.dao.FileDAO Class Reference

### Public Member Functions

- [FileDAO](#) (Connection connection)  
*Constructs a [FileDAO](#) with the specified database connection.*
- `ArrayList< File > getDir (int id) throws SQLException`  
*Retrieves all directories owned by a user.*
- `ArrayList< File > getAll (int id) throws SQLException`  
*Retrieves all files and directories owned by a user.*
- `ArrayList< File > getFileFrom (int id, int idDirPadre) throws SQLException`  
*Retrieves all files in a specific directory owned by a user.*
- `File getFile (int id, int idDocu) throws SQLException`  
*Retrieves the details of a specific file.*
- `boolean isDir (int id, int idDirDest) throws SQLException`  
*Checks if a specific ID corresponds to a directory owned by a user.*
- `boolean moveFile (int id, int idPartenza, int idDirDest) throws SQLException`  
*Moves a file to a new directory.*
- `boolean newDocu (int id, String nome, int idDirPadre, String tipo, String descrizione, String data) throws SQLException`  
*Inserts a new file or directory into the database.*

### 6.11.1 Detailed Description

This class allows for various operations on files and directories in the database, such as retrieving, inserting, and moving files, as well as checking directory status.

Definition at line 20 of file FileDAO.java.

### 6.11.2 Constructor & Destructor Documentation

#### 6.11.2.1 FileDAO()

```
it.polimi.tiw.dao.FileDAO.FileDAO (
    Connection connection )
```

Constructs a [FileDAO](#) with the specified database connection.

##### Parameters

<i>connection</i>	the database connection to be used by this DAO.
-------------------	---

Definition at line 28 of file FileDAO.java.

### 6.11.3 Member Function Documentation

#### 6.11.3.1 getAll()

```
ArrayList<File> it.polimi.tiw.dao.FileDAO.getAll (
    int id ) throws SQLException
```

Retrieves all files and directories owned by a user.

##### Parameters

<i>id</i>	the ID of the user whose files and directories are to be retrieved.
-----------	---

##### Returns

a list of File objects representing the user's files and directories.

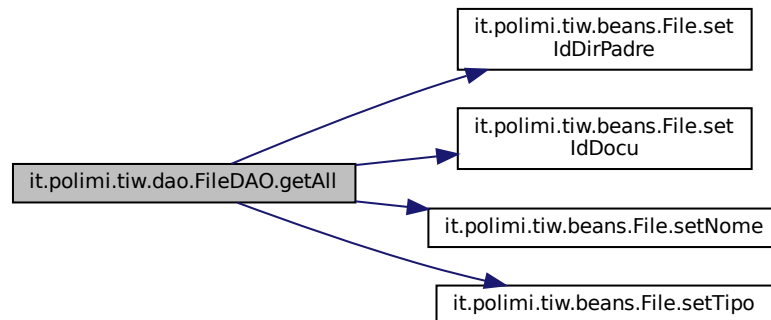
##### Exceptions

<i>SQLException</i>	if there is an error accessing the database.
---------------------	--

This method queries the database for all files and directories owned by the specified user and returns them in an ArrayList, sorted by their parent ID.

Definition at line 103 of file FileDAO.java.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.11.3.2 getDir()

```
ArrayList<File> it.polimi.tiw.dao.FileDAO.getDir (
    int id ) throws SQLException
```

Retrieves all directories owned by a user.

#### Parameters

<i>id</i>	the ID of the user whose directories are to be retrieved.
-----------	---

#### Returns

a list of File objects representing the user's directories.

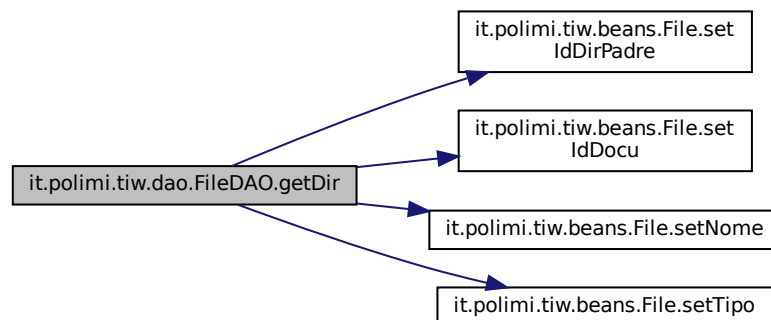
## Exceptions

<i>SQLException</i>	if there is an error accessing the database.
---------------------	--

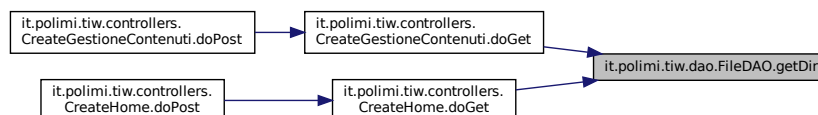
This method queries the database for directories owned by the specified user and returns them in an ArrayList. The directories are sorted by their parent ID.

Definition at line 41 of file FileDAO.java.

Here is the call graph for this function:



Here is the caller graph for this function:



## 6.11.3.3 getFile()

```
File it.polimi.tiw.dao.FileDAO.getFile (
    int id,
    int idDocu ) throws SQLException
```

Retrieves the details of a specific file.

## Parameters

<i>id</i>	the ID of the user who owns the file.
<i>idDocu</i>	the ID of the file to be retrieved.

**Returns**

a File object representing the specified file, or null if not found.

