GWT - HTML WIDGET

Introduction

The **HTML** widget can cantain arbitrary HTML it can be interpreted as HTML. This widget uses a element, causing it to be displayed with block layout.

Class declaration

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

```
public class HTML
   extends Label
      implements HasHTML
```

CSS style rules

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

```
.gwt-HTML { }
```

Class constructors

S.N.	Constructor & Description
1	**HTML** Creates an empty HTML.
2	**protected HTML*Elementelement*** This constructor may be used by subclasses to explicitly use an existing element.
3	**HTML*java. lang. Stringhtml*** Creates a HTML with the specified html contents.
4	**HTML*java. lang. Stringhtml, booleanwordWrap*** Creates an HTML widget with the specified contents, optionally treating it as HTML, and optionally disabling word wrapping.

Class methods

1

java.lang.String getHTML

Gets this object's contents as HTML.

2

void setHTML*java. lang. String**html*

Sets this object's contents via HTML.

3

static HTML wrap*Element**element*

Creates an HTML widget that wraps an existing or element.

Methods inherited

This class inherits methods from the following classes:

- com.google.gwt.user.client.ui.Label
- com.google.gwt.user.client.ui.UIObject
- com.google.gwt.user.client.ui.Widget
- com.google.gwt.user.client.ui.HasText
- java.lang.Object

Html Widget Example

This example will take you through simple steps to show usage of a HTML 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 *HelloWorld* under a package *com.tutorialspoint* as explained in the *GWT - Create Application* chapter.
2	Modify *HelloWorld.gwt.xml*, *HelloWorld.css*, *HelloWorld.html* and *HelloWorld.java* 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-Green-Border{
   border:1px solid green;
}
.gwt-Blue-Border{
   border:1px solid blue;
}
```

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>HTML 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 HTML widget.

```
package com.tutorialspoint.client;

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

public class HelloWorld implements EntryPoint {
   public void onModuleLoad() {
      // create two HTML widgets
      HTML html1 =
      new HTML("This is first GWT HTML widget using <b> tag of html.");
      HTML html2 =
      new HTML("This is second GWT HTML widget using <i> tag of html.");

      // use UIObject methods to set HTML widget properties.
      html1.addStyleName("gwt-Green-Border");
      html2.addStyleName("gwt-Blue-Border");

      // add widgets to the root panel.
      VerticalPanel panel = new VerticalPanel();
      panel.setSpacing(10);
      panel.add(html1);
      panel.add(html2);

      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: