ORACLE
FUSION MIDDLEWARE
APPLICATION DEVELOPMENT
FRAMEWORK

An Oracle White Paper
Oct 2013

ADF Faces Layout Basics

ORACLE

Disclaimer

Overview of Layouts in ADF Faces

Layouts in ADF Faces are built using layout components. In ADF Faces, layout components have a specific behavior in which contained child components are placed on a page or view. There are many layout components offered by ADF Faces out of the box, some of them can be stretched by a parent container, some of them can stretch the components they contain and some of them can do both.

Introduction

Layouts play an important role in user experience and acceptance of application user interfaces. This whitepaper provides basic information about different layout components offered by ADF Faces, by example of reverse engineering of selected popular web user interfaces to explain the usage of different layout components, solutions to some frequent questions and best practices. This whitepaper also shows how to use different layout components in combination to achieve a complex page design.

Geometry Management

Modern web layouts don't treat all areas equally in that some need to stretch to claim maximum space and others don't but scroll instead. We need to allow stretching some of the contents, scroll some of the contents and keep some of the contents static. The resize behavior of ADF Faces components is referred to as geometry management, which

- Ensures components to properly respond to browser resizing.
- Lays out components to fit into the size available on a specific area of the view.
- Handles the stretch behavior based on parent or child component settings.

Resizing and the geometry management of the components can be imagined to one of the puzzle pieces, as shown in Figure 1.

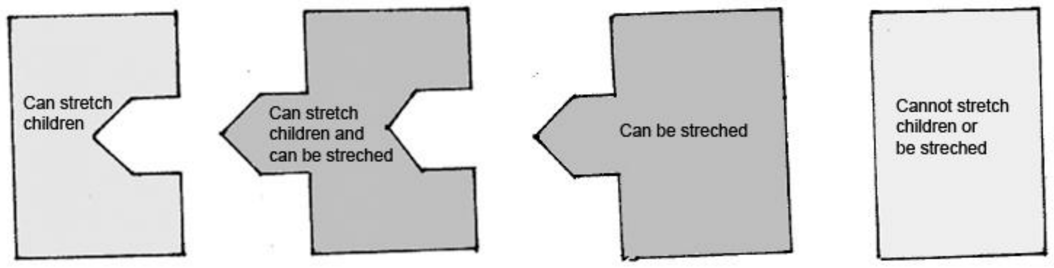

Can stretch children
Can stretch children and can be streched
Can be streched
Cannot stretch children or be streched

Figure 1.Different categories of components for geometry management.

Below table provides comparative overview on layouts which can be stretched by parent and stretch its children.

TABLE 1. GEOMETRY MANAGEMENT OF DIFFERENT ADF LAYOUT COMPONENTS

LAYOUT NAME	STRETCHABLE BY PARENT	STRETCH ITS CHILDREN
panelStretchLayout	Yes	Yes
panelGroupLayout	Yes (if layout is scroll/vertical)	No
panelBorderLayout	No	No
panelFormLayout	No	No
panelGridLayout	Yes	Yes
panelAccordion	Yes	Yes
panelSplitter	Yes	Yes

panelTabbed	Yes	Yes
panelDashboard	Yes	Yes
panelBox	Yes	Yes (only if being stretched by parent)
decorativeBox	Yes	Yes
panelCollection	Yes	Yes
panelHeader	Yes	Yes (only if being stretched by parent)
panelDrawer	Yes	Yes
panelSpringboard	Yes	Yes

Different Layouts provided by ADF Faces

All the ADF Layout components hold the UI components either as direct children or in a special location, which are defined as "facets". The names of these "facets" differ by components and some components don't even have "facets".

Developers new to ADF Faces may be overwhelmed by the choice of layout components wondering which ones to choose for a specific layout to build. Common usage pattern for some of the ADF Faces layout components are described below.

Layout Components

To map available ADF Faces layout components to their use, let us look at frequent developer questions with respect to their stretch-ability.

Stretchable by its parent and stretch its children

- "Which component can I use as the parent container for whole of my page?" or "Which component I can use to stretch the content?"
    - panelStretchLayout : This has 5 facets (top, start, center, end and bottom)
        - Use this layout's center facet to place stretchable content.
        - Can be used as a parent container for the page.
- "How do I put series of contents in different panels, which can individually be expanded/collapsed by the user?"
    - panelAccordion : No facets
        - Use this in order to let the user to expand / collapse the contents.
        - We can place multiple af:showDetailItem inside this component.
- "I want to build a layout as grid. Is there any component I can make use of?"
    - panelGridLayout : This has no facets, but has gridRow and gridCell children

        - Use this layout to construct island contents as grid (rows and columns)
- "I want to split the page into two parts, and user can adjust the size of it."
    - panelSplitter : This has two facets (first and second)
        - Use this to divide the page into two parts (horizontal/vertical).
        - A splitter will be provided, using which the size of the divided portions can be adjusted.
- "My page contains many sub-pages, and I want to show them in different tabs."
    - panelTabbed : No facets
        - Use this layout to show the contents in different tabs.
- "I am building a dashboard with multiple panels inside in it. How do I achieve it?"
    - panelDashboard : No facets
        - Use this layout to tile the panelBoxes / regions.
        - Can leverage the option of dragging the panels (move) to different locations with the help of component drag source tag.
- "I want to display supporting information in the page."
    - panelBox : This has one facet (toolbar)
        - The panelBox is used to place ancillary information on a page, offset by a certain color.
- "I want to display an information to the user and want it to look better than panelBox."
    - decorativeBox : This has two facets (center and top)
        - Use it to give a bordered look for the content
        - Theme can be changed dynamically for the child components to give different look and feel at runtime.
- "My page contains a table. I want to let the users to show/hide columns at runtime."
    - panelCollection : This has 6 facets (afterToolbar, menus, secondaryToolbar, statusbar, toolbar, viewMenu)
        - Surround the table / treeTable inside this layout in order to get some additional features (show/hide columns, freeze columns etc.).
- "Do we have any component, using which we can display information with a header?"
    - panelHeader : This has 6 facets (context, help, info, legend, menuBar, toolbar)
        - This component can also be used to display informational messages in the page.
- "I need to have floating tabs, which should get activated and show details when clicked."
    - panelDrawer : No facets
        - Use this to toggle the contents in the tab (like closing and opening the drawer)
- "I would like to create group of items/icons as a strip and also as a grid. Is there any such components in ADF?"
    - panelSpringboard : No facets
        - Use this to display the group of contents as icons in a Strip / Grid

Stretchable by its parent component and don't stretch children

ADF Faces offers some components which can be stretched by its parent, but cannot stretch its children.

- "I want to make the components in my page to be laid out horizontally / vertically."
    - panelGroupLayout : This has 1 facet (separator)
        - Use this layout to logically group the contents (horizontal/vertical/scroll).

Not stretchable by its parent and don't stretch children

There are some components which cannot be stretched by its parent and cannot stretch its children

- "I want to make the components in my page to be laid out horizontally / vertically without stretching the child components."
    - panelBorderLayout : This has 12 facets (start, innerStart, top, innerTop, left, innerLeft, end, innerEnd, right, innertRight, bottom and innerBottom)
        - Use this layout to logically group the contents that are not stretchable, in different locations.

- "I am building a page with input components as form. Which layout component should I use?"
    - panelFormLayout : This has 1 facet (footer)
        - Use this layout to create a form with aligned labels and form fields (ex. Registration form / Reservation form / Login form etc.).
        - Set the rows & columns properties of this layout to arrange the fields

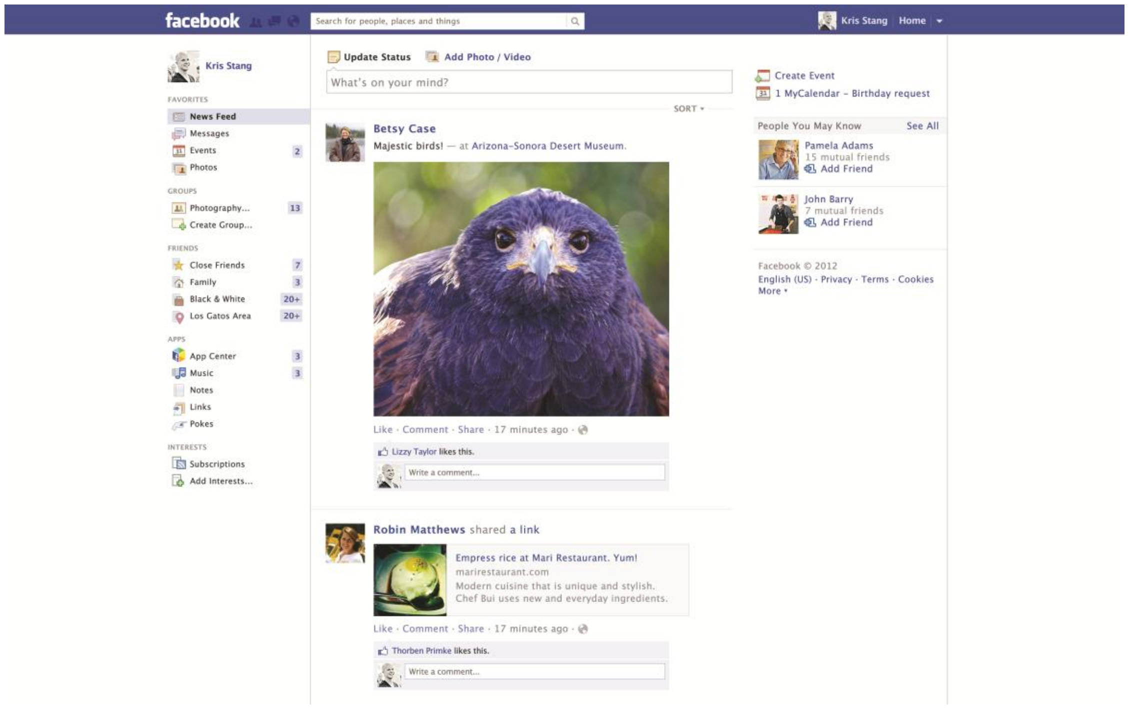
Reverse Engineering

As we've gained some basic knowledge about different layouts and their stretch-ability, we can visualize some of the popular websites and reverse engineer those using ADF Faces. Let us take the following websites for construction

- Facebook.
- Gmail
- Twitter

Reverse Engineering – Facebook

Main page of Facebook contains the layout as shown in Figure 2.



Betsy Case
Majestic birds! — at Arizona-Sonora Desert Museum.

Lizzy Taylor likes this.

Robin Matthews shared a link
Empress rice at Mari Restaurant. Yum!
marirestaurant.com
Modern cuisine that is unique and stylish. Chef Bui uses new and everyday ingredients.

Thorben Primke likes this.

Facebook and the Facebook logo are registered trademarks of Facebook Inc.

Figure 2.Screenshot of Facebook's main page.

If we split up the UI, these are different components

- Stretchable outer container
    1. Non-stretchable/non-scrollable header
    2. Non-stretchable/non-scrollable left panel
    3. Non-stretchable/scrollable center panel
    4. Non-stretchable/non-scrollable right panel
    5. Non-stretchable/scrollable rightmost panel

This can be visualized like the image shown in Figure 3.

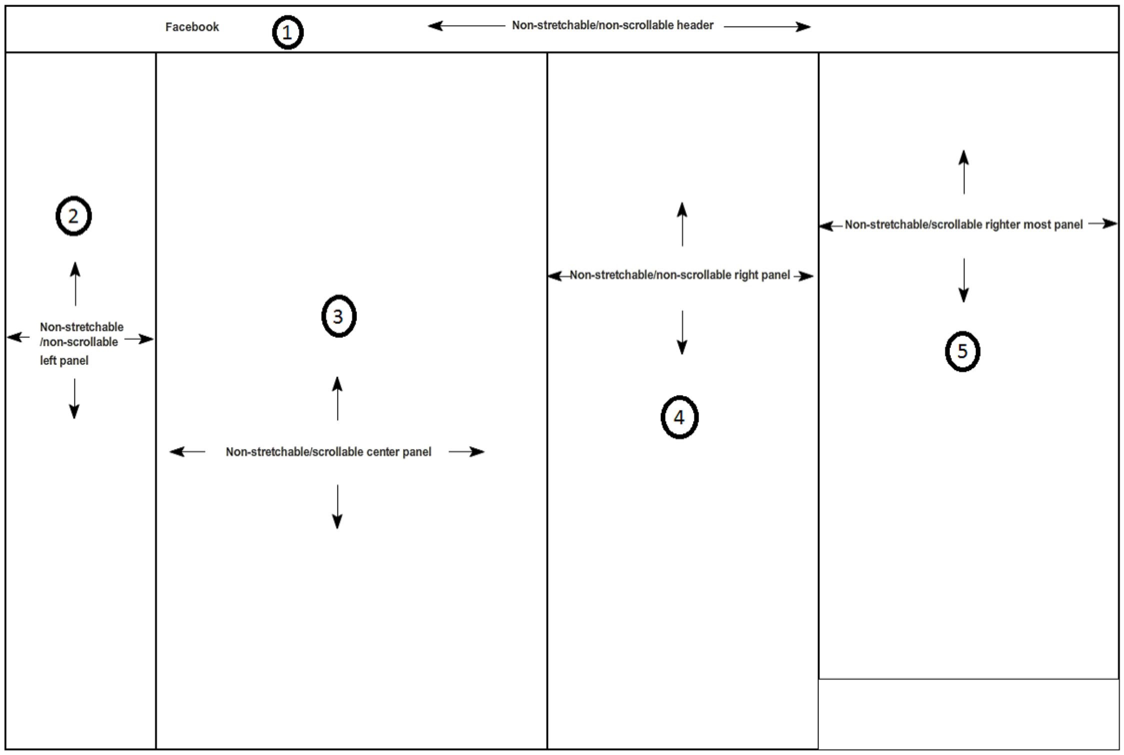

Facebook
1
Non-stretchable/non-scrollable header
2
Non-stretchable /non-scrollable left panel
3
Non-stretchable/scrollable center panel
Non-stretchable/non-scrollable right panel
4
Non-stretchable/scrollable righter most panel
5

Figure 3.Mockup of Facebook's main page.

To create the similar layout in ADF Faces, we can use the combination of following layout components.

- af:panelStretchLayout
    - This will stretch its children
- af:panelGroupLayout
    - layout="scroll" can be used to for the scrollable area
- af:panelGridLayout
    - With gridRow and gridCell(s) for splitting right area into two

And the rough design of it in ADF Faces will be similar to the image shown in Figure 4.

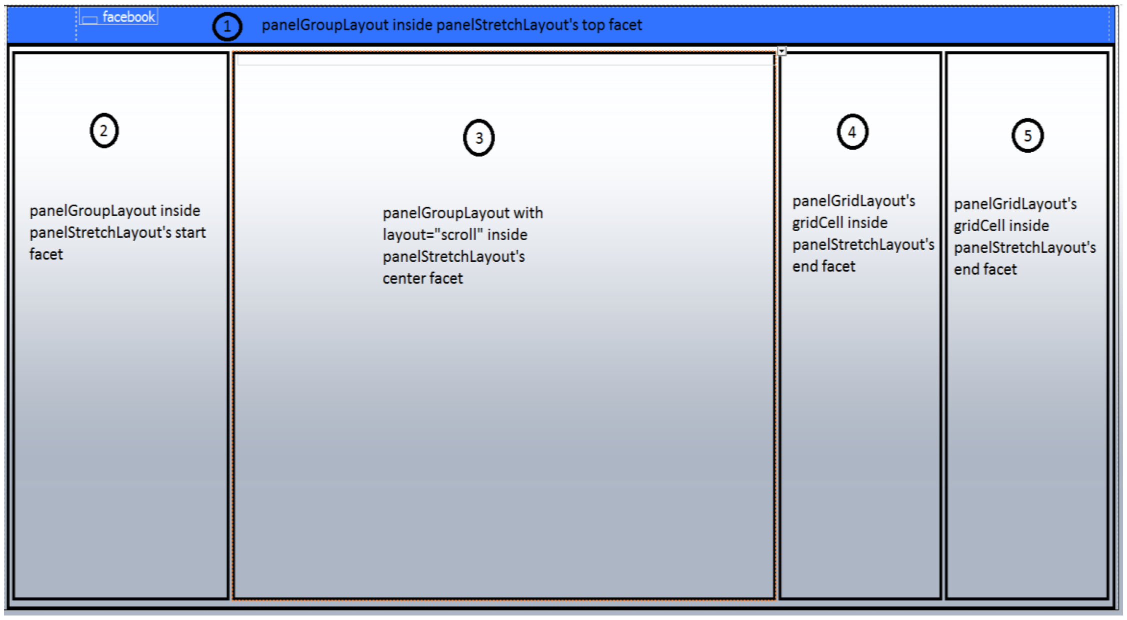

facebook
1 panelGroupLayout inside panelStretchLayout's top facet
2 panelGroupLayout inside panelStretchLayout's start facet
3 panelGroupLayout with layout="scroll" inside panelStretchLayout's center facet
4 panelGridLayout's gridCell inside panelStretchLayout's end facet
5 panelGridLayout's gridCell inside panelStretchLayout's end facet

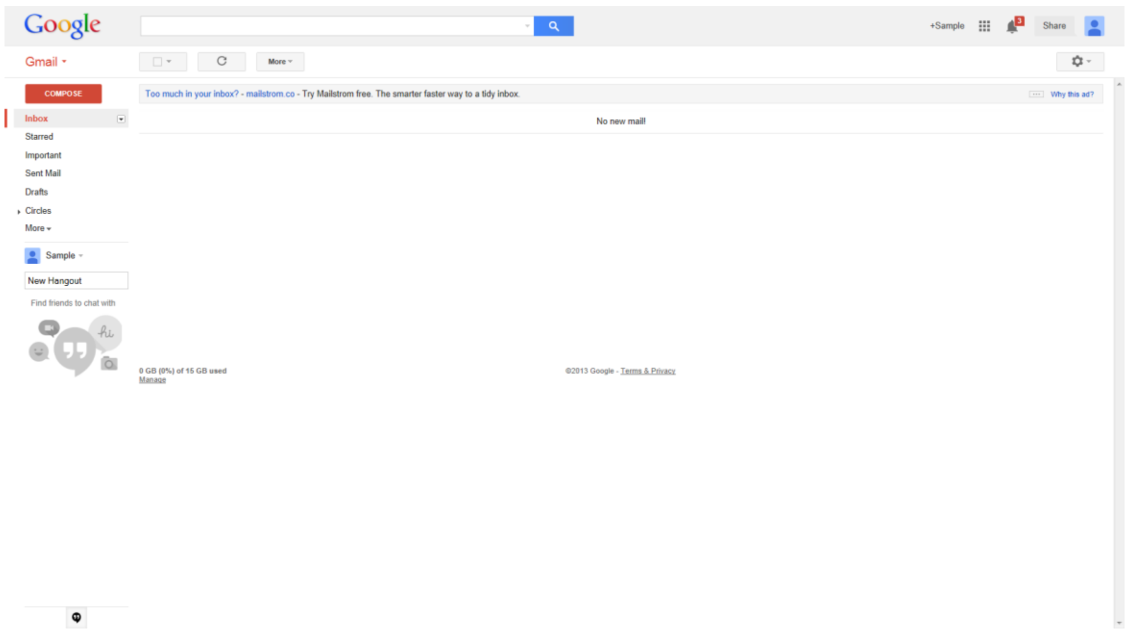
Figure 4.Rough design of Facebook's main page in ADF Faces.

To achieve this, this is how we can design the layout.

- af:panelStretchLayout for the outer stretchable container
    - This will make sure the whole page can be stretched
- af:panelGroupLayout/af:panelGridLayout in the top facet of the parent to get the branding bar
    - Using which, we can place the relevant icons / images / links etc., in required location.
- af:panelGroupLayout inside the start facet.
    - This will allow us to place the contents in the left side of the page.
- af:panelGroupLayout with layout="scroll" in the center facet
    - This will let us have a scrollable container in the middle portion.
- af:panelGridLayout with gridRow and 2 gridCells in the end facet
    - This will split the right portion of the page into two parts.

Reverse Engineering – Gmail

Main page of Gmail look like the image shown in Figure 5.



Too much in your inbox? - mailstrom.co - Try Mailstrom free. The smarter faster way to a tidy inbox.

No new mail!

0 GB (0%) of 15 GB used

©2013 Google - Terms & Privacy

Google and the Google logo are registered trademarks of Google Inc.

Figure 5.Screenshot of Gmail's main page.

This UI has been formed with somewhat similar to following parts.

- Stretchable outer container
  1. Non-stretchable/non-scrollable 1st panel - header (menu bar)
  2. Non-stretchable/non-scrollable 2nd panel – header with branding bar
  3. Non-stretchable/non-scrollable 3rd panel – sub-menu bar
  4. Non-stretchable/non-scrollable 4th panel – advertisement bar
  5. Non-stretchable/ scrollable top left panel
  6. Non-stretchable/ scrollable bottom left panel
  7. Non-stretchable/scrollable right panel

This can be visualized like the image shown in Figure 6.

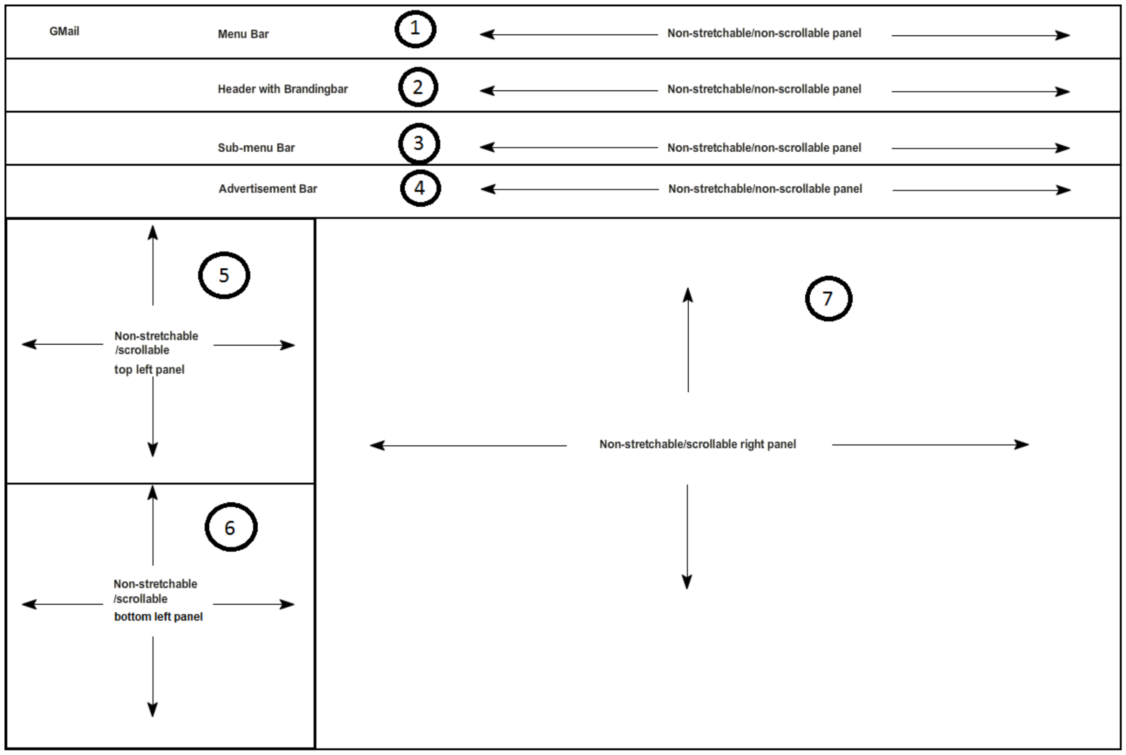

GMail
Menu Bar
1
Non-stretchable/non-scrollable panel
Header with Brandingbar
2
Non-stretchable/non-scrollable panel
Sub-menu Bar
3
Non-stretchable/non-scrollable panel
Advertisement Bar
4
Non-stretchable/non-scrollable panel
5
Non-stretchable /scrollable top left panel
6
Non-stretchable /scrollable bottom left panel
7
Non-stretchable/scrollable right panel

Figure 6.Mockup of Gmail's main page.

