© 2015 Marty Hall

PrimeFaces: Panels

Originals of slides and source code for examples: http://www.coreservlets.com/JSF-Tutorial/primefaces/
Also see the JSF 2 tutorial – http://www.coreservlets.com/JSF-Tutorial/jsf2/
and customized JSF2 and PrimeFaces training courses – http://courses.coreservlets.com/jsf-training.html

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

© 2015 Marty Hall

For live training on JSF 2, PrimeFaces, or other Java EE topics, email hall@coreservlets.com
Marty is also available for consulting and development support

Taught by the author of *Core Servlets and JSP*, this tutorial, and JSF 2.2 version of *Core JSF*. 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, Ajax, jQuery, Spring MVC, JSP, Android, general Java, Java 8 lambdas/streams, GWT, custom topic mix
  - Courses available in any location worldwide. Maryland/DC area companies can also choose afternoon/evening courses.
- Courses developed and taught by coreservlets.com experts (edited by Marty)
  - Hadoop, Spring, Hibernate/JPA, RESTful Web Services

Contact hall@coreservlets.com for details

Topics in This Section

- **Accordion panels**
  - Static content
  - Lazy loading
  - Dynamic/Ajax content
- **Tabbed panels**
  - Static content
  - Lazy loading
  - Dynamic/Ajax content
- **Themed tables: p:panelGrid**
- **Scroll panels**
- **Panels that can be dragged and reordered**

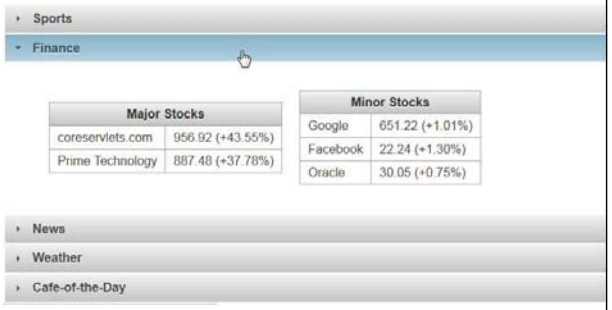
© 2015 Marty Hall

p:accordionPanel – Basics

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

p:accordionPanel: Overview

- **Appearance and behavior**
  – Wrapper for the jQuery UI accordion panel
- **Purpose: for displaying a section at a time of larger content**
  – User can toggle which section is open



Major Stocks	
coreservlets.com	956.92 (+43.55%)
Prime Technology	887.48 (+37.78%)

Minor Stocks	
Google	651.22 (+1.01%)
Facebook	22.24 (+1.30%)
Oracle	30.05 (+0.75%)

Q: What is the difference between a crow and an accordion?
A: One makes a loud and annoying sound. The other is a bird.
Q: What is the definition of a gentleman?
A: Someone who knows how to play an accordion, but doesn't.
Q: What do you call a sunken ship full of accordions?
A: A good start.

p:accordionPanel: Basic Usage

```
<p:accordionPanel>
<p:tab title="Title for First Tab">
    Arbitrary JSF and HTML content
</p:tab>
<p:tab title="Title for Second Tab">
    Arbitrary JSF and HTML content
</p:tab>
…
</p:accordionPanel>
```

p:accordionPanel: Summary of Most Important Attributes

- **<p:accordionPanel .../>...</p:accordionPanel>**
  - activeIndex (*integer* [default is 0])
    - Index of section that should be open initially. A value of -1 means all sections are initially closed.
  - dynamic (true or false [default])
    - Should all sections be loaded on page load? Or should closed tabs only be loaded (via Ajax) when user clicks?
  - cache (true [default] or false)
    - For dynamic tabs, should content be cached, or should clicking again result in a new Ajax request for potentially changed content?
  - Note: no "collapsible" option
    - Lacks the collapsible option supported by jQuery UI. PrimeFaces accordion panels are always collapsible (i.e., clicking on the header of the open tab closes it, so no content is visible).

Example: User Interest Page

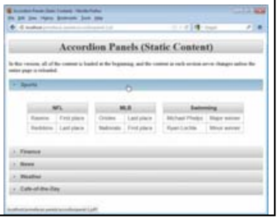
- **Page with headings**
  - Sports
    - Information about sports scores, with data taken from managed beans.
  - Finance
    - Stock values. Data from beans.
  - News
    - Latest headlines. Data from beans. Changes on each request.
  - Weather
    - Weather forecast. Static data.
  - Café-of-the-Day
    - A picture of an internet café. Data from beans. Changes on each request.
- **Uses ui:include**
  - Same sections are used in many examples in this lecture. So, content is put in separate file and inserted into accordion panel sections with ui:include.
    - ui:include is covered in general JSF2 tutorial in section on Page Templating

Main Page

```
<p:accordionPanel>
<p:tab title="Sports">
  <ui:include src="/snippets/sports.xhtml"/>
</p:tab>
<p:tab title="Finance">
  <ui:include src="/snippets/finance.xhtml"/>
</p:tab>
<p:tab title="News">
  <ui:include src="/snippets/news.xhtml"/>
</p:tab>
<p:tab title="Weather">
  <ui:include src="/snippets/weather.xhtml"/>
</p:tab>
<p:tab title="Cafe-of-the-Day">
  <ui:include src="/snippets/cafe.xhtml"/>
</p:tab>
</p:accordionPanel>
```

Sports: Facelets Content (Part I)

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:h="http://java.sun.com/jsf/html"
                xmlns:f="http://java.sun.com/jsf/core"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:p="http://primefaces.org/ui">
<h:panelGrid columns="3" cellspacing="20">

<p:panelGrid columns="2"  >
  <f:facet name="header">NFL</f:facet>
  Ravens
  <h:outputText value="#{sportsBean.ravens}"/>
  Redskins
  <h:outputText value="#{sportsBean.redskins}"/>
</p:panelGrid>
```



Accordion Panels (Static Content)

In this version, all of the content is loaded at the beginning, and the content in each section never changes unless the entire page is reloaded.

Sports

NFL	
Ravens	First place
Redskins	Last place

MLB	
Orioles	Last place
Nationals	First place

Swimming	
Michael Phelps	Major winner
Ryan Lochte	Minor winner

Finance

News

Weather

Cafe-of-the-Day

This content is somewhat arbitrary; any JSF or HTML content can be inside each accordion panel tab.
p:panelGrid is covered in the next section, but, it is basically a theme-aware replacement for h:panelGrid.

Sports: Facelets Content (Part II)

```
<p:panelGrid columns="2">
  <f:facet name="header">MLB</f:facet>
  Orioles
  <h:outputText value="#{sportsBean.orioles}"/>
  Nationals
  <h:outputText value="#{sportsBean.nationals}"/>
</p:panelGrid>
<p:panelGrid columns="2">
  <f:facet name="header">Swimming</f:facet>
  Michael Phelps
  <h:outputText value="#{sportsBean.phelps}"/>
  Ryan Lochte
  <h:outputText value="#{sportsBean.lochte}"/>
</p:panelGrid>

</h:panelGrid>
</ui:composition>
```

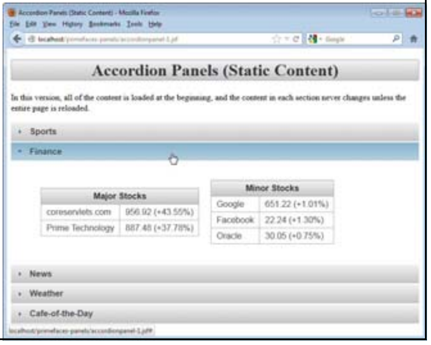
Sports: Bean

```
@ApplicationScoped
@ManagedBean
public class SportsBean {
  public String getRavens() {
    return("First place");
  }
  public String getRedskins() {
    return("Last place");
  }
  public String getOrioles() {
    return("Last place");
  }
  public String getNationals() {
    return("First place");
  }
  public String getPhelps() {
    return("Major winner");
  }
  public String getLochte() {
    return("Minor winner");
  }
}
```

The only reason for this bean is to illustrate that the content of accordion panel sections need not be static HTML, but can also come from EL elements or any other JSF content.

Finance: Facelets Content

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:h="http://java.sun.com/jsf/html"
                xmlns:f="http://java.sun.com/jsf/core"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:p="http://primefaces.org/ui">
<h:panelGrid columns="2" cellspacing="20">
<p:panelGrid columns="2"  >
  <f:facet name="header">Major Stocks</f:facet>
  coreservlets.com
  <h:outputText value="#{financeBean.coreservlets}"/>
  Prime Technology
  <h:outputText value="#{financeBean.prime}"/>
</p:panelGrid>
<p:panelGrid columns="2">
  <f:facet name="header">Minor Stocks</f:facet>
  Google
  <h:outputText value="#{financeBean.google}"/>
  Facebook
  <h:outputText value="#{financeBean.facebook}"/>
  Oracle
  <h:outputText value="#{financeBean.oracle}"/>
</p:panelGrid>
</h:panelGrid>
</ui:composition>
```



Accordion Panels (Static Content)

In this version, all of the content is loaded at the beginning, and the content in each section never changes unless the entire page is reloaded.

