PrimeFaces Menus

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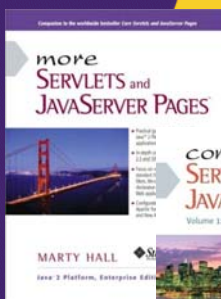
For live training on JSF 2, PrimeFaces, or other Java EE topics, email hall@coreservlets.com
Marty is also available for consulting and development support

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

- **Basic structure**
- **Displaying server results**
- **Redirecting to URLs**
- **Submenus**
- **Icons**
- **Menu options**

5

Basic Structure

Basic Menu Structure

- **Big idea**
  - Start with p:menubar, use p:submenu, then p:menuitem
    - You can also put p:menuitem directly in p:menubar
    - You can replace p:menubar with p:menu for menus that stay open, instead of ones that drop down on clicks
  - p:menuitem can give an action (which submits the form normally) or a url (which redirects to that location)
- **Quick example**

```
<p:menubar>
<p:submenu label="Menu label">
  <p:menuitem value="Item label" .../>
  <!-- More menu items -->
</p:submenu>
<!-- More submenus -->
</p:menubar>
```

Notice the inconsistency where you use "label" for menu labels, but you use "value" for menu item labels.

Related Menu Types

- **Places you can use p:menuitem**
  - p:menubar (shown in this tutorial)
  - p:menu (menu that remains open)
  - p:megaMenu (shows all submenus at once)
  - p:breadcrumb (shows page hierarchy)
  - p:dock (imitates dock from MacOS)
  - p:stack (imitates stacks from MacOS)
  - p:menuButton (a button that pops up a menu)
  - p:splitButton (a button where part of it pops up a menu)
  - p:panelMenu (uses accordion panels to display menus)
  - p:tabMenu (uses tabs to display menu choices)
  - p:slideMenu (slides menus into view)
  - p:tieredMenu (displays submenus as overlays)

Using Menus to Display Server Results

The action Attribute

- **Big idea**
  - You use action just as you would with p:commandButton. You normally also use update and process as discussed in the earlier Ajax section.
    - The menubar must be inside h:form
- **Quick example**

```
<p:menuitem value="Item label"
            action="#{someBean.someMethod}"
            process="@form"
            update="some-id"/>
...
<h:outputText value="#{someBean.someProperty"
              id="some-id"/>
```

Related Attributes

- **update**
  - The id or space-separated set of ids to update
    - Used exactly as in the Ajax lecture
- **process**
  - The id, space-separated set of ids, or special name (usually @form) to designate components to process on server
    - Used exactly as in the Ajax lecture
- **ajax**
  - As with p:commandButton, if you specify ajax="false", then the result is a normal form submission instead of an Ajax update
- **onclick, oncomplete, etc.**
  - Same as with p:commandButton.
  - Useful for popping up dialogs from menus.

Example: HTML Outline

```
<h:form>
<p:menubar>
  <!-- Entries shown on next slide -->
</p:menubar>
<p:messages autoUpdate="true"/>
Enter message:
<p:inputText value="#{messageBean.message}"
             required="true"
             requiredMessage="Message cannot be empty"/>
<h2><h:outputText value="#{messageBean.message}"
                  id="messageDisplay"/>
</h2>
</h:form>
```

Example: HTML for Menu

```
<p:submenu label="Actions">
  <p:menuitem value="Uppercase"
              action="#{messageBean.makeUpperCase}"
              process="@form" update="messageDisplay"/>
  <p:menuitem value="Lowercase"
              action="#{messageBean.makeLowerCase}"
              process="@form" update="messageDisplay"/>
  <p:menuitem value="Doubled"
              action="#{messageBean.makeDoubled}"
              process="@form" update="messageDisplay"/>
  <p:menuitem value="Reversed"
              action="#{messageBean.makeReversed}"
              process="@form" update="messageDisplay"/>
  <p:menuitem value="Show on Separate Page"
              action="#{messageBean.showSeparately}"
              ajax="false"/>
</p:submenu>
```