The rough design of it using ADF Faces can be like Figure 7.

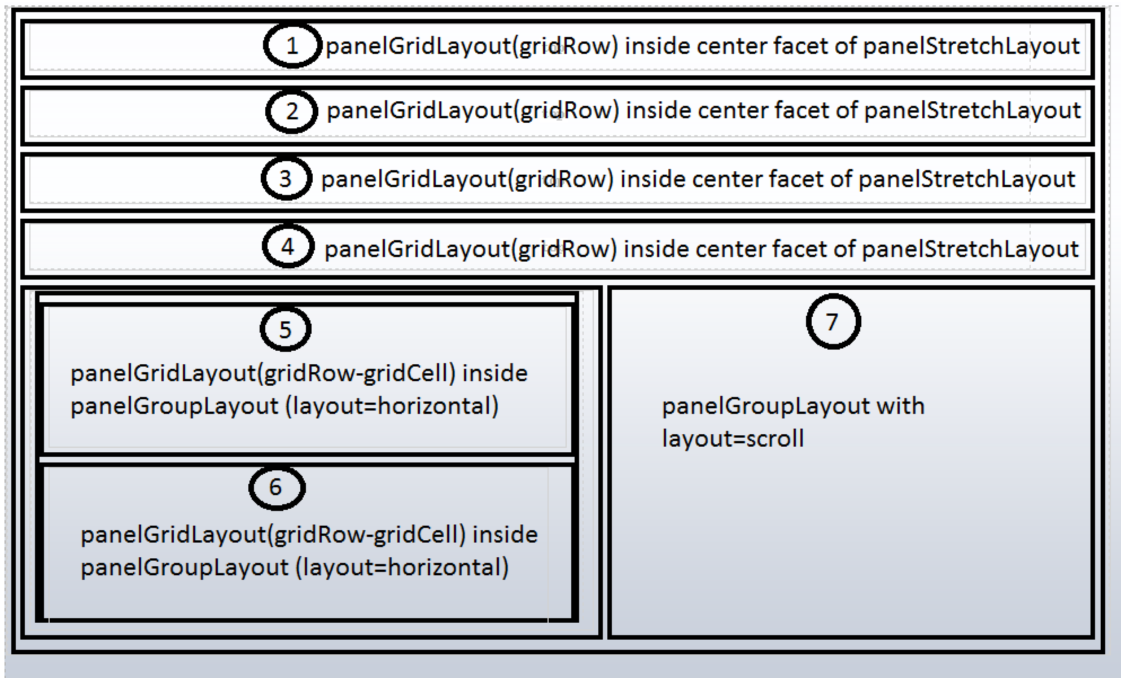

1 panelGridLayout(gridRow) inside center facet of panelStretchLayout
2 panelGridLayout(gridRow) inside center facet of panelStretchLayout
3 panelGridLayout(gridRow) inside center facet of panelStretchLayout
4 panelGridLayout(gridRow) inside center facet of panelStretchLayout
5 panelGridLayout(gridRow-gridCell) inside panelGroupLayout (layout=horizontal)
6 panelGridLayout(gridRow-gridCell) inside panelGroupLayout (layout=horizontal)
7 panelGroupLayout with layout=scroll

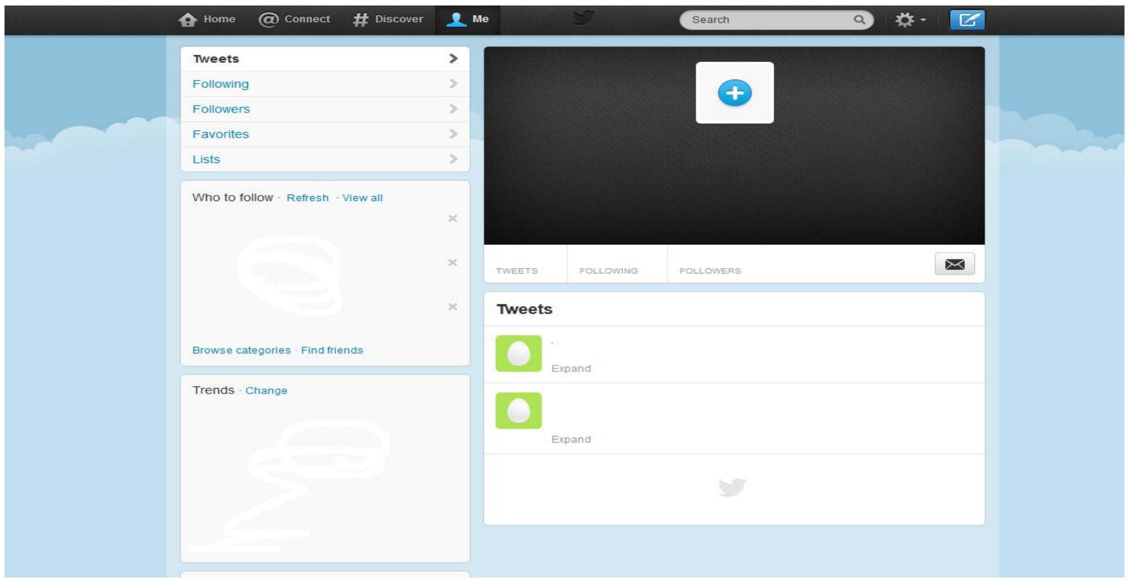
Figure 7.Rough design of Gmail's main page in ADF Faces.