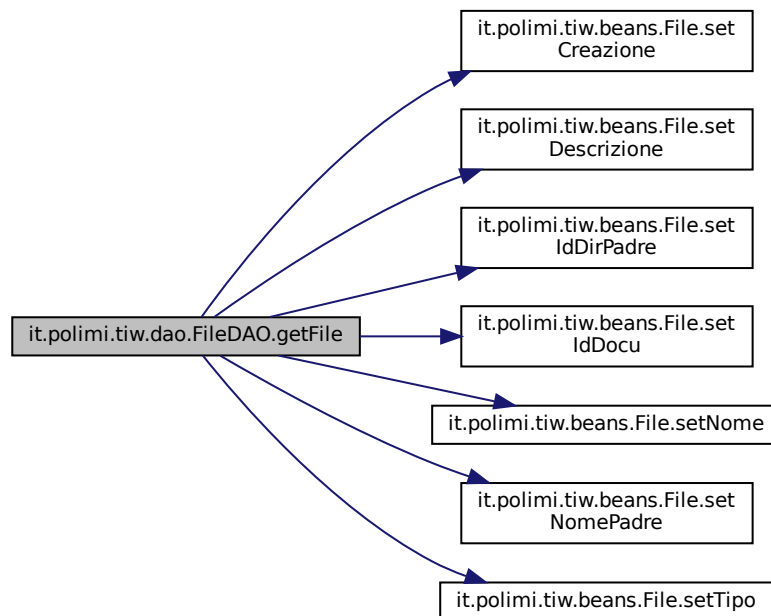
**Exceptions**

<i>SQLException</i>	if there is an error accessing the database.
---------------------	--

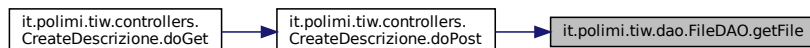
This method queries the database for a specific file's details based on the user ID and file ID.

Definition at line 227 of file FileDAO.java.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.11.3.4 getFileFrom()

```
ArrayList<File> it.polimi.tiw.dao.FileDAO.getFileFrom (
    int id,
    int idDirPadre ) throws SQLException
```

Retrieves all files in a specific directory owned by a user.

**Parameters**

<i>id</i>	the ID of the user whose files are to be retrieved.
<i>idDirPadre</i>	the ID of the parent directory.

**Returns**

a list of File objects representing the files in the specified directory.

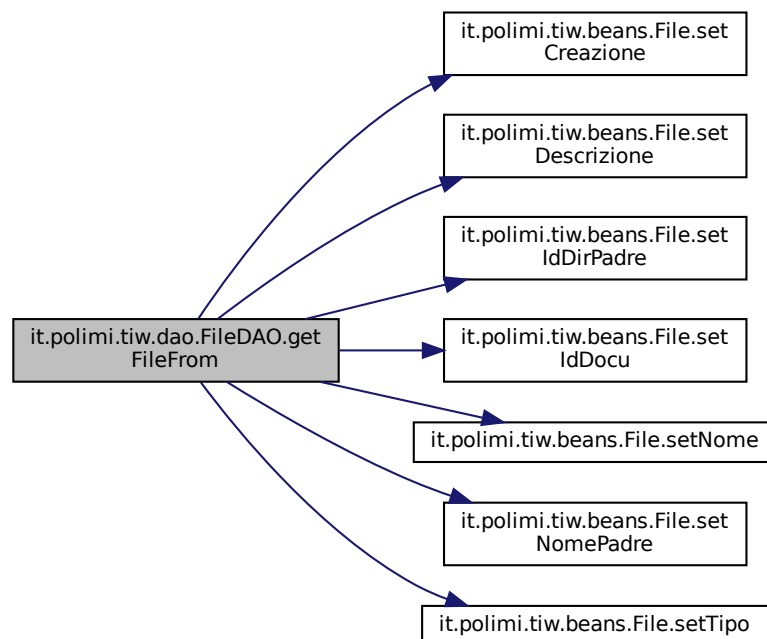
**Exceptions**

<i>SQLException</i>	if there is an error accessing the database.
---------------------	--

This method queries the database for all files that belong to the specified user and are located in the specified directory.

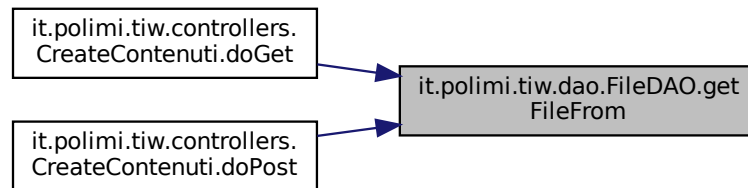
Definition at line 166 of file FileDAO.java.

Here is the call graph for this function:





Here is the caller graph for this function:



### 6.11.3.5 isDir()

```
boolean it.polimi.tiw.dao.FileDAO.isDir (
    int id,
    int idDirDest ) throws SQLException
```

Checks if a specific ID corresponds to a directory owned by a user.

#### Parameters

<i>id</i>	the ID of the user.
<i>idDirDest</i>	the ID of the directory to check.

#### Returns

true if the ID corresponds to a directory, false otherwise.

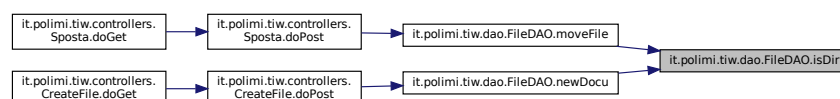
#### Exceptions

<i>SQLException</i>	if there is an error accessing the database.
---------------------	--

This method queries the database to determine if the specified ID corresponds to a directory owned by the user.

Definition at line 285 of file FileDAO.java.

Here is the caller graph for this function:



### 6.11.3.6 moveFile()

```
boolean it.polimi.tiw.dao.FileDAO.moveFile (  
    int id,  
    int idPartenza,  
    int idDirDest ) throws SQLException
```

Moves a file to a new directory.

#### Parameters

<i>id</i>	the ID of the user who owns the file.
<i>idPartenza</i>	the ID of the file to be moved.
<i>idDirDest</i>	the ID of the destination directory.

#### Returns

true if the file was successfully moved, false otherwise.

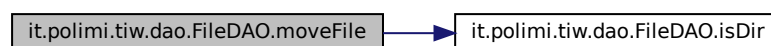
#### Exceptions

<i>SQLException</i>	if there is an error accessing the database.
---------------------	--

This method updates the database to move the specified file to the specified destination directory, ensuring that the destination is a valid directory.

