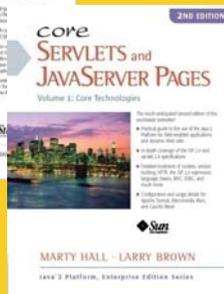
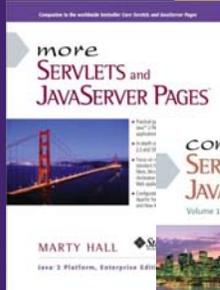




Laying out Windows with Panels: Part I – Simple Panels (GWT 2.5 Version)

Originals of Slides and Source Code for Examples:
<http://courses.coreservlets.com/Course-Materials/gwt.html>

Customized Java EE Training: <http://courses.coreservlets.com/>
GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live Ajax & GWT training, see training courses at <http://courses.coreservlets.com/>.



Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
 - JSF 2, PrimeFaces, servlets/JSP, Ajax, jQuery, Android development, Java 7 or 8 programming, custom mix of topics
 - Courses available in any state or country. Maryland/DC area companies can also choose afternoon/evening courses.
- Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, GWT, Hadoop, HTML5, RESTful Web Services

Contact hall@coreservlets.com for details



Topics in This Section

- **Strategy behind Panel usage**
 - Similar to use of LayoutManagers in desktop Java
- **Basic Panel types**
 - VerticalPanel
 - HorizontalPanel
 - FlowPanel
 - FlowPanel as vertical panel
 - FlowPanel as horizontal panel
 - Grid
 - FlexTable

5

© 2013 Marty Hall & Yaakov Chaikin



Overview

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Layout Strategies

- **HTML-based layout**
 - Write HTML by hand, designate places for individual controls
 - HTML body contains a lot of content
 - Best when GWT is used for
 - A traditional Web app with some pieces of Ajax-enabled content
 - Complex page where HTML layout does not change
 - You need the HTML content indexed by a search engine
- **Java-based layout**
 - Attaches main layout panel to HTML <body> element
 - Java uses Panels to build up overall layout
 - Similar to way LayoutManagers are used in desktop Java
 - Best when GWT is used to
 - Mimic a desktop application
 - Create an application where HTML layout changes on fly
- **Declarative layout – see UiBinder slides**

7

HTML-Based Layout

- **HTML**

```
<body>
Regular HTML <div id="id-1"></div>
Regular HTML <div id="id-2"></div>
Regular HTML <div id="id-3"></div>
Regular HTML
</body>
```
- **Java**

```
public void onModuleLoad() {
    SomeWidget w1 = ...;
    RootPanel.get("id-1").add(w1);
    SomeWidget w2 = ...;
    RootPanel.get("id-2").add(w2);
    SomeWidget w3 = ...;
    RootPanel.get("id-3").add(w3);
}
```

8

Java-Based Layout

- **HTML**

```
<body>
<!-- Possibly history iframe, but few or no HTML elements -->
</body>
```

- **Java**

```
public void onModuleLoad() {
    SomePanel mainPanel = ...;
    mainPanel.setVariousProperties();
    SomeOtherPanel subPanel = ...;
    SomeWidget w1 = ...;
    subPanel.add(w1);
    ...
    mainPanel.add(subPanel);
    ...
    RootLayoutPanel.get().add(mainPanel);
}
```

9

Typical GWT Panel Design

- **Multi-panel types**

- Either the top-level panel or the main region of the top-level panel usually contains a multi-panel type
 - E.g., DeckLayoutPanel, TabLayoutPanel, StackLayoutPanel

- **Sample designs**

- Top-level window is a TabLayoutPanel or StackLayoutPanel
 - Every time the user clicks on a tab or stack (accordion) label, you switch “screens”
- Top-level window is a DeckLayoutPanel
 - Custom controls to navigate from screen to screen
- Top-level window is a DockLayoutPanel
 - And center is a TabLayoutPanel, StackLayoutPanel, or DeckLayoutPanel



10

Tutorial Breakdown

- **This section: basic panels**

- Purpose: organize a group of GUI controls
 - All panels can contain other panels, but the first Panel tutorial covers the simplest panel types that are often used to represent a small group of widgets (e.g., a row or column of buttons or an area of text)
 - Examples: HorizontalPanel, VerticalPanel, Grid, FlexTable, FlowPanel, ScrollPanel

- **Next section: composite panels**

- Purpose: organize other panels
 - The second Panel tutorial covers the panels that are used primarily to group other panels, and are typically the top-level panels in an application.
 - Examples: TabLayoutPanel, StackLayoutPanel, DockLayoutPanel, DeckLayoutPanel

11

Methods Supported by Most Panels

- **add(Widget w)**

- Adds a Widget to the Panel. Where it gets inserted depends on the Panel type.
 - Most Panels support “insert” to put Widget in specific place
 - All Panels support “remove” (one Widget) and “clear” (all)
- The Widget that you add can itself be a Panel
- Many Panels have specialized versions of add with extra args

- **setSize, setWidth, setHeight**

- These take CSS-style size descriptors (String-based)

- **addStyleName**

- For CSS styling. Most widgets also have a predefined CSS style name (e.g., gwt-Button).

- **getChildren, getWidget, getWidgetCount**

- Look up information about widgets already inside the Panel

12

Quirks vs. Standards Mode in Browsers

- **Idea**
 - Some browsers render differently when following CSS and HTML standards than they did traditionally
 - Applies especially to Internet Explorer
 - The traditional behavior is called “Quirks Mode”
 - Table-based layouts assume traditional behavior
 - Has little effect on the simple panels used in this section, but will affect complex panels used in next section.
- **Bottom line for simple panels**
 - Most of the time, you can get away with keeping your page in “standards mode”, and, if needed, set explicit sizes on child widgets to resolve any issues that might come up
 - Since recent versions of GWT rely on “standards mode”, you should always use that mode
 - It might require you to brush up on CSS
 - To go to quirks mode, omit the `<!doctype>` declaration

13

© 2013 Marty Hall & Yaakov Chaikin



Examples: Overview

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Approach for Examples

- **Hybrid between HTML-based and Java-based approaches**
 - One div per sample panel type, for sake of illustration
 - This is to simplify examples that show many different panel types at once
- **Full Java-based approach illustrated in second tutorial section**
 - And also in tutorial section on History handling



15

Top-Level Example Code: Java

```
public class GwtPanels1 implements EntryPoint {  
    public void onModuleLoad() {  
        addHorizontalPanel();  
        addVerticalPanel();  
        addGrid();  
        ...  
    }  
    ...  
}
```

16

Top-Level Example Code: HTML

```
<!doctype html>
<html...><head><title>...</title>...</head>
<body>
<p/>
<fieldset>
<legend>Simple Horizontal Panels</legend>
<div id="horizontal-panel"></div>
</fieldset>
<p/>
<fieldset>
<legend>Simple Vertical Panels</legend>
<div id="vertical-panel"></div>
</fieldset>
...
</body>
```

17

© 2013 Marty Hall & Yaakov Chaikin



HorizontalPanel

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **Purpose**
 - To put widgets side-by-side
 - Similar to Swing's BorderLayout (with a horizontal layout)
- **Main methods**
 - setSpacing(int pixels)
 - Sets inter-widget spacing in pixels
 - add(Widget w)
 - Adds a widget, in left-to-right order.
 - Use "insert" if you want to put widget at specific index
 - setHorizontalAlignment, setVerticalAlignment
 - Sets default alignment

