

Extending Liferay Three Ways

Core Portlets, Plugin Portlets, No Portlets

Jeff Handa

Sr. Training Consultant



San Francisco, USA
October 20-22, 2013

Workshop Goals

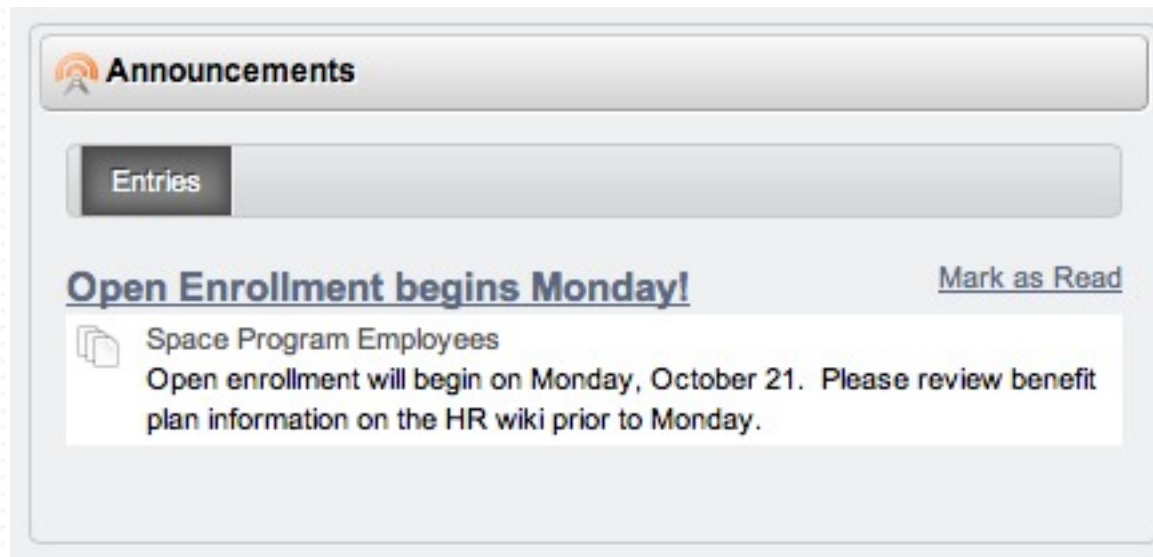
- After completing this workshop, you should:
 - Understand how to customize a core portlet to extend or customize a Liferay feature
 - Understand how to develop and deploy a portlet plugin to extend or customize a Liferay feature
 - Understand how to create Liferay services to extend or customize a Liferay feature

Background

- Liferay is a portal, but's it's also a platform built on an open SOA strategy
- Over 100 portlets available out of the box
- Portlets are consumers of the core services
- Don't like the way we do something? Change it!

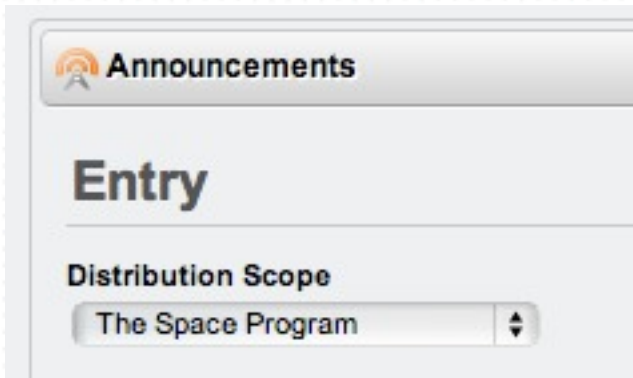
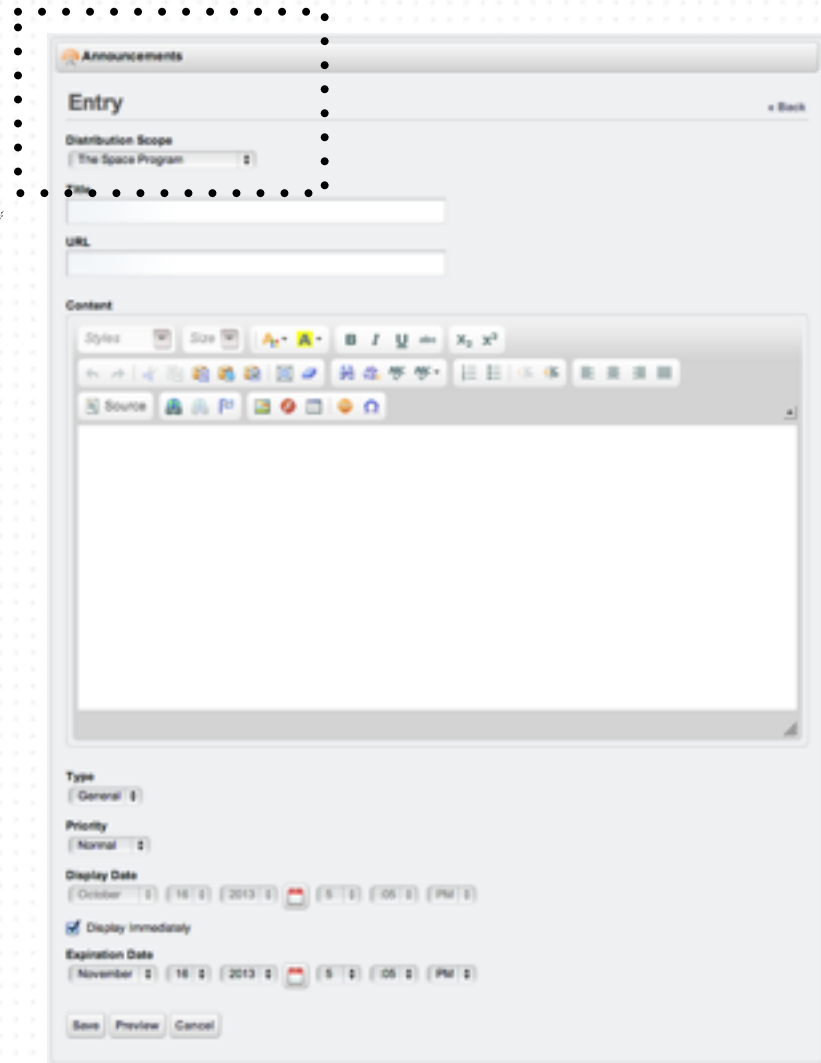
Part 1: Core Portlets

Announcements Portlet




The screenshot displays the Liferay Announcements Portlet. At the top, there is a header with the Liferay logo and the word "Announcements". Below this is a tab labeled "Entries". The main content area features a prominent announcement titled "Open Enrollment begins Monday!" with a "Mark as Read" link to its right. The announcement text, preceded by a document icon, states: "Space Program Employees Open enrollment will begin on Monday, October 21. Please review benefit plan information on the HR wiki prior to Monday."

Announcements Portlet



Announcements Portlet

 **Announcements**

Entry

[← Back](#)

Distribution Scope

Sites

- Classified Mars Info
- The Space Program
- Space Program Employees

Title

First, a little homework . . .

[http://localhost:8080/web/guest/home?
p_p_id=84&p_p_lifecycle=0&p_p_state=normal&p_p_mode
=view&p_p_col_id=column-2&p_p_col_pos=1&p_p_col_cou
nt=2&_84_struts_action=%2Fannouncements
%2Fview&tabs1=manage-entries](http://localhost:8080/web/guest/home?p_p_id=84&p_p_lifecycle=0&p_p_state=normal&p_p_mode=view&p_p_col_id=column-2&p_p_col_pos=1&p_p_col_count=2&_84_struts_action=%2Fannouncements%2Fview&tabs1=manage-entries)

- p_p_id=84
- p_p_lifecycle=0
- p_p_state=normal
- p_p_mode=view
- _84_struts_action=/announcements/view
- tabs1=manage-entries

Look up 84 in portlet-custom.xml

```
<portlet>
  <portlet-name>84</portlet-name>
  <display-name>Announcements</display-name>
  <portlet-class>
    com.liferay.portlet.StrutsPortlet
  </portlet-class>
  . . .
  <init-param>
    <name>view-action</name>
    <value>/announcements/view</value>
  </init-param>
  . . .
</portlet>
```

Look up */announcements/view* in the *struts-config.xml*

```
<action path="/announcements/view"  
        forward="portlet.announcements.view" />
```

Look up *portlet.announcements.view* in the *tiles-defs.xml*

```
<definition name="portlet.announcements.view"  
    extends="portlet.announcements">  
    <put name="portlet_content" value="/portlet/  
        announcements/view.jsp" />  
</definition>
```

view.jsp

```
<c:choose>
  <c:when test='<%=tabs1.equals("entries") %>'>
    <%@ include file="/html/portlet/
      announcements/view_entries.jspf" %>
  </c:when>
  <c:when test='<%=tabs1.equals("manage-entries") %>'>
    <%@ include file="/html/portlet/announcements/
      view_manage_entries.jspf" %>
  </c:when>
</c:choose>
```

view_manage_entries.jspf

```
<auiform action="<%= portletURL.toString() %>"
  method="post" name="fm">

  . . .

  <%
  boolean submitOnChange = true;
  %>

  <%@ include file="/html/portlet/announcements/
    entry_select_scope.jspf" %>

  . . .

</auiform>
```

entry_select_scope.jspf

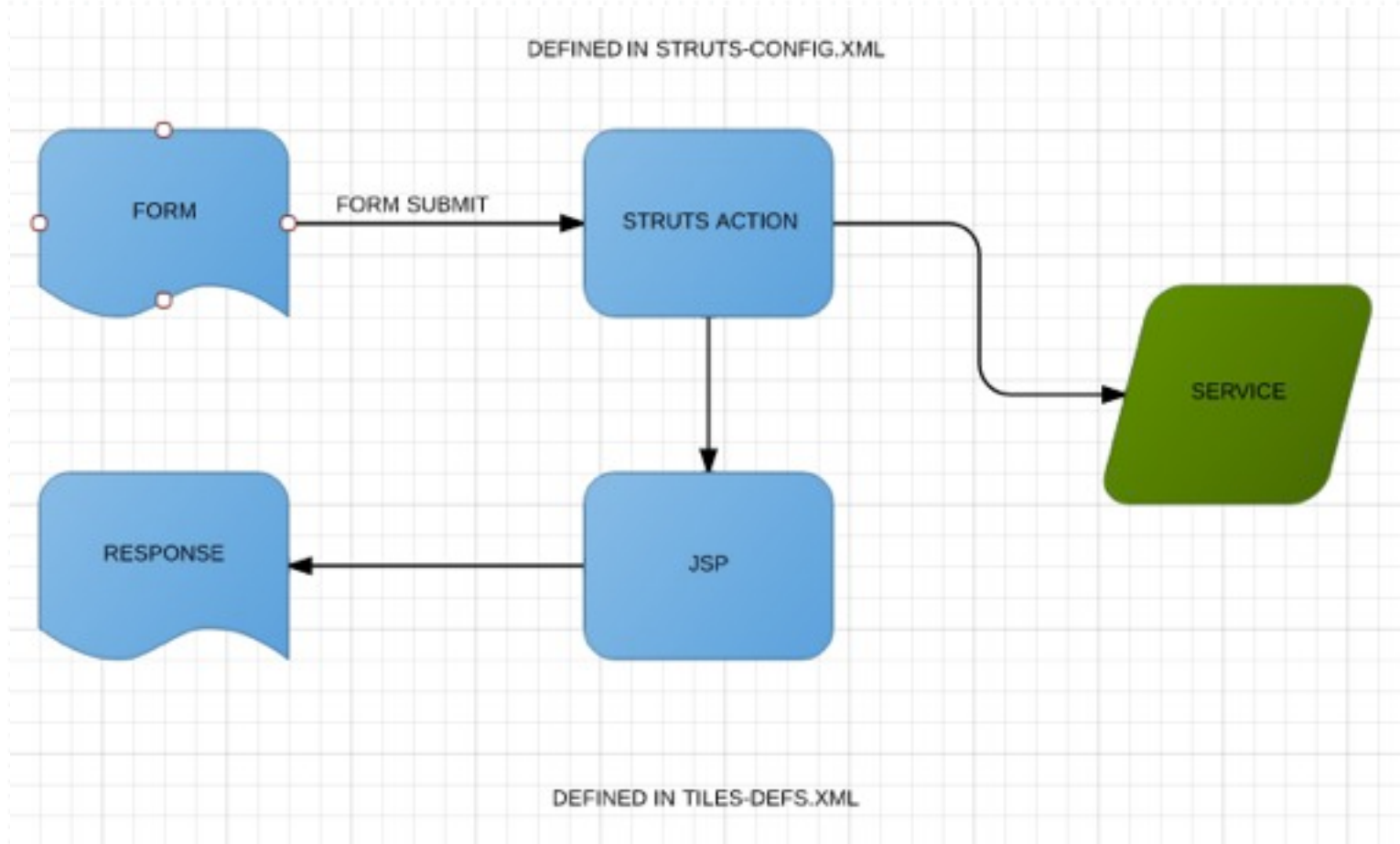
```
<au:select
  changesContext="<%= submitOnChange %>"
  name="distributionScope"
  onChange='<%= submitOnChange ?
    renderResponse.getNamespace() +
    "selectDistributionScope(this.value);" : "" %>'
  showEmptyOption="<%= true %>">
```

entry_select_scope.jspf

```
<auiselect
  multiple="true"
  changesContext="<%= submitOnChange %>"
  name="distributionScope"
  onChange='<%= submitOnChange ?
    renderResponse.getNamespace() +
    "selectDistributionScope(this.value);" : "" %>'
  showEmptyOption="<%= true %>">
```

Could it really be this easy?

Request Lifecycle



edit_entry.jsp

```
function <portlet:namespace />saveEntry() {
    document.<portlet:namespace />fm.action =
        '<portlet:actionURL>
            <portlet:param
                name="struts_action"
                value="/announcements/edit_entry"
            />
        </portlet:actionURL>';
        . . .
    submitForm(document.<portlet:namespace />fm);
}
```

Development Requirements

1. Modify JSPs
2. Modify Struts Action

Development Approach

- Hook Plugin vs Ext Plugin
 - Hook plugins are hot-deployable
 - Hook plugins can be more modular
 - Hook plugins can provide a cleaner upgrade path
 - Ext Plugins give access Liferay class loader
- Use Hook Plugins when you can
- Use Ext Plugins when you need to

Hands On Exercise

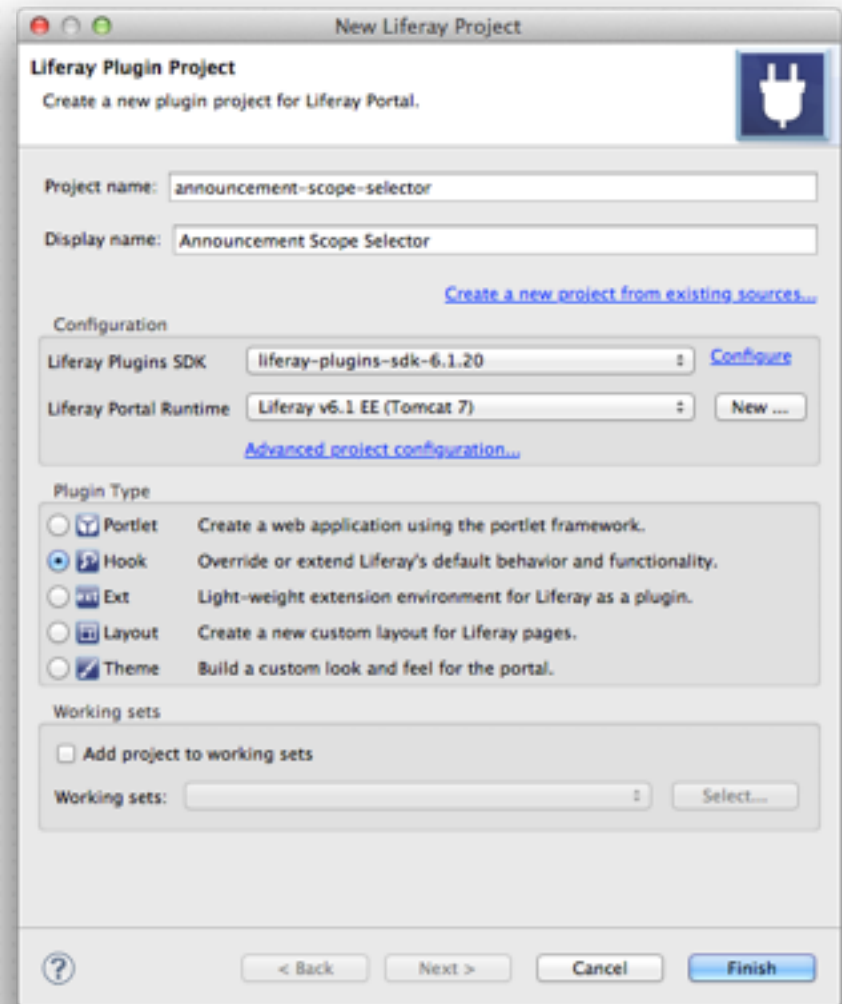
Create New Liferay Project

Project name:
announcement-scope-selector

Display name:
Announcement Scope Selector

Plugin Type: **Hook**

Click **Finish**

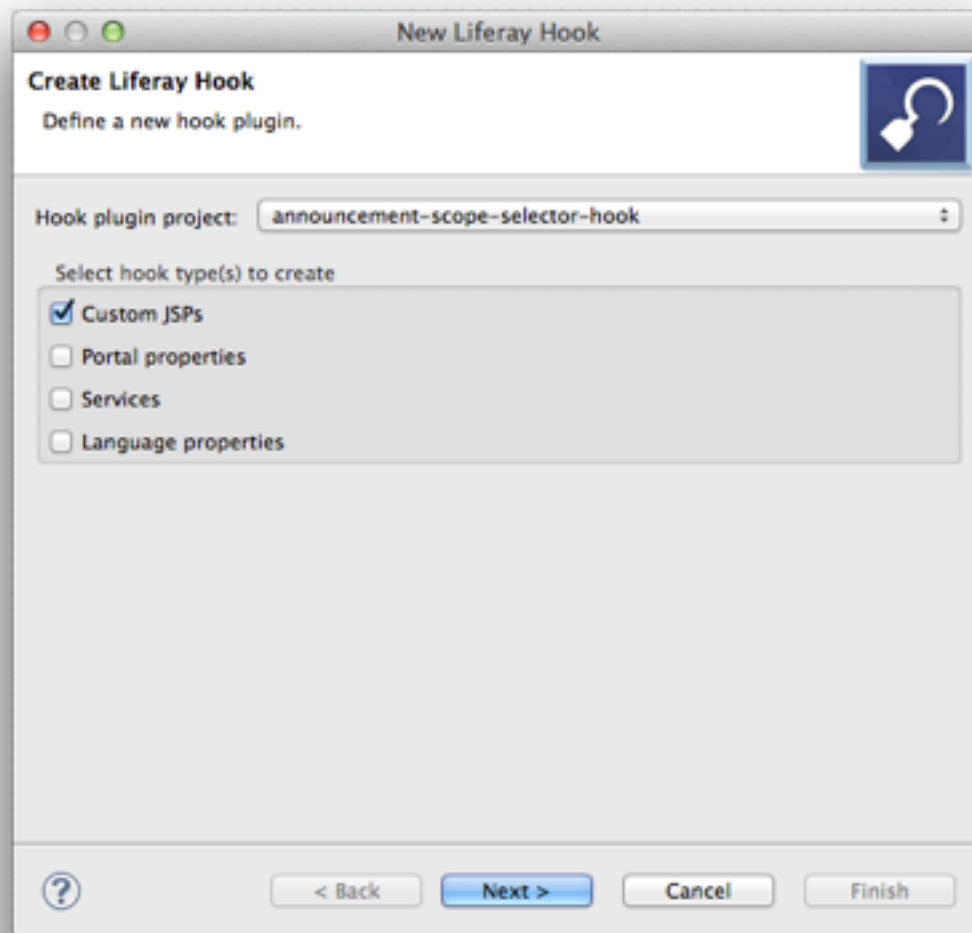


Create Liferay Hook

Right click on project and select
New > Liferay Hook

Select **Custom JSPs**

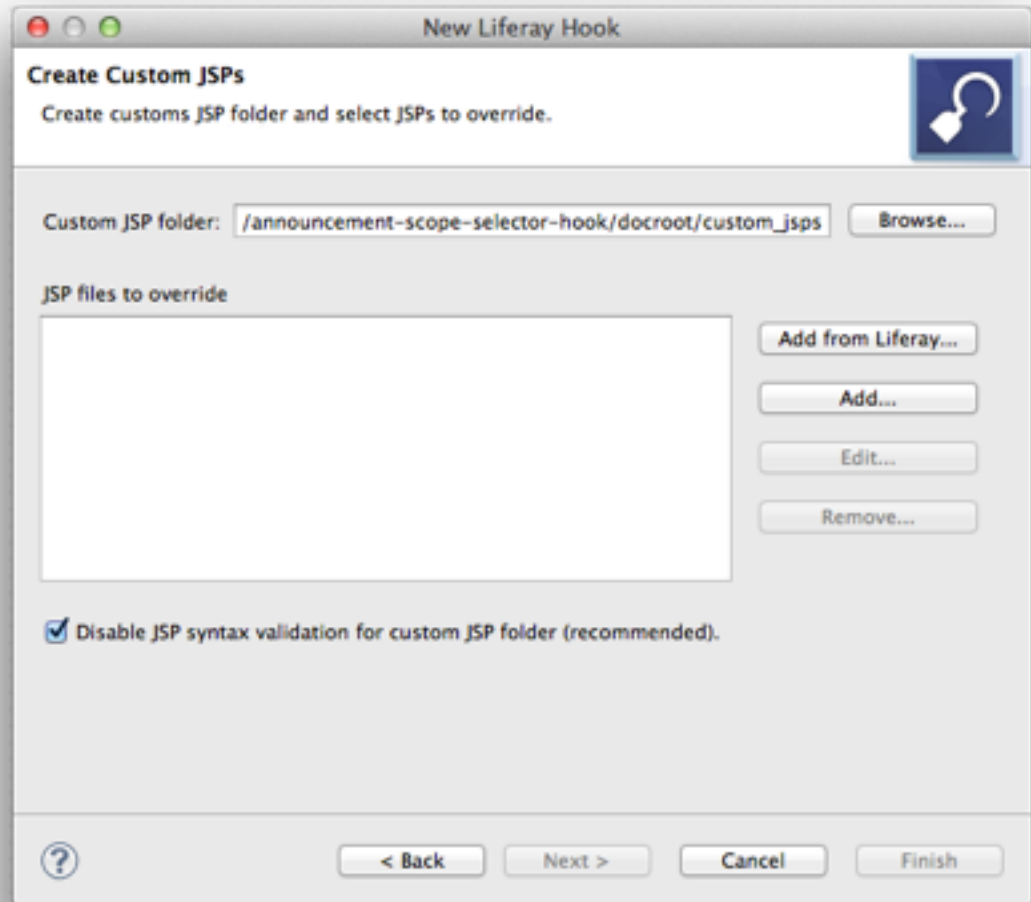
Click **Next**



Create Custom JSPs

Accept default Custom JSP Folder

Click **Add from Liferay ...**



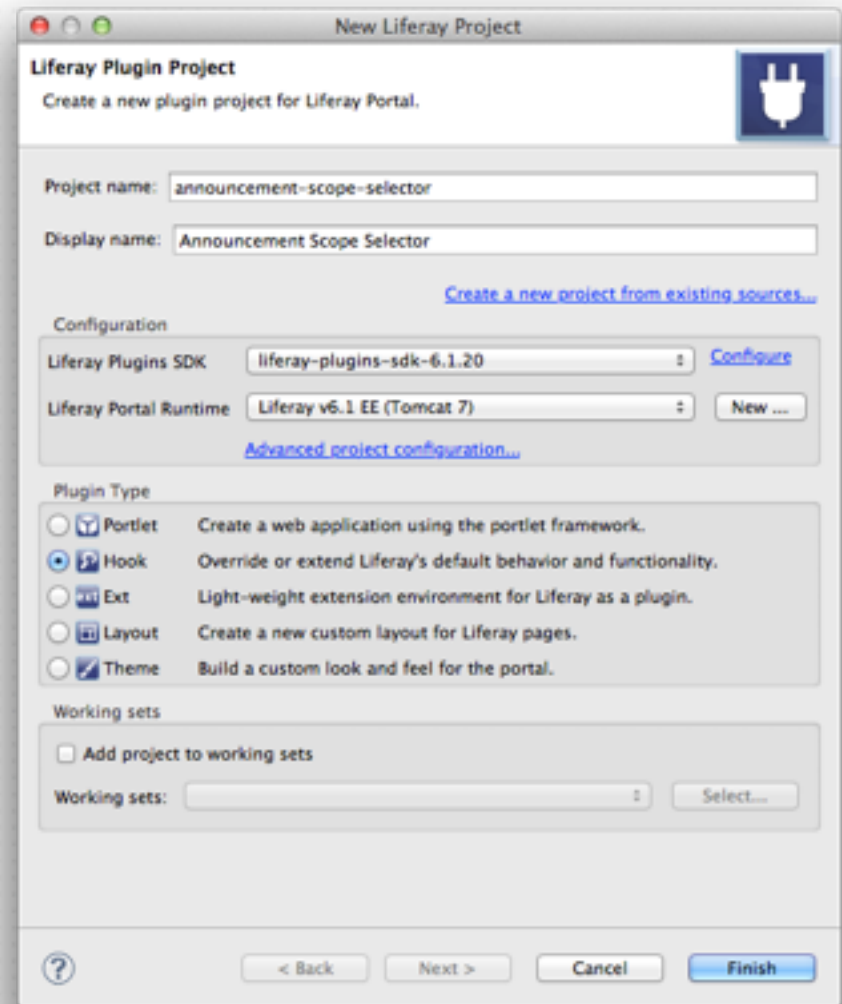
Select JSP to Customize

Browse for:
html/portlet/announcements/

Select
entry_select_scope.jspf

Click **Ok**

Click **Finish**



Update JSP

1. If necessary, open `entry_select_scope.jspf`
2. Insert snippet `01-entry_select_scope.jspf` between `<auri:select` and `changesContext=`
3. Save and close the file

Declare Struts Action Hook

1. Open *docroot/WEB-INF/liferay-hook.xml*
2. Switch to *Source* tab
3. Insert snippet *02-liferay-hook.xml* after the closing `</custom-jsp-dir>` and before the closing `</hook>` tag
4. Save and close the file

Create Struts Action Class - I

1. Create a new class

- Class Name: MyEditEntryAction
- Package: com.liferay.workshop.action
- Superclass: BaseStrutsPortletAction

2. Insert snippet *03-MyEditEntryAction-render* into the body of the class

```
import com.liferay.portal.kernel.struts.StrutsPortletAction;  
import javax.portlet.PortletConfig;  
import javax.portlet.RenderRequest;  
import javax.portlet.RenderResponse;
```

Create Struts Action Class - II

1. Insert snippet *04-MyEditEntryAction-processAction* into the body of the class

```
import com.liferay.portal.kernel.servlet.SessionErrors;
import com.liferay.portal.kernel.util.Constants;
import com.liferay.portal.kernel.util.ParamUtil;
import com.liferay.portal.kernel.util.Validator;
import com.liferay.portal.security.auth.PrincipalException;
import com.liferay.portal.util.PortalUtil;
import com.liferay.portlet.announcements.EntryContentException;
import com.liferay.portlet.announcements.EntryDisplayDateException;
import com.liferay.portlet.announcements.EntryExpirationDateException;
import com.liferay.portlet.announcements.EntryTitleException;
import com.liferay.portlet.announcements.EntryURLException;
import com.liferay.portlet.announcements.NoSuchEntryException;

import javax.portlet.ActionRequest;
import javax.portlet.ActionResponse;
```

Create Struts Action Class - III

1. Insert snippet *05-MyEditEntryAction-updateEntry* into the body of the class

```
import com.liferay.portal.kernel.util.GetterUtil;
import com.liferay.portal.kernel.util.StringUtil;
import com.liferay.portal.kernel.util.WebKeys;
import com.liferay.portal.theme.ThemeDisplay;

import java.util.Calendar;

import
    com.liferay.portlet.announcements.service. AnnouncementsEntryServiceUtil;
```

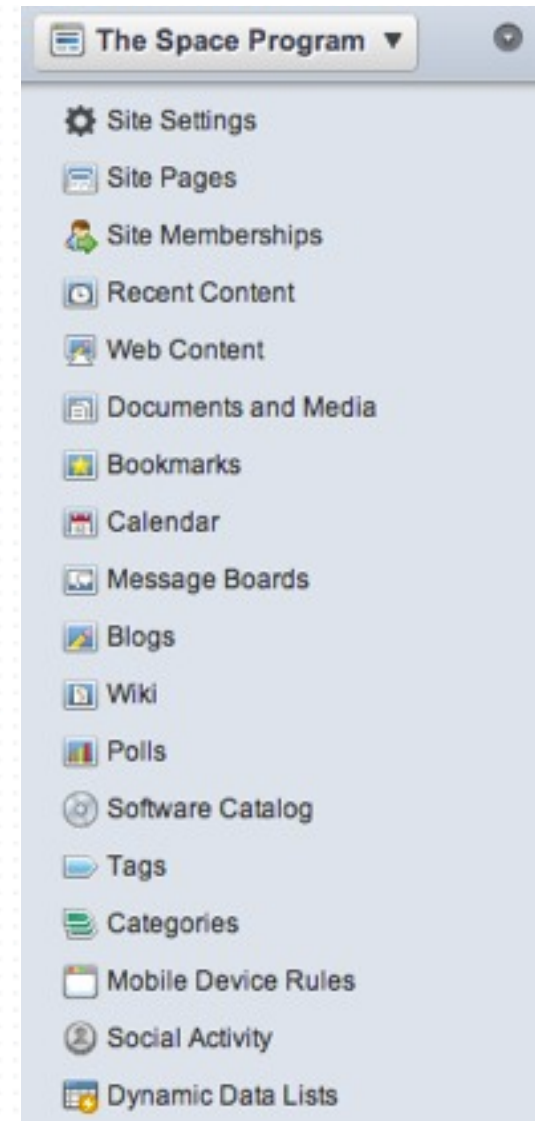
Create Struts Action Class

1. Insert snippet *06-MyEditEntryAction-deleteEntry* into the body of the class
2. Save
3. Deploy the portlet

Part 2: Plugin Portlets

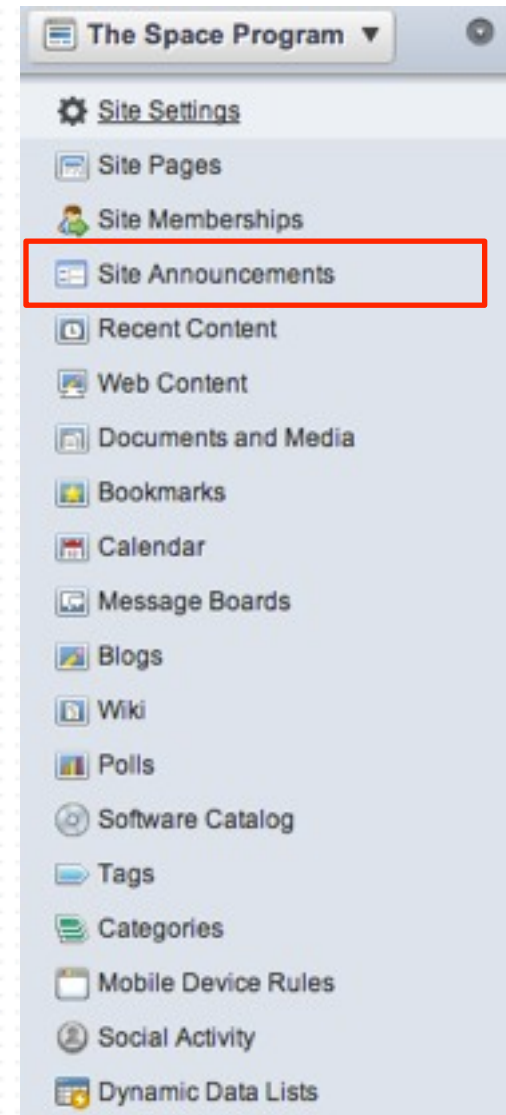
What about sending Announcements from the Control Panel?

Liferay provides a method to add portlets to the control panel, but the current announcement portlet interface doesn't meet our requirements.

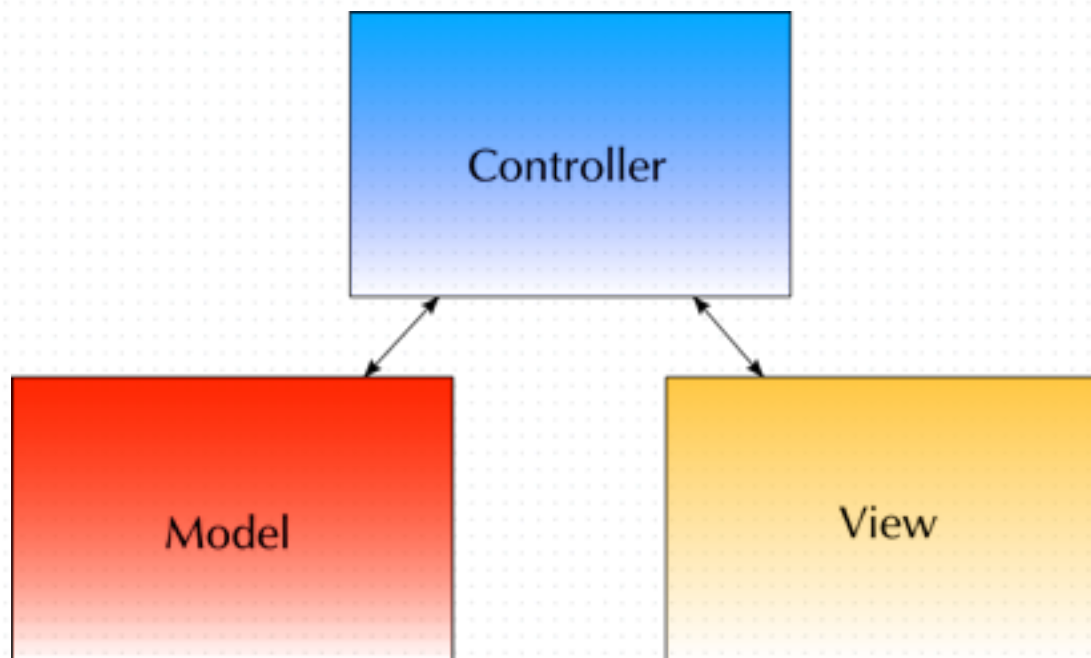


Business Requirements

- Provide Site Administrators with a Site Announcements portlet in the Control Panel
- Portlet should provide a simplified interface for creating, updating, and deleting announcements for current site
- Announcements should be visible in the existing Announcements portlet



Design Approach



- Controller - Liferay MVCPortlet
- Model - Service Builder
- View - JSPs

Liferay MVCPortlet

- MVCPortlet is very lightweight portlet framework
- Boilerplate code is hidden, but accessible when necessary
- Controlling page flows with MVCPortlet is accomplished through the `mvcPath` render parameter
- If the `mvcPath` parameter is present, the portlet dispatches to the page indicated in the parameter

Liferay Service Builder

- Liferay services are built using Liferay Service Builder
- Liferay portlets are consumers of services
- Liferay plugins can consume core portal services or services provided by other plugins

Liferay Service Builder

- Local services can be accessed from the local application server
 - EntityLocalService (Java interface)
 - EntityLocalServiceUtil (Service Locator)
 - EntityLocalServiceImpl (Java class)
- Remote services can be accessed from the local application server or remotely as web services.
 - EntityService (Java interface)
 - EntityServiceUtil (Service Locator)
 - EntityServiceImpl (Java class)

JSP Pages

- Portlets can leverage the same JSP knowledge you've gained working with Servlets
- The JSR-286 specification provides the portlet tag library to accomplish common portlets tasks
- Liferay provides several additional tag libraries including:
 - Liferay UI
 - Liferay Theme
 - Alloy UI

Hands On Exercise

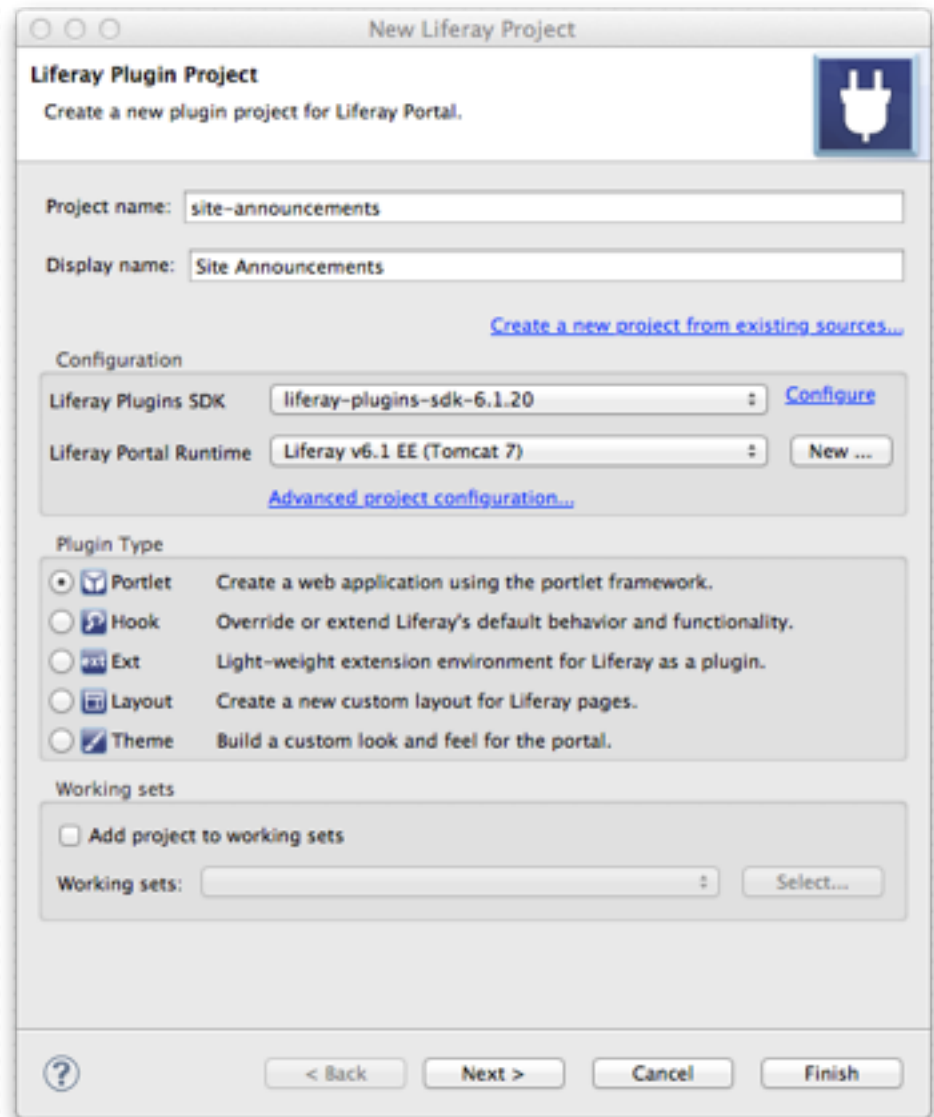
Create new project

Project name: **site-announcements**

Display name: **Site Announcements**

Plugin Type: **Portlet**

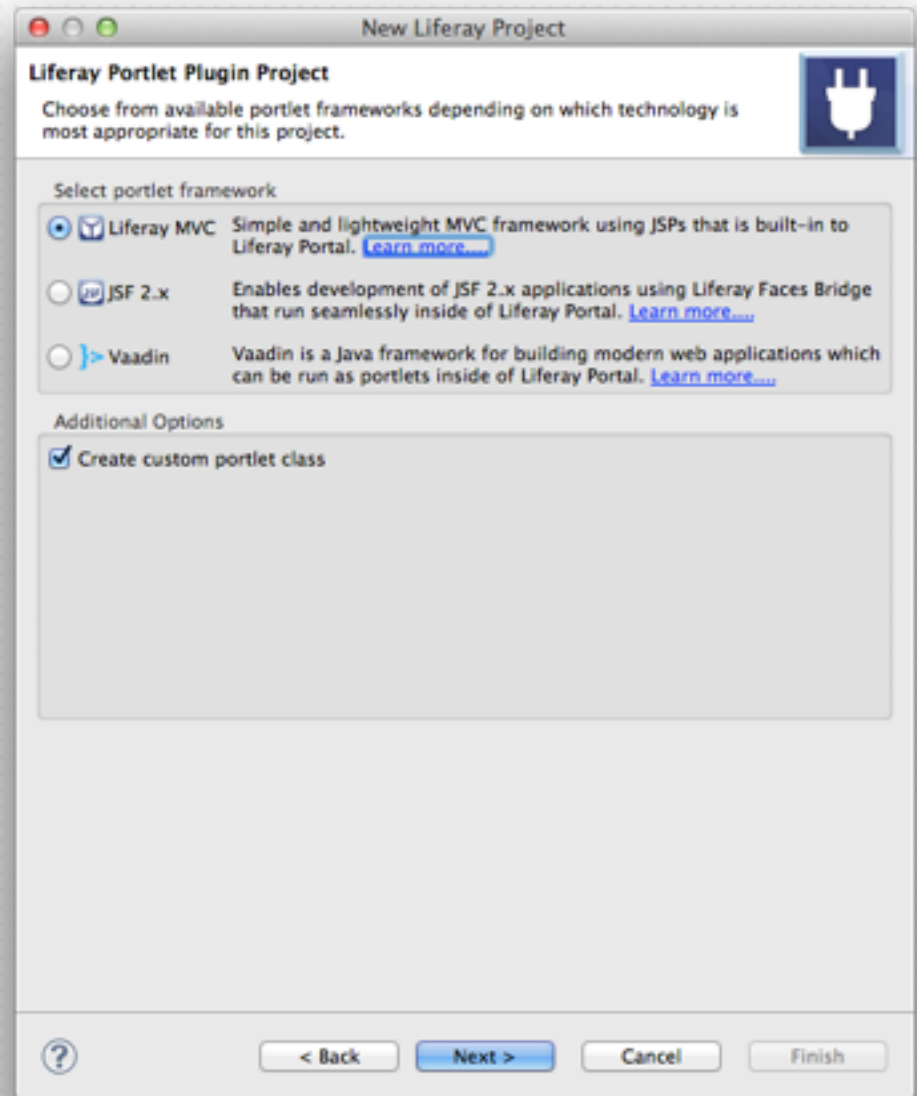
Click **Next**



Select Portlet Framework

Select *Create custom portlet class*

Click **Next**



Create Liferay Portlet

Portlet class:

SiteAnnouncementsPortlet

Java Package:

com.liferay.workshop.portlet

Superclass:

com.liferay.util.bridges.mvc.MVCPortlet

Click **Next**



Create Liferay Portlet

Update Display name & Title
Site Announcements

Click Next

New Liferay Portlet

Create Liferay Portlet
 Specify portlet deployment descriptor details.

Portlet Info

Name:

Display name:

Title:

Portlet Modes

View Edit Help

Liferay Portlet Modes

About Config Edit Defaults Edit Guest Preview Print

Resources

Create JSP files

JSP folder:

Create resource bundle file

Resource bundle file path:

? < Back Next > Cancel Finish

Create Liferay Portlet

Update Display Category:
category.hidden

Select ***Add to Control Panel***

Select ***Content Section***

Update Entry Weight: **3.5**

Click **Finish**

New Liferay Portlet

Create Liferay Portlet
Specify Liferay portlet deployment descriptor details.

Liferay Portlet Info

Icon: /icon.png
 Allow multiple instances

CSS: /css/main.css

JavaScript: /js/main.js

CSS class wrapper: site-announcements-portlet

Liferay Display

Display Category: category.hidden

Add to Control Panel

Entry Category: Content Section

Entry Weight: 3.5

Create Entry Class

Entry Class: SiteAnnouncementsPortletControlPanelEntry

? < Back Next > Cancel Finish

Create init.jsp

1. Create new file under *docroot/html/* called *init.jsp*
2. Insert snippet *01-init.jsp*
3. Save and close the file

Update view.jsp

1. Open *docroot/html/site_announcements/view.jsp*
2. Insert snippet *02-view.jsp-1*
3. Save and close the file

Create edit_announcement.jsp

1. Create new file under *docroot/html/site_announcements/* called *edit_announcement.jsp*
2. Insert snippet *03-edit_announcement.jsp*
3. Save and close the file
4. Deploy the portlet

Implement Update Action

1. Open *SiteAnnouncementsPortlet*
2. Insert snippet *04-SiteAnnouncementsPortlet-updateEntry*
3. Save and close the file

Update view.jsp

1. Open *docroot/html/site_announcements/view.jsp*
2. Insert snippet *05-view.jsp-2 at the bottom of the file*
3. Save and close the file

Create announcement_actions.jsp

1. Create new file under *docroot/html/site_announcements/* called *announcement_actions.jsp*
2. Insert snippet *06-announcement_actions.jsp*
3. Save and close the file

Implement Delete Action

1. Open *SiteAnnouncementsPortlet*
2. Insert snippet *07-SiteAnnouncementsPortlet-deleteEntry*
3. Save and close the file

Part 3: No Portlets

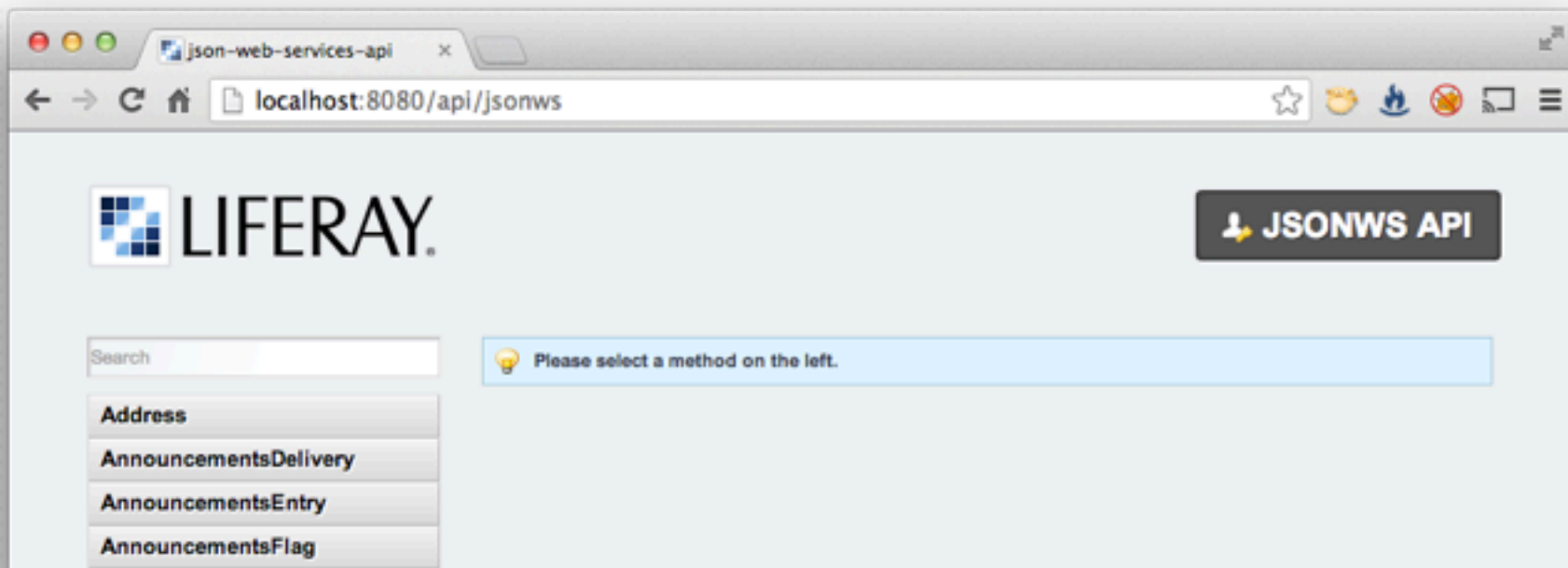
Liferay's Web Services

- Web Services are platform-independent resources which may be called over the HTTP protocol to return data.
- When the database classes are generated by the Service Builder, web services can be generated too
- As a result, nearly all of Liferay's back-end API calls can be made using web services
- Liferay's web services support SOAP, and JSON over HTTP

Accessing Liferay's Web Services


- To access SOAP based web services:
 - <http://localhost:<port>/api/axis> or
 - <http://127.0.0.1:<port>/api/axis>
- To access JSON over HTTP Web Services
 - <http://localhost:<port>/api/jsonws> or
 - <http://127.0.0.1:<port>/api/jsonws>


Announcements Entry Web Service





Announcements Entry Web Service


Address
AnnouncementsDelivery
AnnouncementsEntry
add-entry
add-entry
delete-entry
get-entry
update-entry

 **/announcementsentry/get-entry**
 com.liferay.portlet.announcements.service.[AnnouncementsEntryServiceUtil](#)#**getEntry**

 **Parameters**
 entryId long

 **Return Type**
 com.liferay.portlet.announcements.model.[AnnouncementsEntry](#)

 **Exception**
 com.liferay.portal.kernel.exception.[PortalException](#)
 com.liferay.portal.kernel.exception.[SystemException](#)

 **Execute**
 entryId

 long

Business Requirements

- Provide a web service for displaying announcements for a given user
- Web service should provide option to display either *unread* or *read* messages

Hands On Exercise

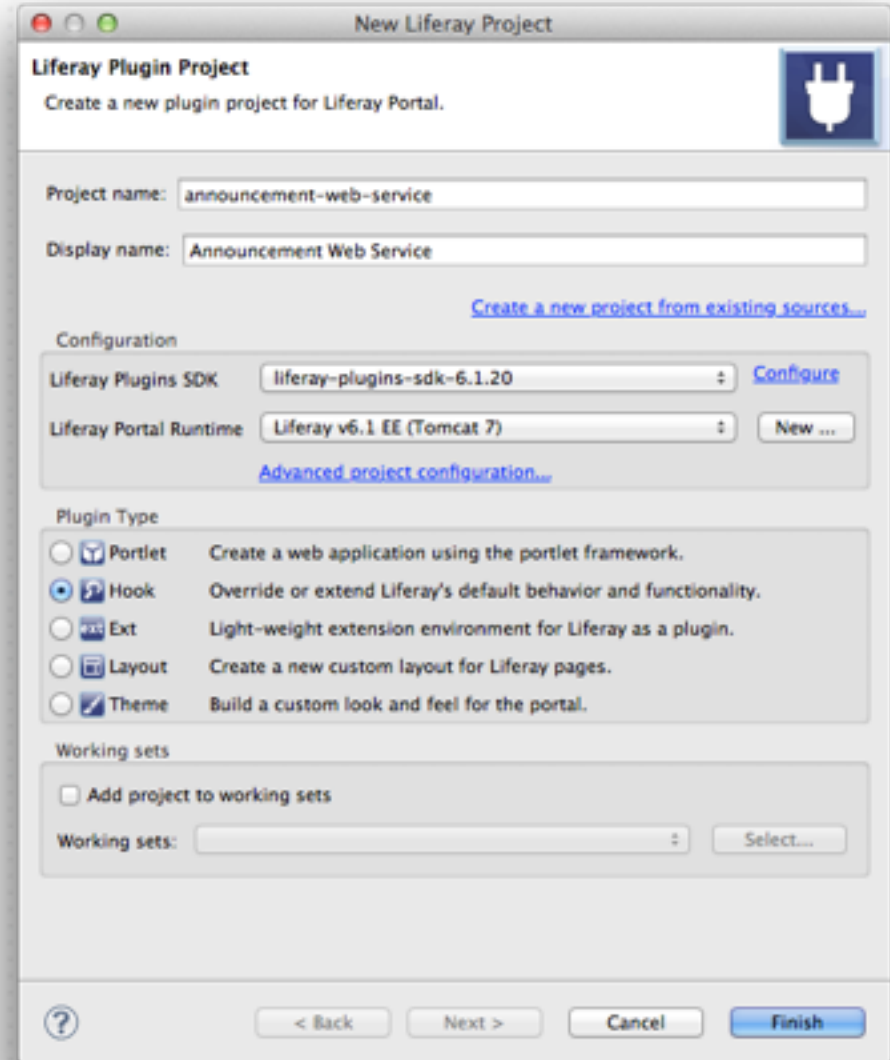
Create new project

Project name:
announcement-web-service

Display name:
Announcement Web Service

Plugin Type: **Hook**

Click **Finish**



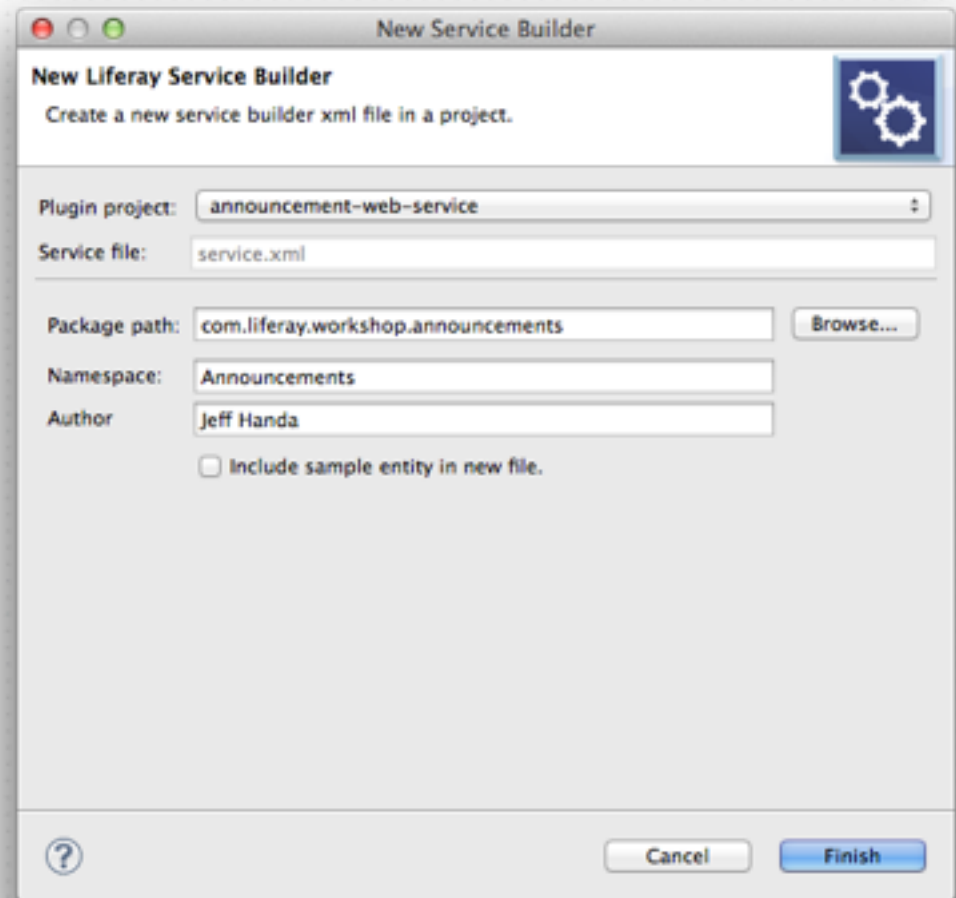
New Liferay Service Builder

Right click on project and select
New > Liferay Service Builder

Package Path:
com.liferay.workshop.announcements

Namespace:
Announcements

Click **Finish**



Create Service

1. Switch to the *Source* tab
2. Replace the empty entity tag with snippet *01-service.xml*
3. Save
4. Build Services

Expose Remote Methods

1. Open *MyAnnouncementsEntryServiceImpl*
2. Insert snippet *02-MyAnnouncementsEntryServiceImpl* into the body of the class
3. Build Services

Fix SOAP Issues

1. Open *MyAnnouncementsEntryServiceSoap*
2. Replace **AnnouncementsEntry[]** with **AnnouncementsEntrySoap[]**
3. Replace **AnnouncementsEntry[returnValue.size()]** with **AnnouncementsEntrySoap[returnValue.size()]**
4. Build WSDD
5. Deploy

* <http://issues.liferay.com/browse/LPS-12629>

Access JSON Services

1. Navigate to:

<http://localhost:8080/announcement-web-service-hook/api/jsonws>

Questions?

#ExtendLRWkshp



San Francisco, USA
October 20-22, 2013