Sports

Finance

Major Stocks	
coreservlets.com	956.92 (+43.55%)
Prime Technology	887.48 (+37.78%)

Minor Stocks	
Google	651.22 (+1.01%)
Facebook	22.24 (+1.30%)
Oracle	30.05 (+0.75%)

News

Weather

Cafe-of-the-Day

Finance: Bean

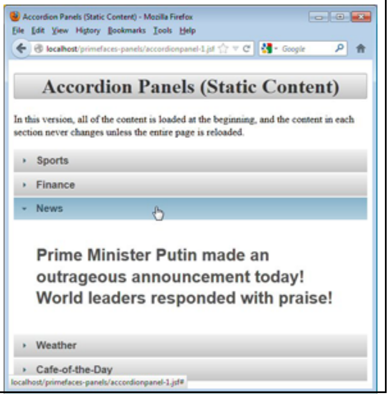
```
@ApplicationScoped
@ManagedBean
public class FinanceBean {
  public String getCoreservlets() {
    return("956.92 (+43.55%)");
  }

  public String getPrime() {
    return("887.48 (+37.78%)");
  }

  public String getGoogle() {
    return("651.22 (+1.01%)");
  }

  public String getFacebook() {
    return("22.24 (+1.30%)");
  }

  public String getOracle() {
    return("30.05 (+0.75%)");
  }
```

Again, the reason for this bean is to illustrate that the content of accordion panel sections need not be static HTML, but can also come from EL elements or any other JSF content.

News: Facelets Content

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:h="http://java.sun.com/jsf/html"
                xmlns:f="http://java.sun.com/jsf/core"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:p="http://primefaces.org/ui">
<h2 style="font-size: #{newsBean.fontSize}">
#{newsBean.headline}
</h2>
</ui:composition>
```

The output of #{newsBean.headline} potentially changes on every request.
This fact will be important when we do dynamic="true" and cache="false".


Accordion Panels (Static Content) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/accordionpanel-1.jsf
Google

Accordion Panels (Static Content)

In this version, all of the content is loaded at the beginning, and the content in each section never changes unless the entire page is reloaded.

- Sports
- Finance
- News

Prime Minister Putin made an outrageous announcement today! World leaders responded with praise!

- Weather
- Cafe-of-the-Day

localhost/primefaces-panels/accordionpanel-1.jsf#

News: Bean

```
@ApplicationScoped
@ManagedBean
public class NewsBean {
  private String template =
    "%s made %s announcement today! World leaders responded with %s!";
  private String[] leaders =
    { "President Obama", "Prime Minister Putin", "Angela Merkel", "President Xi Jinping" };
  private String[] adjectives =
    { "a surprising", "an outrageous", "an expected", "an unexpected" };
  private String[] responses =
    { "anger", "caution", "outrage", "relief", "praise" };

  public String getHeadline() {
    String headline =
      String.format(template,
                    RandomUtils.randomElement(leaders),
                    RandomUtils.randomElement(adjectives),
                    RandomUtils.randomElement(responses));
    return(headline);
  }

  public String getFontSize() {
    int pixels = 20 + (int)(Math.random() * 60);
    return(String.format("%spx", pixels));
  }
}
```

Point: the getHeadline method potentially returns a different value each time.

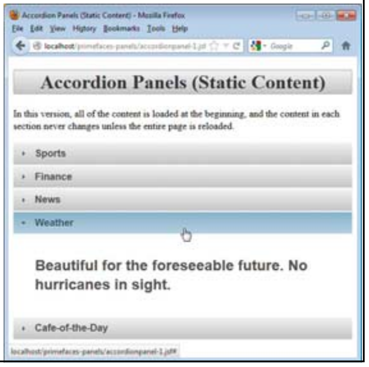
News: Helper Class

```
public class RandomUtils {
  private static Random r = new Random();

  /** Return a random int from 0 to range-1. So, randomInt(4)
   *  returns any of 0, 1, 2, or 3.
   */

  public static int randomInt(int range) {
    return(r.nextInt(range));
  }

  /** Return a random index of an array. */

  public static int randomIndex(Object[] array) {
    return(randomInt(array.length));
  }

  /** Return a random element from an array.
   *  Uses generics, so no typecast is required
   *  for the return value.
   */

  public static <T> T randomElement(T[] array) {
    return(array[randomIndex(array)]);
  }
}
```

Weather: Facelets Content

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:h="http://java.sun.com/jsf/html"
                xmlns:f="http://java.sun.com/jsf/core"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:p="http://primefaces.org/ui">
<h2>
Beautiful for the foreseeable future. No hurricanes in sight.
</h2>
</ui:composition>
```

No bean.


Accordion Panels (Static Content) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/accordionpanel-1.jsf
Google

Accordion Panels (Static Content)

In this version, all of the content is loaded at the beginning, and the content in each section never changes unless the entire page is reloaded.

- Sports
- Finance
- News
- Weather

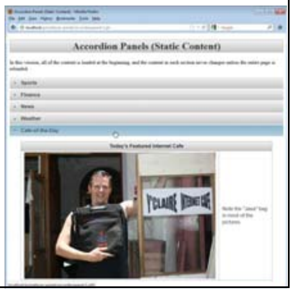
Beautiful for the foreseeable future. No hurricanes in sight.

- Cafe-of-the-Day

localhost/primefaces-panels/accordionpanel-1.jsf#

Café: Facelets Content

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:h="http://java.sun.com/jsf/html"
                xmlns:f="http://java.sun.com/jsf/core"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:p="http://primefaces.org/ui">

<p:panelGrid columns="2"  >
  <f:facet name="header">
    Today's Featured Internet Cafe
  </f:facet>
  <h:graphicImage url="#{imageBean.randomImage}"/>
  Note the "Java" bag in most of the pictures.
</p:panelGrid>

</ui:composition>
```


Accordion Panels (Static Content)
Sports
Finance
News
Weather
Cafe of the Day
Today's Featured Internet Cafe
Note the "Java" bag in most of the pictures.

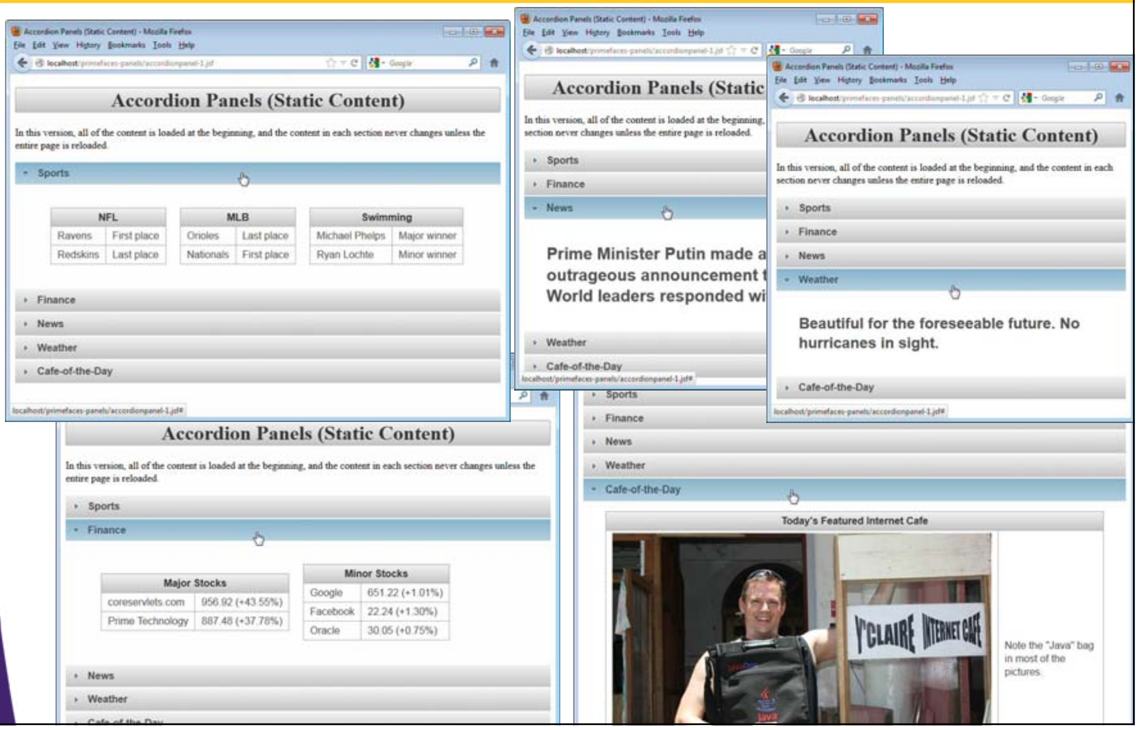
The image URL potentially changes on every request. This fact will be important when we do dynamic="true" and cache="false".

Café: Bean

```
@ApplicationScoped
@ManagedBean
public class ImageBean {
  private int numImages = 22;
  private Random r = new Random();

  public String getRandomImage() {
    int num = r.nextInt(numImages) + 1;
    // Path is relative to WebContent in Eclipse project
    // If you want it relative to WebContent/resources,
    // move "images" to resources and use
    // h:graphicImage name="cafe-x.jpg" library="images/internet-cafes"
    String path =
      String.format("/images/internet-cafes/cafe-%s.jpg", num);
    return(path);
  }
}
```

Point: the getRandomImage method potentially returns a different value each time.

Results



Accordion Panels (Static Content) - Mozilla Firefox

localhost/primefaces-panels/accordionpanel-1.jsf

Accordion Panels (Static Content)

In this version, all of the content is loaded at the beginning, and the content in each section never changes unless the entire page is reloaded.

- Sports

NFL	
Ravens	First place
Redskins	Last place

MLB	
Orioles	Last place
Nationals	First place

Swimming	
Michael Phelps	Major winner
Ryan Lochte	Minor winner

- Finance
- News
- Weather
- Cafe-of-the-Day

localhost/primefaces-panels/accordionpanel-1.jsf#

Accordion Panels (Static Content)

In this version, all of the content is loaded at the beginning, and the content in each section never changes unless the entire page is reloaded.

- Sports
- Finance

Major Stocks	
coreservlets.com	956.92 (+43.55%)
Prime Technology	887.48 (+37.78%)

Minor Stocks	
Google	651.22 (+1.01%)
Facebook	22.24 (+1.30%)
Oracle	30.05 (+0.75%)

- News
- Weather
- Cafe-of-the-Day

Accordion Panels (Static

In this version, all of the content is loaded at the beginning, section never changes unless the entire page is reloaded.

- Sports
- Finance
- News

Prime Minister Putin made a
outrageous announcement t
World leaders responded wi

- Weather
- Cafe-of-the-Day

localhost/primefaces-panels/accordionpanel-1.jsf#

Accordion Panels (Static Content)

In this version, all of the content is loaded at the beginning, and the content in each section never changes unless the entire page is reloaded.

- Sports
- Finance
- News
- Weather

Beautiful for the foreseeable future. No hurricanes in sight.

- Cafe-of-the-Day

localhost/primefaces-panels/accordionpanel-1.jsf#

- Sports
- Finance
- News
- Weather
- Cafe-of-the-Day

Today's Featured Internet Cafe

Note the "Java" bag in most of the pictures.

© 2015 Marty Hall

p:accordionPanel – Lazy Loading

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

Idea

- **dynamic="true"**
  - Designate lazy loading with
    `<p:accordionPanel dynamic="true">…</p:accordionPanel>`
  - This means that only the content for the initially open tab is loaded when the main page is loaded. Content for other tabs is loaded via Ajax only when the user first clicks on the tab.
    - Result: faster initial page load, esp. with many tabs
    - Content is cached (unless you use cache="false" as in the next section), so closing and reopening a tab will not reload content from server.
      - Note: some PrimeFaces versions implemented cache="true" (the default) incorrectly, and Ajax requests were re-issued, just as with cache="false". So, test carefully. Works properly in current release.

Usage Warning: Lazy Loading Applies to GET Requests Only

- **PrimeFaces is very clever**
  - It automatically saves partial page content and makes it available via Ajax (with or without caching)
  - But, this partial page content is associated with the URL
- **Results pages**
  - Use POST with forwards, and thus the URL refers to a totally different page
- **Bottom line**
  - dynamic="true" (regardless of the value of “cache”) works only for GET requests
  - You can use tabbed panels or accordion panels with results pages, but *not* with dynamic=“true"

SEO Warning: Lazy Tabs Will Not Be Indexed

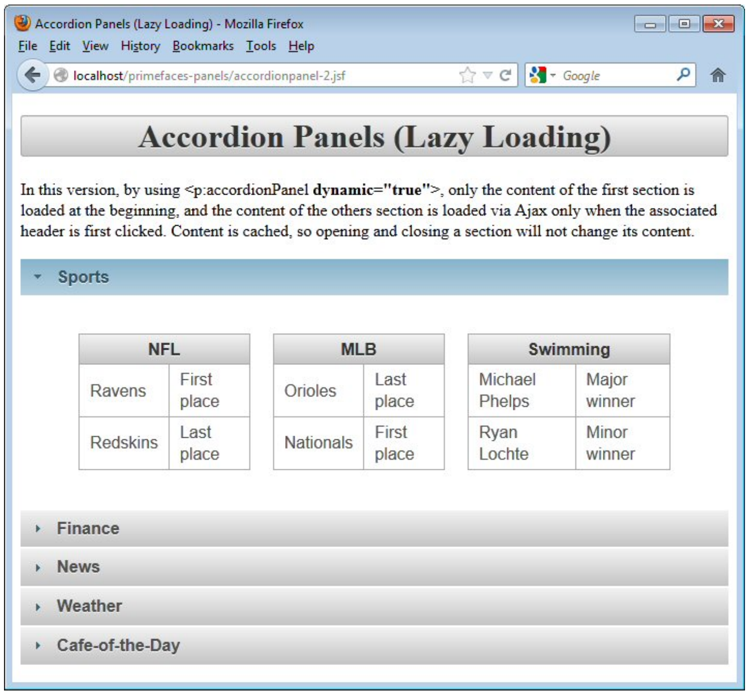
- **Most content not part of page load**
  - Content for other tabs is not part of initial page load.
- **Google and other search engines**
  - *Will* index content outside the accordion panel
  - *Will* index content of the initially-open tab (normally the first, but can be changed with activeIndex)
  - *Will not* index the content of the remaining tabs
    - This usually does not matter for intranet applications, but even there, you might have an internal page indexer, so be aware that it will not see the content of the other tabs

Main Page

```
<p:accordionPanel dynamic="true">
<p:tab title="Sports">
  <ui:include src="/snippets/sports.xhtml"/>
</p:tab>
<p:tab title="Finance">
  <ui:include src="/snippets/finance.xhtml"/>
</p:tab>
<p:tab title="News">
  <ui:include src="/snippets/news.xhtml"/>
</p:tab>
<p:tab title="Weather">
  <ui:include src="/snippets/weather.xhtml"/>
</p:tab>
<p:tab title="Cafe-of-the-Day">
  <ui:include src="/snippets/cafe.xhtml"/>
</p:tab>
</p:accordionPanel>
```

Except for dynamic="true", identical to previous example.

Result (Initial)


Accordion Panels (Lazy Loading) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/accordionpanel-2.jsf
Google

Accordion Panels (Lazy Loading)

In this version, by using <p:accordionPanel **dynamic="true"**>, only the content of the first section is loaded at the beginning, and the content of the others section is loaded via Ajax only when the associated header is first clicked. Content is cached, so opening and closing a section will not change its content.

Sports

NFL	
Ravens	First place
Redskins	Last place

MLB	
Orioles	Last place
Nationals	First place

Swimming	
Michael Phelps	Major winner
Ryan Lochte	Minor winner

Finance

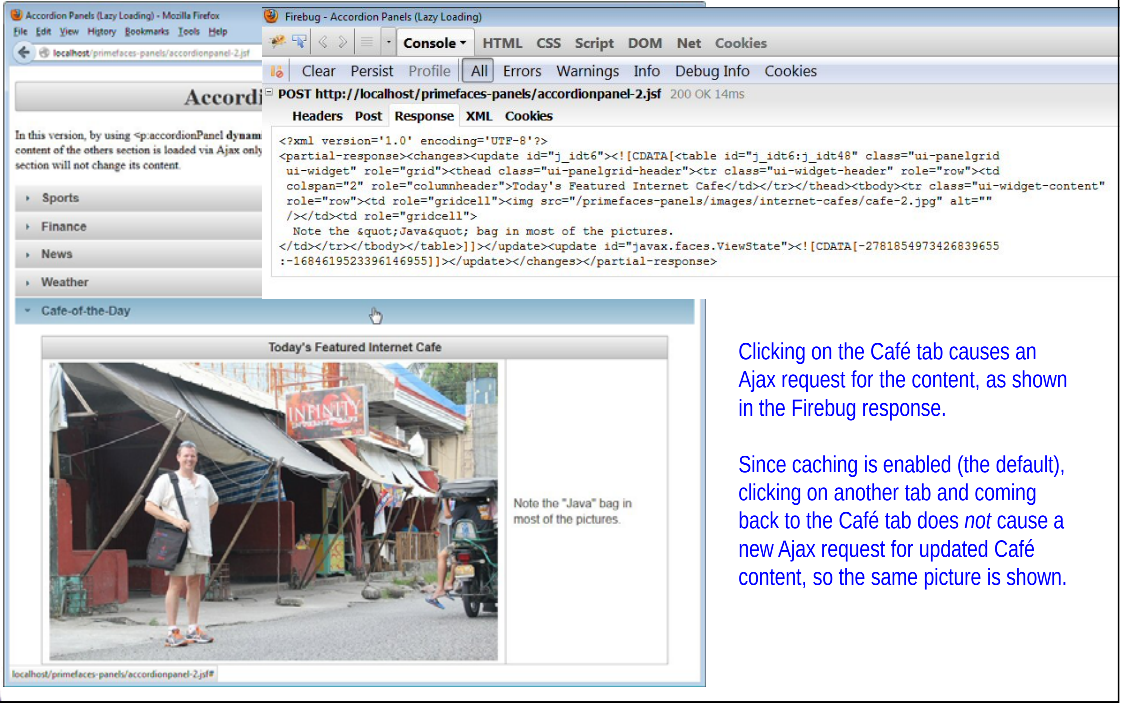
News

Weather

Cafe-of-the-Day

Appearance is the same as with the last example, and the end user is unlikely to notice any difference except that the page loads faster, especially if there are a large number of tabs or if the other tabs contain large amounts of data. You can view source to verify that the content of the Finance, News, Weather, and Café-of-the-Day tabs have not yet been loaded.

Result (Café Tab)


Accordion Panels (Lazy Loading) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/accordionpanel-2.jsf
Accordi
Firebug - Accordion Panels (Lazy Loading)
Console HTML CSS Script DOM Net Cookies
Clear Persist Profile All Errors Warnings Info Debug Info Cookies
Headers Post Response XML Cookies
localhost/primefaces-panels/accordionpanel-2.jsf#

In this version, by using <p:accordionPanel **dynami**
content of the others section is loaded via Ajax only
section will not change its content.

Sports

Finance

News

Weather

Cafe-of-the-Day

POST http://localhost/primefaces-panels/accordionpanel-2.jsf 200 OK 14ms