13

Example: Java

```
@ManagedBean
public class MessageBean {
  private String message="";

  // getMessage and setMessage

  public String makeUpperCase() {
    message = message.toUpperCase();
    return(null);
  }

  // Similar for makeLowerCase, makeDoubled,
  // and makeReversed

  public String showSeparately() {
    return("show-message");
  }
}
```
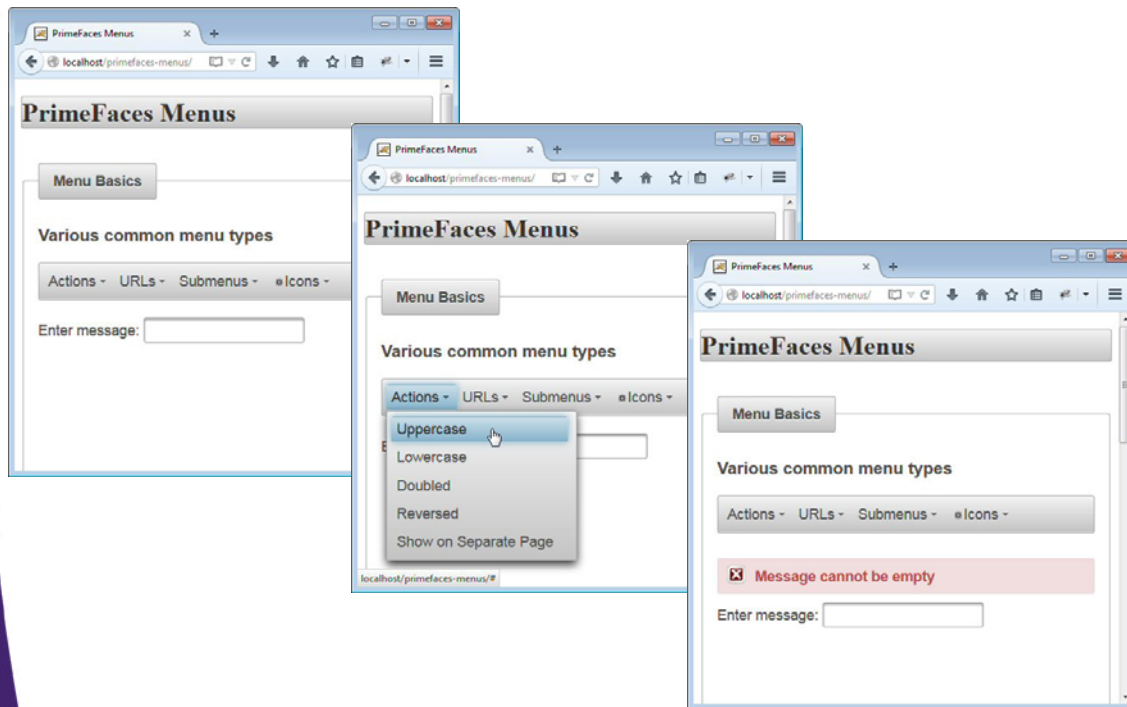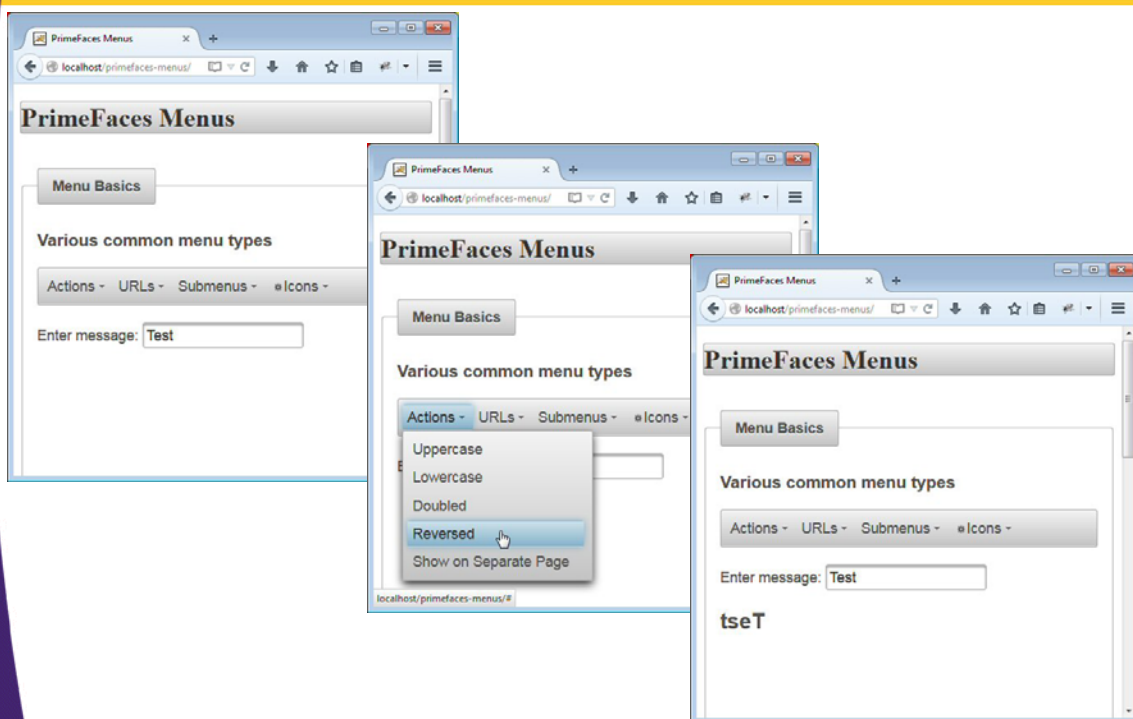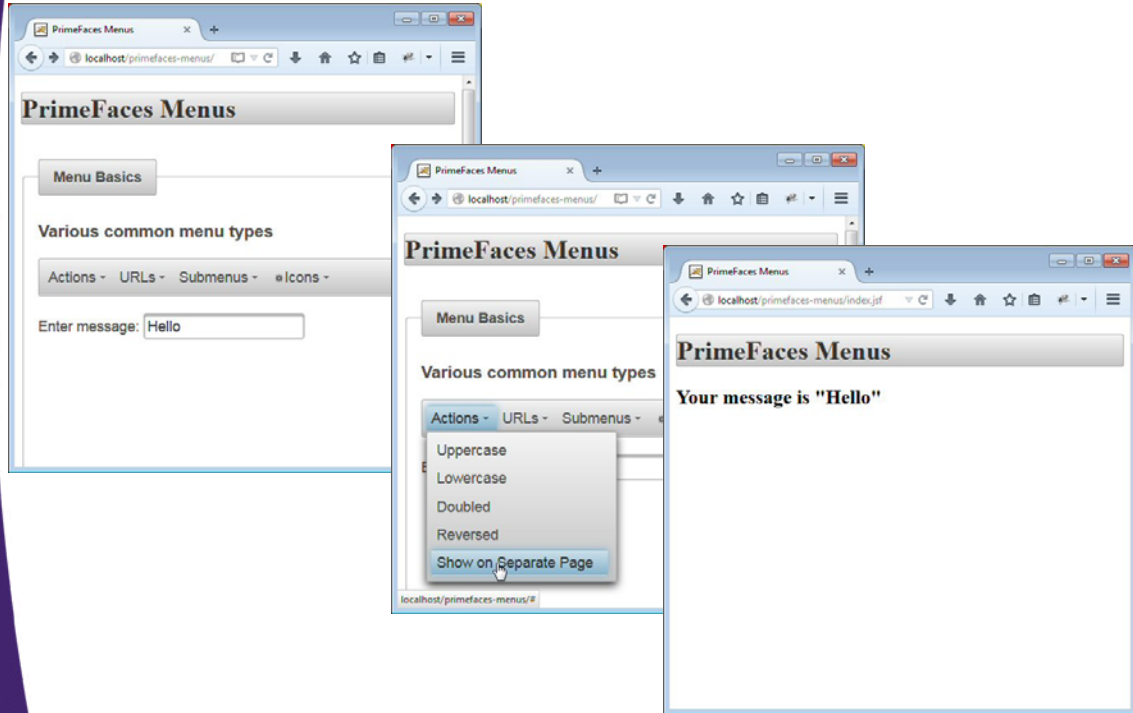
14

