

Introduction

The **Grid** widget represents a rectangular grid that can contain text, html, or a child Widget within its cells. It must be resized explicitly to the desired number of rows and columns.

Class declaration

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

```
public class Grid
    extends HTMLTable
```

Class constructors

S.N.	Constructor & Description
1	Grid Constructor for Grid.
2	Grid <i>introws, intcolumns</i> Constructor for Grid with requested size.

Class methods

S.N.	Function name & Description
1	boolean clearCell <i>introw, intcolumn</i> Replaces the contents of the specified cell with a single space.
2	protected Element createCell Creates a new, empty cell.
3	int getCellCount <i>introw</i> Return number of columns.
4	int getColumnCount Gets the number of columns in this grid.
5	int getRowCount

Return number of rows.

6

int insertRow*intbeforeRow*

Inserts a new row into the table.

7

protected void prepareCell*introw, intcolumn*

Checks that a cell is a valid cell in the table.

8

protected void prepareColumn*intcolumn*

Checks that the column index is valid.

9

protected void prepareRow*introw*

Checks that the row index is valid.

10

void removeRow*introw*

Removes the specified row from the table.

11

void resize*introws, intcolumns*

Resizes the grid.

12

void resizeColumns*intcolumns*

Resizes the grid to the specified number of columns.

13

void resizeRows*introws*

Resizes the grid to the specified number of rows.

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

Grid Widget Example

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

public class HelloWorld implements EntryPoint {

    public void onModuleLoad() {
        // Create a grid
        Grid grid = new Grid(2, 2);

        // Add images to the grid
        int numRows = grid.getRowCount();
        int numColumns = grid.getColumnCount();
        for (int row = 0; row < numRows; row++) {
            for (int col = 0; col < numColumns; col++) {
                grid.setWidget(row, col,
                    new Image("http://www.tutorialspoint.com/images/gwt-mini.png"));
            }
        }

        DecoratorPanel decoratorPanel = new DecoratorPanel();
        decoratorPanel.add(grid);
        // Add the widgets to the root panel.
        RootPanel.get().add(decoratorPanel);
    }
}
```

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:

