

GWT - PASSWORDTEXTBOX WIDGET

http://www.tutorialspoint.com/gwt/gwt_passwordtextbox_widget.htm

Copyright © tutorialspoint.com

Introduction

The **PasswordTextBox** widget represents a standard single-line text box that visually masks its input to prevent eavesdropping.

Class declaration

Following is the declaration for **com.google.gwt.user.client.ui.PasswordTextBox** class:

```
public class PasswordTextBox  
    extends TextBox
```

CSS style rules

Following default CSS Style rules will be applied to all the PasswordTextBox widget. You can override it as per your requirements.

```
.gwt-PasswordTextBox {}  
.gwt-PasswordTextBox-readonly {}
```

Class constructors

S.N.	Constructor & Description
1	PasswordTextBox Creates an empty password text box.
2	PasswordTextBoxElementelement This constructor may be used by subclasses to explicitly use an existing element.

Class methods

S.N.	Function name & Description
1	static PasswordTextBox wrapElementelement Creates a PasswordTextBox widget that wraps an existing <input type='password'> element.

Methods inherited

This class inherits methods from the following classes:

- com.google.gwt.user.client.ui.UIObject
- com.google.gwt.user.client.ui.Widget

- com.google.gwt.user.client.ui.FocusWidget
- com.google.gwt.user.client.ui.TextBoxBase
- com.google.gwt.user.client.ui.TextBox
- java.lang.Object

PasswordTextBox Widget Example

This example will take you through simple steps to show usage of a PasswordTextBox Widget in GWT. Follow the following steps to update the GWT application we created in *GWT - Create Application* chapter:

Step	Description
1	Create a project with a name <i>HelloWorld</i> under a package <i>com.tutorialspoint</i> as explained in the <i>GWT - Create Application</i> chapter.
2	Modify <i>HelloWorld.gwt.xml</i> , <i>HelloWorld.css</i> , <i>HelloWorld.html</i> and <i>HelloWorld.java</i> as explained below. Keep rest of the files unchanged.
3	Compile and run the application to verify the result of the implemented logic.

Following is the content of the modified module descriptor **src/com.tutorialspoint/HelloWorld.gwt.xml**.

```
<?xml version="1.0" encoding="UTF-8"?>
<module rename-to='helloworld'>
  <!-- Inherit the core Web Toolkit stuff. -->
  <inherits name='com.google.gwt.user.User' />

  <!-- Inherit the default GWT style sheet. -->
  <inherits name='com.google.gwt.user.theme.clean.Clean' />

  <!-- Specify the app entry point class. -->
  <entry-point class='com.tutorialspoint.client.HelloWorld' />

  <!-- Specify the paths for translatable code -->
  <source path='client' />
  <source path='shared' />

</module>
```

Following is the content of the modified Style Sheet file **war/HelloWorld.css**.

```
body{
  text-align: center;
  font-family: verdana, sans-serif;
}
h1{
  font-size: 2em;
  font-weight: bold;
  color: #777777;
  margin: 40px 0px 70px;
  text-align: center;
}
.gwt-PasswordTextBox {
  color: green;
}
.gwt-PasswordTextBox-readonly {
  background-color: yellow;
}
```

Following is the content of the modified HTML host file **war/HelloWorld.html**.

```
<html>
<head>
<title>Hello World</title>
  <link rel="stylesheet" href="HelloWorld.css"/>
  <script language="javascript" src="helloworld/helloworld.nocache.js">
  </script>
</head>
<body>

<h1>PasswordTextBox Widget Demonstration</h1>
<div ></div>

</body>
</html>
```

Let us have following content of Java file **src/com.tutorialspoint/HelloWorld.java** which will demonstrate use of PasswordTextBox widget.

```
package com.tutorialspoint.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.RootPanel;
import com.google.gwt.user.client.ui.PasswordTextBox;
import com.google.gwt.user.client.ui.VerticalPanel;

public class HelloWorld implements EntryPoint {
    public void onModuleLoad() {
        //create password textboxes
        PasswordTextBox passwordTextBox1 = new PasswordTextBox();
        PasswordTextBox passwordTextBox2 = new PasswordTextBox();

        //add text to text box
        passwordTextBox2.setText("hell@W@rld");

        //set textbox as readonly
        passwordTextBox2.setReadOnly(true);

        // Add text boxes to the root panel.
        VerticalPanel panel = new VerticalPanel();
        panel.setSpacing(10);
        panel.add(passwordTextBox1);
        panel.add(passwordTextBox2);

        RootPanel.get("gwtContainer").add(panel);
    }
}
```

Once you are ready with all the changes done, let us compile and run the application in development mode as we did in [GWT - Create Application](#) chapter. If everything is fine with your application, this will produce following result:



Loading [MathJax]/jax/output/HTML-CSS/jax.js