

JEE

Java Entreprise Edition

J. Saraydaryan

CPE - Lyon



JSF

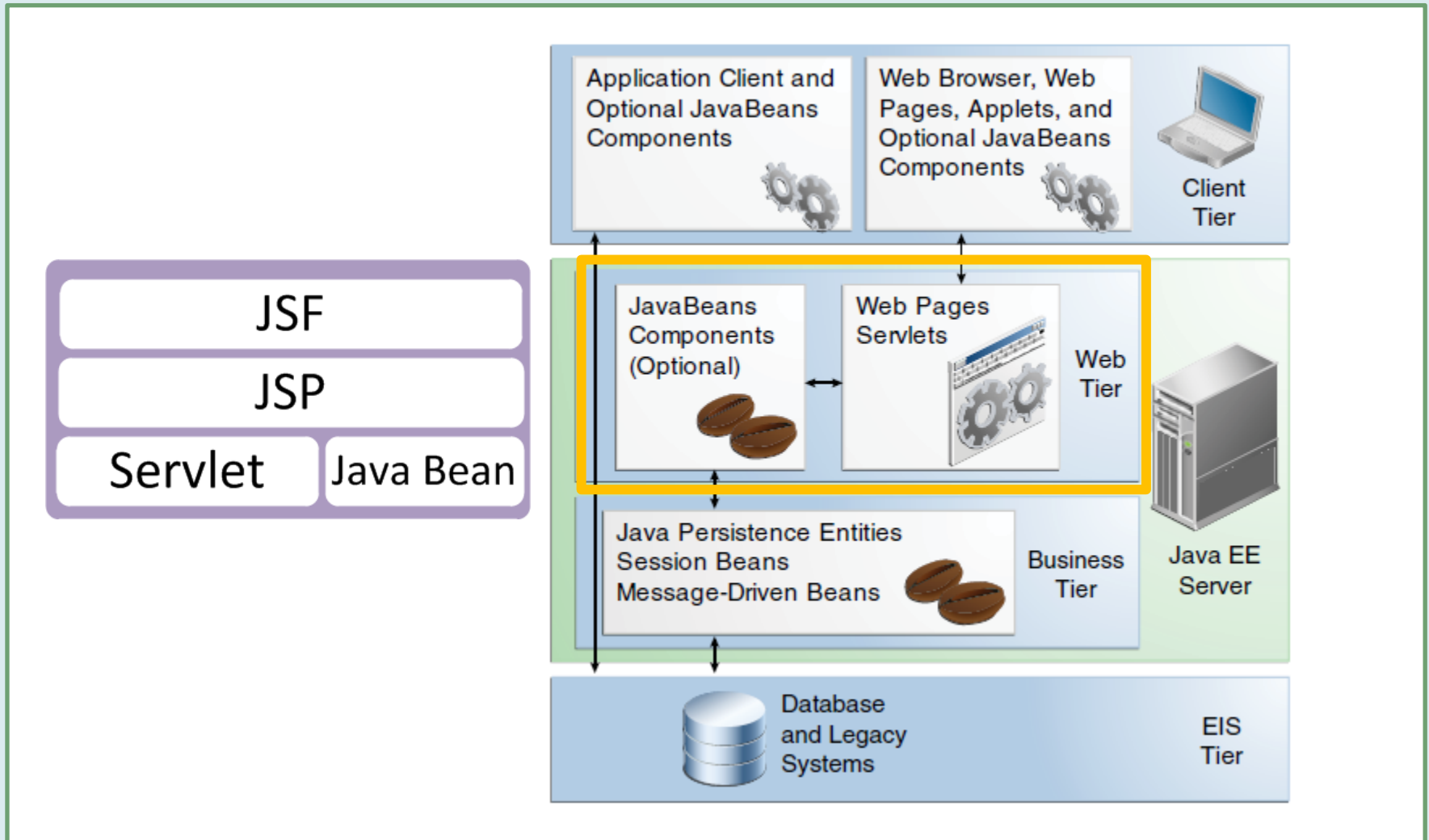
Java Server Faces

J. Saraydaryan

Introduction

JEE- Fonctionnement

- Rappel



JEE- Fonctionnement

• Besoins

❑ Programmation JSP/Servlet Limitée

- Utilisation des Beans dans JSP trop verbeuse
- Optimisation des appels aux serveurs (AJAX) complexe (javascript)
- Besoin d'éléments réutilisables
- Mise à jour des contenus peu élégante

❑ Besoin d'outils facilitant l'écriture et permettant une séparation **Model-Vue-Contrôleur**



JEE- Fonctionnement

• Frameworks

Utilisation de Framework

- **Java Server Faces (JSF)**

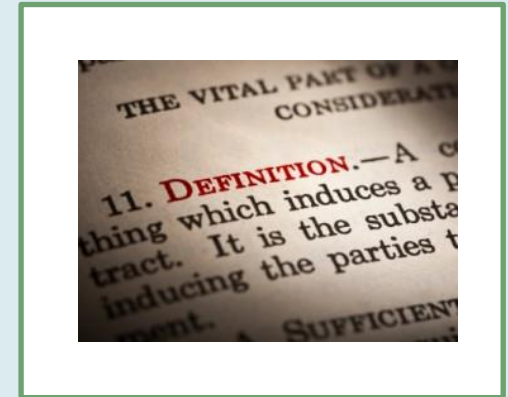
Seule Solution existantes ? NON

- Struts
- Ext-JS
- GWT (Google Web Toolkit)
- Angular.JS



JEE- Fonctionnement

- Définitions
 - ❑ Framework
 - Ensemble de libraires fournissant des outils
 - ❑ Handler/Contrôleur GUI
 - Pre-build, tag html assurant la gestion des évènements
 - ❑ Framework GUI indépendant du support
 - Génération graphique différente du HTML possible
 - ❑ Un Framework orienté MVC
 - Séparation du traitement de l'affichage, validation des données, affichage des résultats
 - ❑ Une librairie Ajax
 - Ensemble d'outils facilitant l'utilisation d'Ajax (mise à jour partielle)



JEE- Fonctionnement

• Comment l'utilise-t-on?

❑ Configuration

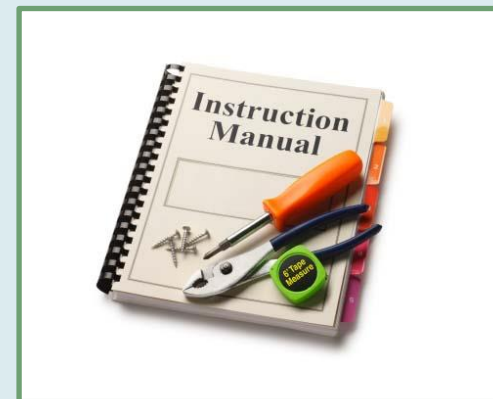
- Importation des librairies dans le projet ET dans le serveur web
- [Javax-faces-<version>.jar](#), [javax.servlet.jsp.jstl-<version>.jar](#)

❑ Présentation

- Utilisation de préférence des fichiers d'extension .xhtml
- Ajout des noms de domaines relatifs à jsf:
 - [xmlns:h=http://java.sun.com/jsf/html](#)
 - [xmlns:f=http://java.sun.com/jsf/core](#)

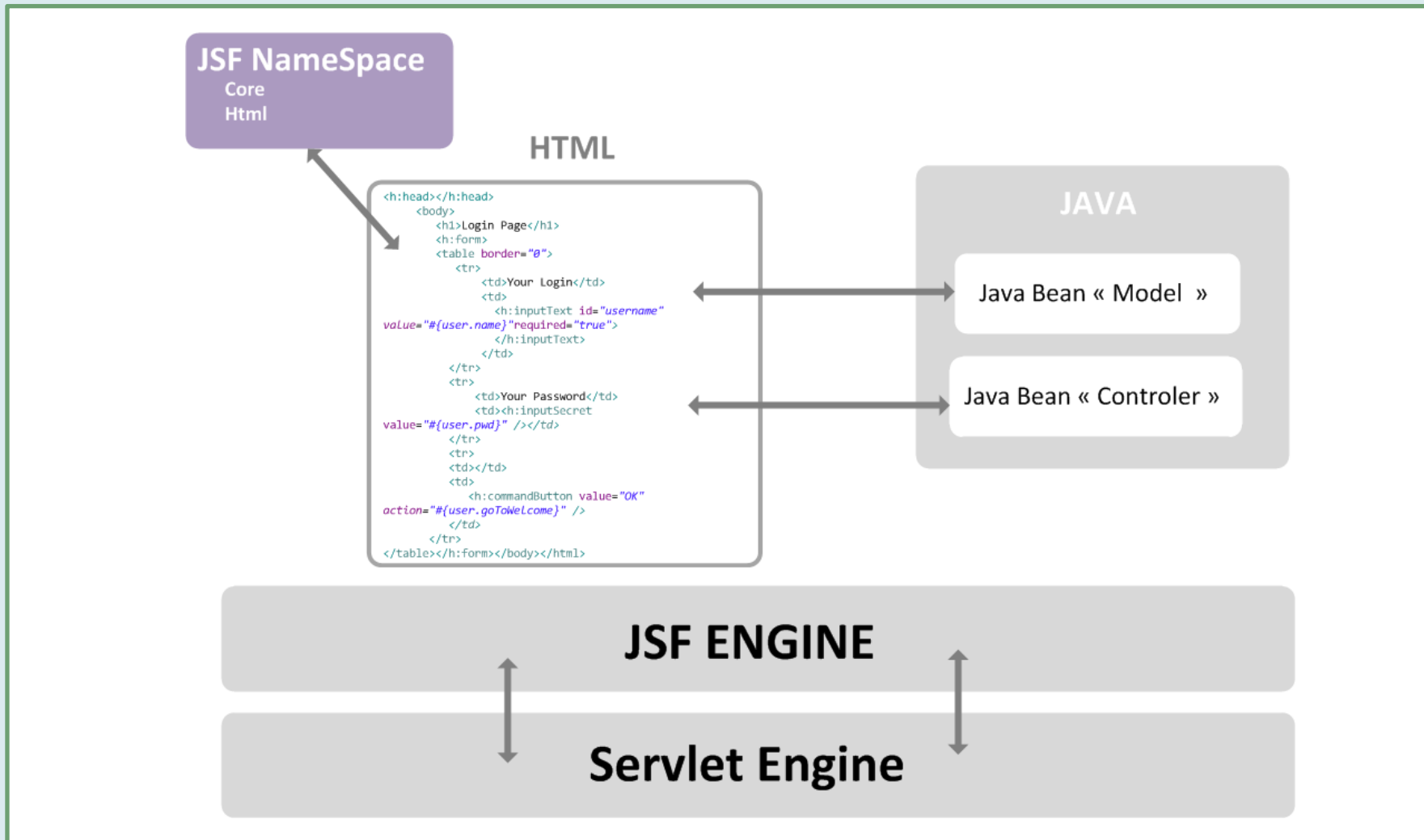
❑ Traitement

- Jsf v1.x utilisation d'un fichier de configuration [faces-config.xml](#) permettant de définir les javaBean utilisé par jsf
- Jsf v2.x utilisation des annotation [@ManagerBean](#) (surcharge [faces-config.xml](#) toujours possible)



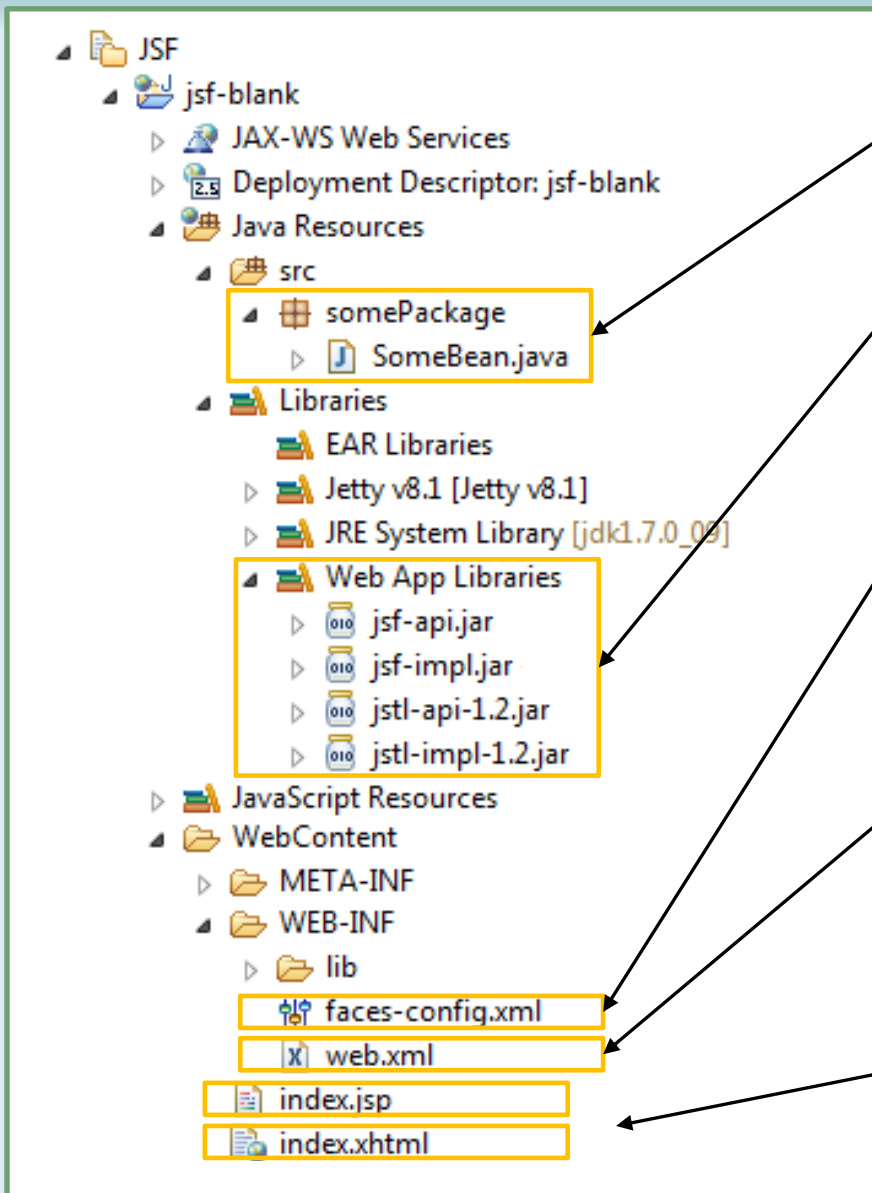
JEE- Fonctionnement

- Fonctionnement



Configuration et première utilisation

Java Server Faces



Java bean utilisés par JSF

Librairies permettant l'utilisation des outils fournis par JSF

Fichier de configuration de JSF permettant de décrire les javaBean utilisés par JSF (v1.x)