Example: Results (Missing Message)

Example: Results (Ajax Options)

Example: Results (Non-Ajax Option)

Using Menus to Display URLs

The url Attribute

- **Big idea**
  - You use url to designate a URL that should be displayed when the menu item is selected.
  - You can also use target to specify the frame cell (or new window with "_new")

- **Quick example**

```
<p:menuitem value="Item label"
            url="http://www.example.com/blah"/>

...
```
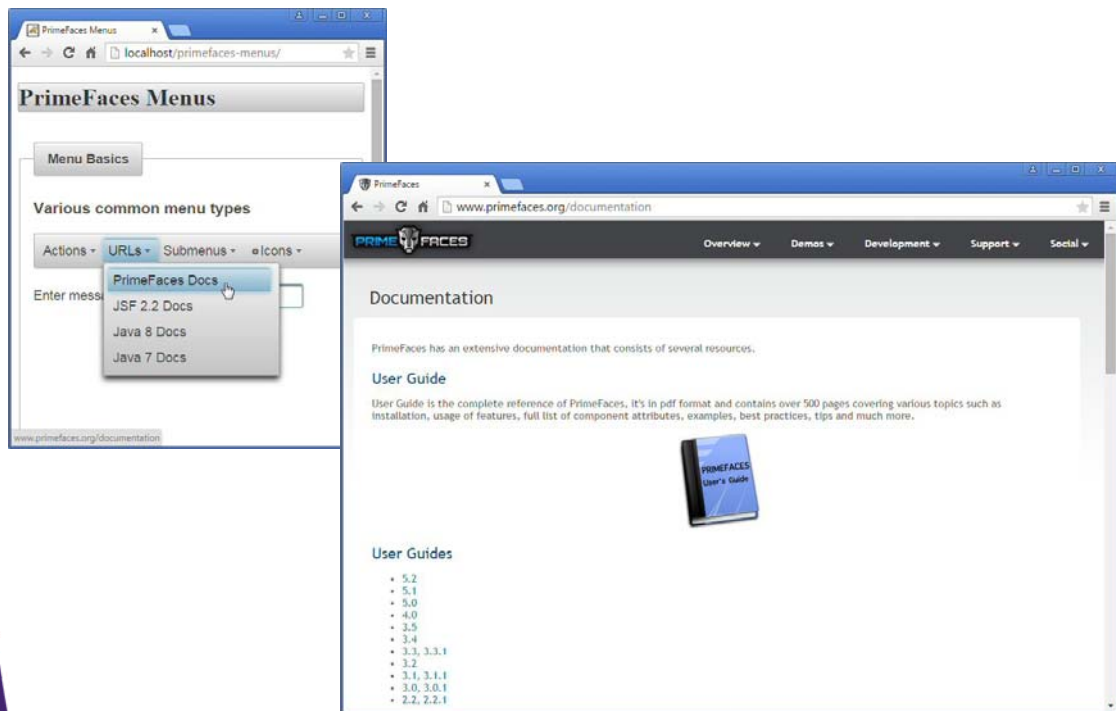
Example: HTML for Menu

```
<p:submenu label="URLs">
  <p:menuitem value="PrimeFaces Docs"
     url="http://www.primefaces.org/documentation"/>
  <p:menuitem value="JSF 2.2 Docs"
     url="https://javaserverfaces.java.net/docs/2.2/"/>
  <p:menuitem value="Java 8 Docs"
     url="http://docs.oracle.com/javase/8/docs/api/"/>
  <p:menuitem value="Java 7 Docs"
     url="http://docs.oracle.com/javase/7/docs/api/"/>
</p:submenu>
```

