

Rich Portlet Development

Jennifer Bourey

Jasig Spring Conference, March 9, 2010

© Copyright Unicon, Inc., 2006. This work is the intellectual property of Unicon, Inc. Permission is granted for this material to be shared for non-commercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of Unicon, Inc. To disseminate otherwise or to republish requires written permission from Unicon, Inc.

Portlet Development Goals

- Make
 - life easier
 - development faster
 - portlets shinier




Create an Accessible 2-Column Layout

Accessible Two-Column Layout

Select People and Groups

Who can add About.com College Life to their layout

Your Selections

-  [Students](#)
-  [Faculty User](#)
-  [Portal Developers](#)

Review

Cancel

Search

Enter a name

Go

Browse

Everyone >

 **Students**

 De-select

Students includes:

People

- [Student User](#)
- [Student Layout Owner](#)

Using the Fluid Skinning System

```
<div class="fl-col-flex2">

  <!-- start: left panel -->
  <div class="fl-col">

    <h1>I'm the left column!</h1>

  </div><!-- end: left panel -->

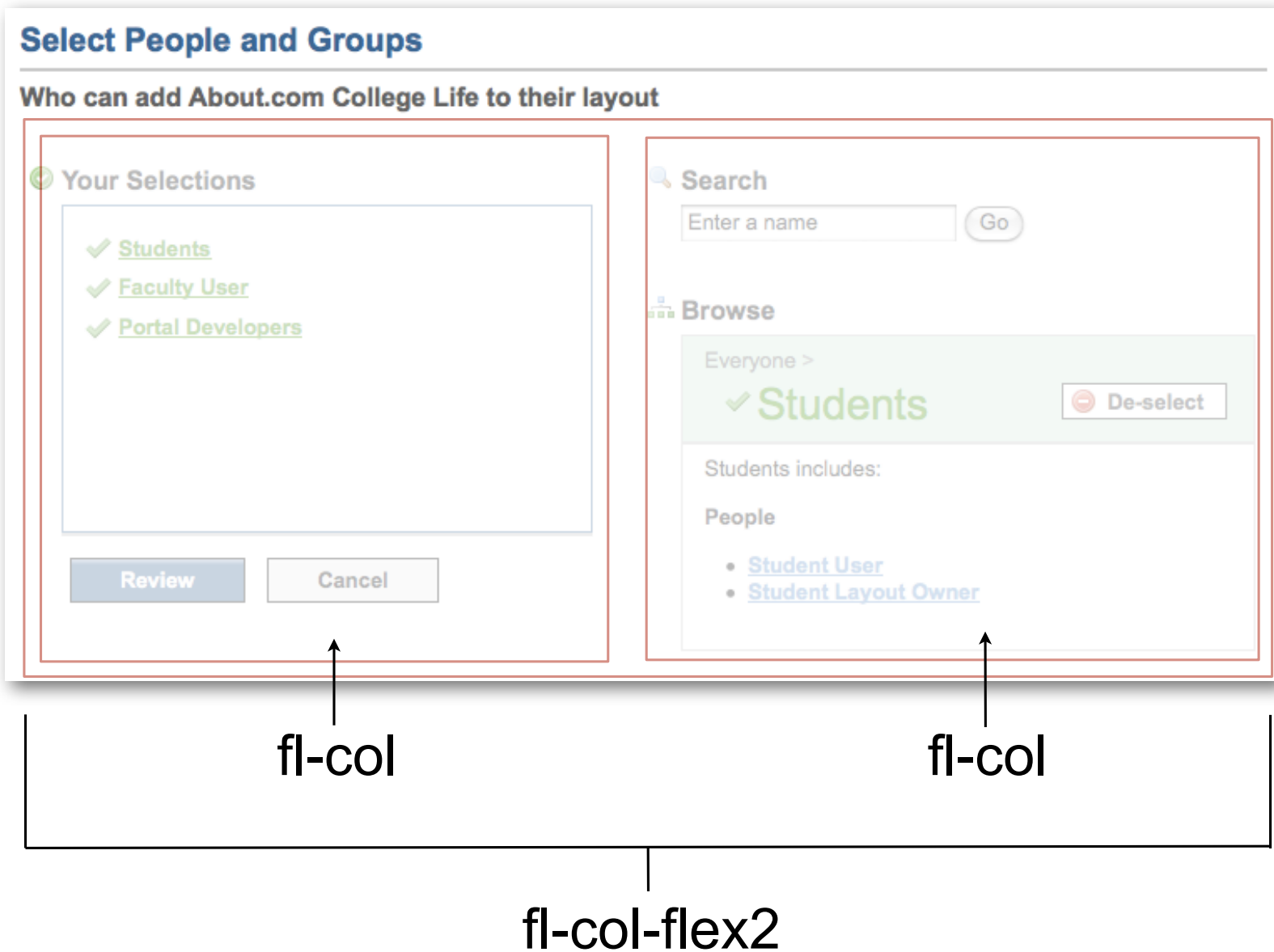
  <!-- start: right panel -->
  <div class="fl-col">

    <h1>I'm the right column!</h1>

  </div><!-- end: right panel -->

</div>
```

Accessible Two-Column Layout



Fluid Skinning System

- Layout wrappers and containers
- Convenience classnames
 - GUI elements: Tabs, menus, icons, widgets
 - font family, size, and formatting
- Theming
- Accessibility
 - easy page linearization
 - tested for usability effectiveness, specifically colorblindness
 - works with Assistive Technology

Allow Users to Page Through a List

Paging Through a List

Portlet Registry

[Register New Portlet](#)

Registered Portlet List

Show

[< previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8 \(last\)](#) [next >](#)

1-10 of 80 items per page

Name	Type	State	Edit	Delete
About.com College Life	RSS	Published	edit	delete
Admin Welcome	Image	Published	edit	delete
Attribute Swapper	Portlet	Published	edit	delete
Bookmarks	Bookmarks Portlet	Published	edit	delete
Cache Manager	Portlet	Published	edit	delete
Calendar	Portlet	Published	edit	delete
CSS Viewer	XML SSL	Published	edit	delete
Daily Business Cartoon	Image	Published	edit	delete
Demo Welcome	Image	Published	edit	delete
Dictionary	Portlet	Published	edit	delete

More from Fluid Infusion

- Components
 - Pager
 - Reorderers (list, grid, and image)
 - Uploader
 - Progressbar
 - Inline text edit
- <http://fluidproject.org/releases/1.1.2/demos/>

Allow Users to Select a Date

Selecting a Date

Automatic Expiration (optional)

Option	Date and Time
Automatic Expiration Date and Time	<input type="text"/>

Review

Cancel

jQuery UI Datepicker

```
<div>
  <label>Start date:</label>
  <input class="cal-datepicker"></input>
</div>

<script type="text/javascript">
  $(document).ready(function() {
    $(".cal-datepicker").datepicker();
  });
</script>
```

jQuery UI Widgets

- Widgets
 - Accordion
 - Datepicker
 - Dialog
 - Progressbar
 - Slider
 - Tabs
- Demos: <http://jqueryui.com/demos/>

Use Javascript Effects

jQuery Effects

Practice Arithmetic

[options](#)

8

- 4

Submit

0 / 0 (%)

jQuery UI

- Effects
 - show, hide, toggle, color animation
 - add/remove/toggle/switch css class
- Demos: <http://jqueryui.com/demos/>

Improve Javascript Performance

Improve UI Performance

- Minify code
- Minimize HTTP requests
- Apply cache headers
- GZip resources

Jasig Resource Server

- Server Webapp
 - Library of popular CSS, JS, and image resources
 - Provide consistent URLs
 - Cache, minify, and gzip hosted resources
- Utilities
 - Filters for caching and gzipping
 - JSP tag for including files from resource server
- Bundled with uPortal 3.1+

Minify code: YUI Maven Plugin

```
<plugin>
  <groupId>net.sf.alchim</groupId>
  <artifactId>yuicompressor-maven-plugin</artifactId>
  <version>0.7.1</version>
  <executions>
    <execution>
      <goals><goal>compress</goal></goals>
    </execution>
  </executions>
  <configuration>
    <excludes>
      <exclude>*/*.min.js</exclude>
    </excludes>
    <linebreakpos>10000</linebreakpos>
    <suffix>.min</suffix>
    <jswarn>>false</jswarn>
  </configuration>
</plugin>
```

Cache Headers: Resource Server

```
<filter>
  <filter-name>CacheExpiresFilter</filter-name>
  <filter-class>
    org.jasig.resourceserver.utils.filter.CacheExpirationFilter
  </filter-class>

  <!-- Length of time to set in cache headers (in seconds) -->
  <init-param>
    <param-name>cacheMaxAge</param-name>
    <param-value>31536000</param-value>
  </init-param>

  <!-- Time between regenerating cache header strings (in millis) -->
  <init-param>
    <param-name>regenerateHeadersInterval</param-name>
    <param-value>1000</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>CacheExpiresFilter</filter-name>
  <url-pattern>/rs/*</url-pattern>
</filter-mapping>
```

GZip CSS and JS

```
<filter>
  <filter-name>pageCachingFilter</filter-name>
  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</
filter-class>
  <init-param>
    <param-name>targetFilterLifecycle</param-name>
    <param-value>>true</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>pageCachingFilter</filter-name>
  <url-pattern>*.js</url-pattern>
</filter-mapping>
<filter-mapping>
  <filter-name>pageCachingFilter</filter-name>
  <url-pattern>*.css</url-pattern>
</filter-mapping>
```

GZip CSS and JS

```
<bean id="cacheManager"  
  class="org.springframework.cache.ehcache.EhCacheManagerFactoryBean">  
  <property name="cacheManagerName" value="ResourceServer.cacheManager" />  
</bean>  
  
<bean id="pageCachingFilter"  
  class="org.jasig.resourceserver.utils.cache.ConfigurablePageCachingFilter">  
  <constructor-arg ref="cacheManager"/>  
  <constructor-arg value="SimplePageCachingFilter"/>  
</bean>
```


GZip HTML: Tomcat

```
<Connector port="8080" protocol="HTTP/1.1"  
  connectionTimeout="20000" redirectPort="8443"  
  emptySessionPath="true"  
  compression="on"  
  compressableMimeType="text/html,text/xml,text/plain" />
```

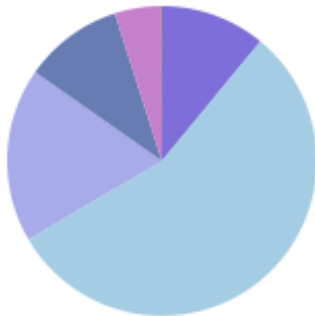
Before

Statistics

The page has a total of **43** HTTP requests and a total weight of **274.1K** bytes with empty cache

WEIGHT GRAPHS

Empty Cache

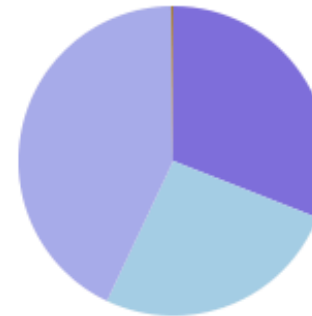


HTTP Requests - 43

Total Weight - 274.1K

1 HTML/Text	30.0K
6 JavaScript File	152.3K
6 Stylesheet File	49.8K
20 CSS Image	28.3K
9 Image	13.3K
1 Favicon	0.1K

Primed Cache



HTTP Requests - 29

Total Weight - 97.1K

1 HTML/Text	30.0K
2 JavaScript File	25.3K
3 Stylesheet File	41.5K
13 CSS Image	0.0K
9 Image	0.04K
1 Favicon	0.1K

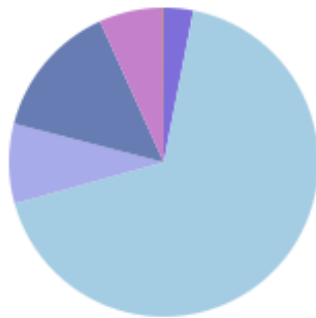
After

Statistics

The page has a total of **42** HTTP requests and a total weight of **199.4K** bytes with empty cache

WEIGHT GRAPHS

Empty Cache

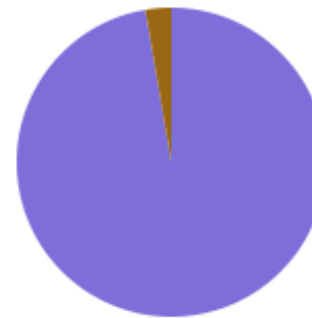


HTTP Requests - 42

Total Weight - 199.4K

■	1 HTML/Text	6.1K
■	6 JavaScript File	134.8K
■	6 Stylesheet File	16.6K
■	20 CSS Image	28.3K
■	8 Image	13.3K
■	1 Favicon	0.1K

Primed Cache



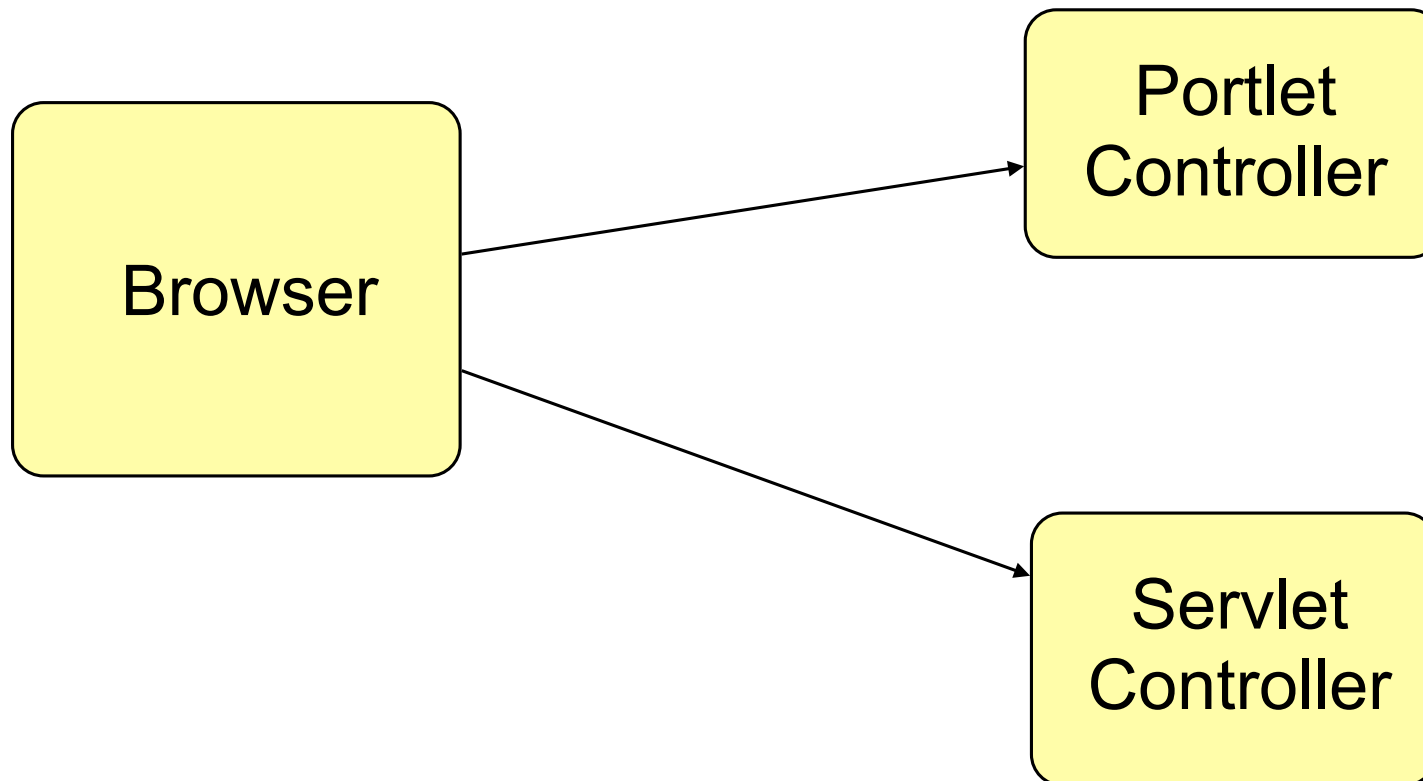
HTTP Requests - 23

Total Weight - 6.3K

■	1 HTML/Text	6.1K
■	13 CSS Image	0.0K
■	8 Image	0.0K
■	1 Favicon	0.1K

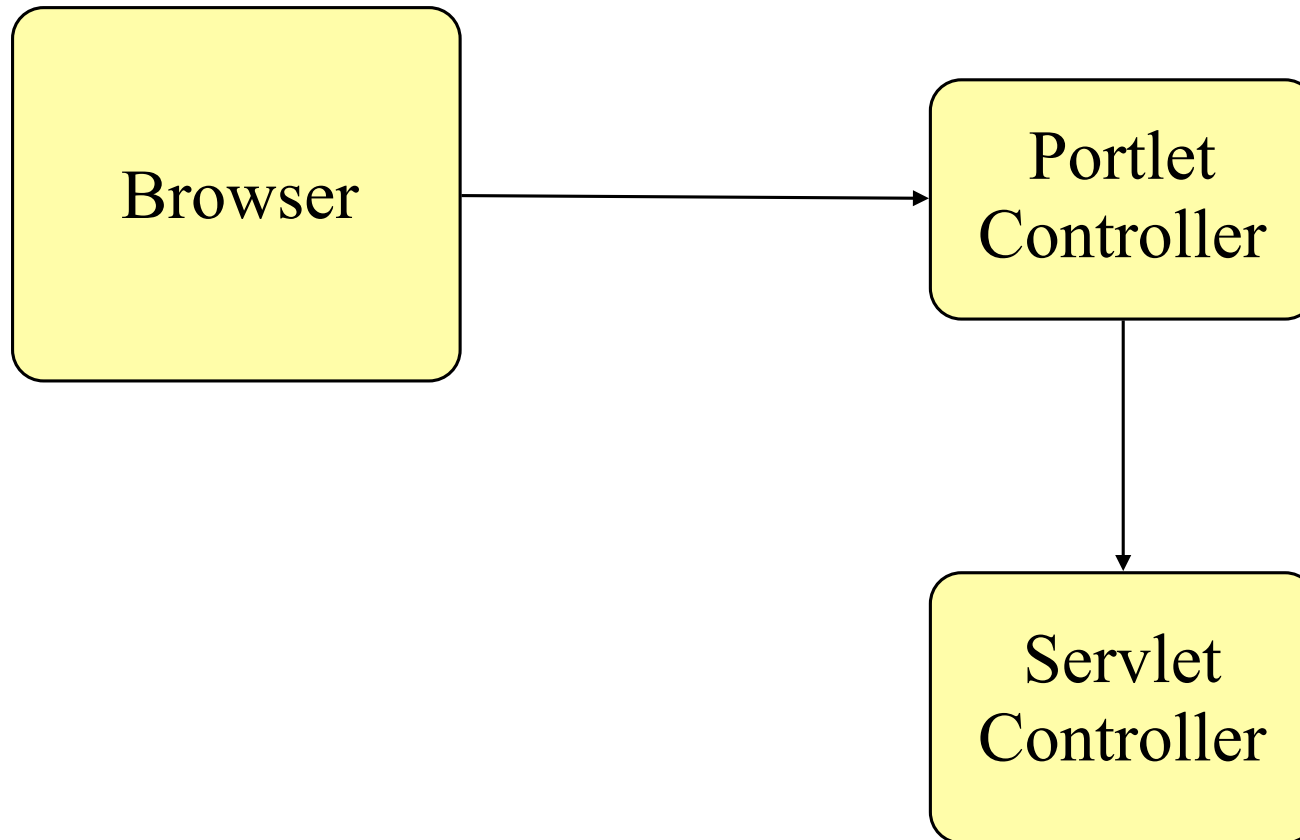
Interact with Portlet Session and Preferences via AJAX

AJAX Servlet

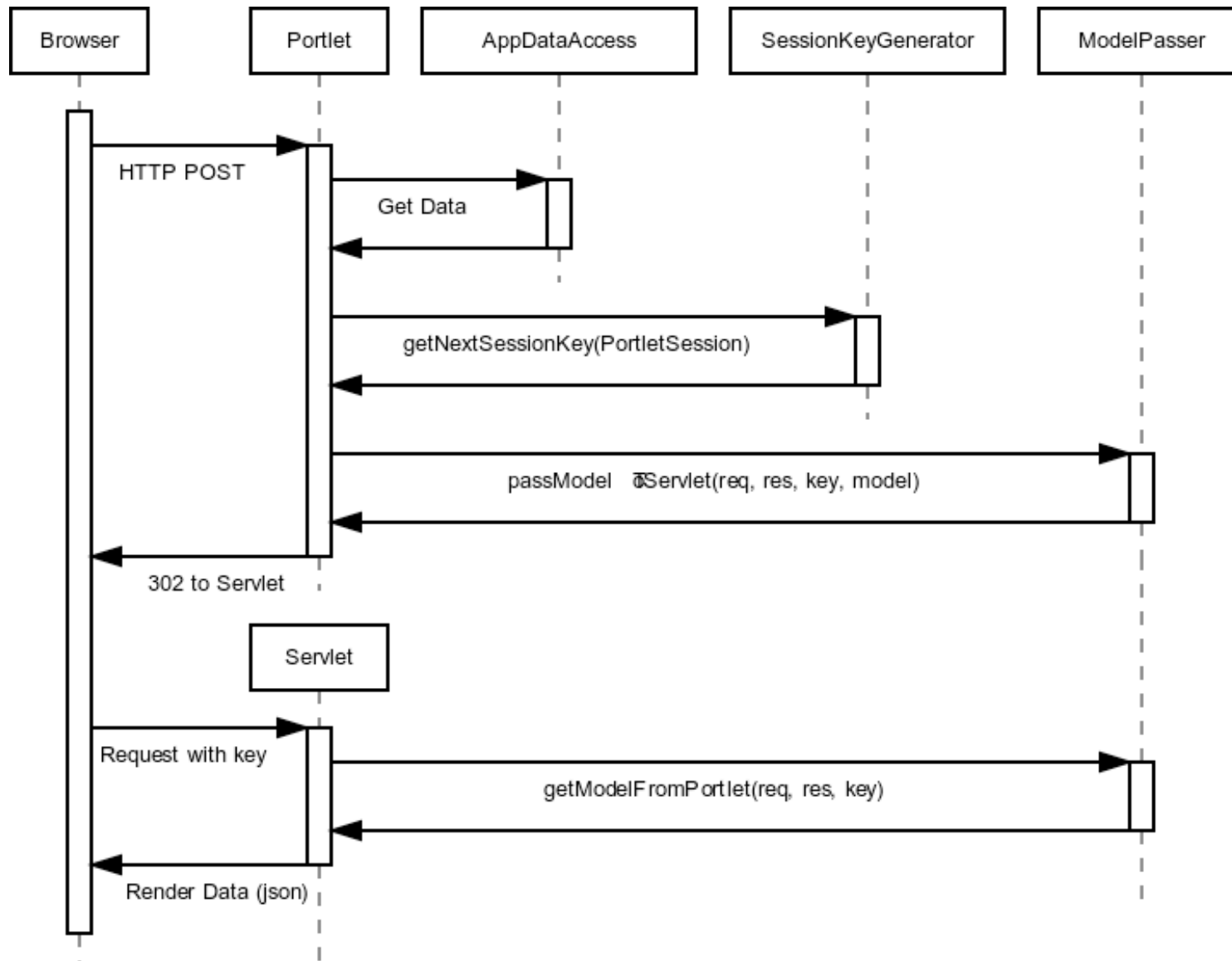


No portlet preferences access
No portlet session access

Jasig Ajax Portlet Support



Jasig Ajax Portlet Support



Weather Portlet Example

- Save location preferences via AJAX POST
- Return success or failure

Editing Portlet Preferences via Ajax

```
@RequestMapping(params = {"action=delete"})
public void deleteCity(
    @RequestParam(value = "locationCode") String locationCode,
    ActionRequest request, ActionResponse response) throws Exception {

    Map<Object, Object> model = new HashMap<Object, Object>();

    // remove location from preferences
    final PortletPreferences prefs = request.getPreferences();
    this.weatherService.deleteWeatherLocation(prefs, locationCode);

    // return success
    model.put("status", "success");
    this.ajaxPortletSupport
        .redirectAjaxResponse("ajax/json", model, request, response);
}
```

Ajax Response Controller

```
@Controller
public class AjaxResponseController {

    private AjaxPortletSupport ajaxPortletSupport;

    @Autowired
    public void setAjaxPortletSupport(
        AjaxPortletSupport ajaxPortletSupport) {
        this.ajaxPortletSupport = ajaxPortletSupport;
    }

    @RequestMapping("/json")
    public ModelAndView renderAjaxResponse(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletExceptionBindingException, IOException {

        final Map<Object, Object> model = this.ajaxPortletSupport
            .getAjaxModel(request, response);

        return new ModelAndView("jsonView", model);
    }
}
```

Proxy an AJAX Target

Jasig Dictionary Portlet

- Uses Aonaware Dictionary XML REST service
- Want to make AJAX requests DictService server
- Single Origin Policy prevents request
 - need same protocol, domain, and port

Using Jasig's ProxyView

```
@RequestMapping(method = RequestMethod.GET)
public ModelAndView getProxyView(HttpServletRequest request,
    @RequestParam(value="searchTerm") String searchTerm) {

    Map<String, Object> model = new HashMap<String, Object>();

    // build the URL to be proxied and add it to our model
    String url = "http://www.proxied.com/url";
    model.put(ProxyView.URL, url.toString());

    // return a proxy view of the URL
    return new ModelAndView("proxyView", map);
}
```

Prevent Cross-side Scripting Attacks

Cross-Site Scripting

The screenshot shows a web browser window titled "uPortal by JA-SIG" with the address bar displaying "http://localhost:8080/uPortal/render.userLayoutRo". The browser's menu bar includes "Latest Headlines", "Music", "Computing News", "Fun", "Programming", "Most Visited", and "Readability". The toolbar shows "Disable", "Cookies", "CSS", "Forms", "Images", "Information", and "Miscellaneous". The page content includes the "uPortal" logo, a "Welcome Admin User. Sign Out" message, and a navigation menu with "Welcome", "News", "Development", "Testing", and "Admin To". Below the menu, there are two portlets: "My Grades" and "Insecure Portlet".

New Tab

My Grades

CPSC 101: A
ENGL 101: C

Insecure Portlet

Insecure content:

Unescaped Javascript

```
<script type="text/javascript">  
    document.write('${title}');  
</script>
```



```
<script type="text/javascript">  
    document.write('' + alert('pwnd!') + '');  
</script>
```


Escaped Javascript

```
<script type="text/javascript">
  document.write ('<spring:escapeBody
    javaScriptEscape="true">${title}</spring:escapeBody>');
</script>
```



```
<script type="text/javascript">
  document.write ('\ ' + alert ('\pwnd! \') + \ ' ');
</script>
```

Unescaped HTML

```
<h1>  
  ${ title }  
</h1>
```



```
<h1>  
  <script type="text/javascript">alert('pwnd!');</script>  
</h1>
```

Escaped HTML

```
<h1>
  <spring:escapeBody htmlEscape="true">
    ${ title }
  </spring:escapeBody>
</h1>
```



```
<h1>
  &lt;script type="text/javascript"&gt;alert
  ('pwnd!');&lt;/script&gt;
</h1>
```

XSS: Dictionary Portlet

```
// parse the definition from the response
in = get.getResponseBodyAsStream();
String def = service.getDefinitionFromXml(in);

// escape any HTML characters
return StringEscapeUtils.escapeHtml(def);
```

XSS: News Reader Portlet

- Display RSS feeds in a portlet
- Allow HTML, but maintain security
- Might need different policies for different sites

OWASP AntiSamy

- Java API for filtering HTML/CSS
- Define allowed tags and attributes through XML
- Clean input by stripping unwanted tags
 - strip malicious scripts from external content
 - improve presentation by removing images, etc.

News Reader: OWASP AntiSamy

```
// retrieve
InputStream in = get.getResponseBodyAsStream();

//parse
SyndFeed feed = input.build(new XmlReader(in));

//clean
AntiSamy as = new AntiSamy();

List<SyndEntry> a = feed.getEntries();

for (SyndEntry entry : a) {
    SyndContent description = entry.getDescription();

    Policy policy = Policy.getInstance(policyFile);

    CleanResults cr = as.scan(description.getValue(), policy);
    description.setValue(cr.getCleanHTML());
    entry.setDescription(description);
    log.info("Feed " + url + " cleaned in " +
            cr.getScanTime() + " seconds");
}
```

XSS: Calendar Portlet

- Custom AntiSamy JSP tag
- Wraps Spring-configured string cleaning utility

```
<antisamy:clean value="${ event.description.value }"/>
```


AntiSamy-based String Cleaning

```
// clean a string using the configured AntiSamy file
public String getCleanString(String dirtyInput) throws ScanException,
    PolicyException {

    AntiSamy as = new AntiSamy();
    CleanResults cr = as.scan(dirtyInput, this.policy);
    return cr.getCleanHTML();

}
```

AntiSamy-based JSP Tag

```
<%@ tag isELIgnored="false" dynamic-attributes="attributes" body-
content="empty" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ attribute name="value" required="true" %>
<%@ tag import="org.springframework.web.context.WebApplicationContext" %>
<%@ tag
import="org.springframework.web.context.support.WebApplicationContextUtils"
%>
<%@ tag import="org.jasig.portlet.calendar.util.StringCleaningUtil" %>
<%
    WebApplicationContext ctx =
        WebApplicationContextUtils.getWebApplicationContext(application);
    StringCleaningUtil util =
        (StringCleaningUtil) ctx.getBean("stringCleaningUtil");
%>
<%= util.getCleanString(value) %>
```

Preventing Non-Portlet Access

```
HttpSession session = request.getSession(false);
if (session == null ||
    !((Boolean) session.getAttribute("hasDictionarySession"))) {
    return new ModelAndView("jsonView", map);
}
```

Work with JSON and XML

Data as XML

- Javascript parsing
 - jQuery
- Java output and parsing
 - Maven JAXB plugin
 - XStream, etc.
- Groovy

Dictionary Parsing in Groovy

Groovy Implementation

```
import java.io.InputStream
import groovy.xml.StreamingMarkupBuilder
import org.jasig.portlet.widget.service.IDictionaryParsingService

class DictionaryParsingServiceImpl implements IDictionaryParsingService {

    String getDefinitionFromXml(InputStream xml) {
        def response = new XmlSlurper().parse(xml)
        return response.Definitions.Definition.WordDefinition
    }
}
```

Spring Wiring

```
<lang:groovy id="dictionaryParsingService"
script-source="classpath:org/jasig/portlet/widget/service/
DictionaryParsingServiceImpl.groovy"/>
```

