

Introduction

The **FlexTable** widget represents a flexible table that creates cells on demand. It can be jagged *that is, each row can contain a different number of cells* and individual cells can be set to span multiple rows or columns.

Class declaration

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

```
public class FlexTable  
    extends HTMLTable
```

Class constructors

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

- | | |
|---|---|
| 1 | FlexTable
Constructor for empty Flex Table. |
|---|---|

Class methods

S.N.	Function name & Description
------	-----------------------------

- | | |
|---|---|
| 1 | void addCell <i>int row</i>
Appends a cell to the specified row. |
| 2 | int getCellCount <i>int row</i>
Gets the number of cells on a given row. |
| 3 | FlexTable.FlexCellFormatter getFlexCellFormatter
Explicitly gets the FlexTable.FlexCellFormatter. |
| 4 | int getRowCount
Gets the number of rows. |
| 5 | void insertCell <i>int beforeRow, int beforeColumn</i>
Inserts a cell into the FlexTable. |

int insertRow*intbeforeRow*

Inserts a row into the FlexTable.

7

protected void prepareCell*introw, intcolumn*

Ensure that the cell exists.

8

protected void prepareRow*introw*

Ensure that the row exists.

9

void removeAllRows

Remove all rows in this table.

10

void removeCell*introw, intcol*

Removes the specified cell from the table.

11

void removeCells*introw, intcolumn, intnum*

Removes a number of cells from a row in the table.

12

void removeRow*introw*

Removes the specified row from the table.

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
- com.google.gwt.user.client.ui.HTMLTable
- java.lang.Object

FlexTable Widget Example

This example will take you through simple steps to show usage of a FlexTable 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;
}
.flexTable td {
    border: 1px solid #BBBBBB;
    padding: 3px;
}
.flexTable-buttonPanel td {
    border: 0px;
}
.fixedWidthButton {
    width: 150px;
}
```

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>FlexTable 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 FlexTable 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.ui.Button;
import com.google.gwt.user.client.ui.DecoratorPanel;
import com.google.gwt.user.client.ui.FlexTable;
import com.google.gwt.user.client.ui.FlexTable.FlexCellFormatter;
import com.google.gwt.user.client.ui.HasHorizontalAlignment;
import com.google.gwt.user.client.ui.HasVerticalAlignment;
import com.google.gwt.user.client.ui.Image;
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 Flex Table
        final FlexTable flexTable = new FlexTable();
        FlexCellFormatter cellFormatter = flexTable.getFlexCellFormatter();
        flexTable.addStyleName("flexTable");
        flexTable.setWidth("32em");
        flexTable.setCellSpacing(5);
        flexTable.setCellPadding(3);

        // Add some text
        cellFormatter.setHorizontalAlignment(
            0, 1, HasHorizontalAlignment.ALIGN_LEFT);
        flexTable.setHTML(0, 0, "This is a FlexTable that supports"
            + " <b>colspans</b> and <b>rowspans</b>."
            + " You can use it to format your page"
            + " or as a special purpose table.");
        cellFormatter.setColSpan(0, 0, 2);

        // Add a button that will add more rows to the table
        Button addRowButton = new Button("Add a Row");
        addRowButton.addClickHandler(new ClickHandler() {
            @Override
            public void onClick(ClickEvent event) {
                addRow(flexTable);
            }
        });

        addRowButton.addStyleName("fixedWidthButton");

        // Add a button that will add more rows to the table
        Button removeRowButton = new Button("Remove a Row");
        removeRowButton.addClickHandler(new ClickHandler() {
            @Override
            public void onClick(ClickEvent event) {
                removeRow(flexTable);
            }
        });

        removeRowButton.addStyleName("fixedWidthButton");

        VerticalPanel buttonPanel = new VerticalPanel();
        buttonPanel.setStyleName("flexTable-buttonPanel");
        buttonPanel.add(addRowButton);
        buttonPanel.add(removeRowButton);
        flexTable.setWidget(0, 1, buttonPanel);
        cellFormatter.setVerticalAlignment(0, 1,
            HasVerticalAlignment.ALIGN_TOP);
    }
}
```

```

// Add two rows to start
addRow(flexTable);
addRow(flexTable);

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

/**
 * Add a row to the flex table.
 */
private void addRow(FlexTable flexTable) {
    int numRows = flexTable.getRowCount();
    flexTable.setWidget(numRows, 0,
        new Image("http://www.tutorialspoint.com/images/gwt-mini.png"));
    flexTable.setWidget(numRows, 1,
        new Image("http://www.tutorialspoint.com/images/gwt-mini.png"));
    flexTable.getFlexCellFormatter().setRowSpan(0, 1, numRows + 1);
}

/**
 * Remove a row from the flex table.
 */
private void removeRow(FlexTable flexTable) {
    int numRows = flexTable.getRowCount();
    if (numRows > 1) {
        flexTable.removeRow(numRows - 1);
        flexTable.getFlexCellFormatter().setRowSpan(0, 1, numRows - 1);
    }
}
}

```

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:

