JavaServer Faces 2.0

Implementing Ajax components

David Geary

Clarity Training, Inc.

David Geary

Struts　Apache **Tiles**　*JSF*　**Shale** APACHE PROJECT

JavaOne

NFJS No Fluff Just Stuff.com

JSF·one
The One Event Exclusively
Focused on the JSF Ecosystem
Sept. 4-6, Vienna, VA
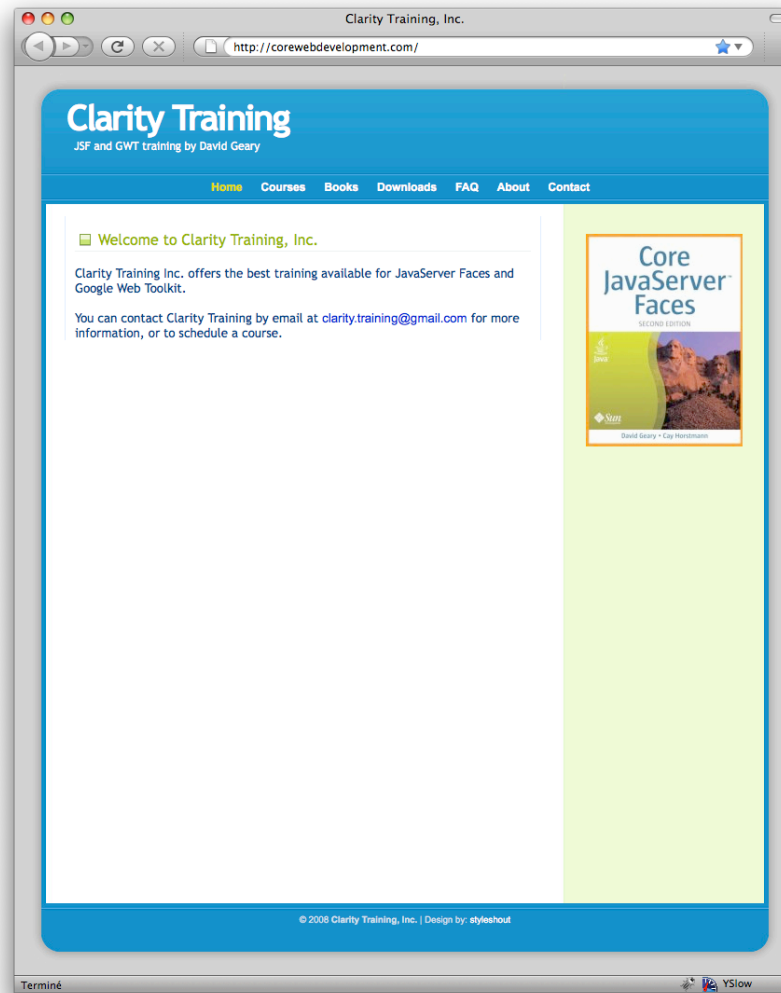JSFCentral NFJS
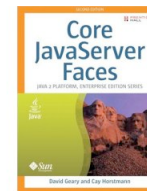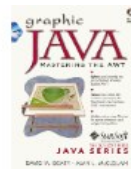
JAOO
conference
- for Developers by Developers

TheServerSide.COM
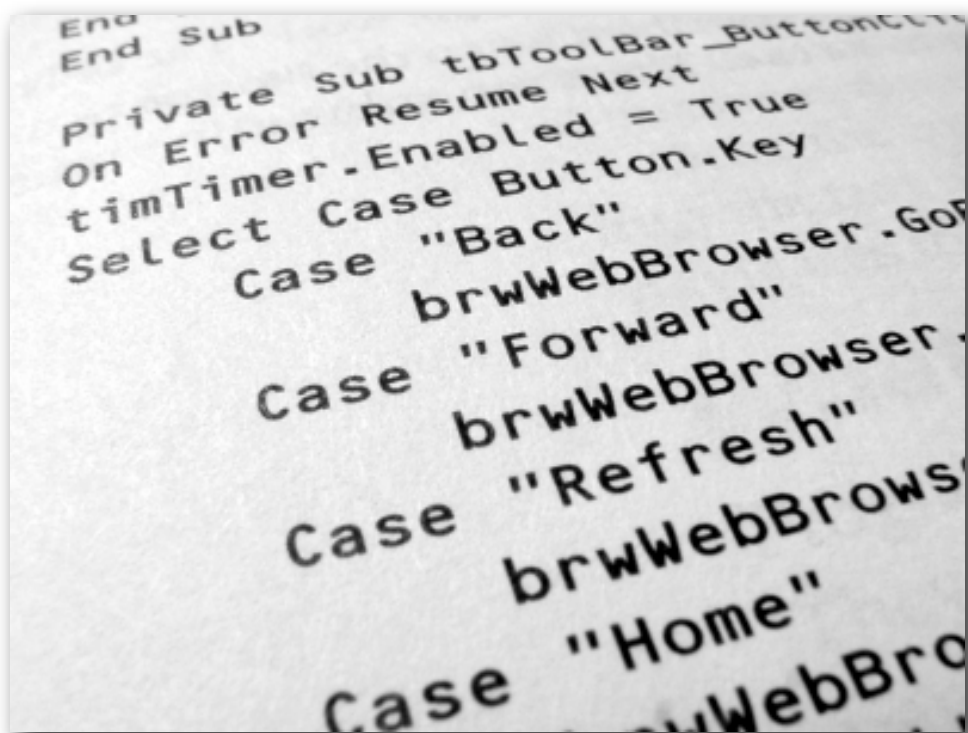Your Enterprise Java Community

Clarity Training

corewebdevelopment.com

Based on

Code

http://code.google.com/p/geary-nfjs

svn co http://geary-nfjs.googlecode.com/svn/trunk/ geary-nfjs-read-only

Demonstration
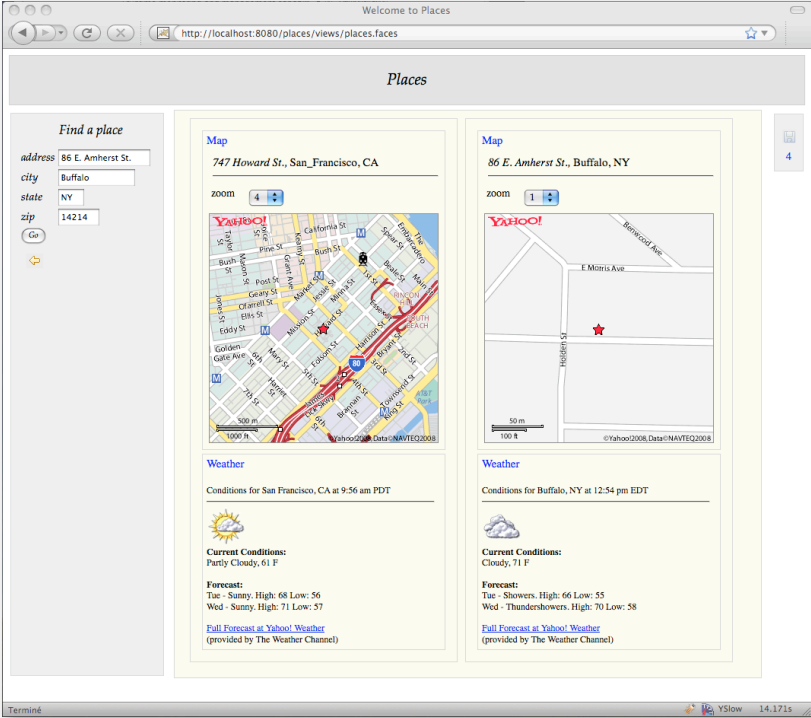
This session

- Composite components
- Ajax with JSF 2
- Ajax validation
- Ajax progress bar
- Ajax components

Composite components

- Composite components…

    - ...are simple to implement

    - ...require no configuration

    - ...support fast iterative development (without redeploying the application)

Using composite components

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:util="http://java.sun.com/jsf/composite/components/util">
  ...
  <util:icon .../>
  ...
</html>
```

```
▼ 📂 resources
    ▼ 📂 components
        ▶ 📂 places
        ▼ 📂 util
              📄 datePicker.xhtml
              📄 icon.xhtml
              📄 login.js
              📄 login.xhtml
```
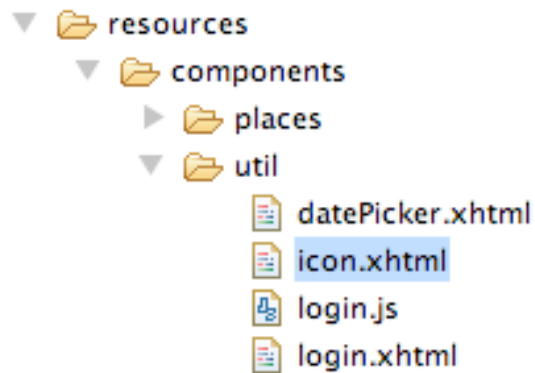
Using an icon

```
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:util="http://java.sun.com/jsf/composite/components/util">

  <util:icon image="#{resource['images:disk-icon.png']}"
     actionMethod="#{sourceViewer.showSource}
        styleClass="icon"/>

</html>
```

Implementing the icon: attributes

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
  xmlns:composite="http://java.sun.com/jsf/composite">

  <composite:interface>
    <composite:attribute name="image"        required="true" />
    <composite:attribute name="styleClass"    default="icon" />
    <composite:attribute name="actionMethod" method-signature="java.lang.String action()" />
  </composite:interface>

  <composite:implementation>
    <h:form>
      <h:commandLink action="#{cc.attrs.actionMethod}">

        <h:graphicImage url="#{cc.attrs.image}"
                styleClass="#{cc.attrs.styleClass}" />

      </h:commandLink>
    </h:form>
  </composite:implementation>

</html>
```
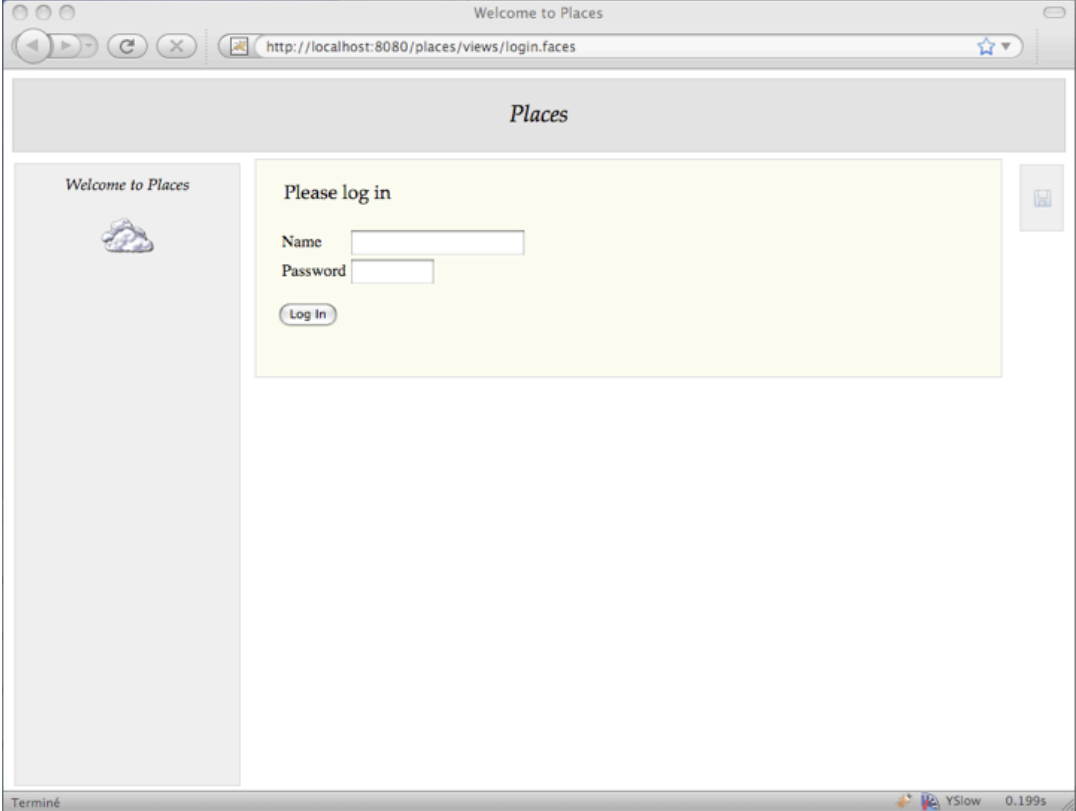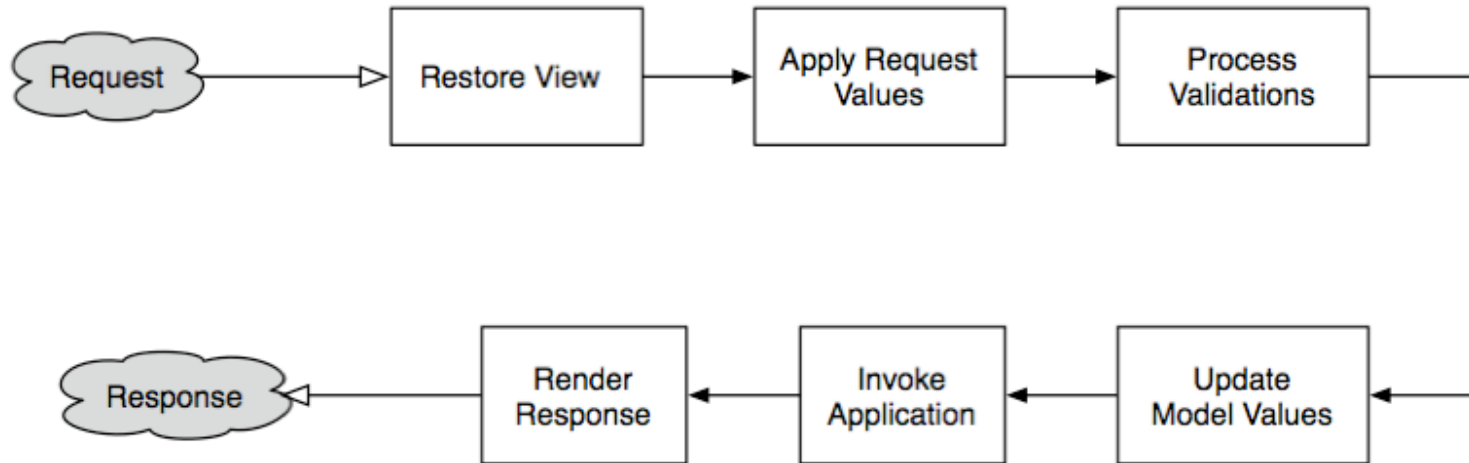
Demonstration

This session
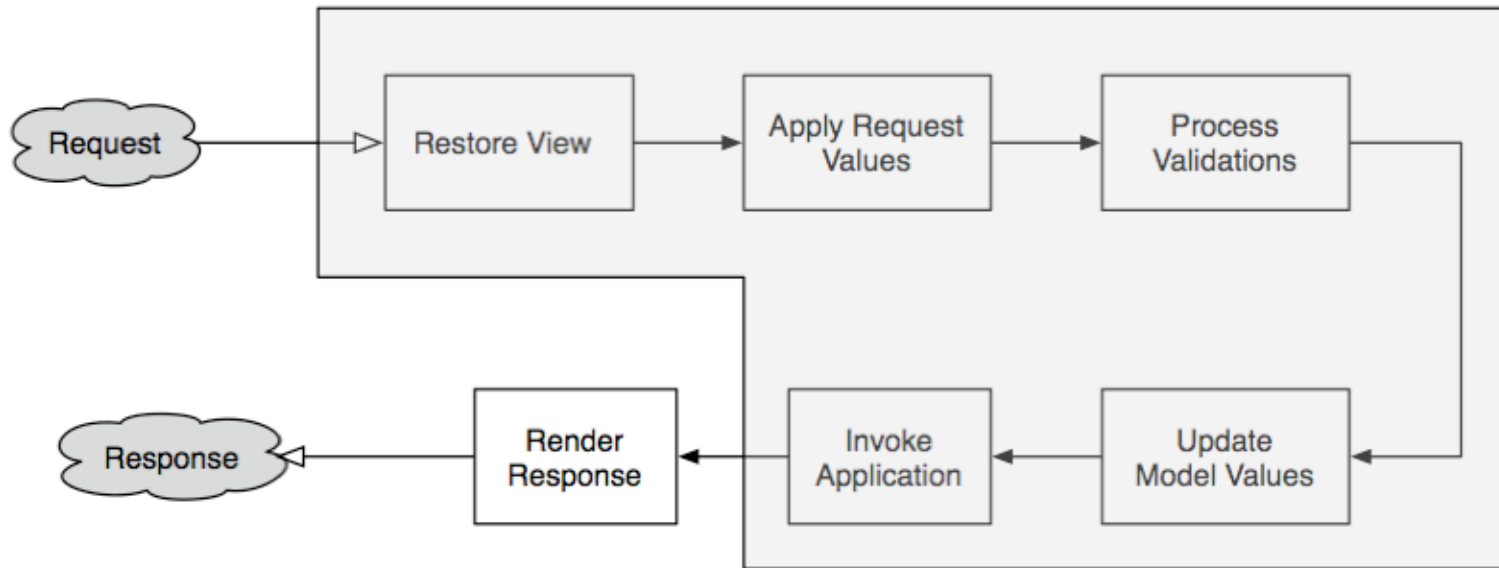
- Composite components
- Ajax with JSF 2
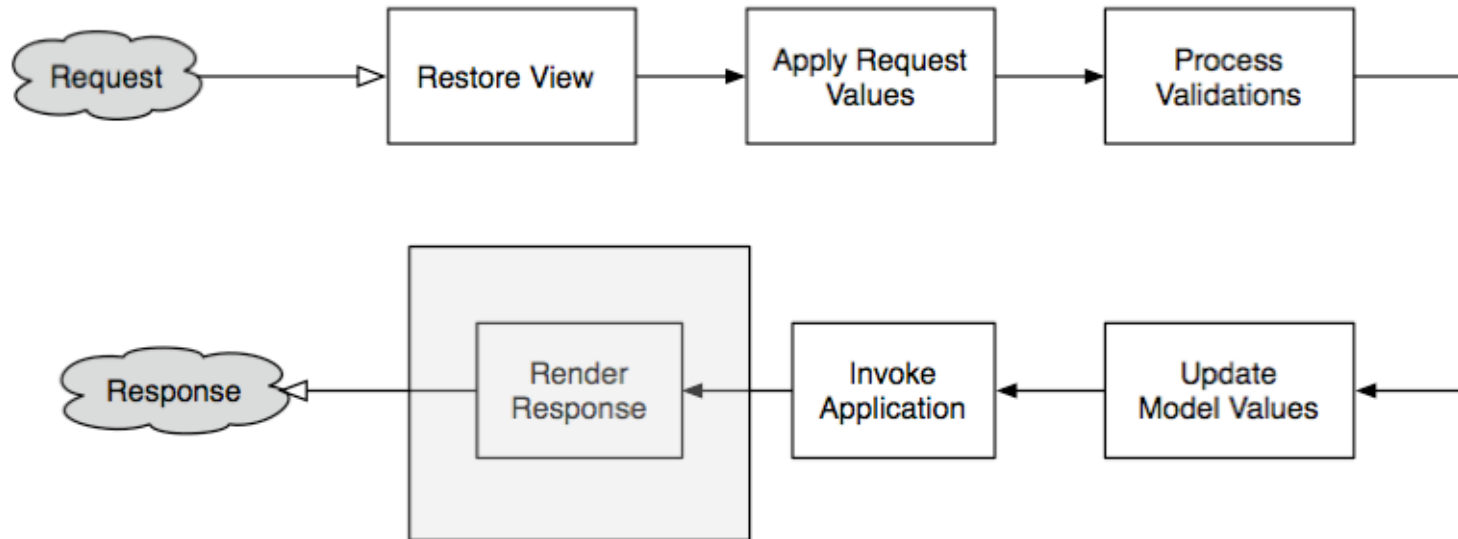- Ajax validation
- Ajax progress bar
- Ajax components