Dictionary Parsing in Groovy

```
import java.io.InputStream
import groovy.xml.StreamingMarkupBuilder
import org.jasig.portlet.widget.service.IDictionaryParsingService

class DictionaryParsingServiceImpl implements IDictionaryParsingService {

    String getDefinitionFromXml(InputStream xml) {
        def response = new XmlSlurper().parse(xml)
        return response.Definitions.Definition.WordDefinition
    }

}
```

Data as JSON

- Javascript parsing
 - jQuery
- Java output and parsing
 - JSONlib
 - net.sf.json's JsonView

JsonView

Controller code

```
@RequestMapping("/json")
public ModelAndView renderAjaxResponse(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletExceptionBindingException, IOException {

    final Map<Object, Object> model = this.ajaxPortletSupport
        .getAjaxModel(request, response);

    return new ModelAndView("jsonView", model);
}
```

views.xml

```
<bean id="jsonView"
    class="net.sf.json.spring.web.servlet.view.JsonView">
    <property name="contentType" value="application/json" />
</bean>
```

Response

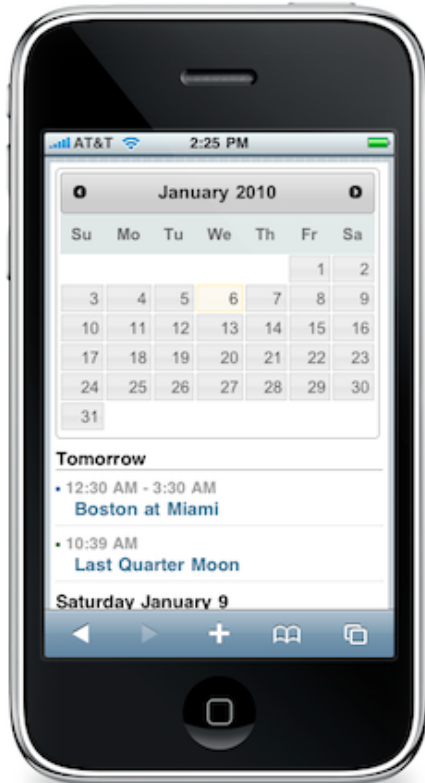
```
{ response: success }
```

Present Different Views

Example: Calendar Portlet

- Three views
 - Mobile View
 - Narrow Column View
 - MAXIMIZED View
- Simple interface for view selection

Calendar Views



Mobile

My Calendars Edit Maximize Remove

View: [Day](#) | Week | [Month](#)

December 2009

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Tomorrow

- All day
[New Year's Day](#)

Sunday January 3

- 11:00 AM - 2:00 PM
[Patriots at Texans](#)

[View more events](#) →

← Column

Maximized

[Edit calendar preferences](#) | [Calendar administration](#)

January 2010

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Boston at Atlanta

Date:
Saturday January 9
1:00 AM to 1:00 AM

Description:
TV: ESPN

[← Return to event list](#)

Show calendar:

- ✓ Boston Celtics
- ✓ Moon Phases
- ✓ US Holidays

Calendar View Selection

```
public String getCalendarViewName(PortletRequest request) {
    String userAgent = request.getProperty("user-agent");

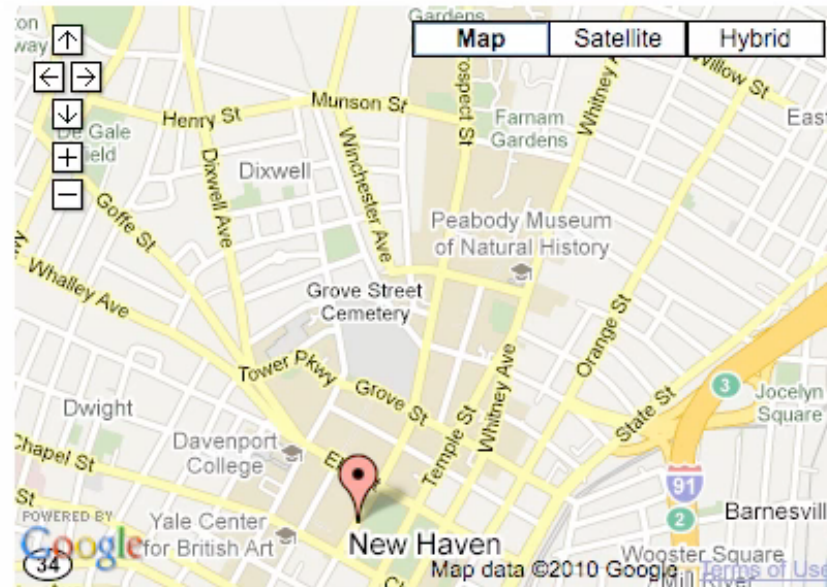
    // check to see if this is a mobile device
    if (this.mobileDeviceRegexes != null && userAgent != null) {
        for (Pattern regex : this.mobileDeviceRegexes) {
            if (regex.matcher(userAgent).matches()) {
                return CALENDAR_MOBILE_VIEW;
            }
        }
    }

    // otherwise check the portlet window state
    WindowState state = request.getWindowState();
    if (WindowState.MAXIMIZED.equals(state)) {
        return CALENDAR_WIDE_VIEW;
    } else {
        return CALENDAR_NARROW_VIEW;
    }
}
```

Mashups

Yale Maps Portlet

Search Yale for buildings (e.g. "Bass" or "library") or for local businesses.



