

# GWT - PUSHBUTTON WIDGET

[http://www.tutorialspoint.com/gwt/gwt\\_pushbutton\\_widget.htm](http://www.tutorialspoint.com/gwt/gwt_pushbutton_widget.htm)

Copyright © tutorialspoint.com

## Introduction

The **PushButton** widget represents a standard push button with custom styling..

## Class declaration

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

```
public class PushButton
    extends CustomButton
```

## CSS style rules

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

```
.gwt-PushButton-up {}
.gwt-PushButton-down {}
.gwt-PushButton-up-hovering {}
.gwt-PushButton-down-hovering {}
.gwt-PushButton-up-disabled {}
.gwt-PushButton-down-disabled {}
```

## Class constructors

| S.N. | Constructor & Description |
|------|---------------------------|
|------|---------------------------|

- |   |                                                                                                                     |
|---|---------------------------------------------------------------------------------------------------------------------|
| 1 | <b>PushButton</b><br>Constructor for PushButton.                                                                    |
| 2 | <b>PushButtonImageupImage</b><br>Creates a PushButton with up state image.                                          |
| 3 | <b>PushButtonImageupImage, ClickListenerlistener</b><br>Creates a PushButton with up state image and clickListener. |
| 4 | <b>PushButtonImageupImage, ImagedownImage</b><br>Creates a PushButton with up state image.                          |
| 5 | <b>PushButtonImageupImage, ImagedownImage, ClickListenerlistener</b><br>Creates a PushButton with up state image.   |
| 6 | <b>PushButtonjava. lang. StringupText</b>                                                                           |

Creates a PushButton with up state text.

7

**PushButton***java. lang. StringupText, ClickListenerlistener*

Creates a PushButton with up state text and clicklistener.

8

**PushButton***java. lang. StringupText, java. lang. StringdownText*

Creates a PushButton with up state and down state text.

8

**PushButton***java. lang. StringupText, java. lang. StringdownText, ClickListenerlistener*

Creates a PushButton with up state, down state text and click listener.

## Class methods

| S.N. | Function name & Description                                                                                                                                                         |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1    | <b>protected void onClick</b><br>Called when the user finishes clicking on this button.                                                                                             |
| 2    | <b>protected void onClickCancel</b><br>Called when the user aborts a click in progress; for example, by dragging the mouse outside of the button before releasing the mouse button. |
| 3    | <b>protected void onClickStart</b><br>Called when the user begins to click on this button.                                                                                          |

## 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.CustomWidget
- java.lang.Object

## PushButton Widget Example

This example will take you through simple steps to show usage of a PushButton 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-PushButton{
    color:red;
}
.gwt-PushButton-up {
    color:green;
}
.gwt-PushButton-down {
    color:blue;
}
.gwt-PushButton-up-hovering {
    color:pink;
}
.gwt-PushButton-down-hovering {
    color:aqua;
}
.gwt-PushButton-up-disabled {
    color:lime;
}
.gwt-PushButton-down-disabled {
    color:maroon;
}
```

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>PushButton 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 PushButton widget.

```

package com.tutorialspoint.client;

import com.google.gwt.core.client.EntryPoint;
import com.google.gwt.event.dom.client.ClickEvent;
import com.google.gwt.event.dom.client.ClickHandler;
import com.google.gwt.user.client.Window;
import com.google.gwt.user.client.ui.PushButton;
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 a push button
        PushButton pushButton = new PushButton("Click Me!");

        //create a push button
        PushButton pushButton1 = new PushButton("Click Me!");

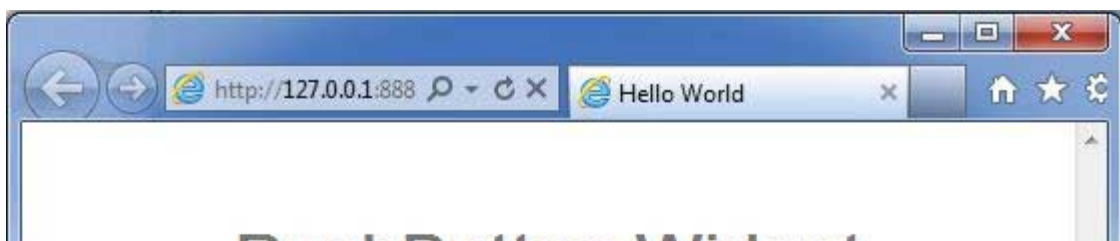
        //disable a push button
        pushButton1.setEnabled(false);

        //add a clickListener to the push button
        pushButton.addClickHandler(new ClickHandler() {
            @Override
            public void onClick(ClickEvent event) {
                Window.alert("Hello World!");
            }
        });

        // Add push buttons to the root panel.
        VerticalPanel panel = new VerticalPanel();
        panel.setSpacing(10);
        panel.add(pushButton);
        panel.add(pushButton1);
        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:



# PushButton Widget Demonstration



When you click **Click Me** button, it will show an alert message **Hello World!**

You can see color of button text and its state will change with your interaction.

- Hover over the button, color will be pink.
- Press the button, color will be aqua.
- Release the button, color will be green.

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