

GWT - TEXTBOX WIDGET

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

Copyright © tutorialspoint.com

Introduction

The **TextBox** widget represents a standard single-line text box.

Class declaration

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

```
public class TextBox
    extends TextBoxBase
    implements HasDirection
```

CSS style rules

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

```
.gwt-TextBox {}
.gwt-TextBox-readonly {}
```

Class constructors

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

Class methods

S.N.	Function name & Description
1	HasDirection.Direction getDirection Gets the directionality of the widget.
2	int getMaxLength Gets the maximum allowable length of the text box.
3	int getVisibleLength Gets the number of visible characters in the text box.

- 4 **void setDirection***HasDirection. Direction**direction*
Sets the directionality for a widget.
- 5 **void setMaxLength***length*
Sets the maximum allowable length of the text box.
- 6 **void setVisibleLength***length*
Sets the number of visible characters in the text box.
- 7 **static TextBox wrap***Element**element*
Creates a TextBox widget that wraps an existing `<input type='text'>` 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`
- `java.lang.Object`

TextBox Widget Example

This example will take you through simple steps to show usage of a TextBox 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-TextBox {
    color: green;
}
.gwt-TextBox-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>TextBox 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 TextBox 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.TextBox;
import com.google.gwt.user.client.ui.VerticalPanel;

public class HelloWorld implements EntryPoint {
    public void onModuleLoad() {
        //create textboxes
        TextBox textBox1 = new TextBox();
        TextBox textBox2 = new TextBox();

        //add text to text box
        textBox2.setText("Hello World!");

        //set textbox as readonly
        textBox2.setReadOnly(true);
    }
}

```

```
// Add text boxes to the root panel.  
VerticalPanel panel = new VerticalPanel();  
panel.setSpacing(10);  
panel.add(textBox1);  
panel.add(textBox2);  
  
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:

