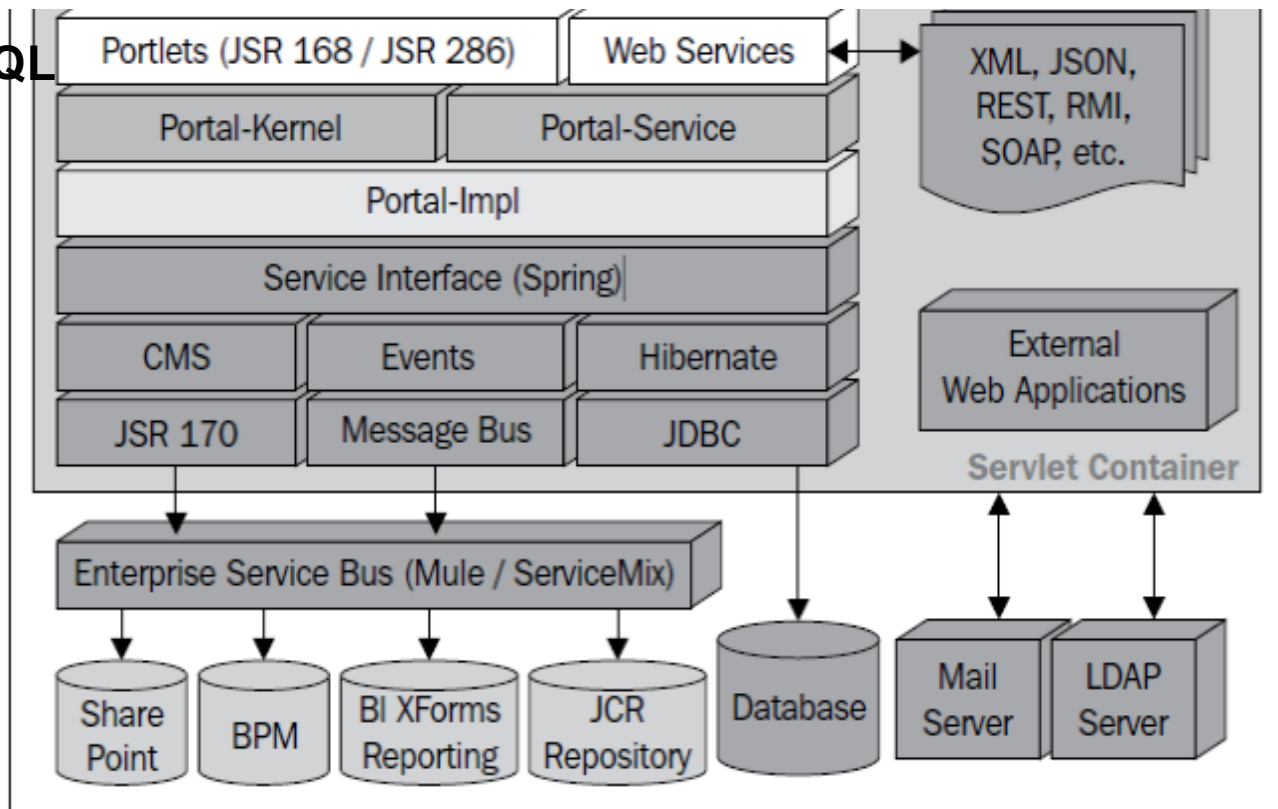




Custom Development on the Liferay Platform

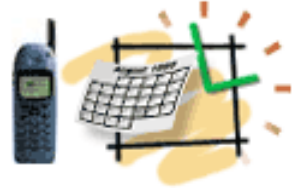
- Repetition:
- **Liferay-Tomcat & MySQL**
- Liferay EXT and SDK
- Ant, Subversion
- Tomcat Plugin
- Get Portal Source
- Build EXT & SDK



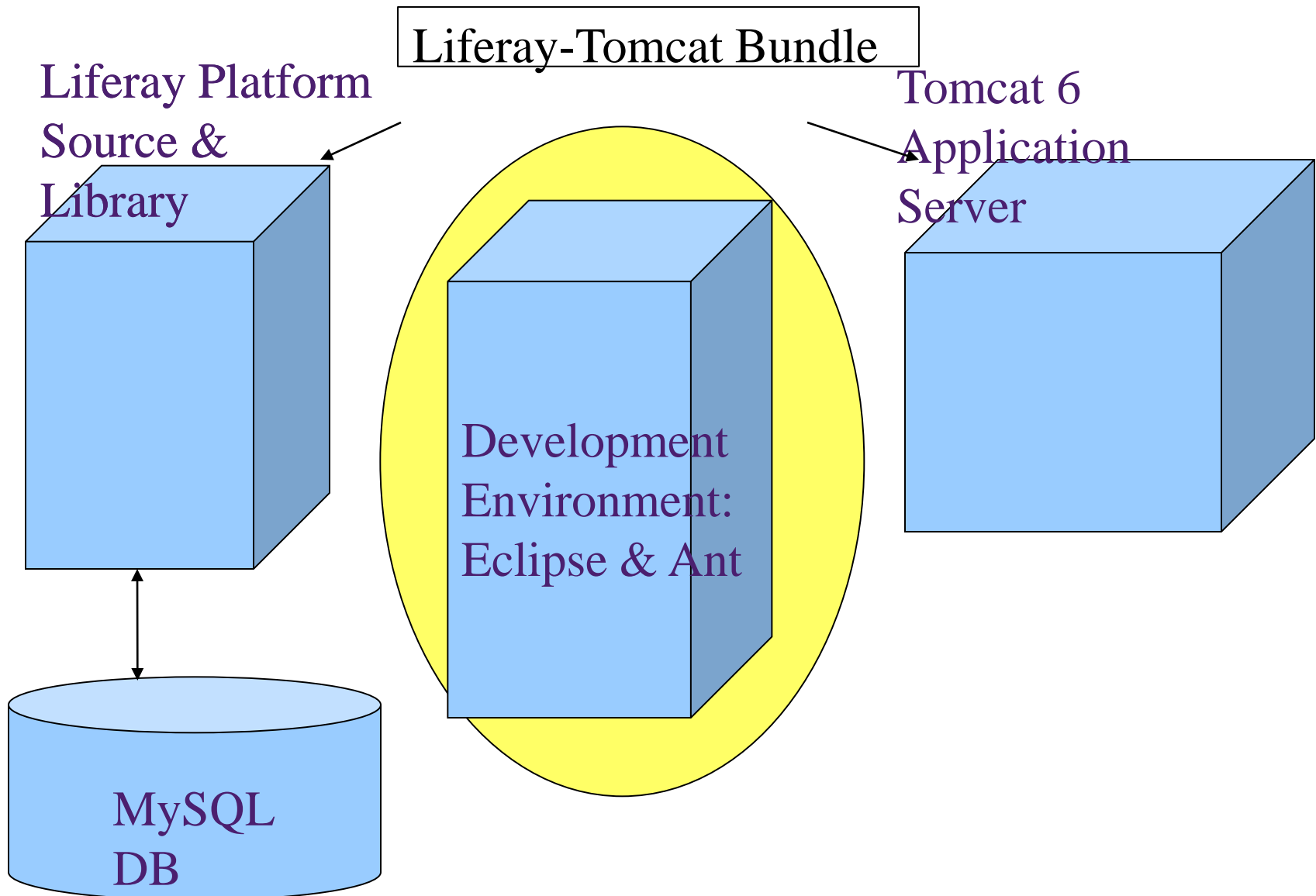
Custom Development

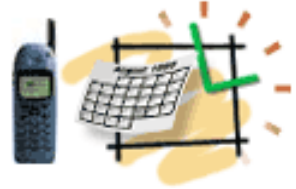
- JSP & Action
- International Msgs

Complementary Semantics (DataService) Frameworks: Data, Services, Rules & Scenarios

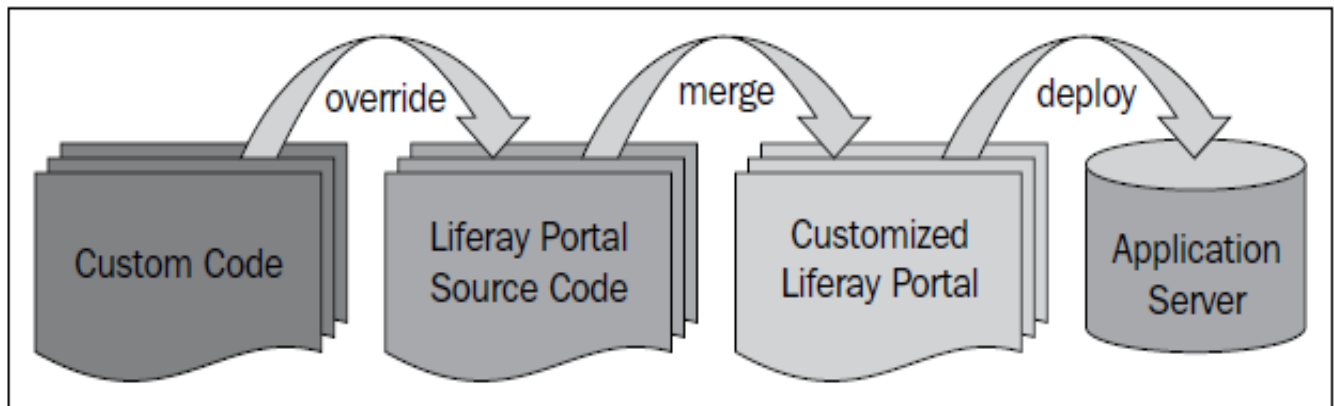
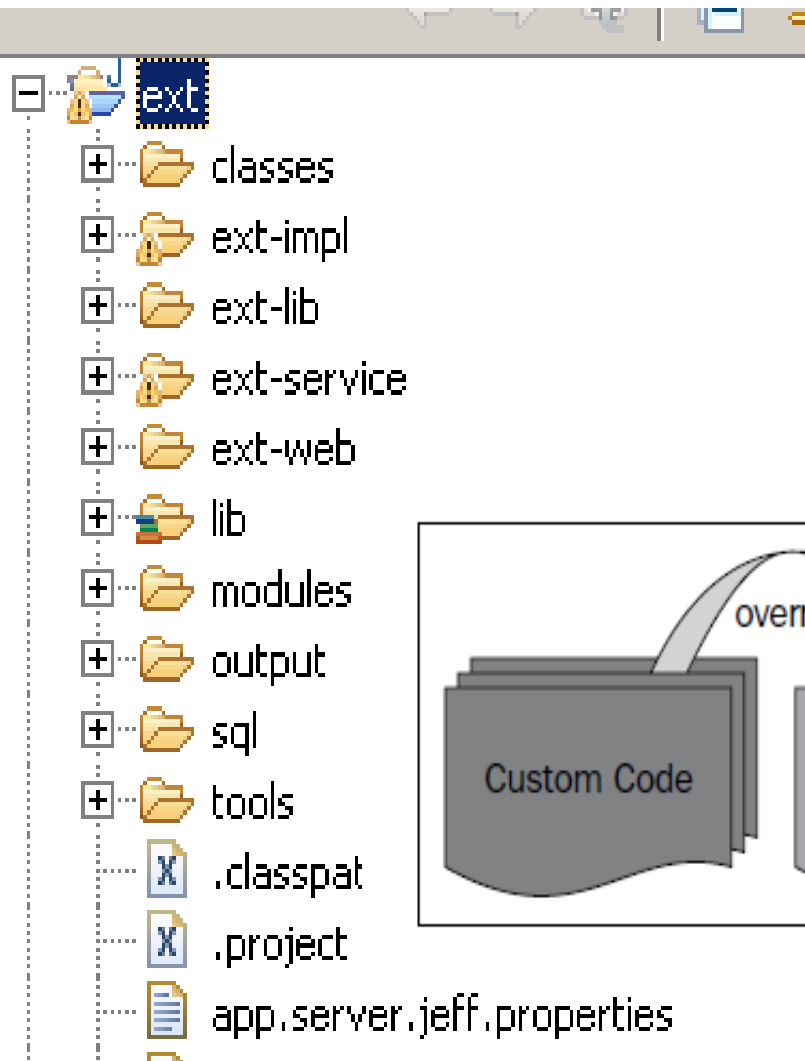


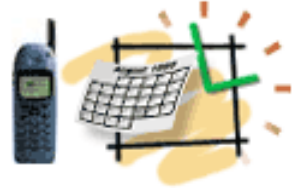
Basic Components





EXT Environment



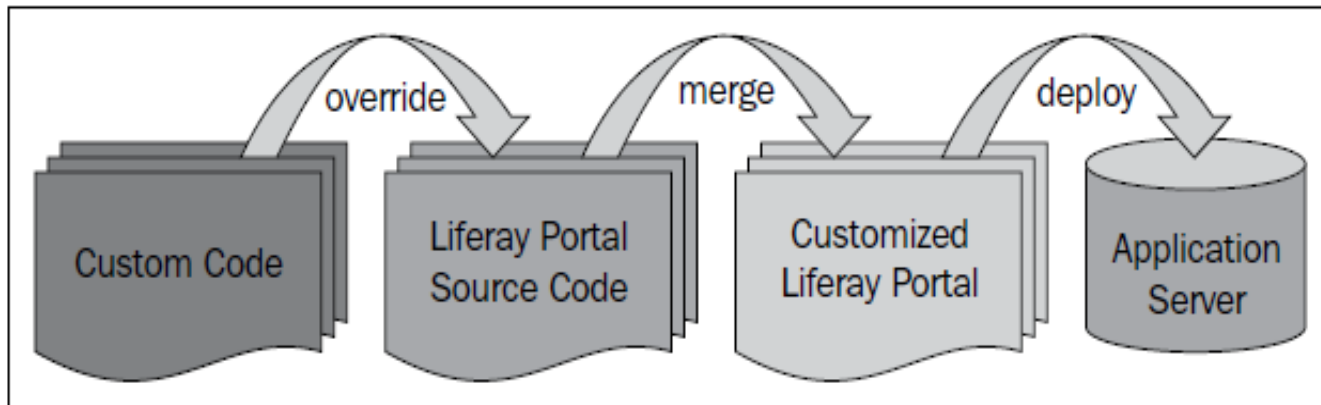


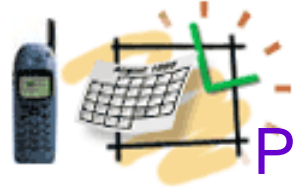
EXT Environment

What is EXT?

Ext is a wrapper for Liferay portal core source directories (that is, ext-impl/, ext-service/, and ext-web/).

EXT allows the users to develop on top of Liferay source





Check the Deployment and Start Liferay!

Portal API Library Jars are deployed in the Tomcat6/lib/ext

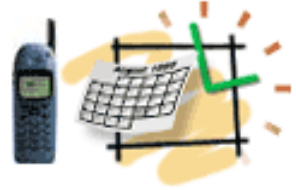
Start Liferay!

Debug Your Application!

Liferay + EXT

Clean and Deploy-Fast (without Re-Start) all files from ext-web

The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays the project structure under CATALINA_HOME, with the 'ROOT' folder expanded to show 'Liferay + EXT'. The Outline view on the right shows the 'ext-web' project with a list of tasks. The 'clean' and 'deploy-fast' tasks are highlighted with red arrows. The 'Start Liferay!' and 'Debug Your Application!' buttons are also highlighted with red arrows.



Play with Portal and Portlets

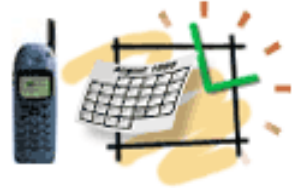
Recollect JSR 286

(Java Specification Request)



A Portlet is an application that presents a fragment of a portal page

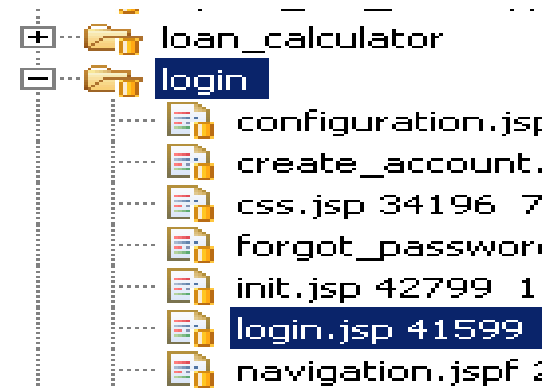
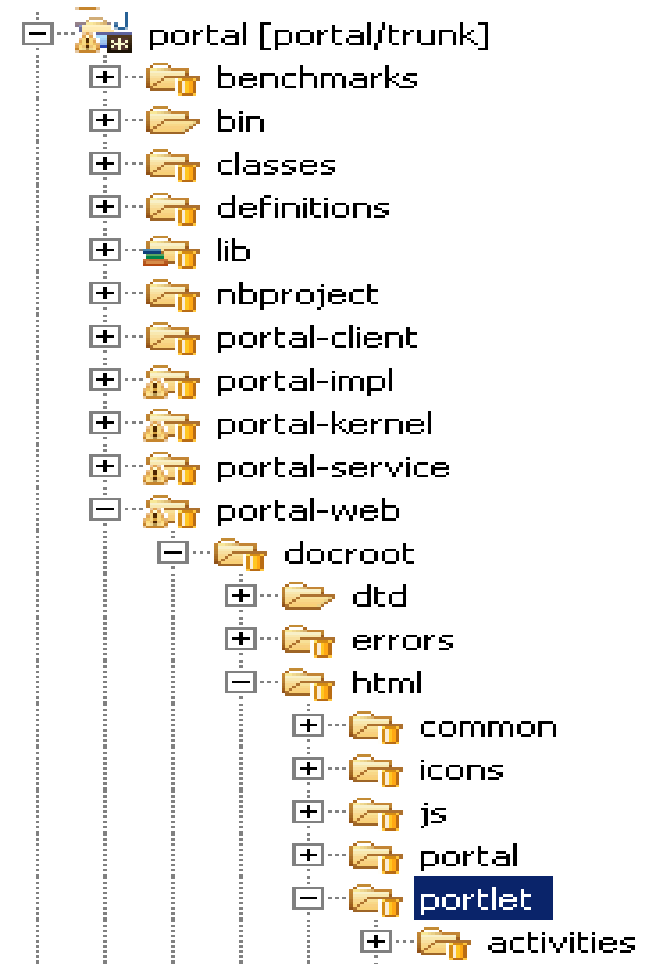
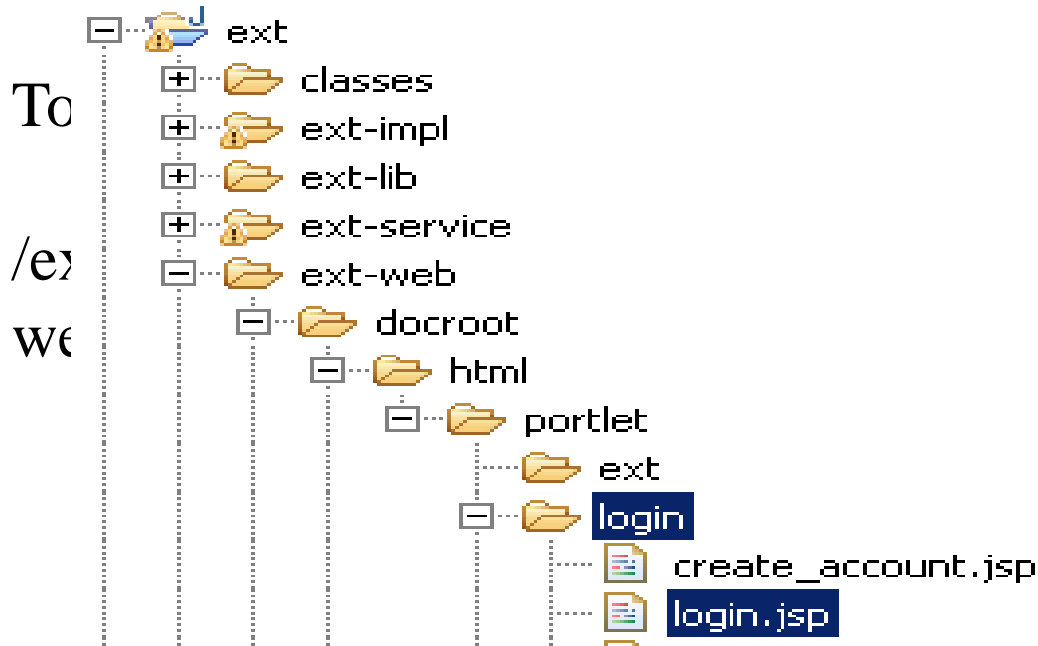
A Portlet Container assembles the portlets into a portal page

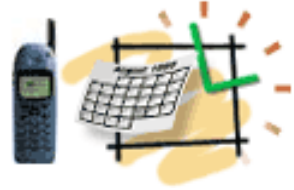


Customize Liferay

Create the folder: /ext/ext-web/docroot/html/portlet/login
Copy one file:

/portal/portal-web/docroot/html/portlet/login/login.jsp





Customize the Login.JSP

Double click and Open in Eclipse the file:

```
/ext/ext-web/docroot/portlet/login/login.jsp
```

Look at the Login screen, we'd like to change the login label to

“Enter your login information”

Find this line in the login.jsp source in Eclipse:

```
<alui:input label="<%= loginLabel %>" name="login" type="text" value="<%= login %>" />
```

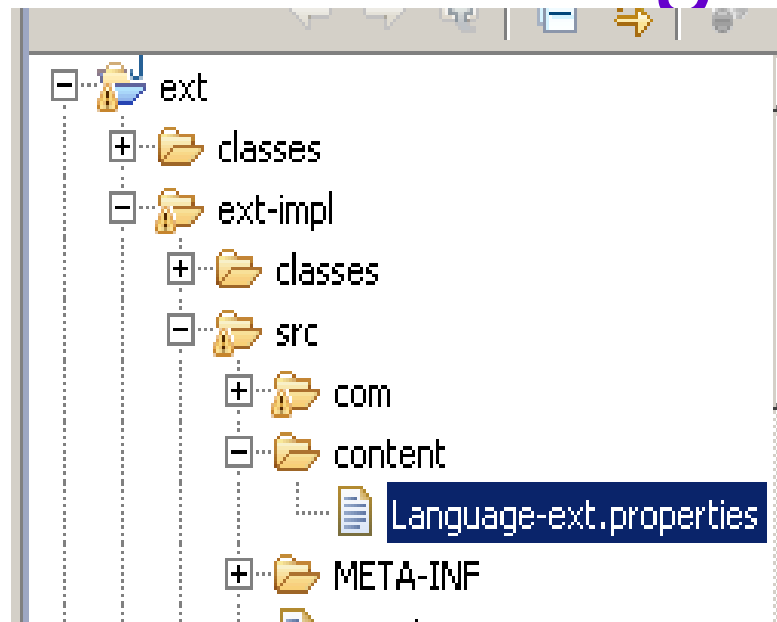
Replace this line with:

```
<alui:input label="enter-your-login-information" name="login" type="text"  
value="<%= login %>" />
```




Customize International Messages

Open in Eclipse the file:



[/ext/ext-impl/src/content/Language-ext.properties](#)

Add to the file this line:

enter-your-login-information=[Enter your login information](#)

Deploy-Fast, Start Liferay and check your updated Login



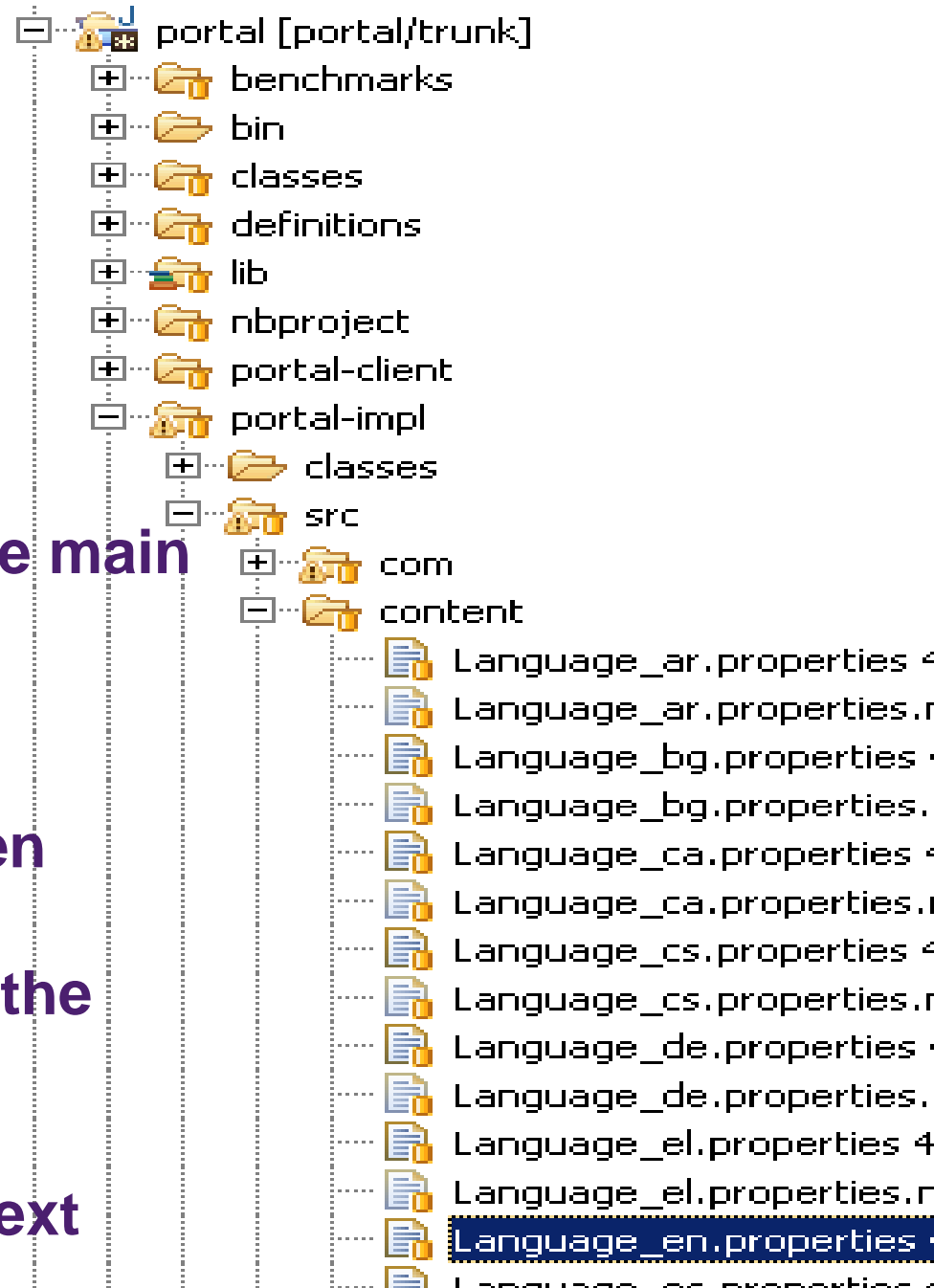
Customize International Messages

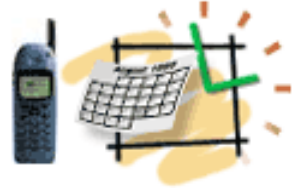
In the future always check the main
file of portal messages

/portal/portal-
impl/src/content/Language_en

before you add a new line to the

/ext/ext-
impl/src/content/Language_ext





Add New Functionality

Student Information Card

login

students

Jeffco ET > osic > Student Information Card

Enrollment: | [UpdateAccount](#) | [Login](#) | [Students](#) | [StudentPrimaryData](#) | [StudentResidence](#) | [StudentEnrollment](#) |

Administration: [Change debug level to HIGH \(more messages\)](#), [Check Data Status](#), [Sync data](#), [Capture rules](#) and [Register Service](#)

student-data

- [Student](#)
- [Residence](#)
- [Enrollment](#)
- [Language](#)
- [Health](#) ← **Link**
- [Contacts](#)

Student Health

💡 This section will be editable after creating the user.

Student Health Data to be replaced by real text fields and values

💡 This section will be editable after creating the user.

`/ext-web/docroot/html/portlet/osicdao/studentHealth.jsp`
`&action-page=StudentHealth`

`/ext-impl/src/com/jeffco/osicdao/action/StudentHealthPortletAction.java`

`/ext-impl/src/com/jeffco/osicdao/beans/StudentHealthBean.java`



Provide the classPath and jspPath in the tomcat/webapps/appNameKeys.xml

```
<appNameKeys>
  <appNames>its,osicdao </appNames>

  <its-customKeys>

newAccountQuestion,true;
customPassword,true;
create-account,new-to-its;
classPath,com.its.service;
jspPath,/html/portlet/its;
comments-label,brand-ip-administration;
dataAdminGroupName,DataAdmin;

  </its-customKeys>

  <osicdao-customKeys>
classPath,com.osicdao.action;
jspPath,/html/portlet/osicdao;
  </osicdao-customKeys>

</appNameKeys>
```

First, list the names of the portlets, in this case: **its** and **osicdao**

Then, for each portlet provide their custom keys that are expected by Liferay or your custom code.

Each key name is followed by the value, separated by comma “,”. Each pair is separated by semi-column “;”.

For example for the JSP Path:
jspPath,/html/portlet/its;

Then you create the yourAction.java and yourJSP code.

No struts-config changes is necessary, if you keep naming conventions provided in the following examples



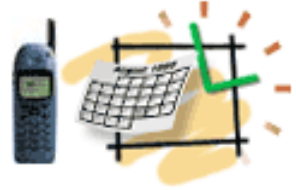
public class StudentHealthPortletAction implements PortletService {

```
/**
 * The run() method is an implementation of the PortletService interface
 */
public String run(RenderRequest renderRequest) throws Exception {
    HttpServletRequest request = PortalUtil.getHttpServletRequest(renderRequest);
    HttpSession session = request.getSession();
    // more code
    String creatorUserId = (String) session.getAttribute("userId");
    String cmd = request.getParameter("cmd"); // can be updateStudentHealth
    String userId = request.getParameter("userId");
    if(userId == null || userId.length() == 0) {
        msg += PortletDataAction.getIntlMsg(session, "create-student-first");
        return msg;
    } else if(cmd != null && cmd.equals("updateStudentHealth")) {
        msg += updateStudentHealth(request); // update, then display
    }
    // prepare to display
    java.util.HashMap map = new java.util.HashMap();
    map.put("userId", userId);
    List beans = DataService.getData("user/getStudentHealth", map,
StudentHealthBean.class, UserService.osicDsName);
    StudentHealthBean bean = (StudentHealthBean) beans.get(0);
    renderRequest.setAttribute("studentHealthBean", bean);
    return msg;
}
```



```
public String updateStudentHealth(HttpServletRequest  
request) {
```