Definition at line 319 of file FileDAO.java.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.11.3.7 newDocu()

```
boolean it.polimi.tiw.dao.FileDAO.newDocu (
    int id,
    String nome,
    int idDirPadre,
    String tipo,
    String descrizione,
    String data ) throws SQLException
```

Inserts a new file or directory into the database.

#### Parameters

<i>id</i>	the ID of the user creating the file or directory.
<i>nome</i>	the name of the file or directory.
<i>idDirPadre</i>	the ID of the parent directory, or 0 if it is a root element.
<i>tipo</i>	the type of the file or directory.
<i>descrizione</i>	a description of the file or directory.
<i>data</i>	the creation date of the file or directory.

#### Returns

true if the file or directory was successfully created, false otherwise.

#### Exceptions

<i>SQLException</i>	if there is an error accessing the database.
---------------------	--

This method inserts a new file or directory into the database with the specified attributes, checking if the parent directory exists if it is not a root element.

Definition at line 377 of file FileDAO.java.

Here is the call graph for this function:



Here is the caller graph for this function:

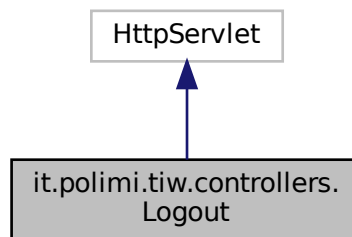


The documentation for this class was generated from the following file:

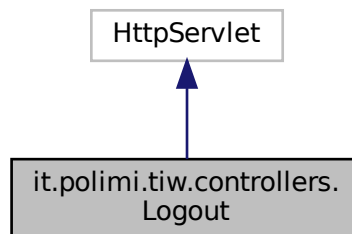
- [tiw/dao/FileDAO.java](#)

## 6.12 it.polimi.tiw.controllers.Logout Class Reference

Inheritance diagram for it.polimi.tiw.controllers.Logout:



Collaboration diagram for it.polimi.tiw.controllers.Logout:



### Public Member Functions

- [Logout](#) ()

### Protected Member Functions

- void [doGet](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles GET requests to log out the user by invalidating the session.*
- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles POST requests by redirecting to the doGet method.*

### 6.12.1 Detailed Description

This class extends `HttpServlet` and processes GET and POST requests to log out a user, ensuring that the session is invalidated and the user is redirected to the home page.

Definition at line 19 of file `Logout.java`.

### 6.12.2 Constructor & Destructor Documentation

#### 6.12.2.1 Logout()

```
it.polimi.tiw.controllers.Logout.Logout ( )
```

Definition at line 21 of file `Logout.java`.

### 6.12.3 Member Function Documentation

#### 6.12.3.1 doGet()

```
void it.polimi.tiw.controllers.Logout.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests to log out the user by invalidating the session.

##### Parameters

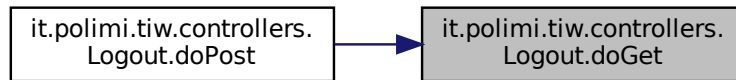
<i>request</i>	the <code>HttpServletRequest</code> object.
<i>response</i>	the <code>HttpServletResponse</code> object.

##### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 32 of file `Logout.java`.

Here is the caller graph for this function:



### 6.12.3.2 doPost()

```

void it.polimi.tiw.controllers.Login.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
  
```

Handles POST requests by redirecting to the doGet method.

#### Parameters

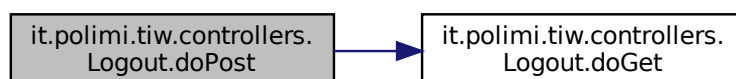
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

#### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 48 of file Login.java.

Here is the call graph for this function:

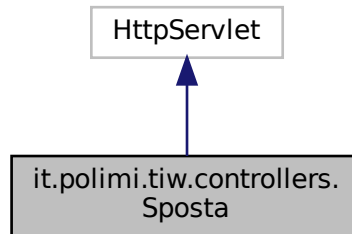


The documentation for this class was generated from the following file:

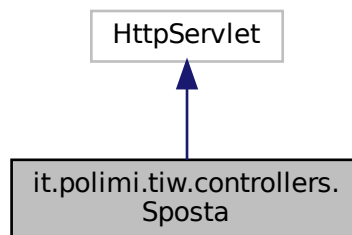
- [tiw/controllers/Login.java](#)

## 6.13 it.polimi.tiw.controllers.Sposta Class Reference

Inheritance diagram for it.polimi.tiw.controllers.Sposta:



Collaboration diagram for it.polimi.tiw.controllers.Sposta:



### Public Member Functions

- [Sposta](#) ()
- void [init](#) () throws ServletException  
*Initializes the servlet by establishing a database connection and setting up the template engine.*
- void [destroy](#) ()  
*Cleans up resources before the servlet is destroyed.*

### Protected Member Functions

- void [doGet](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles GET requests by redirecting to the doPost method.*
- void [doPost](#) (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException  
*Handles POST requests to move a file from one directory to another.*

### 6.13.1 Detailed Description

This class extends `HttpServlet` and processes HTTP requests to move files from one directory to another based on user input. It ensures that the user is authenticated and that necessary parameters are provided.

Definition at line 30 of file `Sposta.java`.

### 6.13.2 Constructor & Destructor Documentation

#### 6.13.2.1 `Sposta()`

```
it.polimi.tiw.controllers.Sposta.Sposta ( )
```

Definition at line 35 of file `Sposta.java`.

### 6.13.3 Member Function Documentation

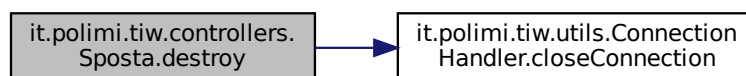
#### 6.13.3.1 `destroy()`

```
void it.polimi.tiw.controllers.Sposta.destroy ( )
```

Cleans up resources before the servlet is destroyed.

Definition at line 116 of file `Sposta.java`.

Here is the call graph for this function:



#### 6.13.3.2 `doGet()`

```
void it.polimi.tiw.controllers.Sposta.doGet (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles GET requests by redirecting to the `doPost` method.