The JSF lifecycle


```
Request  →  Restore View  →  Apply Request Values  →  Process Validations
                                                              ↓
Response  ←  Render Response  ←  Invoke Application  ←  Update Model Values
```

execute

render

Request → Restore View → Apply Request Values → Process Validations → Update Model Values → Invoke Application → Render Response → Response

Using <f:ajax>


```
<h:inputText id="name" value="#{user.name}">
    <f:ajax event="keyup" execute="@this" render="echo"/>
</h:inputText>

…


<h:outputText id="echo" value="#{user.name}"/>
```
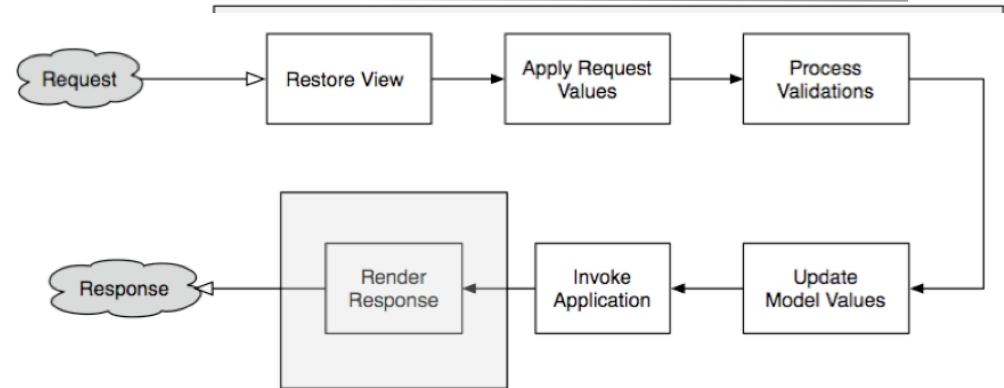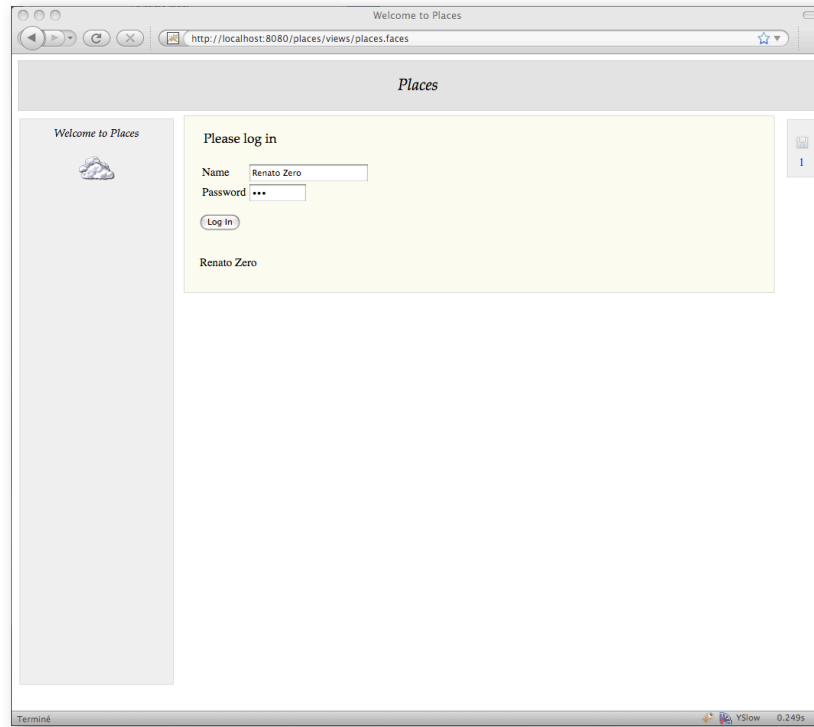
Demonstration