Example: Results

Using Nested Menus

Submenus

- **Big idea**
  - Instead of putting a p:menuitem inside p:submenu, you can put another p:submenu. You can nest further.
    - Analogously, you can use p:menuitem instead of p:submenu at the top level of p:menubar
- **Quick example**

```
<p:menubar>
<p:menuitem value="Item label"/>
<p:submenu label="Menu label">
   <p:menuitem value="Item label"/>
   <p:submenu label="Menu label">
   ...
   </p:submenu>
</p:submenu>
</p:menubar>
```

Example: HTML for Menu
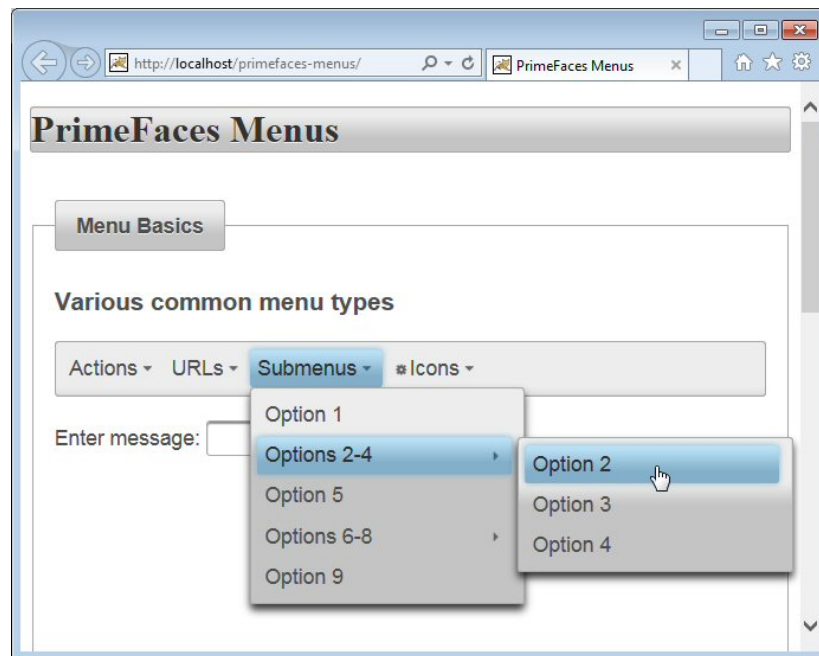
```
<p:submenu label="Submenus">
  <p:menuitem value="Option 1"/>
  <p:submenu label="Options 2-4">
    <p:menuitem value="Option 2"/>
    <p:menuitem value="Option 3"/>
    <p:menuitem value="Option 4"/>
  </p:submenu>
  <p:menuitem value="Option 5"/>
  <p:submenu label="Options 6-8">
    <p:menuitem value="Option 6"/>
    <p:submenu label="Options 7-8">
      <p:menuitem value="Option 7"/>
      <p:menuitem value="Option 8"/>
    </p:submenu>
  </p:submenu>
  <p:menuitem value="Option 9"/>
</p:submenu>
```

Example: Results

Displaying Icons

Icons

- **Big idea**
  - You can add icons to menus and menu items
    - Also to p:commandButton and many other components
    - See next page for legal icon names
  - You can separate related entries with p:separator
- **Quick example**

```
<p:submenu label="Menu label" icon="Icon name">
  <p:menuitem value="Item label" icon="Icon name"/>
  ...
  <p:separator/>
  ...
</p:submenu>
```

Legal Icon Names

- **An icon from jQuery UI**
  - Examples
    - ui-icon-help, ui-icon-disk, ui-icon-trash
  - Official list: http://api.jqueryui.com/theming/icons/
  - Unofficial list:
    http://www.petefreitag.com/cheatsheets/jqueryui-icons/
    - Easier to use, but no guarantee of accuracy in future
- **Your own icon, defined in CSS file**

```
.your-icon-name {
  background: url(my-picture.png) no-repeat;
  height:16px;
  width:16px;
}
```

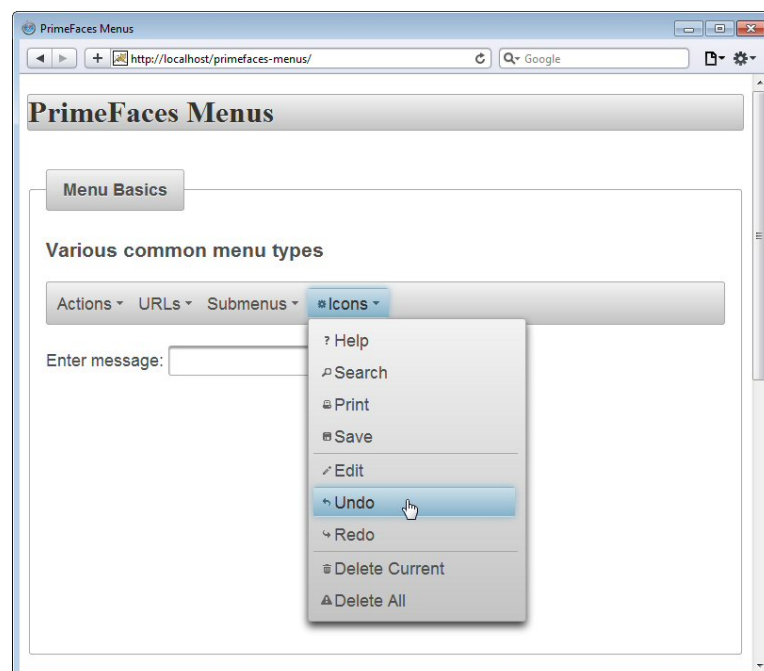# Example: HTML for Menu

```
<p:submenu label="Icons" icon="ui-icon-gear">
   <p:menuitem value="Help" icon="ui-icon-help"/>
   <p:menuitem value="Search" icon="ui-icon-search"/>
   <p:menuitem value="Print" icon="ui-icon-print"/>
   <p:menuitem value="Save" icon="ui-icon-disk"/>
   <p:separator/>
   <p:menuitem value="Edit" icon="ui-icon-pencil"/>
   <p:menuitem value="Undo"
               icon="ui-icon-arrowreturnthick-1-w"/>
   <p:menuitem value="Redo"
               icon="ui-icon-arrowreturnthick-1-e"/>
   <p:separator/>
   <p:menuitem value="Delete Current"
               icon="ui-icon-trash"/>
   <p:menuitem value="Delete All" icon="ui-icon-alert"/>
</p:submenu>
```

29

Example: Results

30

Inserting Custom Menubar Content

Icons

- **Big idea**
  - You can add arbitrary PrimeFaces components to a menubar by using a facet with the name "options".
    - Often used for form elements.