To achieve this, this is how we can design the layout.

- af:panelStretchLayout for the outer stretchable container
  - Using which, the page will be made as stretchable
- af:panelGridLayout in the center facet of the parent to get rows and column layout (with 4 grid rows to simulate top 4 non-scrollable/non-stretchable panels)
  - Can add an af:panelGroupLayout inside the grid cells for wrapping the contents if needed.
- Combination of af:panelGridLayout and af:panelGroupLayout to get the bottom panels.
  - Can use layout=”scroll” for getting the scrollable panel (bottom right)

Reverse Engineering – Twitter

Twitter UI look simple and does not have many complex combinations. Figure 8 shows Twitter main page.



Twitter and the Twitter logo are registered trademarks of Twitter Inc.

Figure 8.Screenshot of Twitter’s main page.

This page can be split into following parts.

- Stretchable outer container

  1. Non-stretchable/non-scrollable top panel - header (menu bar)
  2. 3 non-stretchable/non-scrollable left panels

3. Non-stretchable/non-scrollable right panel
4. Non-stretchable/ scrollable bottom right panel
5. Free spaces on the left and right side of the page

If the split up is visualized, it will look similar to the image shown in Figure 9.

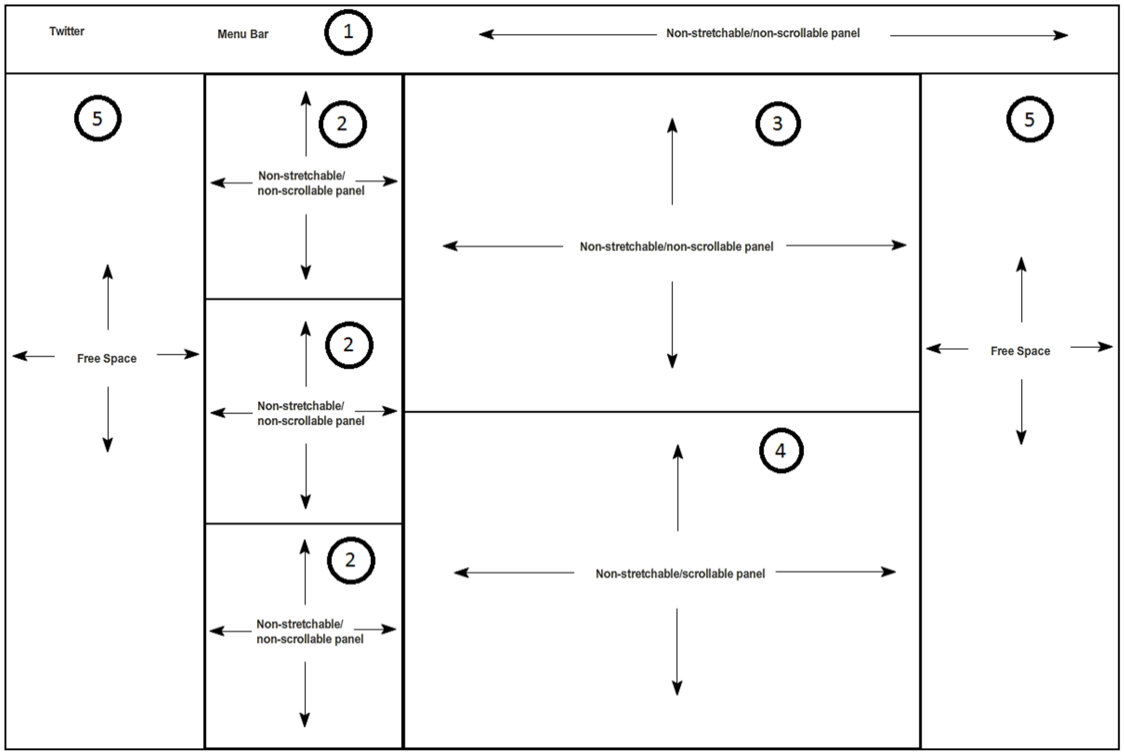

Twitter
Menu Bar
1
Non-stretchable/non-scrollable panel
5
2
Non-stretchable/ non-scrollable panel
3
Non-stretchable/non-scrollable panel
5
Free Space
2
Non-stretchable/ non-scrollable panel
Free Space
4
2
Non-stretchable/scrollable panel
Non-stretchable/ non-scrollable panel

Figure 9.Mockup of Twitter's main page.

The rough design of it using ADF Faces can be like the image shown in Figure 10.