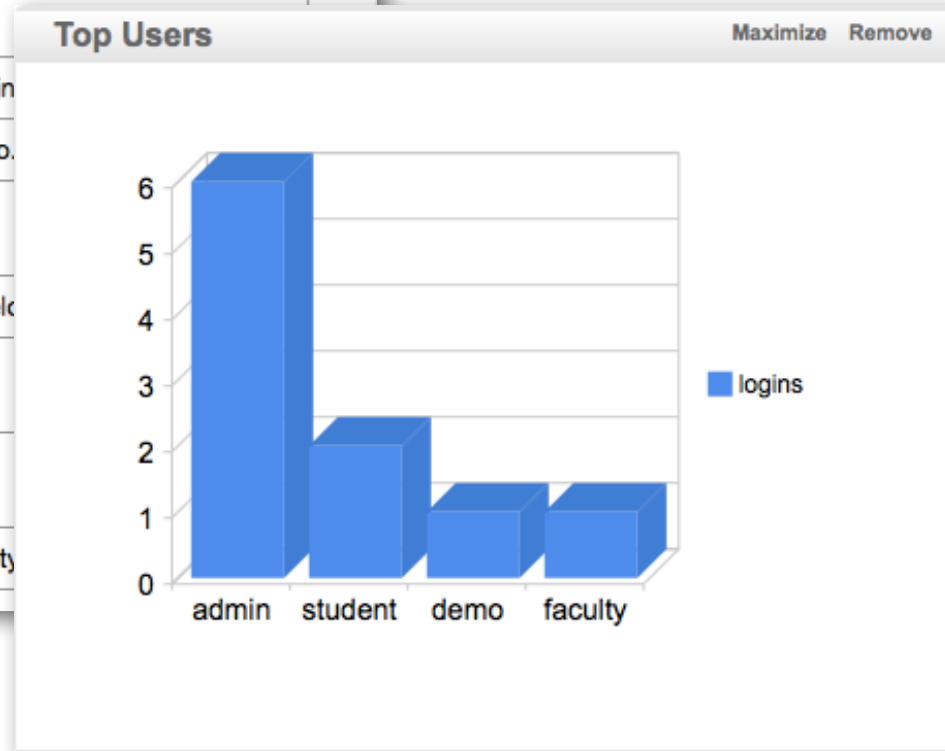
◀ × ▶ scroll through results

Example: SQL Query

Sql Query Maximize Remove

< previous 1 2 (last) next > 1-10 of 17 items 10 per page

username	name	email
admin-lo	Admin Layout Owner	
admin	Admin User	admin
demo	Demo User	demo.
developer-lo	Developer Layout Owner	
developer	Developer User	develo
ent-lo	Entertainment Layout Owner	
faculty-lo	Faculty Layout Owner	
faculty	Faculty User	faculty



Interesting Web APIs

- Google APIs
 - <http://code.google.com/more/>
- Yahoo APIs
 - <http://developer.yahoo.com/everything.html#apis>
- Programmable Web
 - <http://www.programmableweb.com/apis>
- Many Popular Sites
 - Twitter, YouTube, Last.fm

Personal Favorites

- Google Visualizations
- Google Maps
- Yahoo Stock Data
- Yahoo Weather
- Twitter

Looking Ahead

Spring 3.0

- Portlet 2.0 (JSR-286)
 - Resource URLs: `@RequestMapping`
- Spring 3.0
 - REST template
 - JSR-286 support

Ajax in JSR-168 and Spring 2.5

JSP/Javascript

```
$.get('<c:url value="/ajax/dictionary"/>', {}, function() {}, "json");
```

Servlet Controller

```
@RequestMapping(method = RequestMethod.GET)
public ModelAndView getDefinition(HttpServletRequest request,
    @RequestParam(value="word") String word,
    @RequestParam(value="dictId") String dict) throws Exception {

    // stuff goes here
    return new ModelAndView("jsonView", map);

}
```

Ajax with Jasig Library

JSP/Javascript

```
$.get('${dictionaryUrl}', {}, function() {}, "json");
```

Portlet Controller

```
@RequestMapping(params = {"action=delete"})
public ModelAndView getDefinition(ActionRequest request,
    ActionResponse response, @RequestParam(value="word") String word,
    @RequestParam(value="dictId") String dict) throws Exception {

    // stuff goes here
    this.ajaxPortletSupport
        .redirectAjaxResponse("ajax/json", model, request, response);
}
```

Ajax in JSR-286 and Spring 3.0

JSP/Javascript

```
$.get('<portlet:resourceURL/>', {}, function() {}, "json");
```

Portlet Controller

```
@RequestMapping()  
public ModelAndView getDefinition(PortletRequest request,  
    @RequestParam(value="word") String word,  
    @RequestParam(value="dictId") String dict) throws Exception {  
  
    // stuff goes here  
    return new ModelAndView("jsonView", map);  
  
}
```

Example Resources

Presentation Examples

- Jasig Portlet Examples
 - Calendar Portlet
 - News Reader Portlet
 - Weather Portlet
 - Widget Portlets
 - Dictionary
- uPortal Examples
 - Portlet Administration Portlet

Questions?



Jennifer Bourey

jbourey@unicon.net

www.unicon.net