19

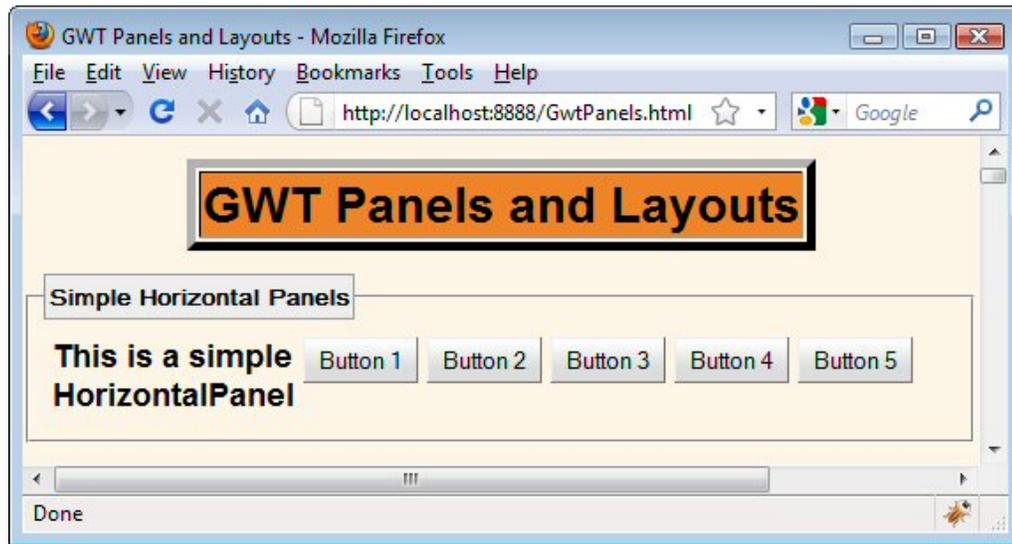
Example Code

```
private void addHorizontalPanel() {
    String text = "<b>This is a simple<br/>HorizontalPanel</b>";
    HorizontalPanel hPanel = makeHorizontalPanel(text, 5);
    RootPanel.get("horizontal-panel").add(hPanel);
}

private HorizontalPanel makeHorizontalPanel(String text,
                                           int numButtons) {
    HorizontalPanel hPanel = new HorizontalPanel();
    hPanel.setSpacing(5);
    hPanel.add(new HTML(text));
    for (int i=1; i <= numButtons; i++) {
        hPanel.add(new Button("Button " + i));
    }
    return(hPanel);
}
```

20

Example Result (Production Mode)



21

Underlying HTML (Generated by GWT)

```
<div id="horizontal-panel">
<table cellpadding="5" cellspacing="0">
<tbody><tr>
  <td align="left" style="vertical-align: top;">
    <div class="gwt-HTML">
      <b>This is a simple<br/>HorizontalPanel</b>
    </div>
  </td>
  <td align="left" style="vertical-align: top;">...</td>
  <td align="left" style="vertical-align: top;">...</td>
  <td align="left" style="vertical-align: top;">...</td>
  <td align="left" style="vertical-align: top;">...</td>
  <td align="left" style="vertical-align: top;">...</td>
</tr></tbody></table>
</div>
```

22



VerticalPanel

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **Purpose**
 - To put widgets on top of each other.
 - Similar to Swing's BorderLayout (with a vertical layout)
- **Main methods**
 - setSpacing(int pixels)
 - Sets inter-widget spacing in pixels
 - add(Widget w)
 - Adds a widget, in top-to-bottom order.
 - Use "insert" if you want to put widget at specific index
 - setHorizontalAlignment, setVerticalAlignment
 - Sets default alignment

