

Web-MI4 et SIMO-MI7

Attribut et javabeau

Francis Brunet-Manquat – SIMO

Hervé Blanchon – AW

Basé sur le cours de Leila Kefi-Khelif et D. Enselme

Objectifs

- ❑ Construire le modèle
- ❑ Récupérer des objets dans une JSP
 - Nous nommerons ces objets des JavaBean
- ❑ Transférer un JavaBean d'une servlet à une autre
 - Utilisation de la notion d'attribut
- ❑ Apprendre de portée (scope) : variable associée à la requête , à la page, à la session ou à l'application

JavaBean

- ❑ Un JavaBean (ou Bean) correspond à une tâche bien définie (**logique métier**) comme un étudiant, un groupe, etc.

- ❑ Un JavaBean est une classe Java constituée :
 - D' un **constructeur public sans argument**
 - De **méthodes getXXX (lecture)** et **setXXX (écriture)** associées aux attributs
 - Les attributs sont aussi appelé les propriétés du JavaBean

Aide : penser à utiliser l'IDE pour générer automatiquement les getter et setter

JavaBean : un exemple

```
package jeu;

public class Partie {

    private String mainJoueur;
    private String mainOrdinateur;

    // Constructeur public sans argument
    public Partie() {
        initMainOrdinateur();
    }

    // Getter et Setter
    public String getMainJoueur() {
        return mainJoueur;
    }

    public void setMainJoueur(String mainJoueur)
    {
        this.mainJoueur = mainJoueur;
    }

    ...
}
```

JavaBean et JSP

- ❑ Un JavaBean est inclus dans une page JSP par la balise

```
<jsp:useBean id="partie" class="jeu.Partie" scope="request"/>
```

Ici un objet de type Partie associé à la requête

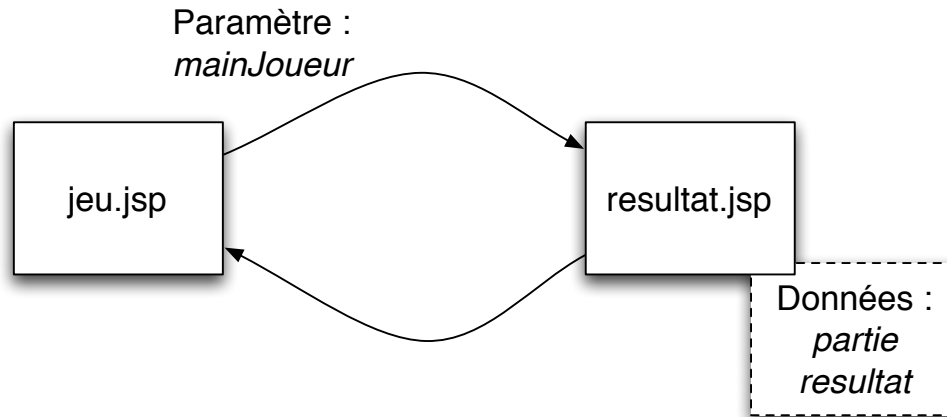
- ❑ Le paramètre d'une requête peut être utilisée directement pour renseigner la propriété d'un bean

```
<jsp:setProperty property="mainJoueur" name="partie"/>
```

Et éviter l'emploi de la méthode `request.getParameter(...)`

Exemple : jeu version 2 (1/2)

Mise en pratique de JavaBean : les classes Partie et Resultat



→ à voir dans le projet Jeu2

Exemple : jeu version 2 (2/2)

- ❑ Dans jeu.jsp, envoi du paramètre mainJoueur

```
<form method="post" action="resultat.jsp">  
  ...  
  <input type="radio" name="mainJoueur" value="pierre"/>  
  ...
```

- ❑ Dans resultat.jsp, création d'un objet de type Partie à partir du paramètre mainJoueur

```
<jsp:useBean id="partie" class="jeu.Partie" scope="request"/>  
<jsp:setProperty property="mainJoueur" name="partie"/>
```

La balise useBean (1/2)

```
<jsp:useBean id = "nomDuBean"  
  scope = "page|request|session|application"  
  class = "nomPackage.nomClasse"  
  type = "typeDuBean">  
</jsp:useBean>
```

- ❑ **id** : Nom de l'instance
- ❑ **class** : classe de l'objet passé en paramètre
- ❑ **type** : classe ou interface de l'objet passé en paramètre
- ❑ **scope** : définit la portée d'un JavaBean
 - page, request, session, application

La balise useBean (2/2)

❑ Différence entre `class` et `type`

- On utilise soit `class` soit `type` dans un `useBean`
- `class` : si l'objet n'hésite pas, il est créé
- `type` : si l'objet n'hésite pas, une erreur est levée
- Par contre `type` autorise le transtypage, par exemple utiliser une interface comme `type`

```
<jsp:useBean id="listeEtudiants" type="java.util.Collection<projet.data.Etudiant>" scope="request"/>
```

❑ Exemple de scope avec la portée session

```
<jsp:useBean id="resultat" class="jeu.Resultat" scope="session"/>
```

