



UNIVERSITÉ
DE MONTPELLIER

Programmation avancée

Aliouache Mohammed

Contexte général

- Application web touristique
- Base de données des différents monuments et célébrités associées
- Utilisation des rôles
- Authentification
- Assurer la sécurité de l'application

Vue Sur l'application

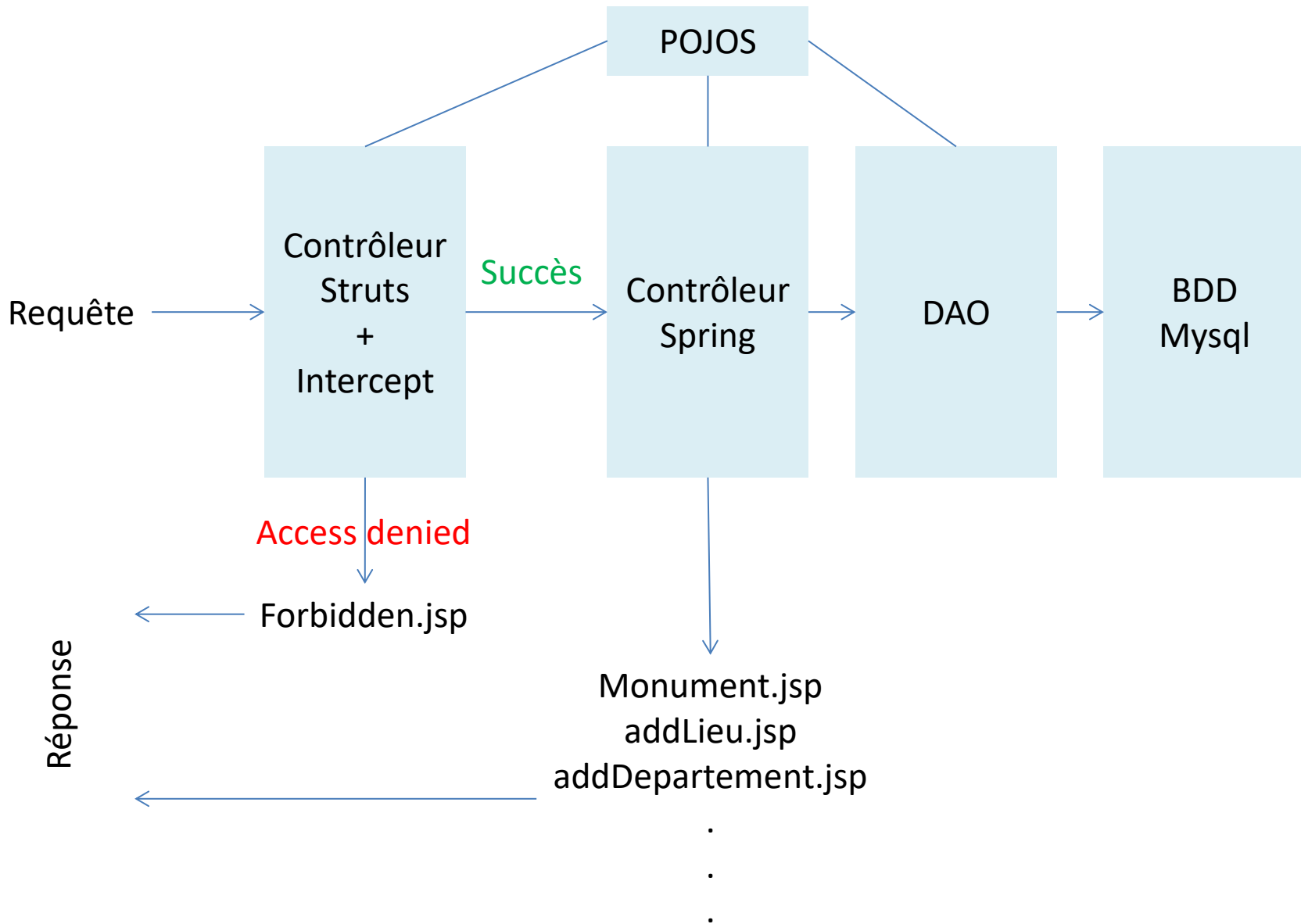
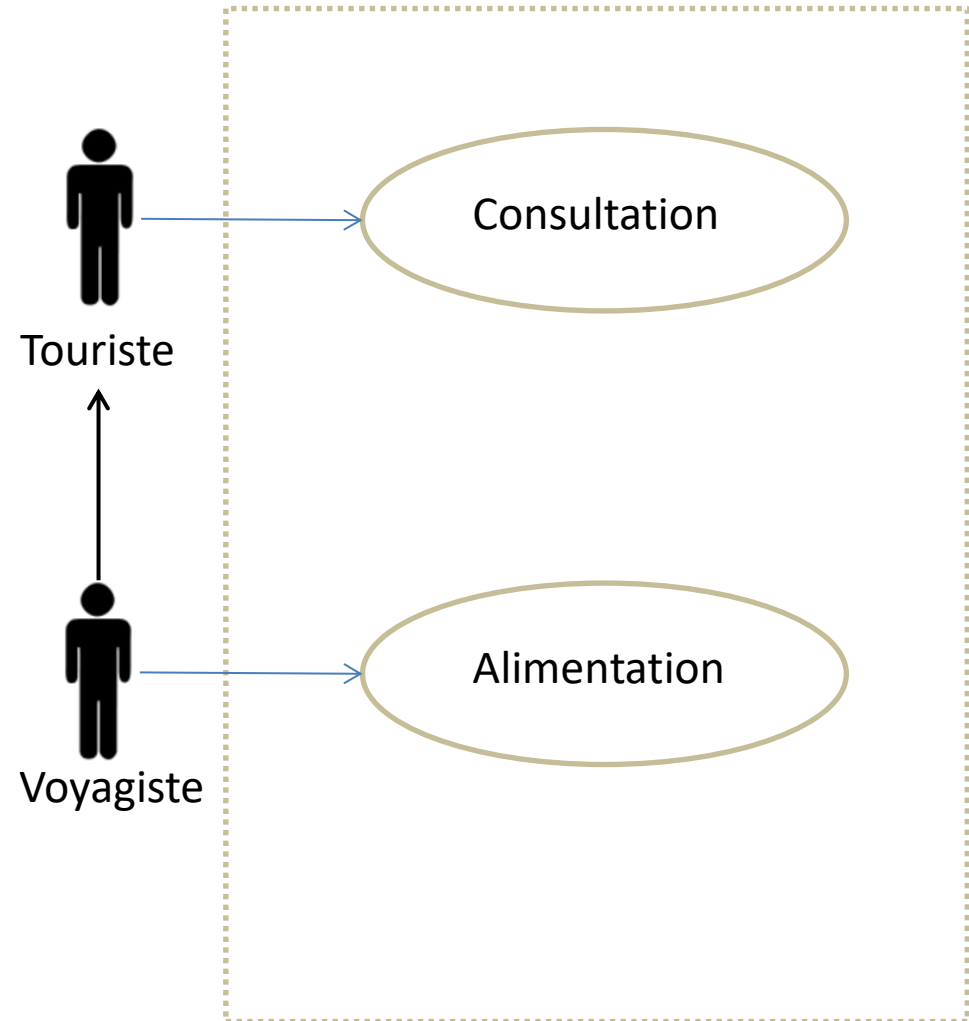


Schéma relationnel et use case

Le schéma des tables est celui
Énoncé

En plus, il y'a la table User

User(idUser, #email, password, nom,
prenom,...)




Partie Vue

<Spring:form **ModelAttribute=...**> mets la liaison avec le pojo

<Spring:input **path=...**> pointe vers les attributs du pojo

<%@taglib uri="http://java.sun.com/jsp/jstl/core" prefix="core"%>



```
<c:forEach  
items="${AllMonuments}"  
var="unMonument">  
...  
...Ici contenu de la boucle  
...  
</c:forEach>
```

```
<core:if test=...>  
...  
...Ici le contenu à afficher si la  
condition dans le test est  
vérifiée  
...  
</core:if>
```

Utilisation des *Includes*
Pour éviter la répétition
Du code

```
<> home.jsp x  
1 <%@ include file="../includes/top.jsp" %>  
2   <div class="container">  
3     <h1>Welcome To JAVA EE Project IPS 2018</h1>  
4   </div>  
5 <%@ include file="../includes/bottom.html" %>
```

Partie Contrôleur

La balise `<mvc:ressources/>` permet de définir un **mapping** pour les ressources : les css, js, image ...

```
class="org.springframework.web.servlet.view.Int
<beans:property name="prefix" value="/pages/" /:
<beans:property name="suffix" value=".jsp" />
beans:bean>
```

```
<annotation-driven/>
```

```
<mvc:resources mapping="/img/**" location="/images/" />
```

Dans la classe `org.springframework.web.servlet.view.InternalResourceViewResolver`, il faudra configurer la manière d'interprétation des **return** des contrôleurs.

l'annotation **@RequestMapping**

l'annotation **@RequestParam**

La classe **Model** et grâce a sa méthode **addModelAttribute (String s, Object o)**

Interpolation **\${monObjet}**

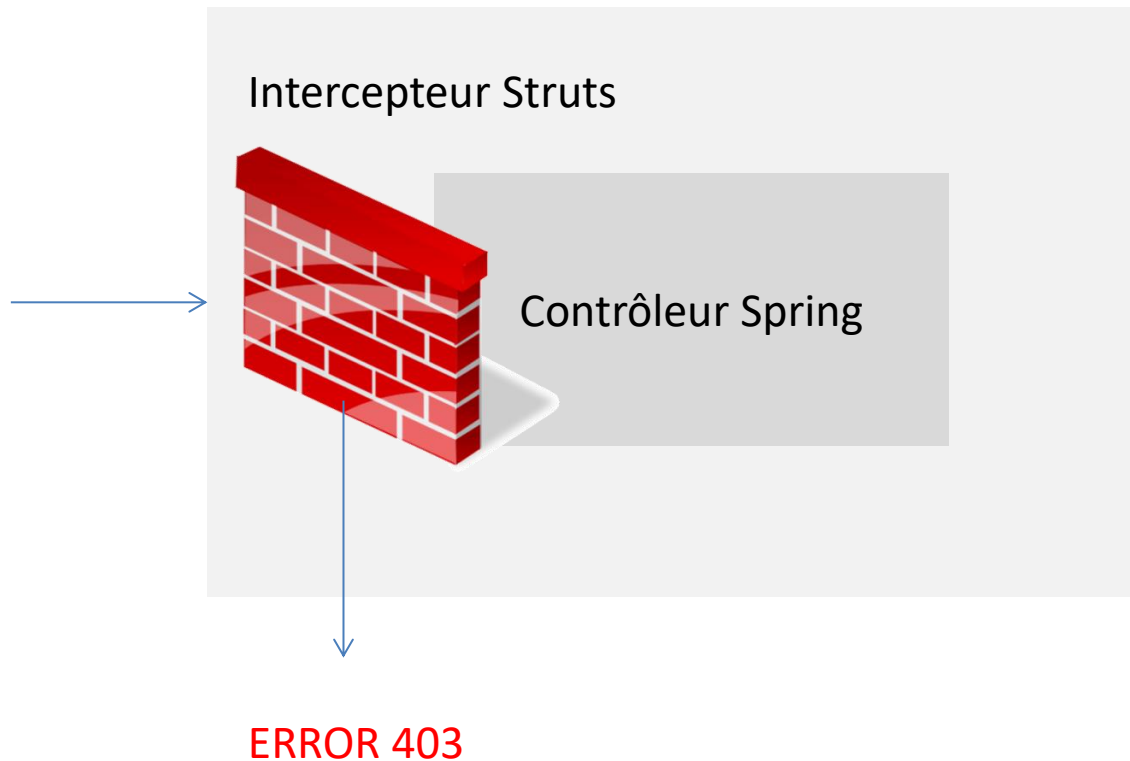
Partie Modèle et persistance

@Annotations

```
@Id
@Column(name="email", length=32, nullable=
@NotEmpty(message="Veuillez saisir une adr
@Pattern(regexp = "[a-zA-Z]+[a-zA-Z0-9\\.
private String email;
@NotEmpty(message="Veuillez mettre un mot
@Column(name="password", length=32, nullab
@Size(min = 8,message="Le mot de passe doi
private String password;
```

- Pour la persistance, utilisation de **Hibernate**

Partie Sécurité



Exemple de www.journaldev.com

Perspectives

- Utilisation des inputs de recherche sur la partie des clés étrangères au lieu d'une liste déroulante
- Profiter de la hiérarchie de la BDD et forcer l'utilisateur à ne pas charger une liste complète
- Circuit touristiques

Merci