Integrazione tra JSF e Liferay

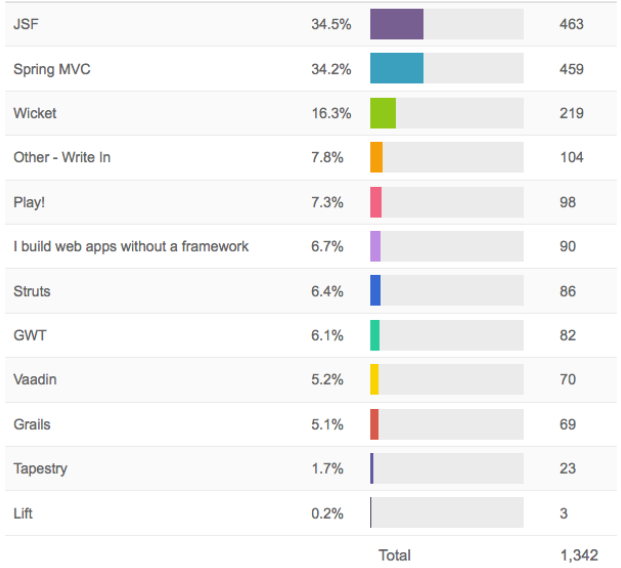
Overwiev around JSF Integration

Gian Marco Vernile, Software Engineer @ Key-Biz

Perchè trattare questo argomento?!

- JSF è una tencologia standard JEE
- Pienamente compatibile con le Portlet 2.0
- Abbattimento tempi di startup
- Knowhow elevato di Javascript , HTML e CSS non richiesto
- Completa integrazione con i componenti di Liferay

JSF prima scelta per gli sviluppatori



JSF	34.5%	463
Spring MVC	34.2%	459
Wicket	16.3%	219
Other - Write In	7.8%	104
Play!	7.3%	98
I build web apps without a framework	6.7%	90
Struts	6.4%	86
GWT	6.1%	82
Vaadin	5.2%	70
Grails	5.1%	69
Tapestry	1.7%	23
Lift	0.2%	3
	Total	1,342

https://dzone.com/articles/poll-what-java-jvm-frameworks-do-you-use

JCP

La Java Community Process esiste per lo sviluppo di standard per tecnologie Java

http://www.jcp.org

JSF è …

- un framework per la realizzazione di web app
- basato sul pattern MVC
- Computed Oriented
- Event Driven
- User Interface (UI) Development
- Validazione ed Error Feedback

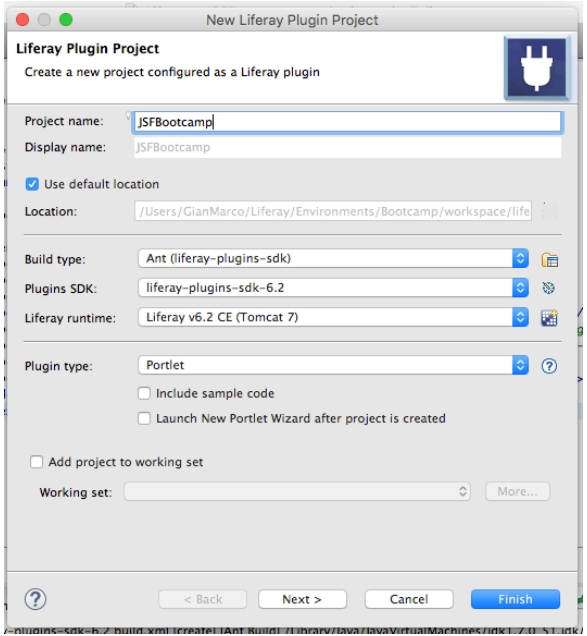
Liferay Faces: Portal 6.2 Support

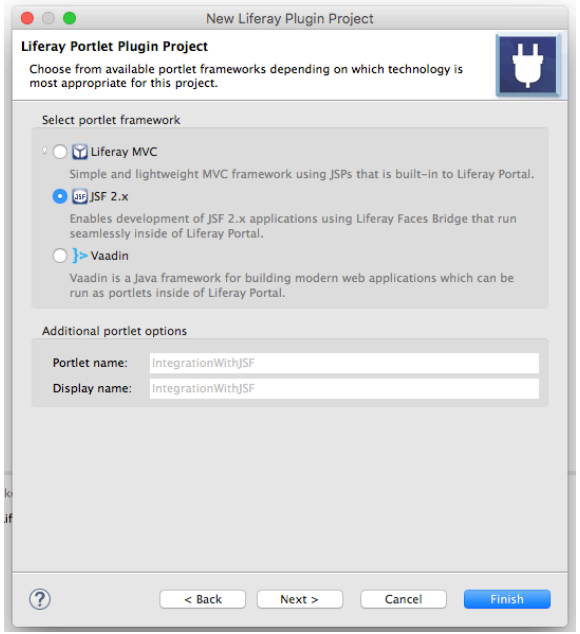
- Liferay Faces GA6 rilasciata l' 11 Agosto 2015
- Nuova suite Liferay Faces Alloy (per AlloyUI 2.0 / Liferay Portal 6.2)
- Showcase online : http://www.liferayfaces.org/web/guest/showcase

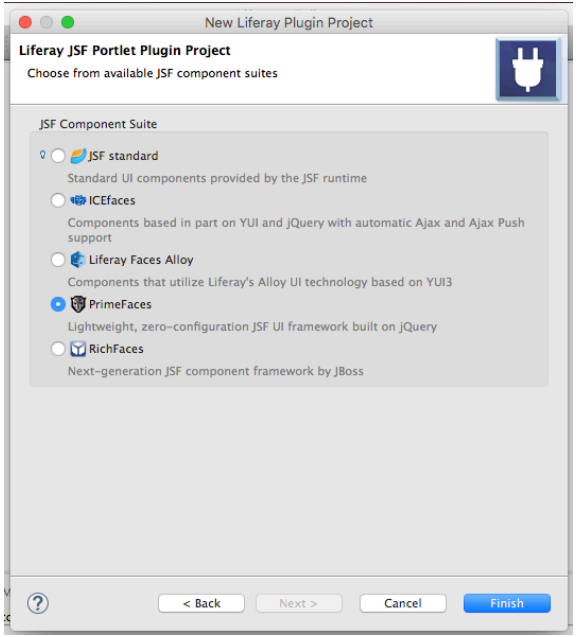
Liferay Faces : overview

- Liferay Faces Alloy
  alloy:
- Liferay Faces Bridge
  bridge:
  portlet:
- Liferay Faces Portal
  portal:

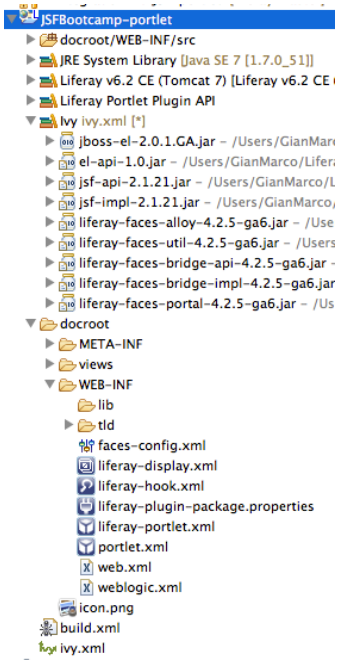
Let’s go!


New Liferay Plugin Project
Liferay Plugin Project
Create a new project configured as a Liferay plugin
Project name: JSFBootcamp
Display name: JSFBootcamp
Use default location
Location: /Users/GianMarco/Liferay/Environments/Bootcamp/workspace/life
Build type: Ant (liferay-plugins-sdk)
Plugins SDK: liferay-plugins-sdk-6.2
Liferay runtime: Liferay v6.2 CE (Tomcat 7)
Plugin type: Portlet
Include sample code
Launch New Portlet Wizard after project is created
Add project to working set
Working set:
More...
< Back
Next >
Cancel
Finish


New Liferay Plugin Project
Liferay Portlet Plugin Project
Choose from available portlet frameworks depending on which technology is most appropriate for this project.
Select portlet framework
Liferay MVC
Simple and lightweight MVC framework using JSPs that is built-in to Liferay Portal.
JSF 2.x
Enables development of JSF 2.x applications using Liferay Faces Bridge that run seamlessly inside of Liferay Portal.
Vaadin
Vaadin is a Java framework for building modern web applications which can be run as portlets inside of Liferay Portal.
Additional portlet options
Portlet name: IntegrationWithJSF
Display name: IntegrationWithJSF
< Back
Next >
Cancel
Finish


New Liferay Plugin Project
Liferay JSF Portlet Plugin Project
Choose from available JSF component suites
JSF Component Suite
JSF standard
Standard UI components provided by the JSF runtime
ICEfaces
Components based in part on YUI and jQuery with automatic Ajax and Ajax Push support
Liferay Faces Alloy
Components that utilize Liferay's Alloy UI technology based on YUI3
PrimeFaces
Lightweight, zero-configuration JSF UI framework built on jQuery
RichFaces
Next-generation JSF component framework by JBoss
< Back
Next >
Cancel
Finish

Si, ma cosa è successo?




```
<?xml version="1.0"?>

<ivy-module
    version="2.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="http://ant.apache.org/ivy/schemas/ivy.xsd"
>
    <info module="portlet_primefaces_tmpl" organisation="com.liferay" />

    <dependencies defaultconf="default">
        <dependency name="jboss-el" org="org.jboss.el" rev="2.0.1.GA" />
        <dependency name="jsf-api" org="com.sun.faces" rev="2.1.21" />
        <dependency name="jsf-impl" org="com.sun.faces" rev="2.1.21" />
        <dependency name="liferay-faces-bridge-api" org="com.liferay.faces" rev="3.2.4-ga5" />
        <dependency name="liferay-faces-bridge-impl" org="com.liferay.faces" rev="3.2.4-ga5" />
        <dependency name="liferay-faces-portal" org="com.liferay.faces" rev="3.2.4-ga5" />
        <dependency name="liferay-faces-util" org="com.liferay.faces" rev="3.2.4-ga5" />
        <dependency name="primefaces" org="org.primefaces" rev="5.2" />
    </dependencies>
</ivy-module>
```

```
<context-param>
  <param-name>com.sun.faces.expressionFactory</param-name>
  <param-value>org.jboss.el.ExpressionFactoryImpl</param-value>
</context-param>
<context-param>
  <param-name>javax.faces.PROJECT_STAGE</param-name>
  <param-value>Development</param-value>
</context-param>
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Liferay Faces Version Scheme

Liferay Faces segue il seguente schema : *Major1.Major2.Minor-Type*

`Major1`:

- `1` = Portlet 1.0 Bridge for JSF 1.2 (JSR 301) *(Note that Liferay Faces does not support Portlet 1.0)*
- `2` = Portlet 2.0 Bridge for JSF 1.2 (JSR 329)
- `3` = Portlet 2.0 Bridge for JSF 2.1 (JSR TBA)
- `4` = Portlet 2.0 Bridge for JSF 2.2 (JSR TBA)

`Major2`:

- `0-legacy` = Liferay `5.2`
- `0` = Liferay `6.0`
- `1` = Liferay `6.1`
- `2` = Liferay `6.2`

`Minor`:

- May contain bug fixes, improvements, and new features.

`Type`:

- GA (General Availability)
- RC (Release Candidate)
- BETA (Beta Quality)
- ALPHA (Alpha Quality)

Portlet JSF – configurazione

portlet.xml

```
<portlet>
    <portlet-name>jsf-form</portlet-name>
    <display-name>JSF Form</display-name>
    <portlet-class>javax.portlet.faces.GenericFacesPortlet</portlet-class>
    <init-param>
        <name>javax.portlet.faces.defaultViewId.view</name>
        <value>/views/jsf-form/view.xhtml</value>
    </init-param>
    <expiration-cache>0</expiration-cache>
    <supports>
        <mime-type>text/html</mime-type>
    </supports>
    <portlet-info>
        <title>JSF Form</title>
        <short-title>JSF Form</short-title>
        <keywords>JSF Form</keywords>
    </portlet-info>
    <security-role-ref>
        <role-name>administrator</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>guest</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>power-user</role-name>
    </security-role-ref>
    <security-role-ref>
        <role-name>user</role-name>
    </security-role-ref>
</portlet>
```

liferay-portlet.xml

```
<portlet>
    <portlet-name>jsf-form</portlet-name>
    <icon>/icon.png</icon>
    <instanceable>true</instanceable>
    <requires-namespaced-parameters>false</requires-namespaced-parameters>
    <ajaxable>false</ajaxable>
    <header-portlet-css>/css/main.css</header-portlet-css>
    <css-class-wrapper>jsf-form</css-class-wrapper>
</portlet>
<portlet>
    <portlet-name>jsf-table</portlet-name>
    <icon>/icon.png</icon>
    <instanceable>true</instanceable>
    <requires-namespaced-parameters>false</requires-namespaced-parameters>
    <ajaxable>false</ajaxable>
    <header-portlet-css>/css/main.css</header-portlet-css>
    <css-class-wrapper>jsf-table</css-class-wrapper>
</portlet>
```

Portlet JSF – simple form

```
<h:form id="id-form">
    <p:growl id="msgs" showDetail="true" />

    <h:panelGrid columns="2" cellpadding="5">
        <p:outputLabel value="Nominativo" for="acnominativo" />
        <p:autoComplete id="acnominativo" value="#{jSFFormMangedBean.personaSelectedName}"
            completeMethod="#{jSFFormMangedBean.completePersona}" required="true"
        />