## Parameters

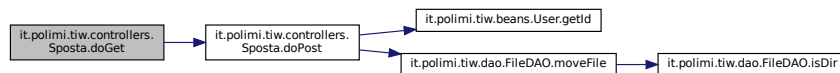
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

## Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 61 of file Sposta.java.

Here is the call graph for this function:



### 6.13.3.3 doPost()

```
void it.polimi.tiw.controllers.Sposta.doPost (
    HttpServletRequest request,
    HttpServletResponse response ) throws ServletException, IOException [protected]
```

Handles POST requests to move a file from one directory to another.

## Parameters

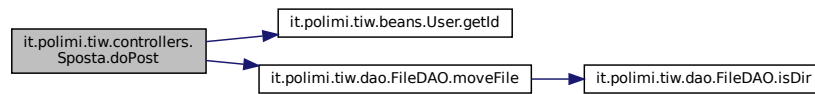
<i>request</i>	the HttpServletRequest object.
<i>response</i>	the HttpServletResponse object.

## Exceptions

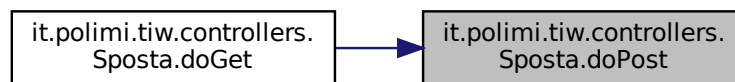
<i>ServletException</i>	if a servlet-related error occurs.
<i>IOException</i>	if an I/O error occurs during the processing of the request.

Definition at line 72 of file Sposta.java.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 6.13.3.4 init()

```
void it.polimi.tiw.controllers.Sposta.init ( ) throws ServletException
```

Initializes the servlet by establishing a database connection and setting up the template engine.

##### Parameters

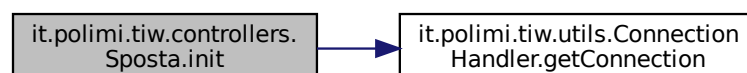
<i>servletContext</i>	the ServletContext for the application.
-----------------------	---

##### Exceptions

<i>ServletException</i>	if a servlet-related error occurs.
-------------------------	------------------------------------

Definition at line 44 of file `Sposta.java`.

Here is the call graph for this function:



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

- tiw/controllers/[Sposta.java](#)

## 6.14 it.polimi.tiw.utils.TreeOrder Class Reference

### Static Public Member Functions

- static [File](#) [getOrder](#) (ArrayList< [File](#) > lista)  
*Organizes a list of files into a tree structure.*

#### 6.14.1 Detailed Description

This utility class provides methods to arrange files in a hierarchical structure based on their parent directory relationships.

Definition at line 14 of file TreeOrder.java.

#### 6.14.2 Member Function Documentation

##### 6.14.2.1 getOrder()

```
static File it.polimi.tiw.utils.TreeOrder.getOrder (  
    ArrayList< File > lista ) [static]
```

Organizes a list of files into a tree structure.

##### Parameters

<i>lista</i>	an ArrayList of File objects to be organized.
--------------	---

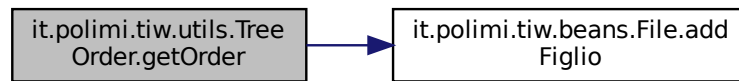
##### Returns

a File object representing the root of the organized tree.

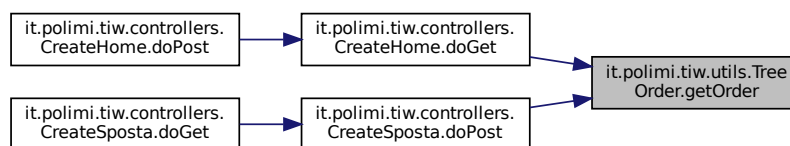
This method processes the input list of files, establishing parent-child relationships based on the directory IDs. Files with a parent directory ID of 0 are considered root elements and are added directly to the root of the tree. Other files are added as children to their respective parent directories.

Definition at line 26 of file TreeOrder.java.

Here is the call graph for this function:



Here is the caller graph for this function:



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

- [tiw/utlis/TreeOrder.java](#)

## 6.15 it.polimi.tiw.beans.User Class Reference

Represents a user in the system.

### Public Member Functions

- `int getId ()`  
*Gets the unique identifier for the user.*
- `String getMail ()`  
*Gets the email address of the user.*
- `String getNome ()`  
*Gets the name of the user.*
- `void setId (int id)`  
*Sets the unique identifier for the user.*
- `void setMail (String mail)`  
*Sets the email address of the user.*
- `void setNome (String nome)`  
*Sets the name of the user.*

### 6.15.1 Detailed Description

Represents a user in the system.

This class encapsulates the properties of a user, including their ID, email address, and name.

Definition at line 12 of file User.java.

### 6.15.2 Member Function Documentation

#### 6.15.2.1 getId()

```
int it.polimi.tiw.beans.User.getId ( )
```

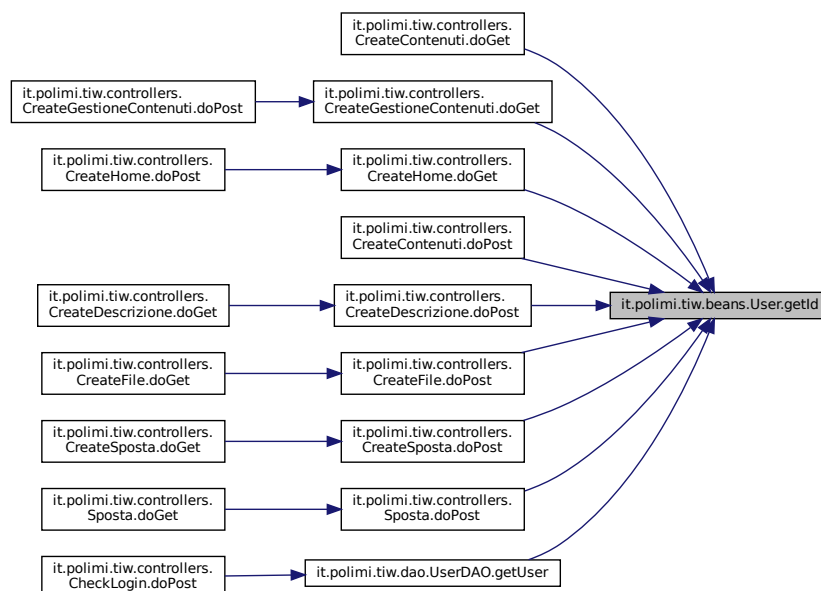
Gets the unique identifier for the user.