```
<?xml version='1.0' encoding='UTF-8'?>
<partial-response><changes><update id="j_idt6"><![CDATA[<table id="j_idt6:j_idt48" class="ui-panelgrid
 ui-widget" role="grid"><thead class="ui-panelgrid-header"><tr class="ui-widget-header" role="row"><td
 colspan="2" role="columnheader">Today's Featured Internet Cafe</td></tr></thead><tbody><tr class="ui-widget-content"
 role="row"><td role="gridcell"><img src="/primefaces-panels/images/internet-cafes/cafe-2.jpg" alt=""
 /></td><td role="gridcell">
  Note the &quot;Java&quot; bag in most of the pictures.
</td></tr></tbody></table>]]></update><update id="javax.faces.ViewState"><![CDATA[-2781854973426839655
:-1684619523396146955]]></update></changes></partial-response>
```

Today's Featured Internet Cafe	
	Note the "Java" bag in most of the pictures.

Clicking on the Café tab causes an Ajax request for the content, as shown in the Firebug response.

Since caching is enabled (the default), clicking on another tab and coming back to the Café tab does *not* cause a new Ajax request for updated Café content, so the same picture is shown.

© 2015 Marty Hall

p:accordionPanel – Dynamic Ajax Content

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

Idea

- **dynamic="true" with cache="false"**
  - Designate dynamic loading with

```
<p:accordionPanel dynamic="true" cache="false">
…
</p:accordionPanel>
```

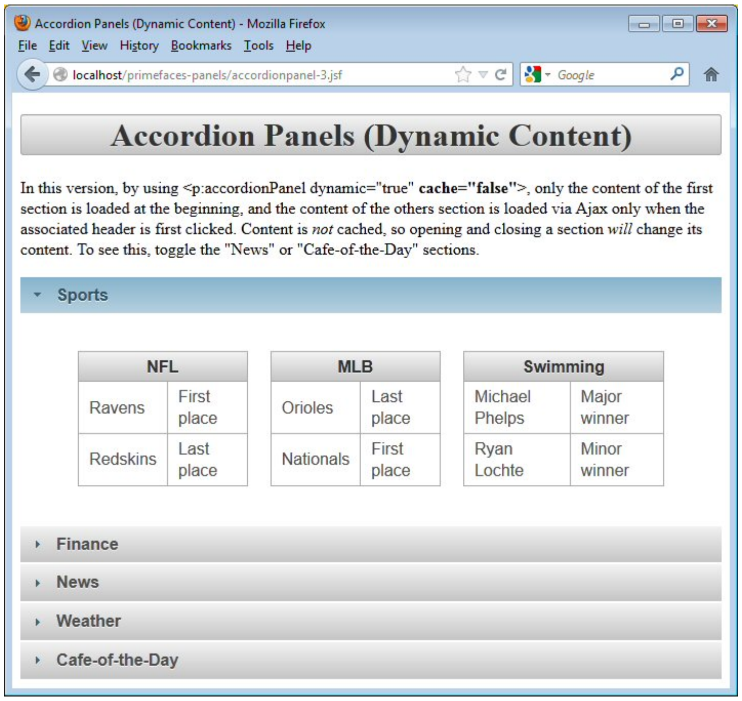
  - This means that only the content for the initially open tab is loaded when the main page is loaded. Content for other tabs is loaded via Ajax *every time* user clicks on the tab.
    - Result 1: faster initial page load, esp. with many tabs
    - Result 2: closing and reopening a tab will reload content from server, potentially changing tab content
- **SEO warning**
  - Same as before: content of other tabs won't be indexed

Main Page

```
<p:accordionPanel dynamic="true" cache="false">
<p:tab title="Sports">
  <ui:include src="/snippets/sports.xhtml"/>
</p:tab>
<p:tab title="Finance">
  <ui:include src="/snippets/finance.xhtml"/>
</p:tab>
<p:tab title="News">
  <ui:include src="/snippets/news.xhtml"/>
</p:tab>
<p:tab title="Weather">
  <ui:include src="/snippets/weather.xhtml"/>
</p:tab>
<p:tab title="Cafe-of-the-Day">
  <ui:include src="/snippets/cafe.xhtml"/>
</p:tab>
</p:accordionPanel>
```

Except for cache="false", identical to previous example.

Result (Initial)


Accordion Panels (Dynamic Content) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/accordionpanel-3.jsf
Google

Accordion Panels (Dynamic Content)

In this version, by using <p:accordionPanel dynamic="true" **cache="false"**>, only the content of the first section is loaded at the beginning, and the content of the others section is loaded via Ajax only when the associated header is first clicked. Content is *not* cached, so opening and closing a section *will* change its content. To see this, toggle the "News" or "Cafe-of-the-Day" sections.

Sports

NFL	
Ravens	First place
Redskins	Last place

MLB	
Orioles	Last place
Nationals	First place

Swimming	
Michael Phelps	Major winner
Ryan Lochte	Minor winner

Finance

News

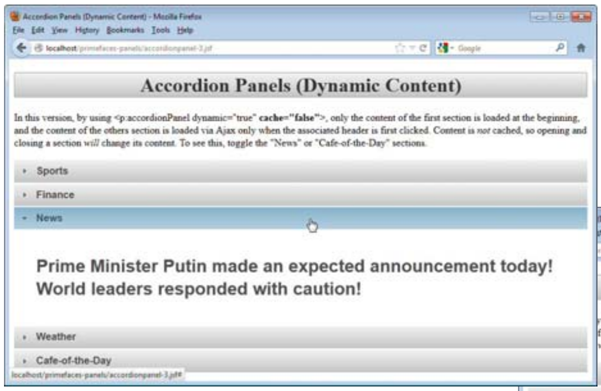
Weather

Cafe-of-the-Day

Initial appearance and behavior are the same as with the last example: only the content for the initial tab is part of the page load.

The difference is if you click on a tab more than once: an Ajax request will be made every time.

Result (News Tab)


Accordion Panels (Dynamic Content) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/accordionpanel-3.jsf

Accordion Panels (Dynamic Content)

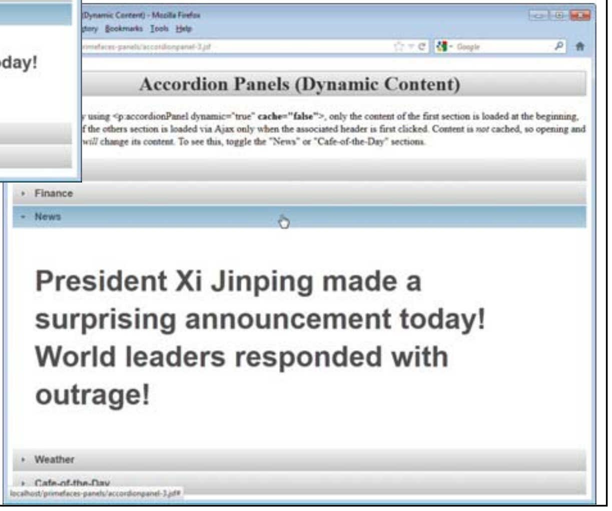
In this version, by using <p:accordionPanel dynamic="true" **cache="false"**>, only the content of the first section is loaded at the beginning, and the content of the others section is loaded via Ajax only when the associated header is first clicked. Content is *not* cached, so opening and closing a section *will* change its content. To see this, toggle the "News" or "Cafe-of-the-Day" sections.

- Sports
- Finance
- News

Prime Minister Putin made an expected announcement today! World leaders responded with caution!

- Weather
- Cafe-of-the-Day

Every time News tab is opened, a new Ajax request is made. Since the underlying bean property has different output for each request, the content of News tab changes each time it is reopened.



Accordion Panels (Dynamic Content)

- Finance
- News

President Xi Jinping made a surprising announcement today! World leaders responded with outrage!

- Weather
- Cafe-of-the-Day

The names, adjectives, and font sizes are selected randomly. See NewsBean in the initial Accordion Panel section earlier.

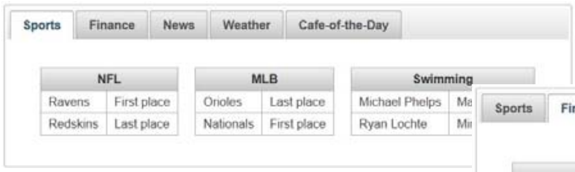
© 2015 Marty Hall

p:tabView – Basics

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

p:tabView: Overview

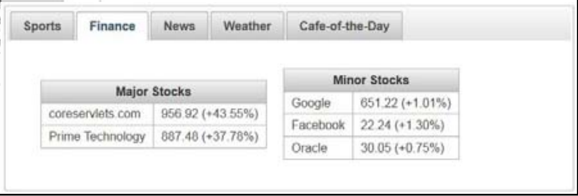
- **Appearance and behavior**
  - Wrapper for the jQuery UI tabbed panel
    - Note the minor inconsistency in names: it is called tabView, not tabPanel or tabbedPanel.
- **Purpose: for displaying a section at a time of larger content**
  - Accordion panels and tabbed panels serve the same general purpose: the tabs are just arranged differently



NFL	
Ravens	First place
Redskins	Last place

MLB	
Orioles	Last place
Nationals	First place

Swimming	
Michael Phelps	Ma
Ryan Lochte	Mi



Major Stocks	
coreservlets.com	956.92 (+43.55%)
Prime Technology	887.48 (+37.78%)

Minor Stocks	
Google	651.22 (+1.01%)
Facebook	22.24 (+1.30%)
Oracle	30.05 (+0.75%)

p:tabView: Basic Usage