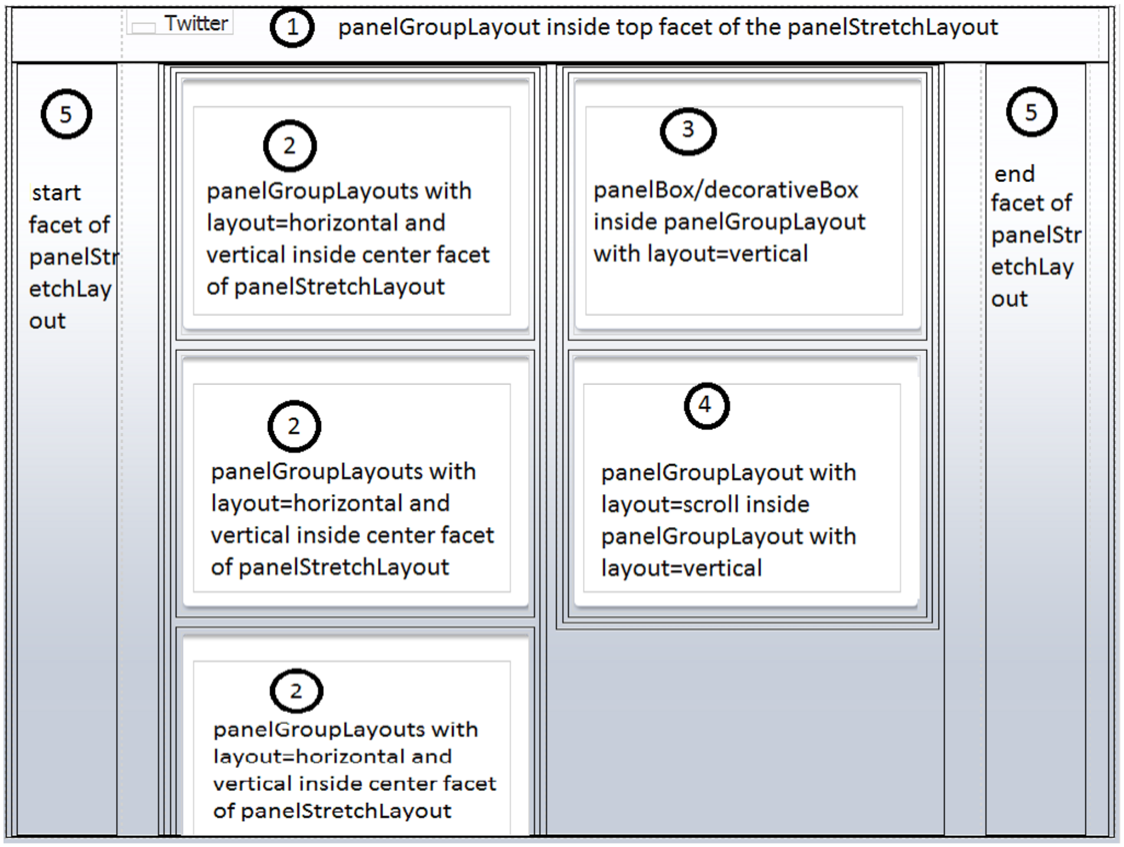

Twitter
1
panelGroupLayout inside top facet of the panelStretchLayout
5
start facet of panelStretchLayout
2
panelGroupLayouts with layout=horizontal and vertical inside center facet of panelStretchLayout
3
panelBox/decorativeBox inside panelGroupLayout with layout=vertical
5
end facet of panelStretchLayout
2
panelGroupLayouts with layout=horizontal and vertical inside center facet of panelStretchLayout
4
panelGroupLayout with layout=scroll inside panelGroupLayout with layout=vertical
2
panelGroupLayouts with layout=horizontal and vertical inside center facet of panelStretchLayout

Figure 10.Rough design of Twitter's main page in ADF Faces.

To achieve this, this is how we can design the layout.

- af:panelStretchLayout for the outer stretchable container
    - For stretching the entire page.
- Combination of af:panelGroupLayout(s) and af:panelStretchLayout to form the inner containers.
    - Can use nested af:panelGroupLayouts with layout as horizontal and vertical to form two columns of panels.
- af:panelBox(s)/af:decorativeBox(s) inside the af:panelGroupLayout(s) for forming the individual islands.
    - Parent container's start and end facets used for the free space on the left and right side of the page.

Above examples are purely on designing the outer layer of the pages. For laying out the inner layers, we can use combination of other layout components like af:panelTabbed, af:panelList, af:panelBox, af:decorativeBox, af:panelSplitter, af:panelAccordion etc.

Solutions to frequent layout scenarios

As we've now designed some pages, let us take couple of frequently used layout scenarios and see how to implement them with ADF Faces.

Aligning a form at the center of the page

Frequent requirement of the developer is "how to align a form (ex: login form) at the center of my page". We can achieve this with a combination of af:panelStretchLayout, af:panelGroupLayout and af:panelFormLayout.

- Design the main container using af:panelStretchLayout.
- Add an af:panelGroupLayout in the center facet of the main container.
- Set its halign property to center and valign property to middle.
- Add an af:panelFormLayout inside this to construct your form.
- Add empty af:panelGroupLayouts in the top and bottom facets of the main container as fillers.
- Set start&endWidth, top&bottomHeight properties of the main container according to your need (ex: 100px)
- This can further be decorated with an af:panelBox / af:decorativeBox as needed.

Alternatively, this can also be achieved using panelGridLayout and panelFormLayout

- Create a panelGridLayout with one row and a column.
- Set the height of the gridRow to 100% and width of the gridCell to 100%
- Set the valign property of the gridCell to middle and halign to center
- Add a panelFormLayout inside the gridCell and add the UI Components inside in it.

After designing it will roughly look like the image shown in Figure 11.