##### Returns

The unique identifier for the user.

Definition at line 30 of file User.java.

Here is the caller graph for this function:



### 6.15.2.2 `getMail()`

```
String it.polimi.tiw.beans.User.getMail ( )
```

Gets the email address of the user.

#### Returns

The email address of the user.

Definition at line 38 of file User.java.

### 6.15.2.3 `getNome()`

```
String it.polimi.tiw.beans.User.getNome ( )
```

Gets the name of the user.

#### Returns

The name of the user.

Definition at line 46 of file User.java.

### 6.15.2.4 `setId()`

```
void it.polimi.tiw.beans.User.setId (
    int id )
```

Sets the unique identifier for the user.

#### Parameters

<i>id</i>	The new unique identifier for the user.
-----------	---

Definition at line 57 of file User.java.

Here is the caller graph for this function:



### 6.15.2.5 setMail()

```
void it.polimi.tiw.beans.User.setMail (
    String mail )
```

Sets the email address of the user.

#### Parameters

<i>mail</i>	The new email address for the user.
-------------	-------------------------------------

Definition at line 65 of file User.java.

Here is the caller graph for this function:



### 6.15.2.6 setNome()

```
void it.polimi.tiw.beans.User.setNome (
    String nome )
```

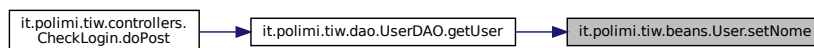
Sets the name of the user.

#### Parameters

<i>nome</i>	The new name for the user.
-------------	----------------------------

Definition at line 73 of file User.java.

Here is the caller graph for this function:



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

- [tiw/beans/User.java](#)

## 6.16 it.polimi.tiw.dao.UserDAO Class Reference

Data Access Object for User entity.

### Public Member Functions

- [UserDAO](#) (Connection connection)  
*Constructs a [UserDAO](#) with the given database connection.*
- [User getUser](#) (String mail, String psw) throws SQLException  
*Retrieves a user from the database using their email and password.*
- boolean [isExistingMail](#) (String mail) throws SQLException  
*Checks if a given email exists in the database.*
- boolean [isExistingUsername](#) (String nome) throws SQLException  
*Checks if a given username exists in the database.*
- int [addUser](#) (String nome, String mail, String psw) throws SQLException  
*Adds a new user to the database.*

### 6.16.1 Detailed Description

Data Access Object for User entity.

This class provides methods for interacting with the user data stored in the database. It allows for operations like retrieving user details, checking existing email and username, and adding a new user.

Definition at line 21 of file UserDAO.java.

### 6.16.2 Constructor & Destructor Documentation

#### 6.16.2.1 UserDAO()

```
it.polimi.tiw.dao.UserDAO.UserDAO (
    Connection connection )
```

Constructs a [UserDAO](#) with the given database connection.

#### Parameters

<i>connection</i>	The database connection to be used by the DAO.
-------------------	--

Definition at line 30 of file UserDAO.java.

### 6.16.3 Member Function Documentation



### 6.16.3.1 addUser()

```
int it.polimi.tiw.dao.UserDAO.addUser (
    String nome,
    String mail,
    String psw ) throws SQLException
```

Adds a new user to the database.

This method registers a new user if the username and email are not already in use.

#### Parameters

<i>nome</i>	The username of the new user.
<i>mail</i>	The email of the new user.
<i>psw</i>	The password of the new user.

#### Returns

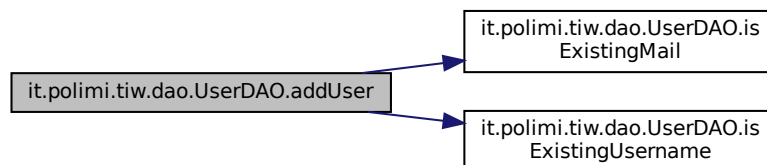
0 if the user was added successfully, 1 if the email already exists, 2 if the username already exists.

#### Exceptions

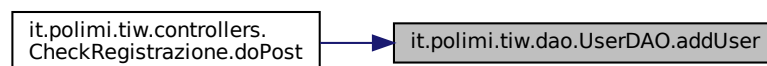
<i>SQLException</i>	If an error occurs while accessing the database.
---------------------	--

Definition at line 207 of file UserDAO.java.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.16.3.2 getUser()

```
User it.polimi.tiw.dao.UserDAO.getUser (  
    String mail,  
    String psw ) throws SQLException
```

Retrieves a user from the database using their email and password.

This method searches for a user with the specified email and password.

#### Parameters

<i>mail</i>	The email of the user.
<i>psw</i>	The password of the user.

#### Returns

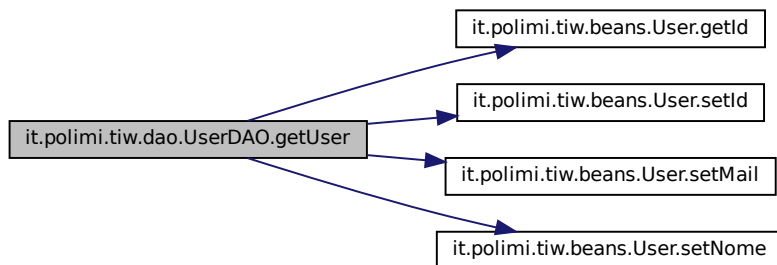
The User object if found, otherwise null.

#### Exceptions

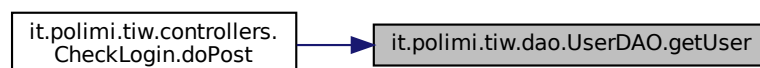
<i>SQLException</i>	If an error occurs while accessing the database.
---------------------	--

Definition at line 44 of file UserDAO.java.