        <p:outputLabel value="Evento" for="acevento" />
        <p:autoComplete id="acevento" value="#{jSFFormMangedBean.eventoSelectedName}"
            completeMethod="#{jSFFormMangedBean.completeEvento}" required="true"
        />

        <p:outputLabel value="Numero Biglietti" for="acbiglietti" />
        <p:inputText type="number" id="acbiglietti" value="#{jSFFormMangedBean.numeroBiglietti}"
            required="true"
        />

        <p:commandButton value="Acquista" actionListener="#{jSFFormMangedBean.acquista}" oncomplete="updateTable();"   />
    </h:panelGrid>
</h:form>


<script type="text/javascript">
    function updateTable(){
        Liferay.fire(
            'updateTable'
        );
    }

</script>
```

Liferay Faces Bridge - IPC

```
<portlet>
    <portlet-name>2</portlet-name>
    <display-name>JSF Table</display-name>
    ...
    <init-param>
        <name>javax.portlet.faces.bridgeEventHandler</name>
        <value>it.keybiz.bootcamp.jsf.bridge.CustomBridgeEventHandler</value>
    </init-param>
    ...
    <supported-processing-event>
        <qname xmlns:x="http://liferay.com/events">x:ipc.updateTicket</qname>
    </supported-processing-event>
</portlet>

<portlet>
    <portlet-name>3</portlet-name>
    <display-name>JSF Update</display-name>
    ...
    <supported-publishing-event>
        <qname xmlns:x="http://liferay.com/events">x:ipc.updateTicket</qname>
    </supported-publishing-event>
</portlet>
...
<event-definition>
    <qname xmlns:x="http://liferay.com/events">x:ipc.updateTicket</qname>
    <value-type>java.lang.String</value-type>
</event-definition>
```

Liferay Faces Bridge - EventHandler

```
QName qName = new QName("http://liferay.com/events", "ipc.updateTicket");
ExternalContext externalContext =
        FacesContext.getCurrentInstance().getExternalContext();
ActionResponse actionResponse = (ActionResponse) externalContext.getResponse();

Map<String, String> params =
        externalContext.getRequestParameterMap();

String idSelected = params.get("idSelected");
actionResponse.setEvent(qName, GetterUtil.getLong(idSelected));
```

```
public class CustomBridgeEventHandler implements BridgeEventHandler{

    @Override
    public EventNavigationResult handleEvent(FacesContext facesContext, Event event) {
        EventNavigationResult eventNavigationResult = null;
        String eventQName = event.getQName().toString();

        if (eventQName.equals("{http://liferay.com/events}ipc.updateTicket")) {
            Serializable value = event.getValue();

            Long selectedId = (Long) value;


            try {
                PersonaEvento personaEvento =
                        PersonaEventoLocalServiceUtil.getPersonaEvento(
                                selectedId);

                UpdateManagedBean managedBean = getUpdateManagedBean(facesContext);
                managedBean.setPersonaEventoUpdate(personaEvento);

            } catch (PortalException e) {
                e.printStackTrace();
            } catch (SystemException e) {
                e.printStackTrace();
            }

            String fromAction = null;
            String outcome = "ipc.updateTicket";
            eventNavigationResult = new EventNavigationResult(fromAction, outcome);
        }

        return eventNavigationResult;
    }
```

Scopes in standard JSF Application

Gli scopes standard sono interpretati differentemente in un portal environment.

JSF Managed Beans hanno scopes definiti nelle specifiche JSF. Le specifiche JSF 1.2 mettono a disposizione 3 scopes:

- **Application**
- **Session**
- **Request**

Con l'introduzione delle specifiche JSF 2.0 sono stati introdotti anche gli scopes **ViewScope** e **PageFlow**

Scopes in Portal Application

In ambiente di portale , proprio per la sua natura, esistono più concetti di scopes rispetto ad una applicazione JSF Standard :

- **Application**
- **Global Session**
- **Portlet Group Session**
- **Portlet Instance Session**
- **Pageflow**
- **View**
- **Portal Aware Request**

To be…

Liferay Faces: Portal 7.0 Support

- ❑ Possibilità di deployare una portlet JSF come un OSGI bundle (.jar)
- ❑ Liferay Faces Alloy (AlloyUI 3.0)
- ❑ Liferay Faces Crystal (Crystal 1.0)

Disclaimer

- http://www.jcp.org
- http://www.liferayfaces.org/web/guest/showcase
- http://www.liferay.com/it/web/neil.griffin/profile
- http://www.primefaces.org/showcase
- http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html

Grazie a tutti!

gvernile@key-biz.it
gvernile
gvernile