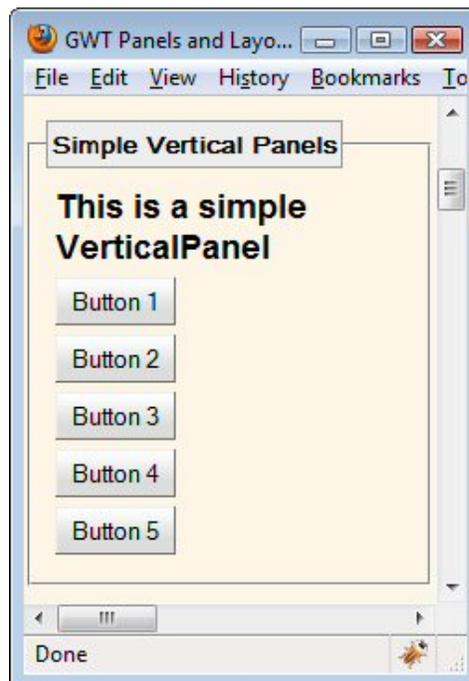
Example Code

```
private void addVerticalPanel() {
    String text =
        "<b>This is a simple<br/>VerticalPanel</b>";
    VerticalPanel vPanel = makeVerticalPanel(text, 5);
    RootPanel.get("vertical-panel").add(vPanel);
}

private VerticalPanel makeVerticalPanel(String text,
                                       int numButtons) {
    VerticalPanel vPanel = new VerticalPanel();
    vPanel.setSpacing(5);
    vPanel.add(new HTML(text));
    for (int i=1; i <= numButtons; i++) {
        vPanel.add(new Button("Button " + i));
    }
    return(vPanel);
}
```

25

Example Result



26

Underlying HTML (Generated by GWT)

```
<div id="vertical-panel">
<table cellpadding="5" cellspacing="0"><tbody>
<tr>
  <td align="left" style="vertical-align: top;">
    <div class="gwt-HTML">
      <b>This is a simple<br/>VerticalPanel</b>
    </div>
  </td>
</tr>
<tr><td align="left" style="vertical-align: top;">...</td></tr>
<tr><td align="left" style="vertical-align: top;">...</td></tr>
<tr><td align="left" style="vertical-align: top;">...</td></tr>
<tr><td align="left" style="vertical-align: top;">...</td></tr>
</tbody></table>
</div>
```

27

© 2013 Marty Hall & Yaakov Chaikin



FlowPanel

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **Purpose**

- To layout widgets according to the natural HTML flow
- Underlying implementation is a simple <div>
- Can be used to simulate vertical and horizontal panels with some CSS
 - Useful when you want guaranteed behavior in standards mode

- **Main methods**

- add(Widget w)
 - Adds a widget, attaching it to the <div> without modification – regular HTML layout
- insert(Widget w, int beforeIndex)
 - Places widget at specific location vis-à-vis the other widgets present in the panel

29

Example Code

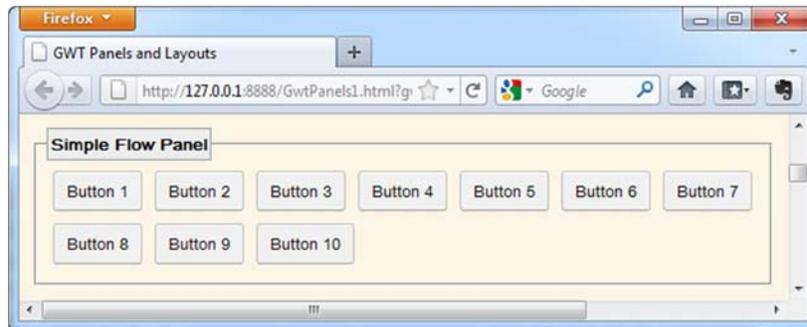
```
private void addSimpleFlowPanel() {
    FlowPanel fPanel = new FlowPanel();
    for (int i = 1; i <= 10; i++) {
        Button button = new Button("Button " + i);
        button.addStyleName("my-button");
        fPanel.add(button);
    }

    RootPanel.get("flow-panel").add(fPanel);
}
```

```
.my-button {
    margin: 5px !important;
}
```

30

Example Result



31

Underlying HTML

```
<div id="flow-panel">  
  <div>  
    <button class="gwt-Button my-button" type="button">  
      Button 1</button>  
    <button class="gwt-Button my-button" type="button">  
      Button 2</button>  
    ...  
    <button class="gwt-Button my-button" type="button">  
      Button 8</button>  
    <button class="gwt-Button my-button" type="button">  
      Button 9</button>  
    <button class="gwt-Button my-button" type="button">  
      Button 10</button>  
  </div>  
</div>
```