```
<p:tabView>
<p:tab title="Title for First Tab">
    Arbitrary JSF and HTML content
</p:tab>
<p:tab title="Title for Second Tab">
    Arbitrary JSF and HTML content
</p:tab>
…
</p:tabView>
```

Except for the name of the main container element (p:tabView instead of p:accordionPanel), the usage is identical to that of p:accordionPanel.

p:tabView: Summary of Most Important Attributes

- **<p:tabView …/>…</p:tabView>**
  - activeIndex, dynamic, cache
    - Same usage as with p:accordionPanel
  - orientation (top [default], bottom, left, right)
    - Location of the tabs relative to the content
  - effect (*animation name*)
    - The jQuery UI animation that should be used to display the tab content when a tab is clicked. For details, see the tutorial section on Date Input.

Example: User Interest Page

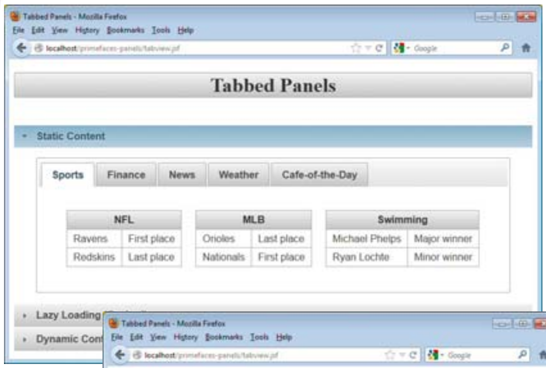
- **Page with headings (same as with p:accordionPanel)**
  - Sports
    - Information about sports scores, with data taken from managed beans.
  - Finance
    - Stock values. Data from beans.
  - News
    - Latest headlines. Data from beans. Changes on each request.
  - Weather
    - Weather forecast. Static data.
  - Café-of-the-Day
    - A picture of an internet café. Data from beans. Changes on each request.
- **Uses ui:include**
  - Same sections are used here as with p:accordionPanel. So, content is put in separate file and inserted into p:tabView sections with ui:include.
- **Has surrounding accordion panel**
  - The three tabbed panel examples are in different accordion panel sections, rather than being in separate files.

Main Page

```
<p:tabView>
<p:tab title="Sports">
  <ui:include src="/snippets/sports.xhtml"/>
</p:tab>
<p:tab title="Finance">
  <ui:include src="/snippets/finance.xhtml"/>
</p:tab>
<p:tab title="News">
  <ui:include src="/snippets/news.xhtml"/>
</p:tab>
<p:tab title="Weather">
  <ui:include src="/snippets/weather.xhtml"/>
</p:tab>
<p:tab title="Cafe-of-the-Day">
  <ui:include src="/snippets/cafe.xhtml"/>
</p:tab>
</p:tabView>
```

Except for changing the name of the main container element from p:accordionPanel to p:tabView, the code is identical to the first p:accordionPanel example.

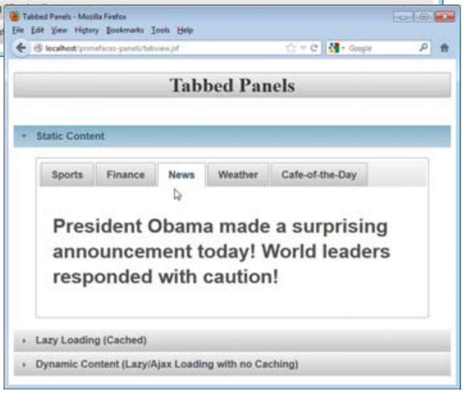
Results


Tabbed Panels - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/tabview.jsf
Tabbed Panels
Static Content
Sports
Finance
News
Weather
Cafe-of-the-Day
Lazy Loading
Dynamic Con

NFL	
Ravens	First place
Redskins	Last place

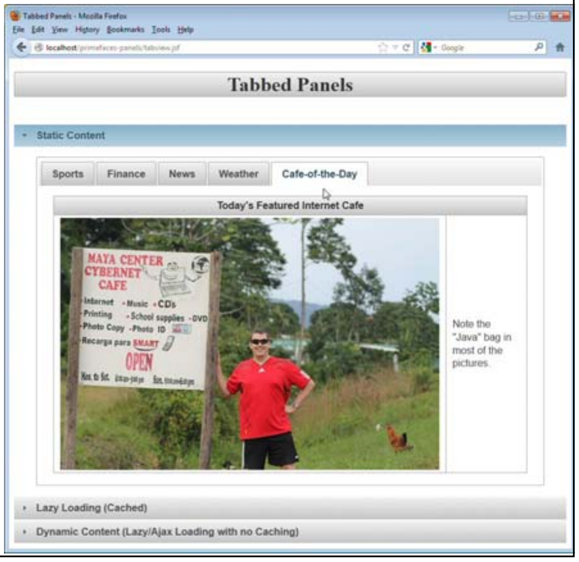
MLB	
Orioles	Last place
Nationals	First place

Swimming	
Michael Phelps	Major winner
Ryan Lochte	Minor winner


Tabbed Panels - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/tabview.jsf
Tabbed Panels
Static Content
Sports
Finance
News
Weather
Cafe-of-the-Day
Lazy Loading (Cached)
Dynamic Content (Lazy/Ajax Loading with no Caching)

President Obama made a surprising announcement today! World leaders responded with caution!

Content of all tabs is loaded on initial page request, so switching back to a previously open tab never changes its content.


Tabbed Panels - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/tabview.jsf
Tabbed Panels
Static Content
Sports
Finance
News
Weather
Cafe-of-the-Day
MAYA CENTER CYBERNET CAFE
OPEN
Lazy Loading (Cached)
Dynamic Content (Lazy/Ajax Loading with no Caching)

Today's Featured Internet Cafe

Note the "Java" bag in most of the pictures.

© 2015 Marty Hall

p:tabView – Lazy Loading

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

Idea (Same as with p:accordionPanel)

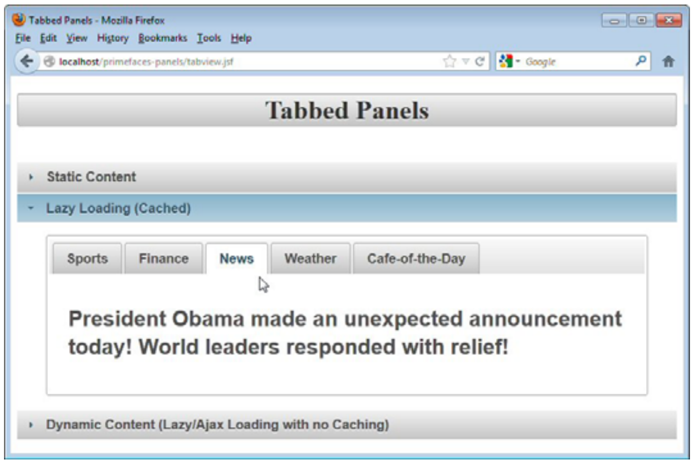
- **dynamic="true"**
  - Designate lazy loading with
    `<p:tabView dynamic="true">…</p:tabView>`
  - This has exactly the same behavior as with accordion panels. And, remember that this applies only to GET requests, not to pages that are the results of POST
- **SEO warning**
  - Since content for other tabs is not part of initial page load, Google and other search engines will index only the content of the tab that is open initially

Main Page

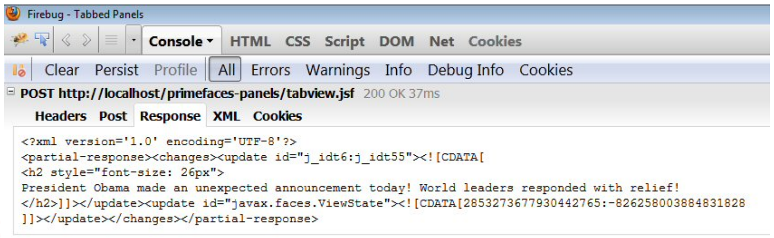
```
<p:tabView dynamic="true">
<p:tab title="Sports">
  <ui:include src="/snippets/sports.xhtml"/>
</p:tab>
<p:tab title="Finance">
  <ui:include src="/snippets/finance.xhtml"/>
</p:tab>
<p:tab title="News">
  <ui:include src="/snippets/news.xhtml"/>
</p:tab>
<p:tab title="Weather">
  <ui:include src="/snippets/weather.xhtml"/>
</p:tab>
<p:tab title="Cafe-of-the-Day">
  <ui:include src="/snippets/cafe.xhtml"/>
</p:tab>
</p:tabView>
```

Except for dynamic="true", identical to previous example.

Result (News Tab)


Tabbed Panels - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/tabview.jsf
Google
Tabbed Panels
Static Content
Lazy Loading (Cached)
Sports
Finance
News
Weather
Cafe-of-the-Day
Dynamic Content (Lazy/Ajax Loading with no Caching)

President Obama made an unexpected announcement today! World leaders responded with relief!



POST http://localhost/primefaces-panels/tabview.jsf 200 OK 37ms

