

GWT - TABPANEL WIDGET

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

Copyright © tutorialspoint.com

Introduction

The **TabPanel** widget represents panel that represents a tabbed set of pages, each of which contains another widget. Its child widgets are shown as the user selects the various tabs associated with them. The tabs can contain arbitrary HTML.

Class declaration

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

```
@Deprecated
public class TabPanel
    extends Composite
        implements TabListener, SourcesTabEvents,
            HasWidgets, HasAnimation, IndexedPanel.ForIsWidget,
                HasBeforeSelectionHandlers<java.lang.Integer>,
                    HasSelectionHandlers<java.lang.Integer>
```

Class constructors

S.N. Constructor & Description

- | | |
|---|--|
| 1 | TabPanel
Deprecated. Creates an empty tab panel. |
|---|--|

Class methods

S.N. Function name & Description

- | | |
|---|---|
| 1 | void addIsWidgetw, IsWidgettabWidget
Deprecated. Convenience overload to allow IsWidget to be used directly. |
| 2 | void addIsWidgetw, java. lang. StringtabText
Deprecated. Convenience overload to allow IsWidget to be used directly. |
| 3 | void addIsWidgetw, java. lang. StringtabText, booleanasHTML
Deprecated. Convenience overload to allow IsWidget to be used directly. |
| 4 | void addWidgetw
Deprecated. Adds a child widget. |
| 5 | void addWidgetw, java. lang. StringtabText |

Deprecated. Adds a widget to the tab panel.

6

void addWidget*w, java. lang. String tabText, boolean asHTML*

Deprecated. Adds a widget to the tab panel.

7

void addWidget*w, Widget tabWidget*

Deprecated. Adds a widget to the tab panel.

8

Handler Registration add Before Selection Handler
BeforeSelectionHandler < java. lang. Integer > handler

Deprecated. Adds a BeforeSelectionEvent handler.

9

Handler Registration add Selection Handler *SelectionHandler < java. lang. Integer > handler*

Deprecated. Adds a SelectionEvent handler.

10

void add Tab Listener*TabListener listener*

Deprecated. Use add Before Selection Handler
com. google. gwt. event. logical. shared. BeforeSelectionHandler and add Selection Handler
com. google. gwt. event. logical. shared. SelectionHandler instead

11

void clear

Deprecated. Removes all child widgets.

12

protected SimplePanel createTabTextWrapper

Deprecated. Create a SimplePanel that will wrap the contents in a tab.

13

DeckPanel getDeckPanel

Deprecated. Gets the deck panel within this tab panel.

14

TabBar getTabBar

Deprecated. Gets the tab bar within this tab panel.

15

Widget getWidget*int index*

Deprecated. Gets the child widget at the specified index.

16

int getWidgetCount

Deprecated. Gets the number of child widgets in this panel.

17

int getWidgetIndex*IsWidgetchild*

Deprecated. Convenience overload to allow IsWidget to be used directly.

18

int getWidgetIndex*Widgetwidget*

Deprecated. Gets the index of the specified child widget.

19

void insert*IsWidgetwidget, IsWidgettabWidget, intbeforeIndex*

Deprecated. Convenience overload to allow IsWidget to be used directly.

20

void insert*IsWidgetwidget, java. lang. StringtabText, booleanasHTML, intbeforeIndex*

Deprecated. Convenience overload to allow IsWidget to be used directly.

21

void insert*IsWidgetwidget, java. lang. StringtabText, intbeforeIndex*

Deprecated. Convenience overload to allow IsWidget to be used directly.

22

void insert*Widgetwidget, java. lang. StringtabText, booleanasHTML, intbeforeIndex*

Deprecated. Inserts a widget into the tab panel.

23

void insert*Widgetwidget, java. lang. StringtabText, intbeforeIndex*

Deprecated. Inserts a widget into the tab panel.

24

void insert*Widgetwidget, WidgettabWidget, intbeforeIndex*

Deprecated. Inserts a widget into the tab panel.

25

boolean isAnimationEnabled

Deprecated. Returns true if animations are enabled, false if not.

26

java.util.Iterator<Widget> iterator

Deprecated. Gets an iterator for the contained widgets.

27

boolean on Before Tab Selected*SourcesTabEventssender, inttabIndex*

Deprecated. Use Before Selection Handler.on Before Selection
com. google. gwt. event. logical. shared. BeforeSelectionEvent instead

28

protected void onEnsureDebugId*java. lang. StringbaseID*

Deprecated. Affected Elements: -bar = The tab bar. -bar-tab# = The element containing the content of the tab itself. -bar-tab-wrapper# = The cell containing the tab at the index. -bottom = The panel beneath the tab bar.

- 29
void onTabSelected*SourcesTabEventssender, inttabIndex*
Deprecated. Use SelectionHandler.onSelection
com. google. gwt. event. logical. shared. SelectionEvent instead
- 30
boolean remove*intindex*
Deprecated. Removes the widget at the specified index.
- 31
boolean remove*Widgetwidget*
Deprecated. Removes the given widget, and its associated tab.
- 32
void removeTabListener*TabListenerlistener*
Deprecated. Use the HandlerRegistration.removeHandler method on the object returned by and add*Handler method instead
- 33
void selectTab*intindex*
Deprecated. Programmatically selects the specified tab and fires events.
- 34
void selectTab*intindex, booleanfireEvents*
Deprecated. Programmatically selects the specified tab.
- 35
void setAnimationEnabled*booleanenable*
Deprecated. Enable or disable animations.

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.Composite
- java.lang.Object

TabPanel Widget Example

This example will take you through simple steps to show usage of a TabPanel 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>TabPanel 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 TabPanel 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.HTMLPanel;
import com.google.gwt.user.client.ui.Label;
import com.google.gwt.user.client.ui.RootPanel;
import com.google.gwt.user.client.ui.TabPanel;

public class HelloWorld implements EntryPoint {

    public void onModuleLoad() {
        //Create an empty tab panel
        TabPanel tabPanel = new TabPanel();

        //create contents for tabs of tabpanel
        Label label1 = new Label("This is contents of TAB 1");
        label1.setHeight("200");
        Label label2 = new Label("This is contents of TAB 2");
        label2.setHeight("200");
        Label label3 = new Label("This is contents of TAB 3");
        label3.setHeight("200");

        //create titles for tabs
        String tab1Title = "TAB 1";
        String tab2Title = "TAB 2";
        String tab3Title = "TAB 3";

        //create tabs
        tabPanel.add(label1, tab1Title);
        tabPanel.add(label2, tab2Title);
        tabPanel.add(label3, tab3Title);

        //select first tab
        tabPanel.selectTab(0);

        //set width if tabpanel
        tabPanel.setWidth("400");

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

```

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