This session

- Composite components
- Ajax with JSF 2
- Ajax validation
- Ajax progress bar
- Ajax components

Ajax validation

Please log in

Name   No_underscores   Name cannot contain underscores

Password   •••••

Log In

markup

Please log in

Name   No_underscores   Name cannot contain underscores

Password   •••••

[Log In]

```
<h:inputText value="#{user.name}"
    valueChangeListener="#{user.validateName}">

    <f:ajax event="blur" render="nameError"/>

</h:inputText>


<h:outputText id="nameError"
            value="#{user.nameError}"
            style="color: red"/>
```

validator

```java
public class User {
  private String nameError;

  public void setNameError(String error) {nameError = error;}
  public String getNameError() {nameError;}

  public void validateName(ValueChangeEvent e) {
    UIInput nameInput = (UIInput)e.getComponent();

    // validate nameInput's value, and store error msg in nameError
  }
}
```
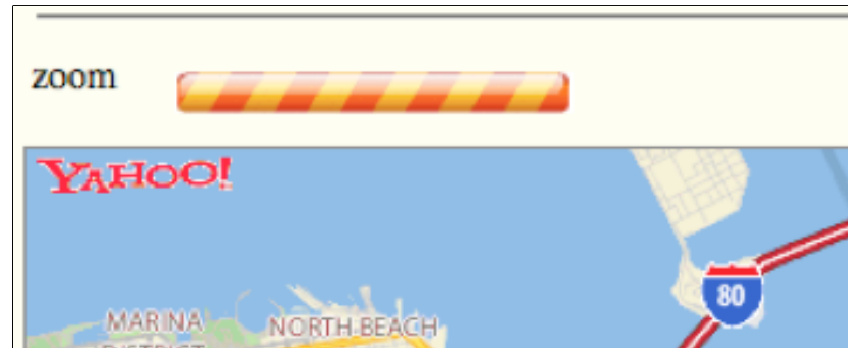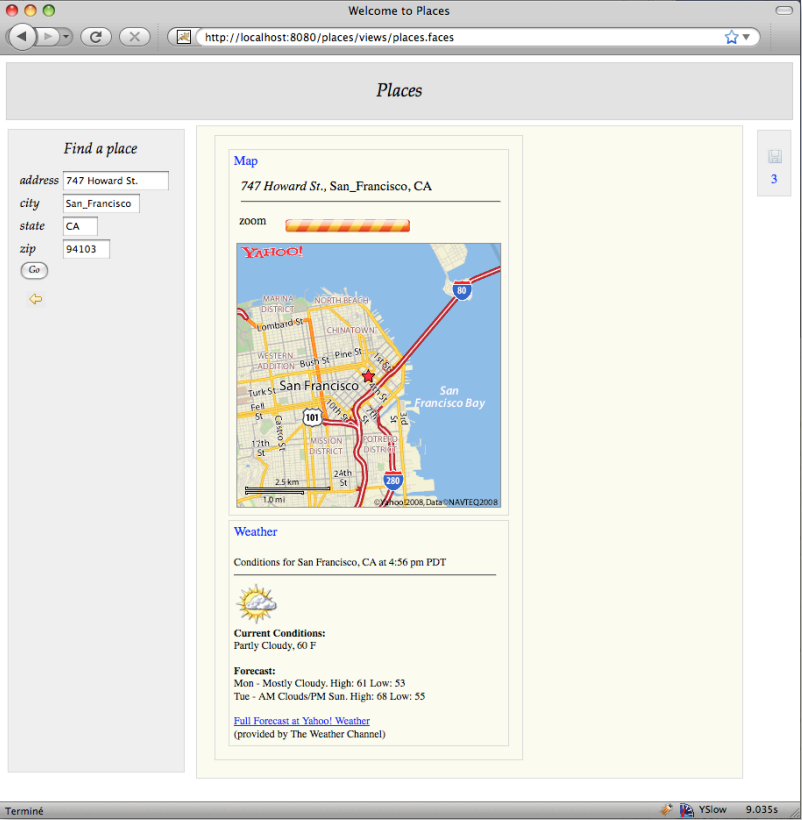
Demonstration
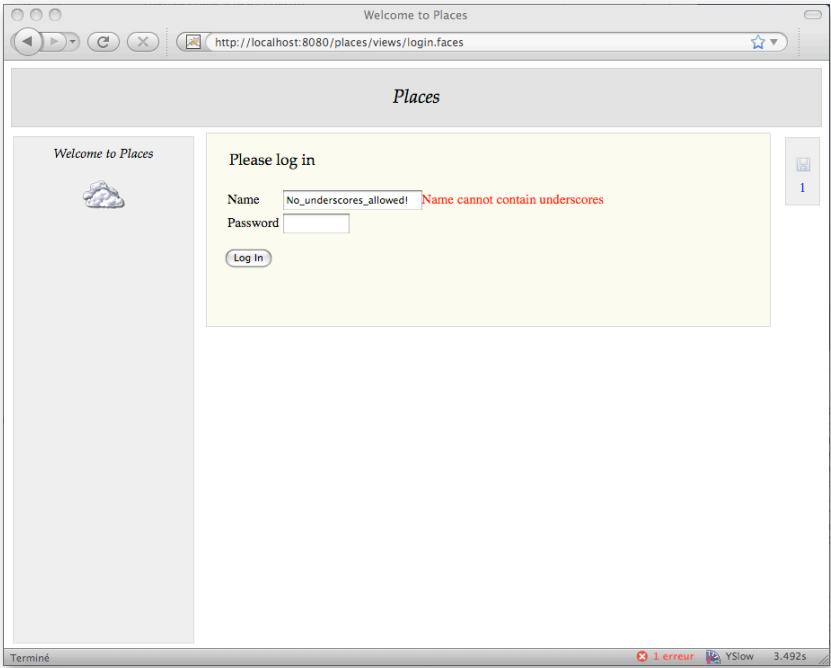
This session

- Composite components
- Ajax with JSF 2
- Ajax validation
- Ajax progress bar
- Ajax components

Ajax progress bar

setting the event handler

```
<h:selectOneMenu id="menu"
    value="#{place.zoomIndex}">

    <f:ajax render="image" onevent="zoomChanging"/>
    …

</h:selectOneMenu>
```

the event handler (JavaScript)

```javascript
function zoomChanging(data) {
  …

  if (data.status == "begin") {
    // hide menu and show progress bar
  }
  else if (data.status == "success") {
    // hide progress bar and show menu
  }
}
```

Demonstration

This session

- Composite components
- Ajax with JSF 2
- Ajax validation
- Ajax progress bar
- Ajax components