32



FlowPanel as a Vertical Panel

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **Purpose**
 - To layout widgets one top of another (vertically)
 - Preferred by some to **VerticalPanel** because it uses **divs** and **CSS** instead of tables
- **Main idea**
 - Before adding to FlowPanel, wrap each widget in some *block-level* element
 - Or its GWT equivalent
 - Add the wrapper element instead of the widget itself
 - Block-level elements naturally render one on top of another, so there is nothing more to do!
- **Wrapper can be SimplePanel**
 - Renders as simple <div>
 - Holds only one child widget at a time

Example Code

```
private void addVerticalFlowPanel() {
    FlowPanel fPanel = new FlowPanel();
    for (int i = 1; i <= 10; i++) {
        SimplePanel wrapper = new SimplePanel();
        Button button = new Button("Button " + i);
        button.addStyleName("my-button");
        wrapper.add(button);
        fPanel.add(wrapper);
    }

    RootPanel.get("flow-panel-vertical").add(fPanel);
}
```

35

Example Result



36

Underlying HTML

```
<div id="flow-panel-vertical">  
  <div>  
    <div>  
      <button class="gwt-Button my-button" type="button">  
        Button 1</button>  
    </div>  
    <div>  
      <button class="gwt-Button my-button" type="button">  
        Button 2</button>  
    </div>  
    ...  
    <div>  
      <button class="gwt-Button my-button" type="button">  
        Button 10</button>  
    </div>  
  </div>  
</div>
```

FlowPanel

Wrapper SimplePanel

37

© 2013 Marty Hall & Yaakov Chaikin



FlowPanel as a Horizontal Panel

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **Purpose**
 - To layout widgets horizontally
 - Preferred by some to `HorizontalPanel` because it uses `divs` and `CSS` instead of tables
- **Main idea**
 - Set `float:left` on each element added to `FlowPanel`
 - Set the width of `FlowPanel`, so elements don't wrap when window resizes
- **Example**
 - `<button>` is an inline element, so to demonstrate this behavior, continue to wrap each button in `SimplePanel`

39

Example Code

```
private void addHorizontalFlowPanel() {
    FlowPanel fPanel = new FlowPanel();
    fPanel.setWidth("740px");
    for (int i = 1; i <= 10; i++) {
        SimplePanel wrapper = new SimplePanel();
        wrapper.add(new Button("Button " + i));
        wrapper.getElement().
            setAttribute("style", "float: left;");
        fPanel.add(wrapper);
    }

    RootPanel.get("flow-panel-horizontal").add(fPanel);
}
```

40

Example Result



41

Underlying HTML

```
<div id="flow-panel-horizontal">  
  <div style="width: 740px;">  
    <div style="float: left;">  
      <button class="gwt-Button my-button" type="button">  
        Button 1</button>  
    </div>  
    <div style="float: left;">  
      <button class="gwt-Button my-button" type="button">  
        Button 2</button>  
    </div>  
    ...  
    <div style="float: left;">  
      <button class="gwt-Button my-button" type="button">  
        Button 10</button>  
    </div>  
  </div>  
</div>
```