Here is the call graph for this function:



Here is the caller graph for this function:



### 6.16.3.3 isExistingMail()

```
boolean it.polimi.tiw.dao.UserDAO.isExistingMail (  
    String mail ) throws SQLException
```

Checks if a given email exists in the database.

This method verifies if the specified email is already associated with a user in the database.

#### Parameters

<i>mail</i>	The email to check.
-------------	---------------------

#### Returns

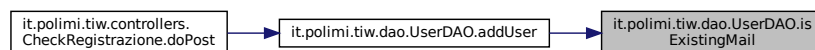
True if the email exists, otherwise false.

#### Exceptions

<i>SQLException</i>	If an error occurs while accessing the database.
---------------------	--

Definition at line 100 of file UserDAO.java.

Here is the caller graph for this function:



### 6.16.3.4 isExistingUsername()

```
boolean it.polimi.tiw.dao.UserDAO.isExistingUsername (  
    String nome ) throws SQLException
```

Checks if a given username exists in the database.

This method verifies if the specified username is already associated with a user in the database.

#### Parameters

<i>nome</i>	The username to check.
-------------	------------------------

### Returns

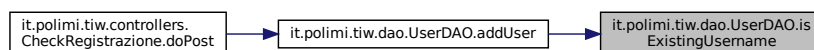
True if the username exists, otherwise false.

### Exceptions

<i>SQLException</i>	If an error occurs while accessing the database.
---------------------	--

Definition at line 152 of file UserDao.java.

Here is the caller graph for this function:



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

- tiw/dao/[UserDAO.java](#)

## Chapter 7

# File Documentation

### 7.1 tiw/beans/File.java File Reference

#### Classes

- class [it.polimi.tiw.beans.File](#)  
*Represents a file or directory in the document management system.*

#### Packages

- package [it.polimi.tiw.beans](#)

### 7.2 tiw/beans/User.java File Reference

#### Classes

- class [it.polimi.tiw.beans.User](#)  
*Represents a user in the system.*

#### Packages

- package [it.polimi.tiw.beans](#)

### 7.3 tiw/controllers/CheckLogin.java File Reference

Servlet that handles user login by checking credentials and managing session creation.

#### Classes

- class [it.polimi.tiw.controllers.CheckLogin](#)  
*A servlet to manage user authentication and session creation.*

## Packages

- package [it.polimi.tiw.controllers](#)

### 7.3.1 Detailed Description

Servlet that handles user login by checking credentials and managing session creation.

This servlet processes login requests, validates user credentials, manages session creation, and handles both GET and POST requests.

## 7.4 tiw/controllers/CheckRegistrazione.java File Reference

Servlet that handles the user registration process by validating input and adding new users to the database.

### Classes

- class [it.polimi.tiw.controllers.CheckRegistrazione](#)  
*Servlet that handles the user registration process.*

## Packages

- package [it.polimi.tiw.controllers](#)

### 7.4.1 Detailed Description

Servlet that handles the user registration process by validating input and adding new users to the database.

This servlet manages both GET requests (to display the registration page) and POST requests (to handle the registration process). It checks the validity of input fields such as name, email, and password, ensuring that they are non-empty and valid. If the input is correct, the servlet adds the user to the database using the `UserDAO` class.

#### Note

It uses the Thymeleaf template engine to render HTML pages.

This servlet also manages errors such as existing usernames and emails.

## 7.5 tiw/controllers/CreateContenuti.java File Reference

This servlet handles the creation and display of contents in a user's directory.

### Classes

- class [it.polimi.tiw.controllers.CreateContenuti](#)

## Packages

- package [it.polimi.tiw.controllers](#)

### 7.5.1 Detailed Description

This servlet handles the creation and display of contents in a user's directory.

## 7.6 tiw/controllers/CreateDescrizione.java File Reference

This servlet handles the display of file descriptions.

## Classes

- class [it.polimi.tiw.controllers.CreateDescrizione](#)

## Packages

- package [it.polimi.tiw.controllers](#)

### 7.6.1 Detailed Description

This servlet handles the display of file descriptions.

## 7.7 tiw/controllers/CreateFile.java File Reference

This servlet handles the creation of new files and directories.

## Classes

- class [it.polimi.tiw.controllers.CreateFile](#)

## Packages

- package [it.polimi.tiw.controllers](#)

### 7.7.1 Detailed Description

This servlet handles the creation of new files and directories.

## 7.8 tiw/controllers/CreateGestioneContenuti.java File Reference

This servlet manages the content creation and display for the user.

### Classes

- class [it.polimi.tiw.controllers.CreateGestioneContenuti](#)

### Packages

- package [it.polimi.tiw.controllers](#)

#### 7.8.1 Detailed Description

This servlet manages the content creation and display for the user.

## 7.9 tiw/controllers/CreateHome.java File Reference

This servlet handles the home page for content management, displaying the user's directories.

### Classes

- class [it.polimi.tiw.controllers.CreateHome](#)

### Packages

- package [it.polimi.tiw.controllers](#)

#### 7.9.1 Detailed Description

This servlet handles the home page for content management, displaying the user's directories.

## 7.10 tiw/controllers/CreateSposta.java File Reference

This servlet handles the file moving process, allowing users to move files between directories.

### Classes

- class [it.polimi.tiw.controllers.CreateSposta](#)



## Packages

- package [it.polimi.tiw.controllers](#)

### 7.10.1 Detailed Description

This servlet handles the file moving process, allowing users to move files between directories.

## 7.11 tiw/controllers/Logout.java File Reference

This servlet handles user logout functionality by invalidating the session.

## Classes

- class [it.polimi.tiw.controllers.Logout](#)

## Packages

- package [it.polimi.tiw.controllers](#)

### 7.11.1 Detailed Description

This servlet handles user logout functionality by invalidating the session.

## 7.12 tiw/controllers/Sposta.java File Reference

This servlet handles the movement of files between directories.

## Classes

- class [it.polimi.tiw.controllers.Sposta](#)

## Packages

- package [it.polimi.tiw.controllers](#)

### 7.12.1 Detailed Description

This servlet handles the movement of files between directories.

## 7.13 tiw/dao/FileDAO.java File Reference

This class provides methods for interacting with the file database.

### Classes

- class [it.polimi.tiw.dao.FileDAO](#)

### Packages

- package [it.polimi.tiw.dao](#)

### 7.13.1 Detailed Description

This class provides methods for interacting with the file database.

## 7.14 tiw/dao/UserDAO.java File Reference

### Classes

- class [it.polimi.tiw.dao.UserDAO](#)  
*Data Access Object for User entity.*

### Packages

- package [it.polimi.tiw.dao](#)

## 7.15 tiw/utils/ConnectionHandler.java File Reference

This class handles the management of database connections.

### Classes

- class [it.polimi.tiw.utils.ConnectionHandler](#)

### Packages

- package [it.polimi.tiw.utils](#)

### 7.15.1 Detailed Description

This class handles the management of database connections.

## 7.16 tiw/Utils/TreeOrder.java File Reference

This class manages the organization of files into a tree structure.

### Classes

- class [it.polimi.tiw.Utils.TreeOrder](#)

### Packages

- package [it.polimi.tiw.Utils](#)

#### 7.16.1 Detailed Description

This class manages the organization of files into a tree structure.



# Index

- addFiglio
  - it.polimi.tiw.beans.File, 50
- addUser
  - it.polimi.tiw.dao.UserDAO, 80
- CheckLogin
  - it.polimi.tiw.controllers.CheckLogin, 12
- CheckRegistrazione
  - it.polimi.tiw.controllers.CheckRegistrazione, 16
- closeConnection
  - it.polimi.tiw.utils.ConnectionHandler, 19
- CreateContenuti
  - it.polimi.tiw.controllers.CreateContenuti, 23
- CreateDescrizione
  - it.polimi.tiw.controllers.CreateDescrizione, 27
- CreateFile
  - it.polimi.tiw.controllers.CreateFile, 32
- CreateGestioneContenuti
  - it.polimi.tiw.controllers.CreateGestioneContenuti, 36
- CreateHome
  - it.polimi.tiw.controllers.CreateHome, 41
- CreateSposta
  - it.polimi.tiw.controllers.CreateSposta, 45
- destroy
  - it.polimi.tiw.controllers.CheckLogin, 12
  - it.polimi.tiw.controllers.CheckRegistrazione, 16
  - it.polimi.tiw.controllers.CreateContenuti, 23
  - it.polimi.tiw.controllers.CreateDescrizione, 27
  - it.polimi.tiw.controllers.CreateFile, 32
  - it.polimi.tiw.controllers.CreateGestioneContenuti, 37
  - it.polimi.tiw.controllers.CreateHome, 41
  - it.polimi.tiw.controllers.CreateSposta, 45
  - it.polimi.tiw.controllers.Sposta, 72
- doGet
  - it.polimi.tiw.controllers.CheckLogin, 13
  - it.polimi.tiw.controllers.CheckRegistrazione, 17
  - it.polimi.tiw.controllers.CreateContenuti, 23
  - it.polimi.tiw.controllers.CreateDescrizione, 28
  - it.polimi.tiw.controllers.CreateFile, 33
  - it.polimi.tiw.controllers.CreateGestioneContenuti, 37
  - it.polimi.tiw.controllers.CreateHome, 41
  - it.polimi.tiw.controllers.CreateSposta, 45
  - it.polimi.tiw.controllers.Logout, 69
  - it.polimi.tiw.controllers.Sposta, 72
- doPost
  - it.polimi.tiw.controllers.CheckLogin, 13
  - it.polimi.tiw.controllers.CheckRegistrazione, 17
  - it.polimi.tiw.controllers.CreateContenuti, 24
  - it.polimi.tiw.controllers.CreateDescrizione, 29
  - it.polimi.tiw.controllers.CreateFile, 33
  - it.polimi.tiw.controllers.CreateGestioneContenuti, 38
  - it.polimi.tiw.controllers.CreateHome, 42
  - it.polimi.tiw.controllers.CreateSposta, 47
  - it.polimi.tiw.controllers.Logout, 70
  - it.polimi.tiw.controllers.Sposta, 73
- FileDAO
  - it.polimi.tiw.dao.FileDAO, 59
- getAll
  - it.polimi.tiw.dao.FileDAO, 59
- getConnection
  - it.polimi.tiw.utils.ConnectionHandler, 20
- getCreazione
  - it.polimi.tiw.beans.File, 50
- getDescrizione
  - it.polimi.tiw.beans.File, 51
- getDir
  - it.polimi.tiw.dao.FileDAO, 60
- getFigli
  - it.polimi.tiw.beans.File, 51
- getFile
  - it.polimi.tiw.dao.FileDAO, 61
- getFileFrom
  - it.polimi.tiw.dao.FileDAO, 62
- getId
  - it.polimi.tiw.beans.User, 77
- getIdDirPadre
  - it.polimi.tiw.beans.File, 51
- getIdDocu
  - it.polimi.tiw.beans.File, 52
- getMail
  - it.polimi.tiw.beans.User, 77
- getNome
  - it.polimi.tiw.beans.File, 52
  - it.polimi.tiw.beans.User, 78
- getNomePadre
  - it.polimi.tiw.beans.File, 52
- getOrder
  - it.polimi.tiw.utils.TreeOrder, 75
- getProprietario
  - it.polimi.tiw.beans.File, 53
- getTipo
  - it.polimi.tiw.beans.File, 53
- getUser