```
<?xml version='1.0' encoding='UTF-8'?>
<partial-response><changes><update id="j_idt6:j_idt55"><![CDATA[
<h2 style="font-size: 26px">
President Obama made an unexpected announcement today! World leaders responded with relief!
</h2>]]></update><update id="javax.faces.ViewState"><![CDATA[2853273677930442765:-8262580038848 31828
]]></update></changes></partial-response>
```

Clicking on the News tab causes an Ajax request for the content, as shown in the Firebug response.

Since caching is enabled (the default), clicking on another tab and coming back to the News tab does *not* cause a new Ajax request for updated News content, so the same text is shown.

© 2015 Marty Hall

p:tabView – Dynamic Ajax Content

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

Idea (Same as with p:accordionPanel)

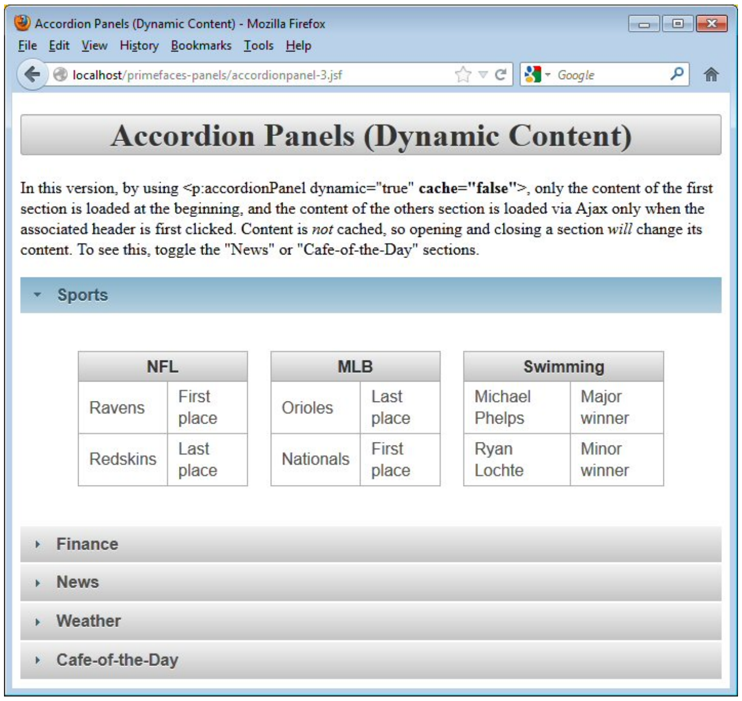
- **dynamic="true" with cache="false"**
  - Designate dynamic loading with
    ```
    <p:tabView dynamic="true" cache="false">
    …
    </p:tabView>
    ```
  - This means that only the content for the initially open tab is loaded when the main page is loaded. Content for other tabs is loaded via Ajax *every time* user clicks on the tab.
    - Result 1: faster initial page load, esp. with many tabs
    - Result 2: closing and reopening a tab will reload content from server, potentially changing tab content
- **SEO warning**
  - Same as before: content of other tabs won't be indexed

Main Page

```
<p:tabView dynamic="true" cache="false">
<p:tab title="Sports">
  <ui:include src="/snippets/sports.xhtml"/>
</p:tab>
<p:tab title="Finance">
  <ui:include src="/snippets/finance.xhtml"/>
</p:tab>
<p:tab title="News">
  <ui:include src="/snippets/news.xhtml"/>
</p:tab>
<p:tab title="Weather">
  <ui:include src="/snippets/weather.xhtml"/>
</p:tab>
<p:tab title="Cafe-of-the-Day">
  <ui:include src="/snippets/cafe.xhtml"/>
</p:tab>
</p:tabView>
```

Except for cache="false", identical to previous example.

Result (Initial)


Accordion Panels (Dynamic Content) - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/accordionpanel-3.jsf
Google

Accordion Panels (Dynamic Content)

In this version, by using <p:accordionPanel dynamic="true" **cache="false"**>, only the content of the first section is loaded at the beginning, and the content of the others section is loaded via Ajax only when the associated header is first clicked. Content is *not* cached, so opening and closing a section *will* change its content. To see this, toggle the "News" or "Cafe-of-the-Day" sections.

Sports

NFL	
Ravens	First place
Redskins	Last place

MLB	
Orioles	Last place
Nationals	First place

Swimming	
Michael Phelps	Major winner
Ryan Lochte	Minor winner

Finance

News

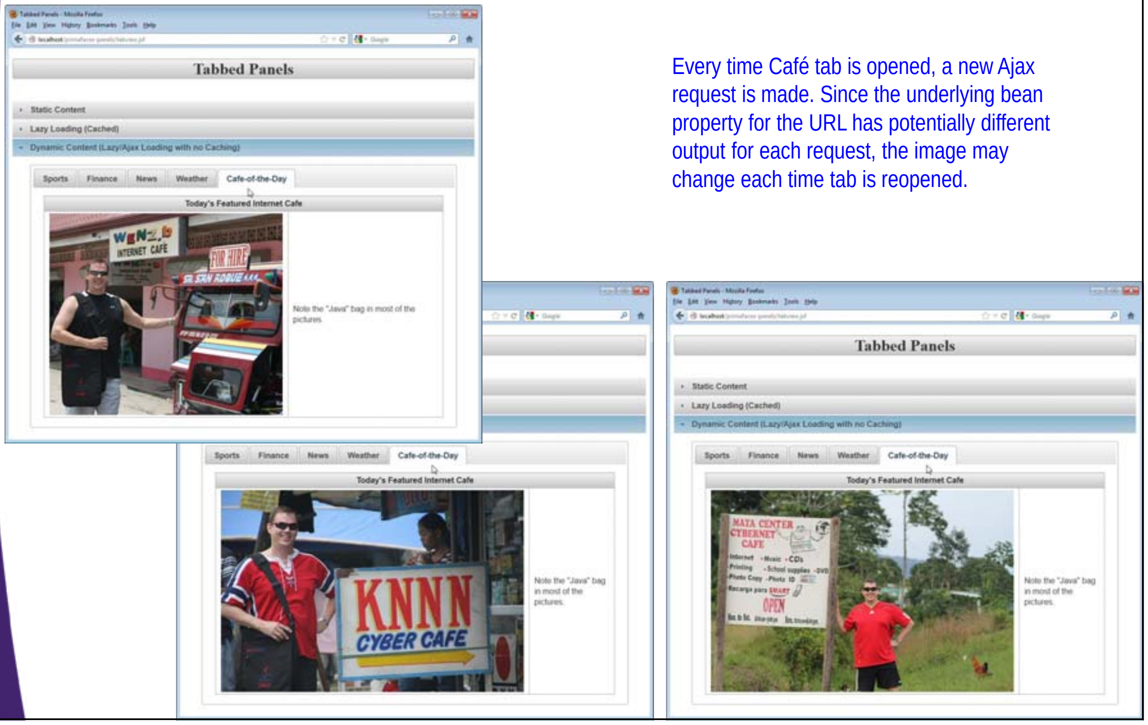
Weather

Cafe-of-the-Day

Initial appearance and behavior are the same as with the last example: only the content for the initial tab is part of the page load.

The difference is if you click on a tab more than once: an Ajax request will be made <u>every</u> time.

Result (Café Tab)


Tabbed Panels
Static Content
Lazy Loading (Cached)
Dynamic Content (Lazy/Ajax Loading with no Caching)
Sports
Finance
News
Weather
Cafe-of-the-Day
Today's Featured Internet Cafe
Note the "Java" bag in most of the pictures.

Every time Café tab is opened, a new Ajax request is made. Since the underlying bean property for the URL has potentially different output for each request, the image may change each time tab is reopened.

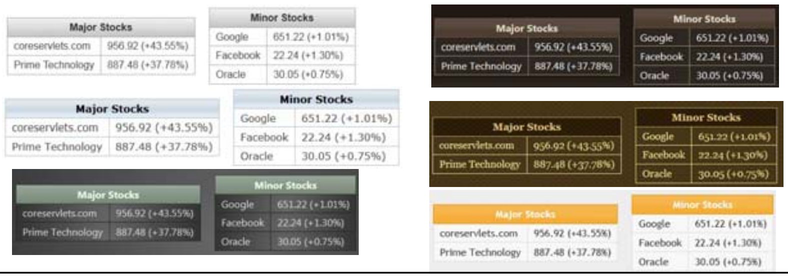
© 2015 Marty Hall

p:panelGrid

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

p:panelGrid: Overview

- **Appearance and behavior**
  - Theme-aware replacement for h:panelGrid. Shortcut for making HTML tables.
- **Purpose: for displaying tables that have visible borders and/or headings**
  - If you want themed headers and borders, use p:panelGrid
  - If you are using tables only for alignment (with no headings or borders), use h:panelGrid



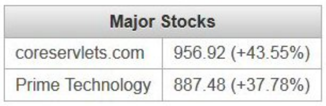
Major Stocks	
coreservlets.com	956.92 (+43.55%)
Prime Technology	887.48 (+37.78%)

Minor Stocks	
Google	651.22 (+1.01%)
Facebook	22.24 (+1.30%)
Oracle	30.05 (+0.75%)

p:panelGrid: Basic Usage

- **Basics are same as h:panelGrid**
  - <p:panelGrid columns="*x*">
    …
    </p:panelGrid>
  - Each element is placed in separate cell, left-to-right, top-to-bottom, based on the number of columns.
    - Main difference is that borders are on by default, with colors and thickness based on current theme
    - Also has “header” facet for heading that spans table

```
<p:panelGrid columns="2"  >
  <f:facet name="header">Major Stocks</f:facet>
  coreservlets.com
  <h:outputText value="#{financeBean.coreservlets}"/>
  Prime Technology
  <h:outputText value="#{financeBean.prime}"/>
</p:panelGrid>
```



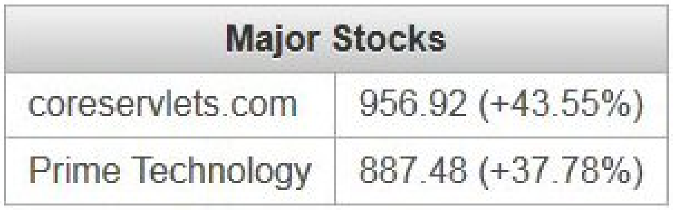
Major Stocks	
coreservlets.com	956.92 (+43.55%)
Prime Technology	887.48 (+37.78%)

Differences from h:panelGrid

- **Not for invisible alignment**
  - Borders are on by default, and color/font of table text may be different from surrounding default
- **Headers and footers**
  - Facets named “header” and “footer”
- **p:row & p:column instead of h:panelGroup**
  - For grouping multiple elements in a single cell
- **Column and row spanning elements**
  - p:column has “rowspan” and “colspan” options
- **Missing pass-through attributes**
  - h:panelGrid has bgcolor, border, cellpadding, cellspacing, etc. p:panelGrid does not.
    - Either use theme defaults or use CSS styles set with style, styleClass, or via the .ui-panelgrid selector.

Example: Basics

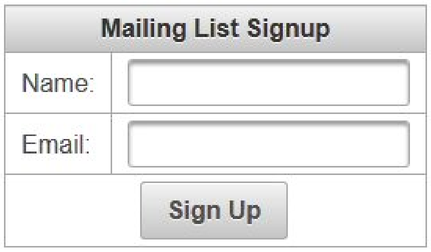
```
<p:panelGrid columns="2"  >
  <f:facet name="header">Major Stocks</f:facet>
  coreservlets.com
  <h:outputText value="#{financeBean.coreservlets}"/>
  Prime Technology
  <h:outputText value="#{financeBean.prime}"/>
</p:panelGrid>
```



Major Stocks	
coreservlets.com	956.92 (+43.55%)
Prime Technology	887.48 (+37.78%)

Example: colspan

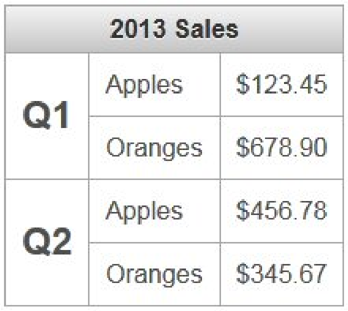
```
<p:panelGrid>
  <f:facet name="header">
    <p:row>
      <p:column colspan="2">Mailing List Signup</p:column>
    </p:row>
  </f:facet>
  <p:row>
    <p:column>Name:</p:column>
    <p:column><p:inputText/></p:column>
  </p:row>
  <p:row>
    <p:column>Email:</p:column>
    <p:column><p:inputText/></p:column>
  </p:row>
  <p:row>
    <p:column colspan="2" style="text-align: center">
      <p:commandButton value="Sign Up"/>
    </p:column>
  </p:row>
</p:panelGrid>
```



Mailing List Signup	
Name:	
Email:	
Sign Up	

Example: rowspan

```
<p:panelGrid>
  <f:facet name="header">
    <p:row>
      <p:column colspan="3">2013 Sales</p:column>
    </p:row>
  </f:facet>
  <p:row>
    <p:column rowspan="2"><h2>Q1</h2></p:column>
    <p:column>Apples</p:column>
    <p:column>$123.45</p:column>
  </p:row>
  <p:row>
    <p:column>Oranges</p:column>
    <p:column>$678.90</p:column>
  </p:row>
  <p:row>
    <p:column rowspan="2"><h2>Q2</h2></p:column>
    <p:column>Apples</p:column>
    <p:column>$456.78</p:column>
  </p:row>
  <p:row>
    <p:column>Oranges</p:column>
    <p:column>$345.67</p:column>
  </p:row>
</p:panelGrid>
```



2013 Sales		
Q1	Apples	$123.45
	Oranges	$678.90
Q2	Apples	$456.78
	Oranges	$345.67

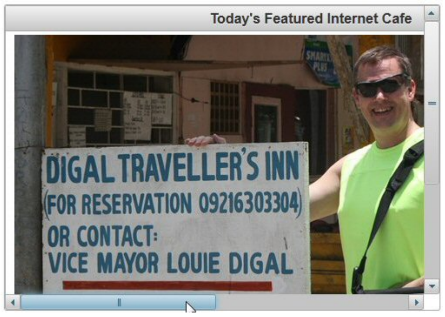
© 2015 Marty Hall

p:scrollPanel

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

p:scrollPanel: Overview

- **Appearance and behavior**
  – Container with theme-aware scrollbars
- **Purpose: for displaying part of a large region and letting user scroll**
  – Similar to using CSS overflow: auto, but the scrollbars adapt to the current theme (very subtle in most themes)



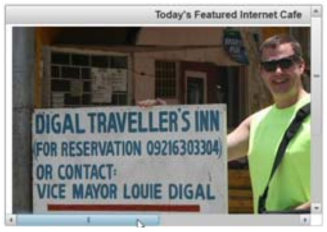
p:scrollPanel: Summary of Most Important Attributes

- **<p:scrollPanel …/>…</p:scrollPanel>**
  - style, styleClass
    - You <u>must</u> give a width and a height, usually with style, but sometimes with styleClass
      - <p:scrollPanel style="width: 250px; height: 200px">
        …
      - </p:scrollPanel>
  - mode (default [default] or native)
    - Whether scroll panel should use theme-aware scrollbars (default) or native browser scrollbars (native).
      - It you set mode="native", there is less reason to use p:scrollPanel instead of normal CSS with "overflow: scroll" or "overflow: auto".

Examples

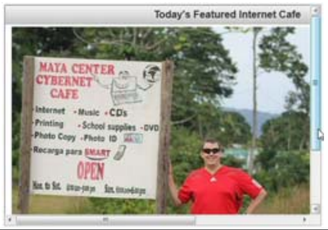
- **Default (theme-aware) scrollbars**

```
<p:scrollPanel style="width: 500px; height: 350px">
    <ui:include src="/snippets/cafe.xhtml"/>
</p:scrollPanel>
```



- **Native scrollbars**

```
<p:scrollPanel mode="native"
               style="width: 500px; height: 350px">
    <ui:include src="/snippets/cafe.xhtml"/>
</p:scrollPanel>
```



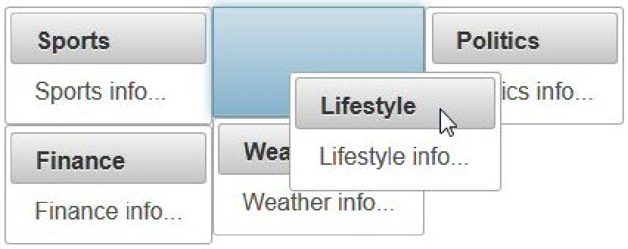
© 2015 Marty Hall

p:dashboard – Basics

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

p:dashboard: Overview

- **Appearance and behavior**
  – Panels that can be dragged and reordered
- **Purpose: for letting user customize layout**
  – Users can move their preferred panels to more prominent positions. If you make the bean Serializable and session-scoped, changes will persist through session.



p:dashboard: Basic Usage (Facelets)

<p:dashboard model="#{someBean.model}">

<p:panel id="id1" header="Header 1">

Arbitrary JSF and HTML content

</p:panel>

<p:panel id="id2" header="Header 2">

Arbitrary JSF and HTML content

</p:panel>

…

</p:dashboard>

p:dashboard: Basic Usage (Bean)

- **Make a DefaultDashboardModel**
  - With getter method that points to the model
- **Make some DefaultDashboardColumns**
  - One DefaultDashboardColumn for each column you want in the display
- **Add panels to the columns**
  - column1.addWidget("some-id");
    - Supply the JSF id of the panel. Added from top down. You are allowed blank spaces at bottom of columns.
- **Add the columns to the model**
  - model.addColumn(column1);

p:dashboard: Sample Bean Outline

```
@SessionScoped
@ManagedBean
public class SomeBean implements Serializable {
  private DashboardModel model;

  public DashboardBean() {
    model = new DefaultDashboardModel();
    DashboardColumn column1 = new DefaultDashboardColumn();
    DashboardColumn column2 = new DefaultDashboardColumn();
    DashboardColumn column3 = new DefaultDashboardColumn();
    column1.addWidget("id1");   // Matches id, not header
    column1.addWidget("id2");
    ...
    column3.addWidget(...);
    model.addColumn(column1);
    model.addColumn(column2);
    model.addColumn(column3);
  }

  public DashboardModel getModel() {
    return(model);
  }
```

Example: Reordarable User Interest Page

- **Page with panels (simple placeholder content)**
  - Sports
  - Finance
  - Lifestyle
  - Weather
  - Politics
- **Uses ui:include**
  - Same panels are used in next example (dashboard with Ajax reorder listener). So, panels are put in separate file and inserted into dashboard with ui:include.

Main Page

```
<p:dashboard model="#{dashboardBean.model}">
  <ui:include src="/snippets/dashboard-panels.xhtml"/>
</p:dashboard>
```

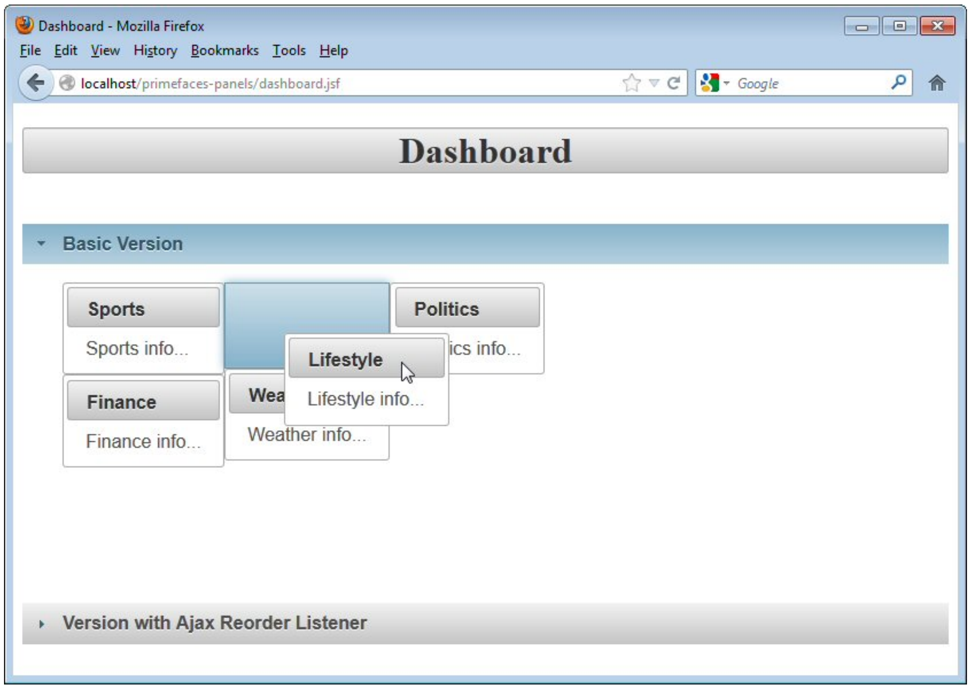
Panels

```
<ui:composition xmlns="http://www.w3.org/1999/xhtml"
                xmlns:ui="http://java.sun.com/jsf/facelets"
                xmlns:p="http://primefaces.org/ui">

<p:panel id="sports" header="Sports">
  Sports info...
</p:panel>
<p:panel id="finance" header="Finance">
  Finance info...
</p:panel>
<p:panel id="lifestyle" header="Lifestyle">
  Lifestyle info...
</p:panel>
<p:panel id="weather" header="Weather">
  Weather info...
</p:panel>
<p:panel id="politics" header="Politics">
  Politics info...
</p:panel>

</ui:composition>
```

Bean

```
@SessionScoped
@ManagedBean
public class DashboardBean implements Serializable {
  private DashboardModel model;

  public DashboardBean() {
    model = new DefaultDashboardModel();
    DashboardColumn column1 = new DefaultDashboardColumn();
    DashboardColumn column2 = new DefaultDashboardColumn();
    DashboardColumn column3 = new DefaultDashboardColumn();
    column1.addWidget("sports");   // Matches id, not header
    column1.addWidget("finance");
    column2.addWidget("lifestyle");
    column2.addWidget("weather");
    column3.addWidget("politics");
    model.addColumn(column1);
    model.addColumn(column2);
    model.addColumn(column3);
  }

  public DashboardModel getModel() {
    return(model);
  }
```

Results



Dashboard - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/dashboard.jsf
Google
Dashboard
Basic Version
Sports
Sports info...
Politics
ics info...
Lifestyle
Lifestyle info...
Wea
Weather info...
Finance
Finance info...
Version with Ajax Reorder Listener

© 2015 Marty Hall

p:dashboard – Ajax Reorder Listener

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

Idea

- **p:ajax supports "reorder" event**
  - <p:ajax event="reorder" .../>
- **Can assign listener for DashboardReorderEvent**
  - <p:ajax event="reorder" listener="#{bean.blah}" .../>
  - public void blah(DashboardReorderEvent event) { ... }
- **Examining state**
  - ID of widget that was moved
    - event.getWidgetId()
  - Index of widget's original column
    - event.getSenderColumnIndex()
  - Index of widget's new column
    - event.getColumnIndex()
  - Index of widget's new row
    - event.getItemIndex()

Main Page

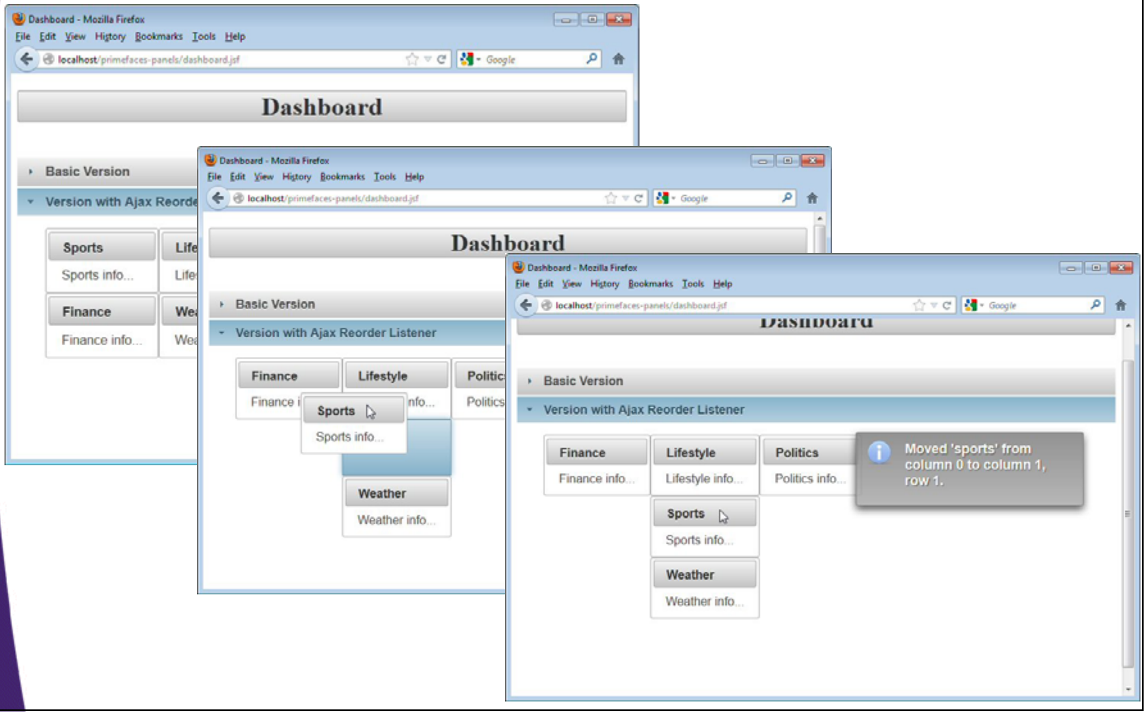
```
<h:form>
<p:dashboard model="#{dashboardBean.model}">
  <p:ajax event="reorder"
          listener="#{dashboardBean.handleReorder}"
          update="growl"/>
  <ui:include src="/snippets/dashboard-panels.xhtml"/>
</p:dashboard>
<p:growl id="growl"/>
</h:form>
```

The content of dashboard-panels.xhtml (p:panel elements with ids that match the Java code) was shown in previous example.

Bean

```
@SessionScoped
@ManagedBean
public class DashboardBean implements Serializable {
  ... // Shown in previous example

  public void handleReorder(DashboardReorderEvent event) {
    String messageTemplate =
      "Moved '%s' from column %s to column %s, row %s.";
    String widgetId = event.getWidgetId();
    Integer oldCol = event.getSenderColumnIndex();
    Integer newCol = event.getColumnIndex();
    if (oldCol == null) {
      oldCol = newCol;
    }
    Integer newRow = event.getItemIndex();
    String message =
      String.format(messageTemplate, widgetId, oldCol, newCol, newRow);
    FacesContext context = FacesContext.getCurrentInstance();
    context.addMessage(null, new FacesMessage(message));
  }
}
```

Results


Dashboard - Mozilla Firefox
File Edit View History Bookmarks Tools Help
localhost/primefaces-panels/dashboard.jsf
Dashboard
Basic Version
Version with Ajax Reorder Listener
Finance
Finance info...
Lifestyle
Lifestyle info...
Politics
Politics info...
Sports
Sports info...
Weather
Weather info...

Moved 'sports' from column 0 to column 1, row 1.

© 2015 Marty Hall

Wrap-Up

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

Other PrimeFaces Container Panels

- **p:layout**
  - Panel with behavior like a Java window that uses BorderLayout
    - http://www.primefaces.org/showcase/ui/layoutHome.jsf
- **p:outputPanel**
  - An initially invisible region that can be displayed later
    - http://www.primefaces.org/showcase/ui/outputPanel.jsf
- **p:panel**
  - Theme-aware grouping component that supports toggling, closing, and options menu. We used it without explanation in dashboard example.
    - http://www.primefaces.org/showcase/ui/panel.jsf
- **p:toolbar**
  - Horizontal grouping component, mostly for buttons
    - http://www.primefaces.org/showcase/ui/toolbar.jsf
- **p:wizard**
  - Component that breaks a single form into pieces with tabs
    - http://www.primefaces.org/showcase/ui/wizard.jsf

Summary

- **Accordion panels**

```
<p:accordionPanel>
<p:tab title="Title for First Tab">
   Arbitrary JSF and HTML content
</p:tab>
…
</p:accordionPanel>
```

  - Also has “dynamic” and “cache” options
    - dynamic="true" – lazy Ajax loading, then cached
    - dynamic="true" cache="false" – Ajax load every time
- **Tabbed panels**
  - Same basic syntax as accordion panels
    - Just replace “p:accordionPanel” with “p:tabView” above

Summary (Continued)

- **p:panelGrid**
  - Simple use: same basic syntax as h:panelGrid, but follows theme, has borders, and supports “header” and “footer” facets
  - Advanced use: use p:row and p:column, then can use “rowspan” and “colspan” attributes of p:column
  - Stick with h:panelGrid for invisible tables used for alignment
- **p:scrollPanel**
  - Requires width and height (usually with “style” attribute)
- **p:dashboard**

```
<p:dashboard model="#{someBean.model}">
<p:panel id="id1" header="Header 1">
  Arbitrary JSF and HTML content
</p:panel>
…
</p:dashboard>
```

© 2015 Marty Hall

Questions?

More info:
http://www.coreservlets.com/JSF-Tutorial/jsf2/ – JSF 2.2 tutorial
http://www.coreservlets.com/JSF-Tutorial/primefaces/ – PrimeFaces tutorial
http://courses.coreservlets.com/jsf-training.html – Customized JSF and PrimeFaces training courses
http://coreservlets.com/ – 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/
Java 7, Java 8, JSF 2, PrimeFaces, Android, JSP, Ajax, jQuery, Spring MVC, RESTful Web Services, GWT, Hadoop.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.