42



Grid

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

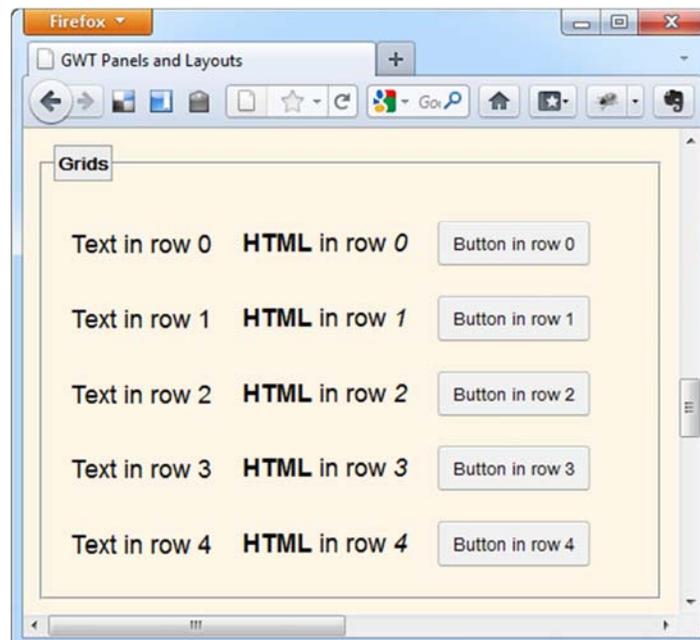
- **Purpose**
 - To put widgets in tabular layout with fixed cell sizes
 - Similar to AWT's GridLayout. Also see FlexTable.
- **Constructor**
 - new Grid(int rows, int columns)
 - Note that class is called "Grid", not "GridPanel"
- **Main methods**
 - setWidget(int row, int col, Widget w)
 - Inserts widget into cell. Note that you do not use "add"
 - setHTML(int row, int col, String htmlText)
 - Wraps htmlText in HTML widget, then inserts
 - setText(int row, int col, String text)
 - Inserts plain text into cell
 - setCellPadding(int pixels), setCellSpacing(int pixels)
 - If there are no cell borders, these two amount to same thing
 - getRowCount, getColumnCount
 - resize(rows, cols), resizeRows(rows), resizeColumns(cols)

Example Code

```
private void addGrid() { // Class is Grid, not GridPanel
    Grid grid = new Grid(5, 3);
    grid.setCellPadding(10);
    for(int i=0; i<grid.getRowCount(); i++) {
        grid.setText(i, 0, "Text in row " + i);
        grid.setHTML(i, 1, "<b>HTML</b> in row <i>" + i + "</i>");
        grid.setWidget(i, 2, new Button("Button in row " + i));
    }
    RootPanel.get("grid").add(grid);
}
```

45

Example Result



46

Underlying HTML

```
<div id="grid">  
<table>  
<colgroup><col/></colgroup>  
<tbody>  
<tr>  
  <td>Text in col 0</td>  
  <td>Text in col 1</td>  
  <td>Text in col 2</td>  
  <td>Text in col 3</td>  
  <td>Text in col 4</td>  
</tr>  
<tr>...</tr>  
<tr>...</tr>  
</tbody>  
</table>  
</div>
```

47

© 2013 Marty Hall & Yaakov Chaikin



FlexTable

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Summary

- **Purpose**

- Same general purpose as Grid: to generate a table
- Allows adding rows and columns on the fly
 - Automatically resizes
- Allows multiple colspan and rowspan
- **Grid is simpler: use Grid if extra functionality not needed**

- **Main methods**

- setWidget(int row, int col, Widget w)
 - Inserts widget into cell. Note that you do not use “add”
- setHTML(int row, int col, String htmlText)
 - Wraps htmlText in HTML widget, then inserts.
 - **If row number is past end of table, it just adds it to bottom.**
- Etc.

