Extension.BackEnd

Class StateManager

```
java.lang.Object
   └ Extension.BackEnd.StateManager
```

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

The StateManager class controls the current state of the extension. It holds all necessary information such as the GUI's and the currently chosen exercise.

Version:

April2015

Author:

Miguel Roman-Roman, Wei Huang, Di Tran, Josh Gillham, Nathan Witt, Thomas Macari

Method Summary

static void	**createDescriptionGUI**() Creates the GUI for the description of the exercise.
static void	**createExerciseListGUI**() Creates the GUI for picking an exercise.
static void	**createTestResultsGUI**() Creates the GUI showing the JUnit tests for the exercise.
static void	**disposeDescriptionGUI**() Disposes the DescriptionGUI
static void	**disposeExerciseGUI**() Disposes the ExerciseListGUI
static void	**disposeTestResultsGUI**() Disposes the TestResultsGUI
static bluej.extensions.BlueJ	**getBlueJ**() Returns the BlueJ instance held
static Extension.BackEnd.Exercise	**getCurrentExercise**() Returns the current exercise chosen by the user.
static Extension.GUI.DescriptionGUI	**getDescriptionGUI**() Returns the held instance of the DescriptionGUI
static java.util.ArrayList<Extension.BackEnd.Exercise>	**getExerciseList**() Initializes the exercise list.
static Extension.GUI.ExerciseListGUI	**getExerciseListGUI**() Returns the held instance of the ExerciseListGUI
static java.lang.String	**getExercisesLocation**() Returns the String location of the exercises.
static java.io.File	**getTempDir**() Returns a File for the location of the Temporary Directory created by this extension.

static Extension.GUI.TestResultsGUI	**getTestResultsGUI**() Returns the held instance of the TestResultGUI
static void	**parseExercises**() Parses the exercises in the location specified and populates the exercise list
static void	**setCurrentExercise**(Extension.BackEnd.Exercise ex) Sets the current exercise chosen by the user.
static void	**setDescriptionGUI**(Extension.GUI.DescriptionGUI gui) Sets the intance of DescriptionGUI
static void	**setExerciseListGUI**(Extension.GUI.ExerciseListGUI gui) Sets the instance of ExerciseListGUI
static void	**setExercisesLocation**(java.lang.String location) Sets the location of the exercises.
static void	**setTestResultsGUI**(Extension.GUI.TestResultsGUI gui) Sets the instance of TestResultsGUI
static void	**setup**(bluej.extensions.BlueJ bluej) Used to set up this State Manager and extension.
static void	**showErrorMessage**(java.lang.String msg) Used to show an error message.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Method Detail

createDescriptionGUI

public static void **createDescriptionGUI**()

Creates the GUI for the description of the exercise.

---

createExerciseListGUI

public static void **createExerciseListGUI**()

Creates the GUI for picking an exercise.

---

createTestResultsGUI

public static void **createTestResultsGUI**()

Creates the GUI showing the JUnit tests for the exercise.

---

disposeDescriptionGUI

public static void **disposeDescriptionGUI**()

Disposes the DescriptionGUI

disposeExerciseGUI

```
public static void disposeExerciseGUI()
```

>   Disposes the ExerciseListGUI

disposeTestResultsGUI

```
public static void disposeTestResultsGUI()
```

>   Disposes the TestResultsGUI

getBlueJ

```
public static bluej.extensions.BlueJ getBlueJ()
```

>   Returns the BlueJ instance held
>
>   **Returns:**
>   >   the BlueJ instance

getCurrentExercise

```
public static Extension.BackEnd.Exercise getCurrentExercise()
```

>   Returns the current exercise chosen by the user.
>
>   **Returns:**
>   >   the currently selected exercise

getDescriptionGUI

```
public static Extension.GUI.DescriptionGUI getDescriptionGUI()
```

>   Returns the held instance of the DescriptionGUI
>
>   **Returns:**
>   >   the DescriptionGUI

getExerciseList

```
public static java.util.ArrayList<Extension.BackEnd.Exercise> getExerciseList()
```

>   Initializes the exercise list.
>
>   **Returns:**
>   >   exercises the list of exercises.

getExerciseListGUI

```
public static Extension.GUI.ExerciseListGUI getExerciseListGUI()
```

>   Returns the held instance of the ExerciseListGUI

Returns:

    the ExerciseListGUI

---

getExercisesLocation

```
public static java.lang.String getExercisesLocation()
```

Returns the String location of the exercises.

Returns:

    the path of the exercises

---

getTempDir

```
public static java.io.File getTempDir()
```

Returns a File for the location of the Temporary Directory created by this extension.

Returns:

    the location of the Temporary Directory

---

getTestResultsGUI

```
public static Extension.GUI.TestResultsGUI getTestResultsGUI()
```

Returns the held instance of the TestResultGUI

Returns:

    the TestResultsGUI

---

parseExercises

```
public static void parseExercises()
```

Parses the exercises in the location specified and populates the exercise list

---

setCurrentExercise

```
public static void setCurrentExercise(Extension.BackEnd.Exercise ex)
```

Sets the current exercise chosen by the user.

Parameters:

    ex - the exercise

---

setDescriptionGUI

```
public static void setDescriptionGUI(Extension.GUI.DescriptionGUI gui)
```

Sets the intance of DescriptionGUI

Parameters:

    gui - the DescriptionGUI

---

setExerciseListGUI

`public static void `**`setExerciseListGUI`**`(Extension.GUI.ExerciseListGUI gui)`

Sets the instance of ExerciseListGUI

Parameters:

`gui` - the ExerciseListGUI

---

setExercisesLocation

`public static void `**`setExercisesLocation`**`(java.lang.String location)`

Sets the location of the exercises.

Parameters:

`location` - the location of the exercises

---

setTestResultsGUI

`public static void `**`setTestResultsGUI`**`(Extension.GUI.TestResultsGUI gui)`

Sets the instance of TestResultsGUI

Parameters:

`gui` - the TestResultsGUI

---

setup

`public static void `**`setup`**`(bluej.extensions.BlueJ bluej)`

Used to set up this State Manager and extension. Creates proper folders in the BlueJ config directory. Parses the exercises from known location.

Parameters:

`bluej` - the BlueJ instance

---

showErrorMessage

`public static void `**`showErrorMessage`**`(java.lang.String msg)`

Used to show an error message. Displays a JOptionPane with given message.

Parameters:

`msg` - the message to display