Fichier de description de l'utilisation des URL modification nécessaire pour la prise en compte de l'appel *.jsf

Fichiers de présentation utilisant ou non des propriétés JSF

Java Server Faces

WEB.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
  http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  version="2.5">

  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-
class>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <url-pattern>*.jsf</url-pattern>
  </servlet-mapping>

  <context-param>
    <param-name>javax.faces.PROJECT_STAGE</param-name>
    <param-value>Development</param-value>
  </context-param>

  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

Description classique des namespaces

Déclaration du traitement particulier effectué sur les appels de fichier *.jsf

Ajout d'information de configuration pour JSF

Liste des pages de démarrage

Java Server Faces

index.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml"  
xmlns:h="http://java.sun.com/jsf/html">
```

```
<h:head>  
</h:head>
```

```
<h:body>  
<h1>You will be redirect to another age</h1>
```

```
<h:form>
```

Click to see

```
<h:commandButton value="Go to a New Page"  
action="#{pageManager.goToPage}" />
```

```
</h:form>
```

```
</h:body>
```

```
</html>
```

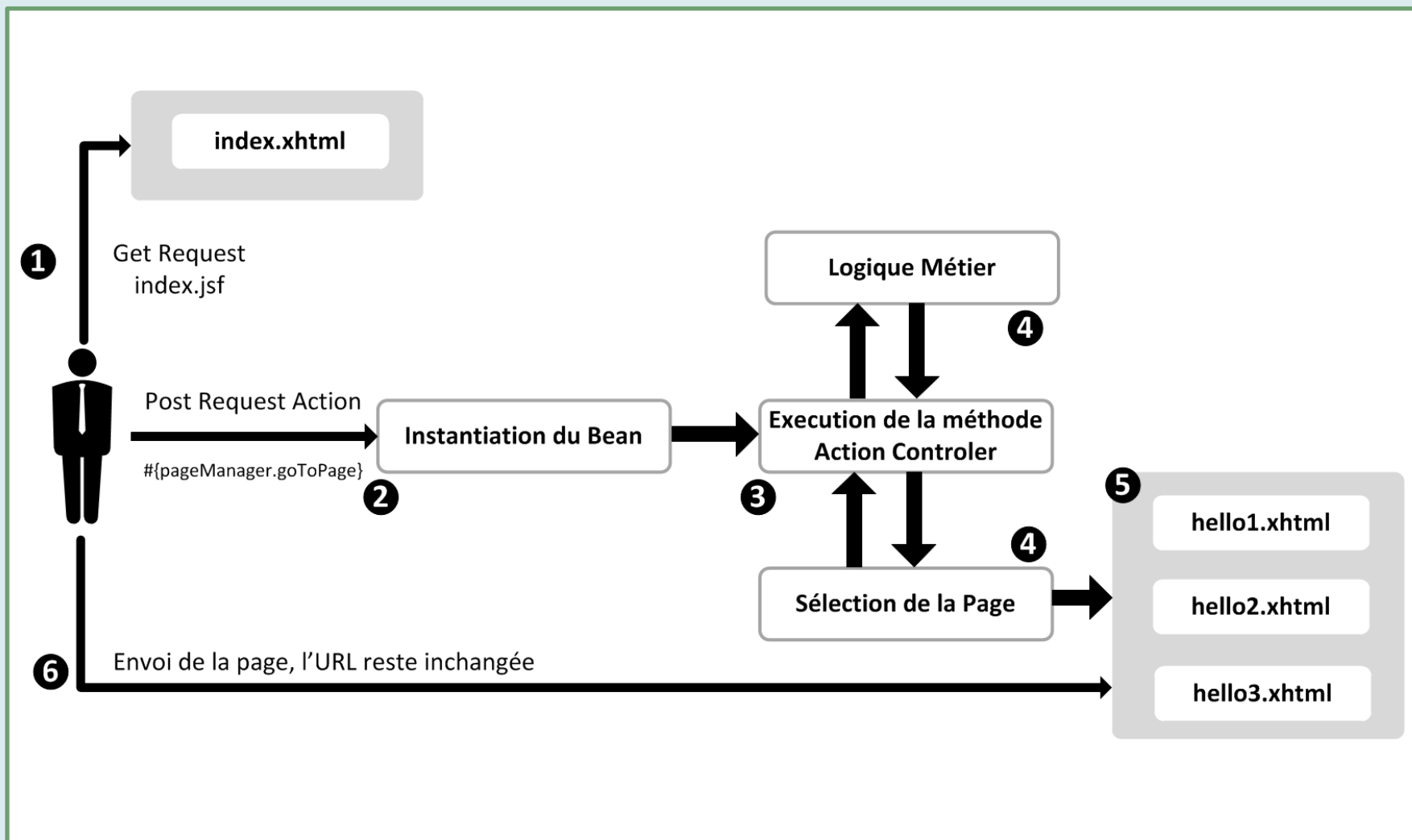
Déclaration xhtml + import
namespace sdes outils JSF

Utilisation des balises JSF

Appel d'une méthode d'un
bean

Java Server Faces

• Workflow d'exécution



Java Server Faces

- **Workflow d'exécution**

The screenshot illustrates a three-page workflow in a web browser. Each page is shown with its address bar and navigation controls.

- Page 1:** The address bar shows `http://localhost:8080/jsf-blank/index.jsf`. The main content is the text **You will be redirect to another Page**. Below this text is a button labeled **Go to a New Page**.
- Page 2:** The address bar shows `http://localhost:8080/jsf-blank/index.jsf`. The main content is the text **Welcome 1**. Below this text is a purple link labeled [go back](#).
- Page 3:** The address bar shows `http://localhost:8080/jsf-blank/index.jsf`. The main content is the text **Welcome 2**. Below this text is a purple link labeled [go back](#).

Welcome 3

[go back](#)

JSF

Usage Basic

Java Server Faces

- Utilisation

- Utilisation des beans dans la présentation

- `<h:inputText value=" #{someBean.someProp} " />`
 - Association automatique de l'input à la propriété du bean
 - Appel la méthode `setSomProp` de la classe `SomeBean`
 - Validation/conversion puis soumission de la nouvelle valeur au bean

- `" #{someBean.someProp} "` directement ou `<h:outputText value=" #{someBean.someProp} " />`

- Association automatique à la propriété du bean
- Appel la méthode `getSomProp` de la classe `SomeBean`



Java Server Faces

- Utilisation

- Les types de Bean

- **None** utiliser en cas d'appel d'un autre bean
- **View** persiste durant l'interaction de l'utilisateur avec une page
- **Request** (default) persiste durant une simple requête HTTP
- **Session** persiste durant la session de l'utilisateur
- **Application** persiste entre les différentes pages des différents utilisateurs



Java Server Faces

- Utilisation

```
<h:head></h:head>
  <body>
    <h1>Login Page</h1>
    <h:form>
      <table border="0">
        <tr>
          <td>Your Login</td>
          <td>
            <h:inputText id="username" value="#{user.name}" required="true">
            </h:inputText>
          </td>
        </tr>
        <tr>
          <td>Your Password</td>
          <td><h:inputSecret value="#{user.pwd}" /></td>
        </tr>
        <tr>
          <td></td>
          <td>
            <h:commandButton value="OK" action="#{user.goToWelcome}" />
          </td>
        </tr>
      </table></h:form></body></html>
```

loginInit.xhtml

Java Server Faces

- Utilisation

```
@ManagedBean  
@SessionScoped
```

User.java

```
public class User implements Serializable {  
    private static final long serialVersionUID = 1L;  
    private String name;  
    private String pwd;  
  
    public User() {}  
  
    public String getName() { return name; }  
  
    public void setName(String name) { this.name = name; }  
  
    public String getPwd() {return pwd;}  
  
    public void setPwd(String pwd) { this.pwd = pwd;}  
  
    public String goToWelcome() { return ("jsf/WelcomeUser.xhtml");  
    }  
}}
```

Java Server Faces

- Utilisation

WelcomeUser.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
  <html xmlns="http://www.w3.org/1999/xhtml"  
        xmlns:h="http://java.sun.com/jsf/html">  
    <h:head>  
    </h:head>  
    <body>  
  
        <h1>Welcome #{user.name}</h1>  
        <a href="loginInit.jsf">go back</a>  
  
    </body>  
</html>
```

Java Server Faces

- Utilisation

← → ↻ localhost:8080/jsf-blank/loginInit.jsf

Login Page

Your Login

Your Password

OK

← → ↻ localhost:8080/jsf-blank/loginInit.jsf

Welcome JDoe

[go back](#)

Java Server Faces

- Validation des inputs

- Récupération des messages d'erreur

- `<h:message>` messages d'un paramètre en particulier
- `<h:messages>` messages d'erreurs globaux

```
<h:inputText id="userName" value="#{userBean.userName}" />

<h:message for="userName" showSummary="true"/>
...
<h:messages showDetail="true"/>
```

- Valideur Build-in

- `f:validateDoubleRange`
- `f:validateLength`
- `f:validateLongRange`

Java Server Faces

- Validation Custom

- Appel d'un validateur créé

```
<h:inputText value="#{userBean.name}"  
             validator="#{userBean.checkUsername}">  
</h:inputText>
```

- Ecriture d'un validateur

```
import javax.faces.component.UIComponent;  
import javax.faces.context.FacesContext;  
import javax.faces.validator.Validator;  
import javax.faces.validator.ValidatorException;  
...  
...  
public void validate(FacesContext context, UIComponent  
                    component, Object value)  
                    throws ValidatorException {}
```


Java Server Faces

- Validation Custom

```
<h:head></h:head>
<body>
  <h1>Login Page</h1>
  <h:form>
    <table border="0">
      <tr>
        <td>Your Login</td>
        <td><h:inputText id="username" value="#{user.name}"required="true">
          <f:validator validatorId="validators.username" />
        </h:inputText></td>
        <td><h:message for="username" showDetail="true" style="color:red" />
        </td>
      </tr>
      <tr>
        <td>Your Password</td>
        <td><h:inputSecret value="#{user.pwd}" /></td>
      </tr>
      <tr>
        <td></td>
        <td><h:commandButton value="OK" /></td>
      </tr>
    </table></h:form></body></html>
```

Java Server Faces

- Validation Custom

```
@FacesValidator(value = "validators.username")
public class UsernameValidator implements Validator {
    private static final String USERNAME_PATTERN = "^[_A-Za-z0-9-@]+";
    private Pattern pattern;
    private Matcher matcher;

    public UsernameValidator() {
        pattern = Pattern.compile(USERNAME_PATTERN);
    }
}
```

```
@Override
public void validate(FacesContext context, UIComponent component,
                    Object value) throws ValidatorException {
    matcher = pattern.matcher(value.toString());
    if(!matcher.matches()){
```

```
        FacesMessage msg =
            new FacesMessage("user name validation failed.",
                "User Name Validation failed please follow the constraint "+USERNAME_PATTERN);
        msg.setSeverity(FacesMessage.SEVERITY_ERROR);
        throw new ValidatorException(msg);
    }}
```

Java Server Faces

- Validation Custom

The image displays three sequential screenshots of a web browser showing a login page at `localhost:8080/jsf-blank/login.jsf`. Each screenshot shows the page title "Login Page" and two input fields: "Your Login" and "Your Password", with an "OK" button below them.

- Top-left screenshot:** The "Your Login" field is empty.
- Top-right screenshot:** The "Your Login" field contains the text "JDoe".
- Bottom screenshot:** The "Your Login" field contains the text "JDoe@_\$\$\$é". To the right of the input field, a red error message is displayed: "User Name Validation failed please follow the constraint `^[A-Za-z0-9-@]+`".

JSF

Usage Avancé

Java Server Faces

• Usage Avancé

- Appel de méthode avec paramètres
- Récupération de champ complexe (list)
- Mise à jour d'une valeur



Java Server Faces

- **Appel de méthode avec paramètres**
 - Expression d'une méthode (JSF 2.0)
 - f:param
 - f:attribute
 - f:setPropertyActionListener



Java Server Faces

- **Appel de méthode avec paramètres**

- Expression d'une méthode (JSF 2.0)

- Utilisation d'une sémantique particulière permettant le passage de paramètre

xhtml

```
<h:commandButton action="#{user.editAction('delete')}" />
```

Java

```
@ManagedBean (name="user")  
@SessionScoped  
public class UserBean{  
    public String editAction (String id) {  
        }  
}
```

Java Server Faces

• Appel de méthode avec paramètres

❑ Expression d'une méthode (JSF 2.0)

➤ Appel d'autre javabean possible

xhtml

```
<h:commandButton action="#{user.editAction(actionMgt.action)}"/>
```

Java

```
@ManagedBean(name="user")
@SessionScoped
public class UserBean{
    public String editAction(String id) {
    }
}

@ManagedBean(name="actionMgt")
@SessionScoped
public class ActionBean{
    public String getAction()
    {
    }
}
```