La balise `jsp:getProperty`

- ❑ Permet de récupérer les valeurs d'une instance de JavaBean

```
<p>nombre de victoires : <jsp:getProperty name="resultat" property="nombreVictoire" /></p>  
<p>nombre de défaites : <jsp:getProperty name="resultat" property="nombreDefaite" /></p>  
<p>nombre d'égalités : <jsp:getProperty name="resultat" property="nombreEgalite" /></p>
```

Equivalent à `<%= resultat.getNombreVictoire() %>`

La balise `jsp:setProperty`

❑ Permet de modifier les valeurs d'une instance de JavaBean

- Attribuer aux attributs du JavaBean les valeurs récupérés de la requête

```
<jsp:setProperty property="*" name="jeu"/>
```

- Attribuer une valeur de paramètre à un attribut du bean

```
<jsp:setProperty property="mainJoueur" param="mainJoueurParameter" name="jeu"/>
```

- Attribuer une valeur à un attribut du bean

```
<jsp:setProperty property="mainJoueur" value= "feuille" name="jeu"/>
```

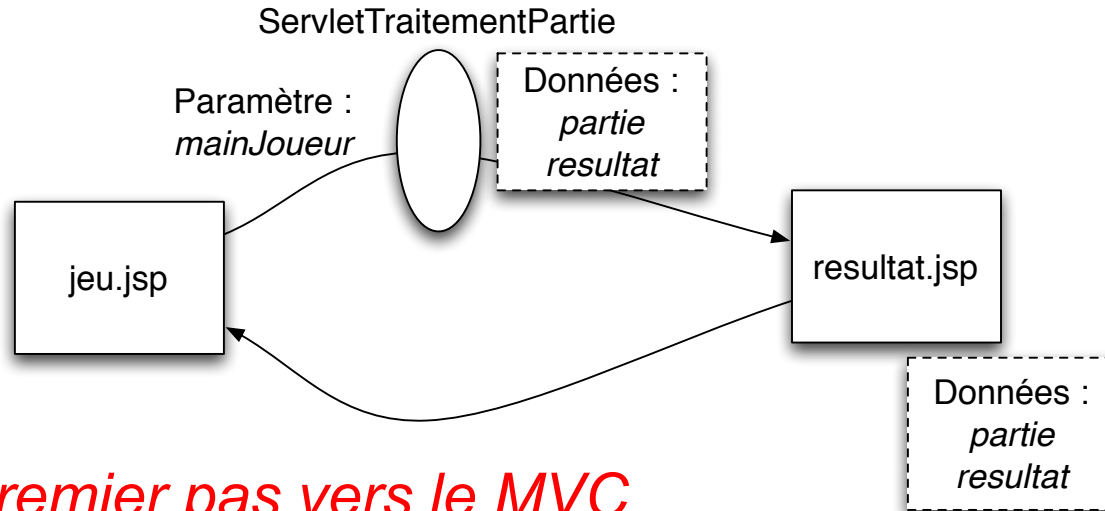
Transmission d'objets entre servlets

Attribut de requête

- ❑ **Envoie** : une servlet envoie un objet vers une autre servlet
On utilise la notion d'attribut de requête ou de session (en distinction des paramètres)
`request.setAttribute("partie", partie);`
- ❑ **Réception** : une JSP reçoit l'attribut sous forme de JavaBean
On utilise l'id partie pour récupérer le bon attribut
`<jsp:useBean id="partie" class="jeu.Partie" scope="request"/>`

Exemple : jeu version 3 (1/3)

- ❑ Une servlet va effectuer le **traitement** et les JSP ne serviront que pour la **présentation** et l'**interaction**



Premier pas vers le MVC

Exemple : jeu version 3 (2/3)

❑ Dans ServletTraitementPartie, **traitement** et **transfère** des résultats

// Créer une objet de type Partie

```
Partie partie = new Partie();
```

```
partie.setMainJoueur(request.getParameter("mainJoueur"));
```

// Mettre l'objet jeu en attribut de requête

```
request.setAttribute("partie", partie);
```

// Méthode qui transfère le contrôle à une autre servlet

```
loadJSP("/resultat.jsp", request, response);
```

Exemple : jeu version 3 (3/3)

❑ Dans resultat.jsp, **récupération** des résultats pour affichage

```
<!-- Récupération des données -->
```

```
<jsp:useBean id="partie" class="jeu.Partie" scope="request"/>
```

```
<jsp:useBean id="resultat" class="jeu.Resultat" scope="session"/>
```

```
<!-- Element de script : Expression -->
```

```
Joueur 
```

```
vs Ordinateur 
```

```
<p>nombre de victoires : <jsp:getProperty name="resultat" property="nombreVictoire" /></p>
```

Votre projet *projet_SIL4.pdf sur Chamilo*

☐ Etape 2 : premier pas vers le MVC

- Utilisation des attributs et des javabeans
- Création d'une servlet ServletTraitementDetails qui prendra en paramètre l'id envoyé par index.jsp, effectuera le traitement (récupérer les détails : l'étudiant et son nombre d'absences) et enverra les informations récupérées à details.jsp pour affichage.

☐ Etape 3 : mise en place du MVC (cours 3)

☐ Etape 4 : mise en place de la persistance (cours 4)