- **Quick example**
  ```
  <p:menubar>
  ...
  <f:facet name="options">
  ...
  </f:facet>
  </p:menubar>
  ```

Example: HTML

```
<h:form>
<p:menubar>

<!-- Normal submenus -->

<f:facet name="options">
   <p:inputText value="#{searchController.searchString}"
                placeholder="Search Text"
                style="margin-right:10px"/>
   <p:commandButton value="Google It"
                    icon="ui-icon-search"
                    action="#{searchController.doSearch}"
                    ajax="false"/>
</f:facet>
</p:menubar>
</h:form>
```

Example: Java

```
@ManagedBean
public class SearchController {
  private String searchString="";
  private final static String GOOGLE_URL =
    "https://www.google.com/#q=";

  // getSearchString and setSearchString

  public String doSearch() throws IOException {
    searchString = URLEncoder.encode(searchString, "utf-8");
    String searchUrl = GOOGLE_URL + searchString;
    ExternalContext context =
      FacesContext.getCurrentInstance().getExternalContext();
    HttpServletResponse response =
      (HttpServletResponse)context.getResponse();
    response.sendRedirect(searchUrl);
    return(null);
  }
}
```
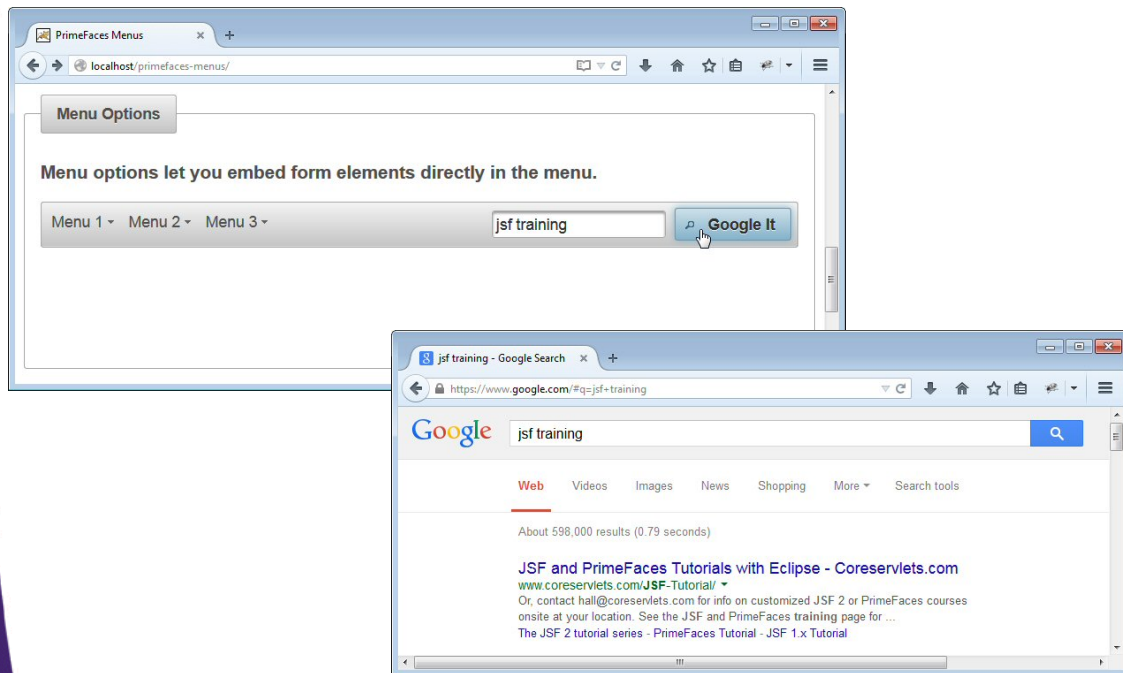
Example: Results

Wrap-Up

Summary

- ## Basic syntax

  ```
  <p:menubar>
  <p:submenu label="Menu label">
    <p:menuitem value="Item label" .../>
    <!-- More menu items -->
  </p:submenu>
  <!-- More submenus -->
  </p:menubar>
  ```

- ## Options
  - action for menuitem: does Ajax or form submission
  - url for menuitem: does redirect
  - Can nest submenus
  - Can use icons (standard or defined in CSS file)
  - Can put custom content in menubar with "options" facet

Questions?

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.