Java Server Faces

• Appel de méthode avec paramètres

❑ F:param

➤ Association d'un paramètre à la requête

```
<h:commandButton action="#{user.editAction}">
    <f:param name="action" value="delete" />
</h:commandButton>
```

xhtml

```
@ManagedBean (name="user")
@SessionScoped
```

Java

```
public class UserBean{

    public String editAction() {

        Map<String,String> params =
            FacesContext.getCurrentInstance().
                getExternalContext().getRequestParameterMap();
        String action = params.get("action");
    }
}
```

Java Server Faces

• Appel de méthode avec paramètres

❑ f:attribute

➤ Association d'un attribut à la requête

```
<h:commandButton action="#{user.editAction}"  
                 actionListener="#{user.attrListener}">  
    <f:attribute name="action" value="delete" />  
</h:commandButton>
```

xhtml

```
@ManagedBean (name="user")  
@SessionScoped  
public class UserBean{  
    String action;  
    public void attrListener(ActionEvent event){  
        action = (String)event.getComponent()  
                .getAttributes().get("action");  
    }  
  
    public String editAction() {  
    }  
}
```

Java

Java Server Faces

- Appel de méthode avec paramètres

- f:setPropertyActionListener

```
<h:commandButton action="#{user.editAction}" >  
  <f:setPropertyActionListener target="#{user.action}" value="delete"/>  
</h:commandButton>
```

xhtml

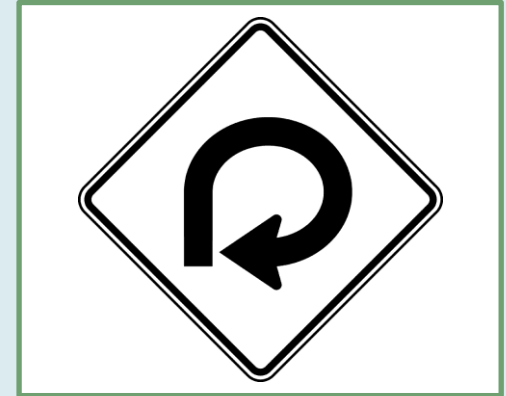
```
@ManagedBean(name="user")  
@SessionScoped
```

Java

```
public class UserBean{  
public String action;  
  public void setAction(String action)  
  {  
    this.action = action;  
  }  
  
  public String editAction() {  
  }  
}
```

Java Server Faces

- **Récupération de champ complexe**
 - ❑ JSTL → foreach
 - ❑ JSF ui:repeat
 - ❑ JSF dataTable



Java Server Faces

• Récupération de champ complexe

□ JSTL → foreach

- Ajout du namespace JSTL

xmlns:c=<http://java.sun.com/jsp/jstl/core>

- Boucle de taille déterminée

```
<c:forEach begin="1" end="10">  
  <div>content</div>  
</c:forEach>
```

- Boucle dépendant de la taille du champ de retour

```
<table>  
  <c:forEach var="person" items="#{people.people}">  
    <tr>  
      <td>#{person.name}</td>  
      <td>#{person.age}</td>  
    </tr>  
  </c:forEach>  
</table>
```

Java Server Faces

- **Récupération de champ complexe**

- ❑ JSF ui:repeat
 - ❑ Fonction similaire à JSTL foreach
 - ❑ Pas de boucle de taille fixe

```
<table>
  <ui:repeat value="#{people.people}" var="person">
    <tr>
      <td>#{person.name}</td>
      <td>#{person.age}</td>
    </tr>
  </ui:repeat >
</table>
```

Java Server Faces

- **Récupération de champ complexe**
 - ❑ JSF dataTable

userList.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core">

<h:head>
  <link href="styles.css" rel="stylesheet" type="text/css"/>
</h:head>
```

Java Server Faces

- Récupération de champ complexe

- JSF dataTable

```
<h1>User List</h1>
<h:dataTable var="usertemp" value="#{userList.userList}"
             styleClass="table-style"
             headerClass="heading-style"
             rowClasses="rowA,rowB">

  <h:column>
    <f:facet name="header">User Name</f:facet>
    #{usertemp.name}
  </h:column>
  <h:column>
    <f:facet name="header">E-Mail</f:facet>
    #{usertemp.email}
  </h:column>
  <h:column>
    <f:facet name="header">Age of User</f:facet>
    #{usertemp.age}
  </h:column>
  <h:column>
    <f:facet name="header">User Password</f:facet>
    *****
  </h:column></h:dataTable>
```

userList.xhtml

Java Server Faces

• Récupération de champ complexe

❑ JSF dataTable

```
@ManagedBean
public class UserList {
    private List<User> userList;

    public UserList() {
        userList = new ArrayList<>();
        User u;
        for (int i = 0; i < 20; i++) {
            u = new User();
            u.setName("joe" + i);
            u.setPwd("pwd" + i);
            u.setEmail("jDoe" + i + "@company.com");
            u.setAge(i);
            userList.add(u);
        }
    }

    public List<User> getUserList() {
        return userList;
    }

    public void setUserList(List<User> userList) {
        this.userList = userList;
    }
}
```

UserList.java

```
styles.css

.table-style{
border:solid;
}

.heading-style{
background-color: black;
color: white;
}

.rowA{
background-color: white;
}

.rowB{
background-color: #689AFF;
}
```

Java Server Faces

- Récupération de champ complexe

- JSF dataTable

← → ↻ localhost:8080/jsf-blank/jsf/userList.jsf

User List

| User Name | E-Mail | Age of User | User Password |
|-----------|--------------------|-------------|---------------|
| joe0 | jDoe0@company.com | 0 | ***** |
| joe1 | jDoe1@company.com | 1 | ***** |
| joe2 | jDoe2@company.com | 2 | ***** |
| joe3 | jDoe3@company.com | 3 | ***** |
| joe4 | jDoe4@company.com | 4 | ***** |
| joe5 | jDoe5@company.com | 5 | ***** |
| joe6 | jDoe6@company.com | 6 | ***** |
| joe7 | jDoe7@company.com | 7 | ***** |
| joe8 | jDoe8@company.com | 8 | ***** |
| joe9 | jDoe9@company.com | 9 | ***** |
| joe10 | jDoe10@company.com | 10 | ***** |
| joe11 | jDoe11@company.com | 11 | ***** |
| joe12 | jDoe12@company.com | 12 | ***** |
| joe13 | jDoe13@company.com | 13 | ***** |
| joe14 | jDoe14@company.com | 14 | ***** |
| joe15 | jDoe15@company.com | 15 | ***** |
| joe16 | jDoe16@company.com | 16 | ***** |
| joe17 | jDoe17@company.com | 17 | ***** |
| joe18 | jDoe18@company.com | 18 | ***** |
| joe19 | jDoe19@company.com | 19 | ***** |

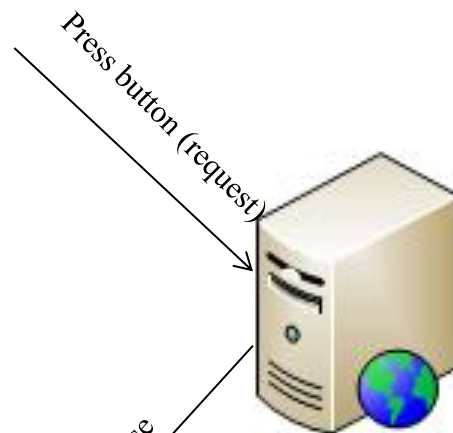
Java Server Faces

- **Récupération de champ complexe**

- Mise à jour partiel du contenu

WEB PAGE 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras nisl lorem, mollis sit amet tempor in, fermentum at erat. Mauris in eleifend nunc.

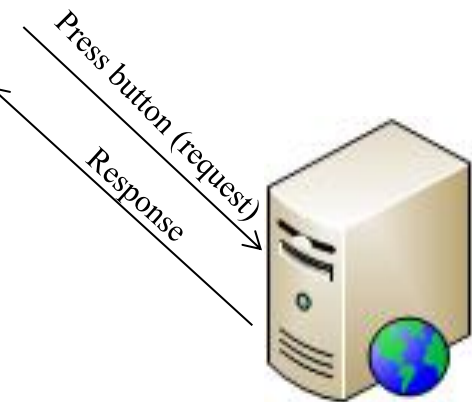


WEB PAGE 2

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras nisl lorem, mollis sit amet tempor in, fermentum at erat. Mauris in eleifend nunc.

WEB PAGE 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras nisl lorem, mollis sit amet tempor in, fermentum at erat. Mauris in eleifend nunc.



Java Server Faces

• Récupération de champ complexe

- ❑ Mise à jour partiel du contenu

```
<h:inputText id="username" value="#{user.name}" required="true"/>
<h:commandButton value="OK" action="myAction">
  <f:ajax render="id1 id2" execute="id3" event="blah" onevent="myJavaScriptFct" />
</h:commandButton>
<h:outputText id="id1" value="#{user.name}" />
```

❑ render

- Défini les ids des blocs JSF à mettre à jour

❑ execute

- Défini les ids des blocs JSF à exécuter

❑ event

- L'événement DOM auquel répondre (keyup,blur) (ne pas mettre le prefix on)

❑ onevent

- L'action à exécuter en cas d'événement

Java Server Faces

• Récupération de champ complexe

- ❑ Mise à jour partiel du contenu

```
<h:inputText id="username" value="#{user.name}" required="true"/>
<h:commandButton value="OK" action="myAction">
  <f:ajax render="id1 id2" execute="« @form" event="blah" onevent="myJavaScriptFct" />
</h:commandButton>
<h:outputText id="id1" value="#{user.name}" />
```

- ❑ **@ this**

- L'élément dans lequel est contenu <f:ajax/>(default)

- ❑ **@ form**

- Le « form » dans lequel est contenu <f:ajax/>

- ❑ **@none**

- Rien n'est renvoyé

- ❑ **@all**

- Tous les éléments UI JSF de la page courante

Java Server Faces

- **Récupération de champ complexe**

- Mise à jour partiel du contenu

```
<h:form>
  <h:inputText id="firstName" value="#{user.firstName}" />
  <h:inputText id="lastName" value="#{user.lastName}" />
  <h:inputText id="password" value="#{user.pwd}" />
  <h:inputText id="age" value="#{user.age}" />
  <h:commandButton value="OK" action="myAction" />
    <f:ajax render="id1" execute="@form" />
  <h:commandButton />
</h:form>

<h:outputText id="id1" value="#{user.name}" />
```

Java Server Faces

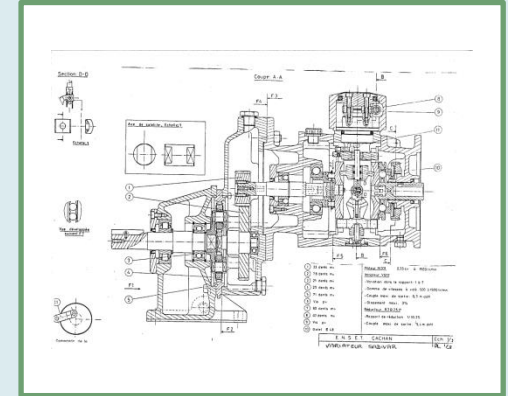
• Création de Template

❑ Objectif

- Éviter la réécriture d'élément identique
- Très utile pour la réutilisation de layout

❑ Et JSP:Include ?

- Page JSF normalement « limite » la réutilisation de tag jsp
- Pas vraiment de template
- Difficile de passer des informations aux pages incluses



Java Server Faces

• Création de Template

Template

Generic Header

Default Section Title

- Section 1
- Section 2
- Section 3
- Section 4
- Custom Section

Default Section Content

Copyright 2048

Usage du Template

Generic Header

My instantiated Title

- Section 1
- Section 2
- Section 3
- Section 4
- Instance 1
- Instance 2

Instantiated body

This is an added paragraph

Copyright 2048

Java Server Faces

• Création de Template

- Définition d'une structure fixe
- Définition de zone de customisation

The diagram illustrates a JSP template structure with the following components:

- Generic Header**: A black header bar at the top.
- Default Section Title**: A white box with a yellow border containing the text "Default Section Title".
- Default Section Content**: A white box with a yellow border containing the text "Default Section Content".
- Section List**: A black box containing a list of sections:
 - Section 1
 - Section 2
 - Section 3
 - Section 4
 - **Custom Section** (highlighted with a yellow border)
- Copyright 2048**: Text at the bottom center.

Java Server Faces

- **Création de Template**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:h="http://java.sun.com/jsf/html"
      xmlns:ui="http://java.sun.com/jsf/facelets">

<h:head>
  <link rel="stylesheet" type="text/css" href="mystyle.css"/>
</h:head>
```

Java Server Faces

• Création de Template

```
<h:body>
<div class="header">
  <h1>Generic Header</h1>
  <h2>
    <ui:insert name="sectionTitle"> Default Section Title</ui:insert>
  </h2>
</div>
<div class="nav">
  <ul>
    <li>Section 1</li><li>Section 2</li><li>Section 3</li><li>Section 4</li>
    <ui:insert name="nav">
      <li>Custom Section</li>
    </ui:insert>
  </ul>
</div>
<div class="corebody">
  <ui:insert name="body"> Default Section Content</ui:insert>
</div>
<div class="footer">
  <h6>Copyright 2048</h6>
</div>
</h:body> </html>
```

Java Server Faces

• Utilisation de Template

- ❑ Réutilisation de la description du template
- ❑ Définition des zones de customisation

Generic Header

My instantiated Title

- Section 1
- Section 2
- Section 3
- Section 4
- **Instance 1**
- **Instance 2**

Instantiated body

This is an added paragraph

enter a value

Copyright 2048

Java Server Faces

- Utilisation de Template

```
<ui:composition
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:h="http://java.sun.com/jsf/html"
    xmlns:ui="http://java.sun.com/jsf/facelets"
    template="/MyTemplate.xhtml">

    <ui:define name="sectionTitle">
        My instantiated Title
    </ui:define>
```