Google maps
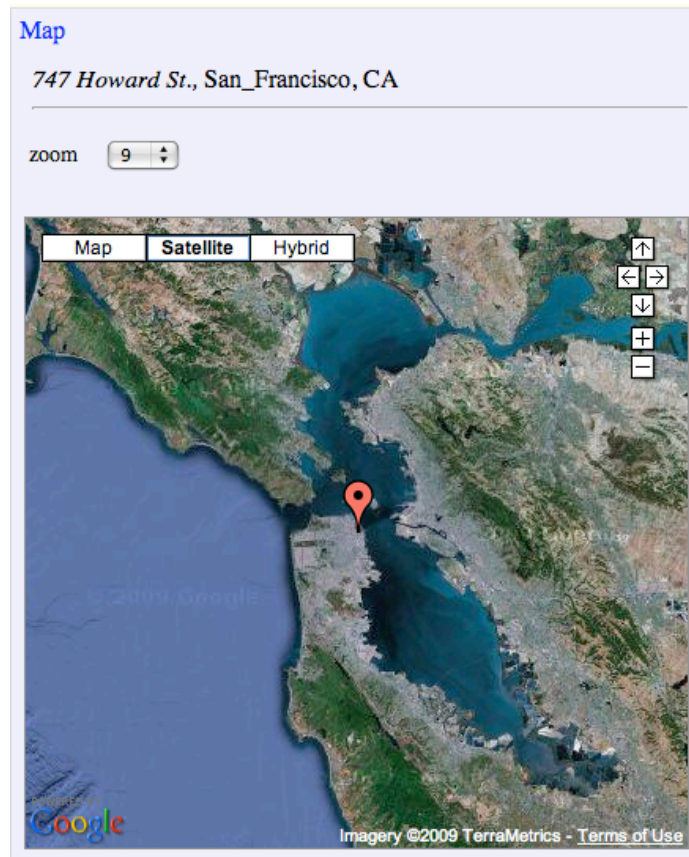
gmaps4jsf

- Blog: jroller.com/HazemBlog
- Download: code.google.com/p/gmaps4jsf
- Documentation: http://code.google.com/p/gmaps4jsf/wiki/TagLibraryDocumentation
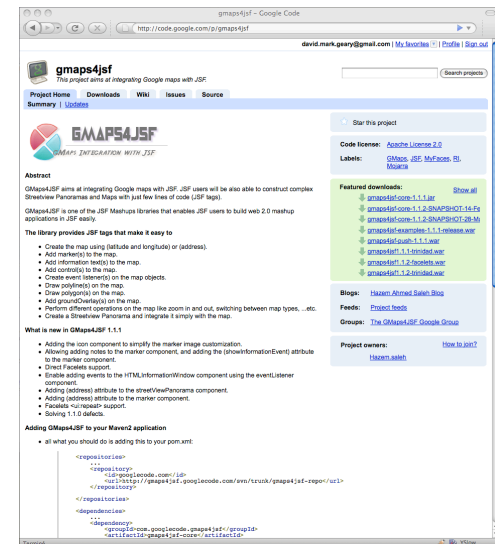
plug in the component
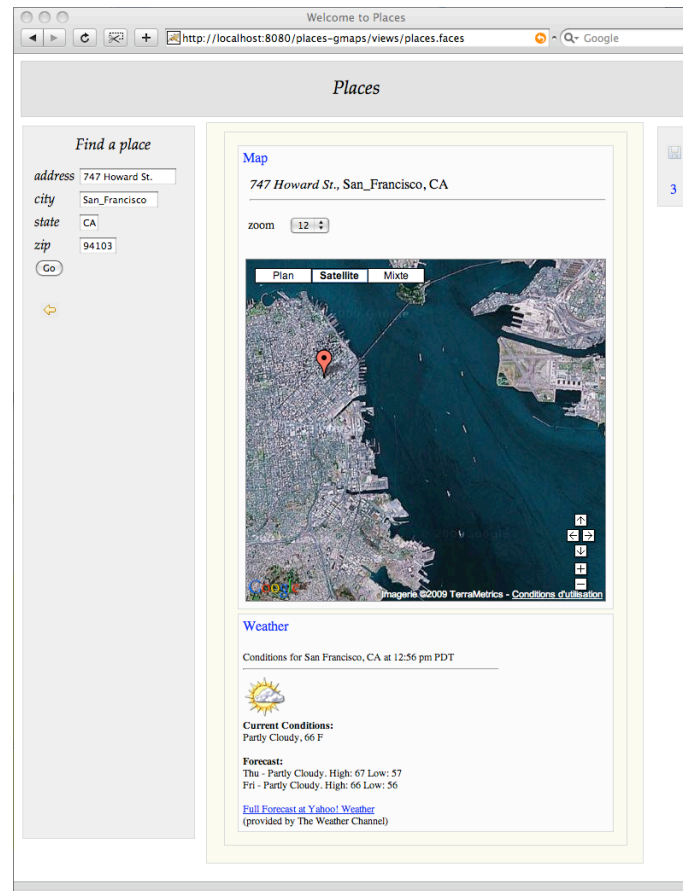
markup

```
<m:map id="placeMap" width="420px" height="400px"
        address="#{location.address}"
        zoom="#{location.zoomIndex}"…>

    <m:mapControl id="smallMapCtrl" name="GSmallMapControl"
            position="G_ANCHOR_BOTTOM_RIGHT"/>

    <m:mapControl  id="smallMapTypeCtrl" name="GMapTypeControl"/>

    <m:marker id="placeMapMarker"/>

</m:map>
```

Demonstration

The End