- it.polimi.tiw.dao.UserDAO, 81
- init
  - it.polimi.tiw.controllers.CheckLogin, 14
  - it.polimi.tiw.controllers.CheckRegistrazione, 18
  - it.polimi.tiw.controllers.CreateContenuti, 25
  - it.polimi.tiw.controllers.CreateDescrizione, 30
  - it.polimi.tiw.controllers.CreateFile, 34
  - it.polimi.tiw.controllers.CreateGestioneContenuti, 39
  - it.polimi.tiw.controllers.CreateHome, 43
  - it.polimi.tiw.controllers.CreateSpota, 48
  - it.polimi.tiw.controllers.Spota, 74
- isDir
  - it.polimi.tiw.dao.FileDAO, 65
- isExistingMail
  - it.polimi.tiw.dao.UserDAO, 82
- isExistingUsername
  - it.polimi.tiw.dao.UserDAO, 83
- it.polimi.tiw.beans, 9
- it.polimi.tiw.beans.File, 49
  - addFiglio, 50
  - getCreazione, 50
  - getDescrizione, 51
  - getFigli, 51
  - getlDirPadre, 51
  - getlDocu, 52
  - getNome, 52
  - getNomePadre, 52
  - getProprietario, 53
  - getTipo, 53
  - setCreazione, 53
  - setDescrizione, 54
  - setFigli, 54
  - setlDirPadre, 55
  - setlDocu, 55
  - setNome, 56
  - setNomePadre, 56
  - setProprietario, 57
  - setTipo, 57
- it.polimi.tiw.beans.User, 76
  - getl, 77
  - getMail, 77
  - getNome, 78
  - setl, 78
  - setMail, 78
  - setNome, 79
- it.polimi.tiw.controllers, 9
- it.polimi.tiw.controllers.CheckLogin, 11
  - CheckLogin, 12
  - destroy, 12
  - doGet, 13
  - doPost, 13
  - init, 14
- it.polimi.tiw.controllers.CheckRegistrazione, 15
  - CheckRegistrazione, 16
  - destroy, 16
  - doGet, 17
  - doPost, 17
- init, 18
- it.polimi.tiw.controllers.CreateContenuti, 22
  - CreateContenuti, 23
  - destroy, 23
  - doGet, 23
  - doPost, 24
  - init, 25
- it.polimi.tiw.controllers.CreateDescrizione, 26
  - CreateDescrizione, 27
  - destroy, 27
  - doGet, 28
  - doPost, 29
  - init, 30
- it.polimi.tiw.controllers.CreateFile, 31
  - CreateFile, 32
  - destroy, 32
  - doGet, 33
  - doPost, 33
  - init, 34
- it.polimi.tiw.controllers.CreateGestioneContenuti, 35
  - CreateGestioneContenuti, 36
  - destroy, 37
  - doGet, 37
  - doPost, 38
  - init, 39
- it.polimi.tiw.controllers.CreateHome, 40
  - CreateHome, 41
  - destroy, 41
  - doGet, 41
  - doPost, 42
  - init, 43
- it.polimi.tiw.controllers.CreateSpota, 44
  - CreateSpota, 45
  - destroy, 45
  - doGet, 45
  - doPost, 47
  - init, 48
- it.polimi.tiw.controllers.Login, 68
  - doGet, 69
  - doPost, 70
  - Login, 69
- it.polimi.tiw.controllers.Spota, 71
  - destroy, 72
  - doGet, 72
  - doPost, 73
  - init, 74
  - Spota, 72
- it.polimi.tiw.dao, 9
- it.polimi.tiw.dao.FileDAO, 58
  - FileDAO, 59
  - getAll, 59
  - getDir, 60
  - getFile, 61
  - getFileFrom, 62
  - isDir, 65
  - moveFile, 65
  - newDocu, 66
- it.polimi.tiw.dao.UserDAO, 80

- addUser, [80](#)
- getUser, [81](#)
- isExistingMail, [82](#)
- isExistingUsername, [83](#)
- UserDAO, [80](#)
- it.polimi.tiw.utils, [10](#)
- it.polimi.tiw.utils.ConnectionHandler, [19](#)
  - closeConnection, [19](#)
  - getConnection, [20](#)
- it.polimi.tiw.utils.TreeOrder, [75](#)
  - getOrder, [75](#)
- Logout
  - it.polimi.tiw.controllers.Logout, [69](#)
- moveFile
  - it.polimi.tiw.dao.FileDAO, [65](#)
- newDocu
  - it.polimi.tiw.dao.FileDAO, [66](#)
- setCreazione
  - it.polimi.tiw.beans.File, [53](#)
- setDescrizione
  - it.polimi.tiw.beans.File, [54](#)
- setFigli
  - it.polimi.tiw.beans.File, [54](#)
- setId
  - it.polimi.tiw.beans.User, [78](#)
- setIdDirPadre
  - it.polimi.tiw.beans.File, [55](#)
- setIdDocu
  - it.polimi.tiw.beans.File, [55](#)
- setMail
  - it.polimi.tiw.beans.User, [78](#)
- setNome
  - it.polimi.tiw.beans.File, [56](#)
  - it.polimi.tiw.beans.User, [79](#)
- setNomePadre
  - it.polimi.tiw.beans.File, [56](#)
- setProprietario
  - it.polimi.tiw.beans.File, [57](#)
- setTipo
  - it.polimi.tiw.beans.File, [57](#)
- Sposta
  - it.polimi.tiw.controllers.Sposta, [72](#)
- tiw/beans/File.java, [85](#)
- tiw/beans/User.java, [85](#)
- tiw/controllers/CheckLogin.java, [85](#)
- tiw/controllers/CheckRegistrazione.java, [86](#)
- tiw/controllers/CreateContenuti.java, [86](#)
- tiw/controllers/CreateDescrizione.java, [87](#)
- tiw/controllers/CreateFile.java, [87](#)
- tiw/controllers/CreateGestioneContenuti.java, [88](#)
- tiw/controllers/CreateHome.java, [88](#)
- tiw/controllers/CreateSposta.java, [88](#)
- tiw/controllers/Logout.java, [89](#)
- tiw/controllers/Sposta.java, [89](#)
- tiw/dao/FileDAO.java, [90](#)
- tiw/dao/UserDAO.java, [90](#)
- tiw/utils/ConnectionHandler.java, [90](#)
- tiw/utils/TreeOrder.java, [91](#)
- UserDAO
  - it.polimi.tiw.dao.UserDAO, [80](#)