

### Lab. EJB 3 : EJB + JSF 2

Maîtrisez la configuration de JBoss Messaging, le module de configuration d'une source de données JMS dans JBoss. Basée sur une infrastructure impliquant Jboss remoting et Jboss Cache, JBoss messaging est un puissant levier pour vos applications asynchrones, avec par exemple un fournisseur de message Active MQ ou WebSphere MQ.

Objectifs :

Comprendre l'architecture d'intégration JSF /EJB

Etape 1 :


Après avoir créer un projet Web classique JSF 2 avec primefaces

Créez le fichier index.xhtml et mettez-y le contenu suivant.

```
<ui:composition>
<h:head>
  <h1><h:outputText value="jsf-ejb example" /></h1>
</h:head>

<h:body>
  <h:form id="form">
    <h:outputLabel value="helloEJB returned message: "/>
    <h:outputText value="#{HelloJSFBean.message}"/>
  </h:form>
</h:body>
</ui:composition>
</html>
```

Etape 2 : Ceéation du managedBea



```
@ManagedBean(name="HelloJSFBean")
@RequestScoped
public class HelloJSFBean implements java.io.Serializable {
    @EJB
    private HelloEJB helloEJB;

    public String getMessage() {
        return helloEJB.hello();
    }
}
```

Etape 3 : Création d'un EJB Singleton

```
@Singleton
public class HelloEJB {
    public String hello() {
        return "Hello from " + this;
    }
}
```

#### Etape 4

Configuration du fichier

```
<servlet>
  <servlet-name>Faces Servlet</servlet-name>
  <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
  <servlet-name>Faces Servlet</servlet-name>
  <url-pattern>/faces/*</url-pattern>
</servlet-mapping>
</web-app>
```

---

#### Etape 5 :