49

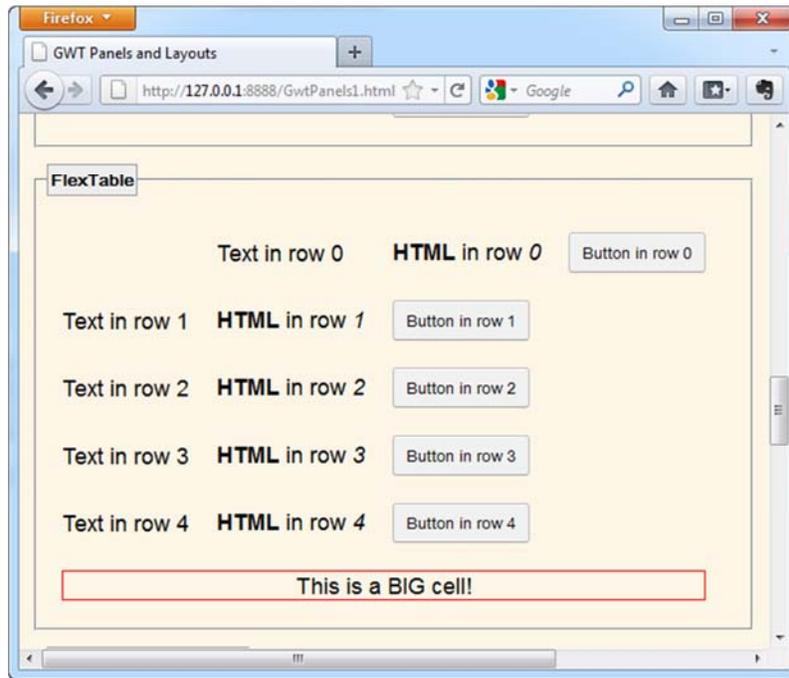
Example Code

```
private void addFlexTable() { Creates cells, adjusts table just by setting a value for a row/col
    FlexTable table = new FlexTable();
    table.setCellPadding(10);
    for (int i = 0; i < 5; i++) {
        table.setText(1, 0, "Text in row " + i);
        table.setHTML(i, 1, "<b>HTML</b> in row <i>" + i + "</i>");
        table.setWidget(i, 2, new Button("Button in row " + i));
    }
    table.insertCell(0, 0); Inserts empty cell at position (0,0)
    String cellText = "<div style='text-align: center;'
        + " border: solid red 1px;'>This is a BIG cell!</div>";
    table.setHTML(5, 0, cellText);
    table.getFlexCellFormatter().setColSpan(5, 0, 4);
    RootPanel.get("flex-table").add(table);
}
```

Spans cell at position (5,0) 4 columns

50

Example Result



51

Underlying HTML

```
<div id="flex-table">
<table cellpadding="10">
...
<tbody>
  <tr>
    <td></td>
    <td>Text in row 0</td>
    <td><b>HTML</b> in row <i>0</i></td>
    <td><button class="gwt-Button" type="button">Button in row
      0</button></td>
  </tr>
  <td colspan="4"><div style="text-align: center; border:
    solid red 1px;">This is a BIG cell!</div></td>
  </tr>
</tbody>
</table>
</div>
```

52



Wrap-Up

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Other Simple Panel Types

- **HTMLPanel**
 - A panel that contains HTML, but where you can add widgets to any HTML sub-element that has an id. Has helper method to create unique ids.
 - Mostly used in declarative approach (see UiBinder slides)
- **AbsolutePanel**
 - A panel that positions all of its children absolutely, allowing them to overlap.
- **ScrollPane**
 - If given a fixed size, will add scrollbars if necessary. Used in this tutorial, but didn't have its own section.

Summary

- **Main layout approaches**
 - HTML-based: best for simple apps with light GWT usage
 - Java-based: best for heavy use of GWT and desktop feel
 - Declarative layout (UiBinder): best for heavy use of GWT but when it is hard to visualize design in Java
- **Simple Panel types**
 - HorizontalPanel, VerticalPanel
 - Like BorderLayout from desktop Java. Uses <table>.
 - FlowPanel
 - Can mimic HorizontalPanel and VerticalPanel if you use extra wrappers and CSS. Does not use <table>.
 - Grid
 - Like GridLayout from desktop Java
 - FlexTable (advanced Grid)

55

© 2013 Marty Hall & Yaakov Chaikin



Questions?

[JSF 2](#), [PrimeFaces](#), [Java 7 or 8](#), [Ajax](#), [jQuery](#), [Hadoop](#), [RESTful Web Services](#), [Android](#), [HTML5](#), [Spring](#), [Hibernate](#), [Servlets](#), [JSP](#), [GWT](#), and other [Java EE training](#).

Customized Java EE Training: <http://courses.coreservlets.com/>

GWT, Java 7 and 8, JSF 2, PrimeFaces, HTML5, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, REST, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.