

GWT - SIMPLEPANEL WIDGET

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

Copyright © tutorialspoint.com

Introduction

The **SimplePanel** widget represents a base class for panels that contain only one widget.

Class declaration

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

```
public class SimplePanel
    extends Panel
    implements HasOneWidget
```

Class constructors

S.N.	Constructor & Description
1	SimplePanel Creates an empty panel that uses a DIV for its contents.
2	protected SimplePanelElementelem Creates an empty panel that uses the specified browser element for its contents.

Class methods

S.N.	Function name & Description
1	void addWidgetw Adds a widget to this panel.
2	protected Element getContainerElement Override this method to specify that an element other than the root element be the container for the panel's child widget.
3	Widget getWidget Gets the panel's child widget.
4	java.util.Iterator<Widget> iterator Gets an iterator for the contained widgets.
5	boolean removeWidgetw Removes a child widget.
6	void setWidgetIsWidgetw Set the only widget of the receiver, replacing the previous widget if there was one.
7	void setWidgetWidgetw Sets this panel's widget.

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.Panel
- java.lang.Object

SimplePanel Widget Example

This example will take you through simple steps to show usage of a SimplePanel 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;
}
```

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>SimplePanel 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 SimplePanel widget.

```
package com.tutorialspoint.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.user.client.ui.DecoratorPanel;
import com.google.gwt.user.client.ui.Label;
import com.google.gwt.user.client.ui.RootPanel;
import com.google.gwt.user.client.ui.SimplePanel;

public class HelloWorld implements EntryPoint {

    public void onModuleLoad() {
        // Create a Simple Panel
        SimplePanel simplePanel = new SimplePanel();
        Label label = new Label("A Simple Label.");
        //add label to simple panel
        simplePanel.add(label);
        //set height and width of simple panel
        simplePanel.setHeight("200");
        simplePanel.setWidth("200");

        DecoratorPanel decoratorPanel = new DecoratorPanel();
        decoratorPanel.add(simplePanel);

        // Add the widgets to the root panel.
        RootPanel.get().add(decoratorPanel);
    }
}
```

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/fonts/TeX/fontdata.js