Java Server Faces

- Utilisation de Template

```
<ui:define name="body">  
  <h3> Instantiated body</h3>  
  <p> This is an added paragraph</p>  
  <h:form>  
    <h:inputText id="username" value="enter a value"  
required="true"></h:inputText>  
    <h:commandButton value="OK"></h:commandButton>  
  </h:form>  
</ui:define>
```

```
<ui:define name="nav">  
  <li class="added">Instance 1</li>  
  <li class="added">Instance 2</li>  
</ui:define>
```

```
</ui:composition>
```

Java Server Faces

- **Tag ui pour le templating**

- ❑ **ui:insert** : définition d'une zone de customisation (identifiable par l'attribut `name=« value »`)
- ❑ **ui:composition** : définition du template à utiliser (ciblé par l'attribut `template=« template_file_name »`)
- ❑ **ui:define** : définition de l'espace de customisation du template (ciblé par l'attribut `name=« value »`)

JSF

PrimeFaces / RicheFaces

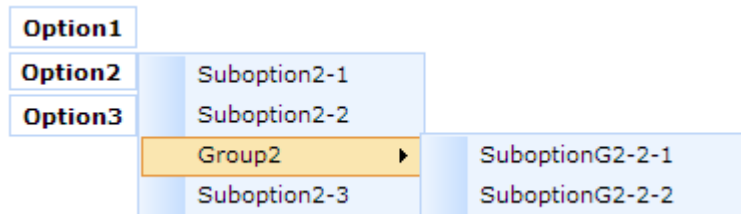
Les boites à outils

Java Server Faces

• RicheFaces



To call the menu - perform the right click on the picture



Java Server Faces

- RicheFaces









Car Store

| | | |
|---|--|--|
| <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Corvette</div> Price: 50520 Mileage: 77719.0 VIN: BLNDJEBJCSYWYWC Stock: GHQSPZW | <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Corvette</div> Price: 32048 Mileage: 20253.0 VIN: OPLKUXZTUUURTWQ Stock: GAABGI | <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Corvette</div> Price: 42031 Mileage: 71926.0 VIN: DTORSJTOWVZWBYT Stock: IIUOTT |
| <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Corvette</div> Price: 25792 Mileage: 36455.0 VIN: YXXDBCSSSOPSPSS Stock: HXAWBO | <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Corvette</div> Price: 18392 Mileage: 71723.0 VIN: IHGMHXZZYVRMOR Stock: YXSUFBY | <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Malibu</div> Price: 29043 Mileage: 42959.0 VIN: HJWQSTQUFZDBADZ Stock: YCTNQRN |
| <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Malibu</div> Price: 38641 Mileage: 6133.0 VIN: XNGGIDGWSXWUZHJ Stock: GLQRRRP | <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Malibu</div> Price: 48913 Mileage: 73809.0 VIN: ABGDEIKQMWCYEFE Stock: RJEFFH | <div style="border: 1px solid #ccc; padding: 5px; background-color: #e6f2ff; margin-bottom: 5px;">Chevrolet Malibu</div> Price: 35300 Mileage: 76397.0 VIN: CXFBHDGGBGACBMK Stock: SORSBF |

«« « 1 2 3 4 5 6 7 8 9 10 » »»

Java Server Faces

- RicheFaces

| Flag | State Name ↓ | State Capital ▲ | Time Zone | |
|--|--------------------------------|----------------------|-----------|--|
| | <input type="text" value="N"/> | <input type="text"/> | | |
|  | New York | Albany | GMT-5 | |
|  | North Dakota | Bismarck | GMT-6 | |
|  | Nevada | Carson City | GMT-8 | |
|  | New Hampshire | Concord | GMT-5 | |
|  | Nebraska | Lincoln | GMT-6 | |
|  | North Carolina | Raleigh | GMT-5 | |
|  | New Mexico | Santa Fe | GMT-7 | |
|  | New Jersey | Trenton | GMT-5 | |

Sort/Selection modes changing

Sort Mode: ▼

Selection Mode: ▼

Currently selected rows:

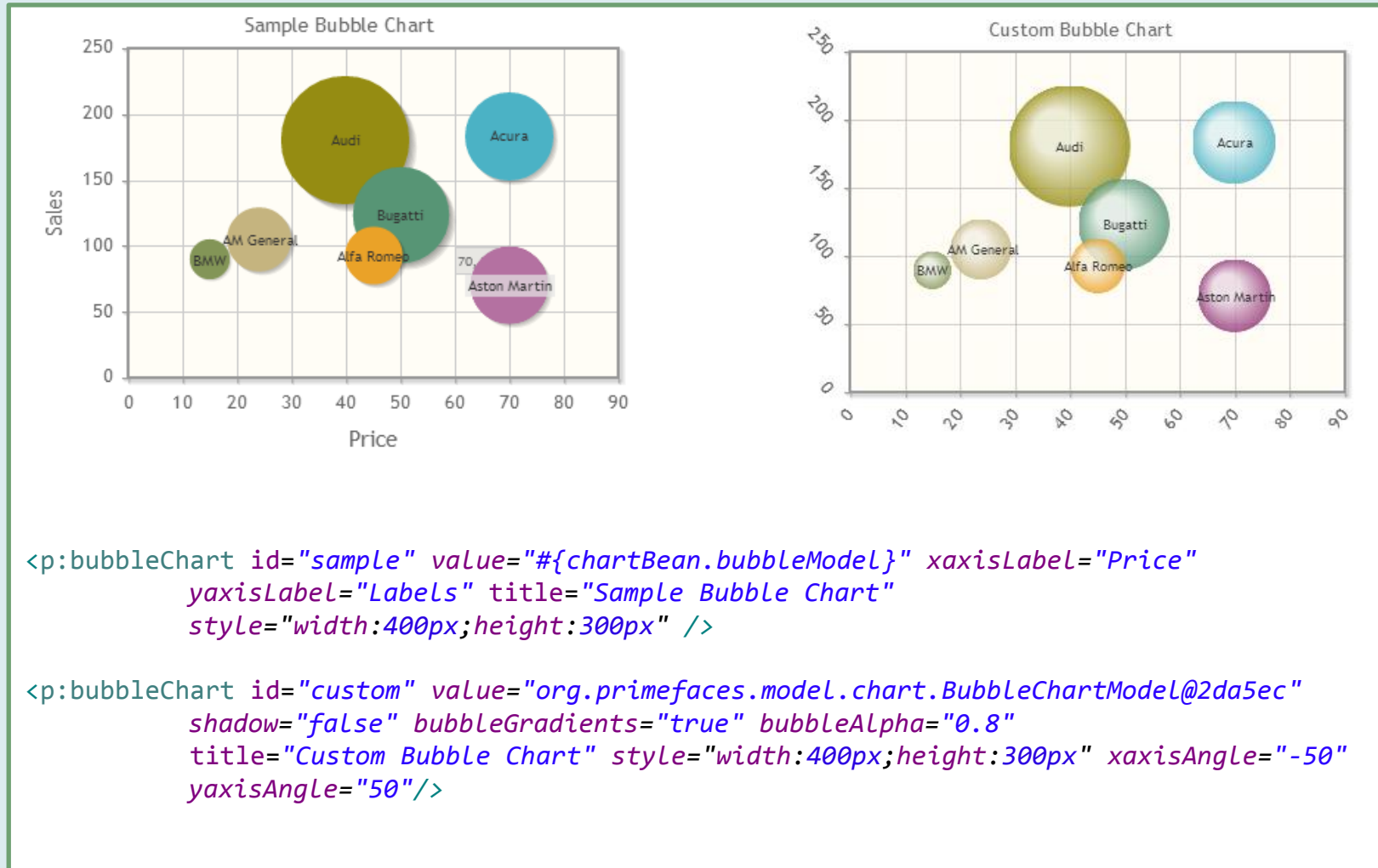
Java Server Faces

- **Qu'est ce que c'est ?**
 - ❑ Bibliothèque d'objet

<http://livedemo.exadel.com/richfaces-demo/>

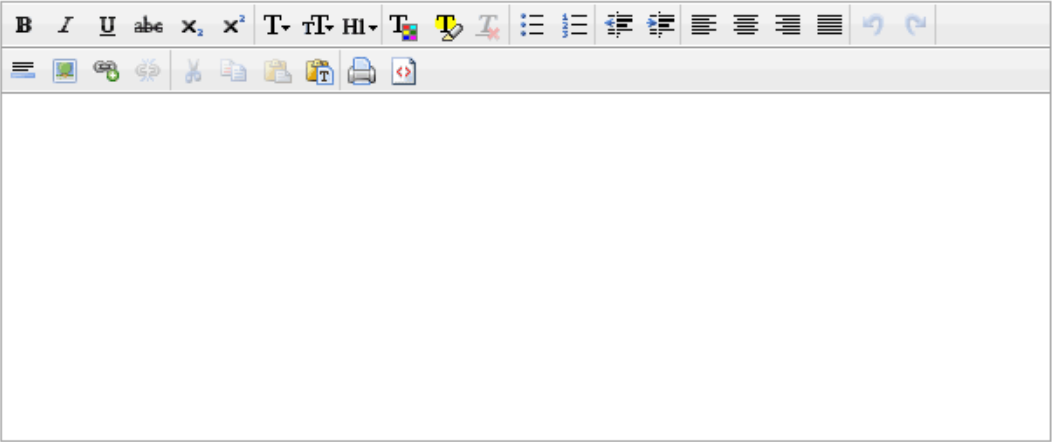
Java Server Faces

• PrimeFaces



Java Server Faces

• PrimeFaces



Masks

Date:

Phone:

Phone with Ext:

taxId:

SSN:

Product Key:

Tooltip

Tooltip has various customization options such as effects, events and custom content support.

Focus/Blur:

Fade: [Fade Effect](#) This tooltip is displayed when input gets the focus

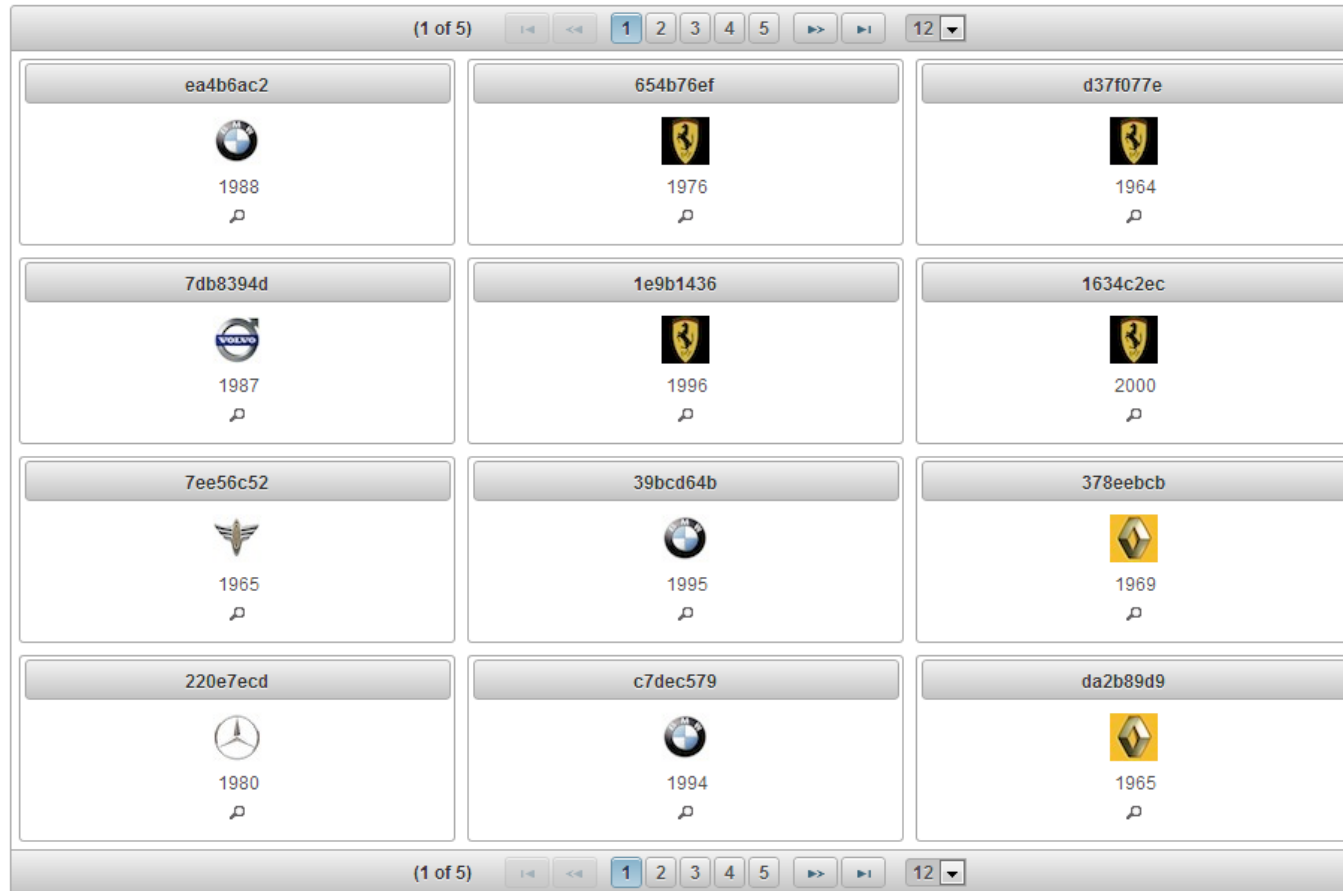
Slide: [Slide Effect](#)

Clip/Explode: [Clip/Explode](#) This tooltip uses slide effect for the animation

Content: [PrimeFaces](#)

Java Server Faces

- PrimeFaces



Java Server Faces

- **Qu'est ce que c'est ?**
 - ❑ Bibliothèque d'objet

<http://www.primefaces.org/showcase/ui/home.jsf>