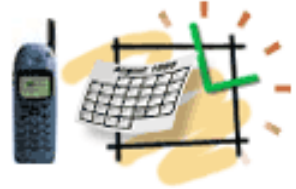
```
    HttpSession session = request.getSession();  
    String appName = (String) session.getAttribute("appName");  
    String msg = "";  
    // get form fields and update student health table  
    String userId = request.getParameter("userId");  
    java.util.HashMap map = new java.util.HashMap();  
    map.put("userId", userId);  
    // more fields to prepare for the update  
    try { // create the file "user/updateHealth.sql" in the WEB-INF/sql  
        DataService.setData("user/updateHealth", map,  
UserService.osicDsName);  
    } catch(Exception e) {  
        Stats.addAppList(appName, "user/updateHealth", "ERROR for  
userId="+userId + ": " +e);  
    }  
    return msg;  
}
```



studentHealth.jsp

```
<%@ include file="/html/portlet/login/init.jsp" %>
<%@ page import="java.util.*" %>
<%@ page import="com.its.util.*" %>
<%@ page import="com.jeffco.osicdao.portlet.UserService" %>
<%@ page import="com.jeffco.osicdao.beans.*" %>

<%
request.setAttribute("currentScreen", "StudentHealth");
String appName = (String) session.getAttribute("appName");
String portalActionLink = Stats.getAppDetailsByKeyName(appName, "actionLink");
String userId = request.getParameter("userId");
String formAction =
portalActionLink+"&action=StudentHealth&appName="+appName;
if(userId == null) {
userId = "";
}
StudentHealthBean bean =
(StudentHealthBean) request.getAttribute("studentHealthBean");
%>
```



studentHealth.jsp (Continue)

```
<%@ include file="/html/portlet/login/osic-header.jsp" %>
```

```
<div id="container">
```

```
<%@ include file="/html/portlet/osicdao/student-nav.jsp" %>
```

```
<div id="content" class="news">
```

```
<h3>
```

```
<liferay-ui:message key="student-health" />
```

```
</h3>
```

```
<%if(userId == null || userId.length() == 0) { %>
```

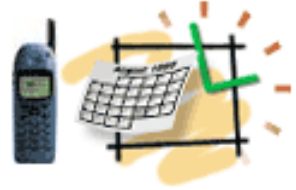
```
    <div class="portlet-msg-info">
```

```
<liferay-ui:message key="this-section-will-be-editable-after-creating-the-user" />
```

```
    </div>
```

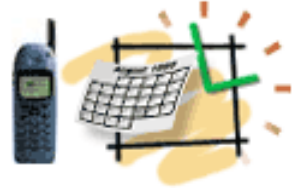
```
<%} %>
```

```
<div class="border" >
```

studentHealth.jsp (End)

```
<form action="<%= formAction %>" method="post" name="fm" onsubmit="return
checkEmptyTextFields(this);">
  <input name="userId" type="hidden" value="<%=userId%>" />
  <input name="cmd" type="hidden" value="updateStudentHealth" />
  <center>
  <h1>Student Health Data to be replaced by real text fields and values</h1>
  </center>
  <%if(userId == null || userId.length() == 0) { %>
    <div class="portlet-msg-info">
  <u>liferay-ui:message key="this-section-will-be-editable-after-creating-the-user" />
    </div>
  <%} else { %>
  <u>auib:button-row>
    <u>auib:button type="submit" />
  </u>
  </u>
  <%} %>
</form>
</div>
</div>
</div>
```



StudentHealthBean.java

```
package com.jeffco.osicdao.beans;
```

```
public class StudentHealthBean {  
private long userId;
```

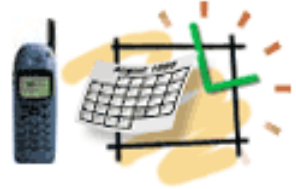
```
// more fields
```

```
}
```



Liferay Custom Development Summary

- **Main components?**
- **Creating an action-page link**
- **Creating a {Name}PortletAction**
- **Creating an {name}JSP**
- **Creating a {Name}Bean**



Liferay Setup Summary

- **What we did first?**
 - **Liferay-Tomcat bundle install**
- **Next?**
 - **MySQL and Tools**
- **Next?**
 - **EXT Environment**