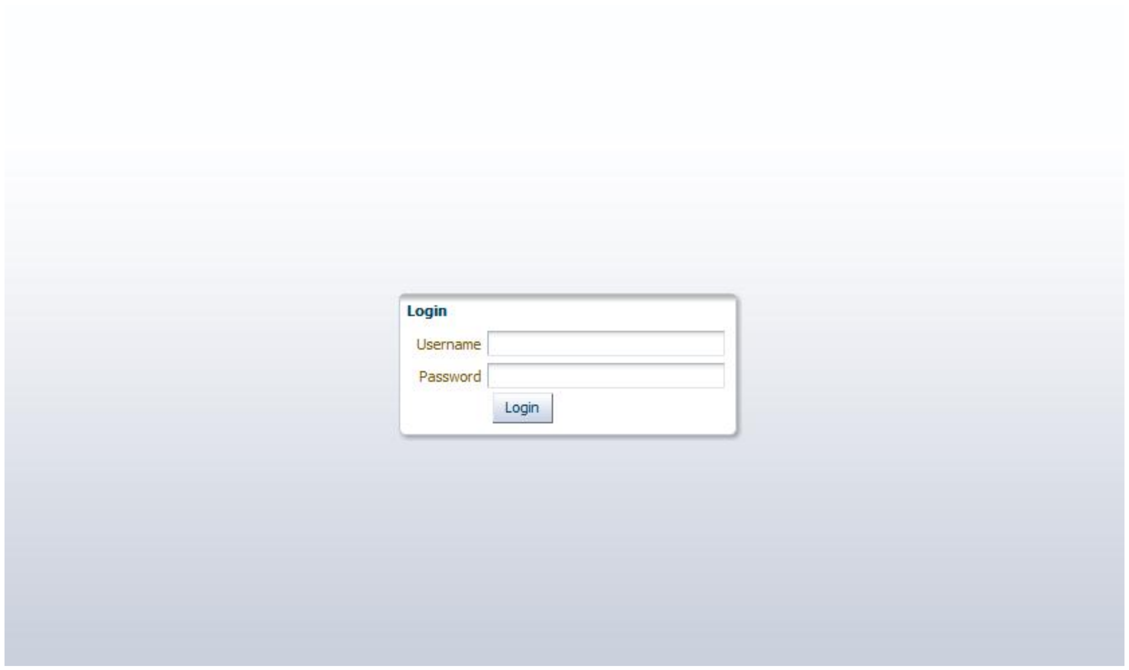


Figure 11.Login form aligned at the center of the page

Designing the table/treeTable

Another frequent requirement is designing the af:table / af:treeTable. To get more out of these components, we can surround them with an af:panelCollection. With the panelCollection, we get some extra options out of the box. Image 12 shows the difference between a table surrounded by panelCollection and the one without it.

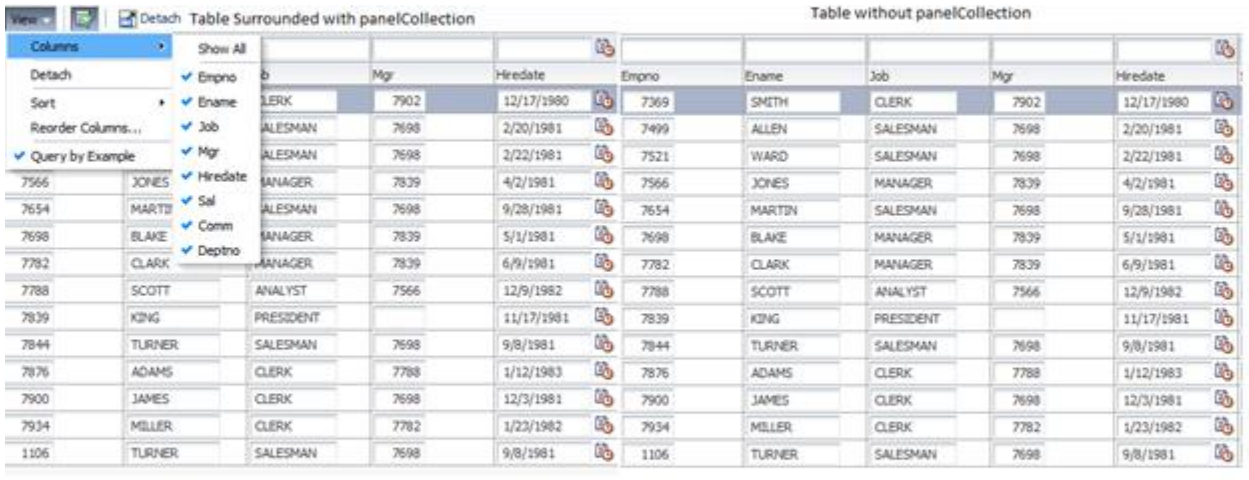


Table Surrounded with panelCollection

		b	Mgr	Hiredate
		LERK	7902	12/17/1980
		ALESMAN	7698	2/20/1981
		ALESMAN	7698	2/22/1981
7566	JONES	ANAGER	7839	4/2/1981
7654	MARTI	ALESMAN	7698	9/28/1981
7698	BLAKE	ANAGER	7839	5/1/1981
7782	CLARK	ANAGER	7839	6/9/1981
7788	SCOTT	ANALYST	7566	12/9/1982
7839	KING	PRESIDENT		11/17/1981
7844	TURNER	SALESMAN	7698	9/8/1981
7876	ADAMS	CLERK	7788	1/12/1983
7900	JAMES	CLERK	7698	12/3/1981
7934	MILLER	CLERK	7782	1/23/1982
1106	TURNER	SALESMAN	7698	9/8/1981

Table without panelCollection

Empno	Ename	Job	Mgr	Hiredate
7369	SMITH	CLERK	7902	12/17/1980
7499	ALLEN	SALESMAN	7698	2/20/1981
7521	WARD	SALESMAN	7698	2/22/1981
7566	JONES	MANAGER	7839	4/2/1981
7654	MARTIN	SALESMAN	7698	9/28/1981
7698	BLAKE	MANAGER	7839	5/1/1981
7782	CLARK	MANAGER	7839	6/9/1981
7788	SCOTT	ANALYST	7566	12/9/1982
7839	KING	PRESIDENT		11/17/1981
7844	TURNER	SALESMAN	7698	9/8/1981
7876	ADAMS	CLERK	7788	1/12/1983
7900	JAMES	CLERK	7698	12/3/1981
7934	MILLER	CLERK	7782	1/23/1982
1106	TURNER	SALESMAN	7698	9/8/1981

Figure 12.Screenshot of table surrounded with panelCollection and without panelCollection

Another key point in table / treeTable is to adjust the width of them to fit the entire page. ADF provides a built-in global style selector AFStretchWidth for this scenario. Refer this documentation to know the global style selectors available in ADF Faces.

Tips

- Use Page Templates for better layout design
- Make use of styleClasses

Summary

From this article, we've now learnt the basic layout components in ADF, their geometry management, how to construct a layout for the page and to the answers for some of the frequently asked layout design questions. We've also learned that

- ADF Faces is a component based UI framework
- Develop UIs in components not markup
- We can build sophisticated layouts in ADF Faces by nesting layout components
- Geometry management allows ADF Faces to automatically adapt to browser real estate changes

Additional Resources

Refer to these documents for more in depth information about the ADF Faces Layout Components.

- ADF Faces Layout Best Practices
- ADF Faces Tag Documentation
- Developing Web User Interfaces with Oracle ADF Faces

ORACLE®

Oracle is committed to developing practices and products that help protect the environment

ADF Faces Layout Basics
Oct 2013
Author: Arunkumar Ramamoorthy

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Hardware and Software, Engineered to Work Together