

# *JSP/Servlets*

## *Enchaînement de pages*

B. Mermet  
2009



# *Spécification d'une adresse cible*

- Adresse absolue  
protocole://adresse
  - Adresse relative à l'application  
/adressePage
  - Adresse relative à la page  
adressePage
- 
-

# Moyens d'enchaînement

- Enchaînements effectués par le serveur (non visibles dans la barre d'adresse)
    - Include
      - Enchaînement temporaire
    - Forward
      - Enchaînement définitif
  - Enchaînements effectués par le client
    - SendRedirect
      - Ne plus rien écrire ensuite dans la réponse
    - Lien dans une page / Bouton
      - A la demande
- 
-

# Mise en oeuvre dans une Servlet

## Include

- methode (... req, ... res) {  
    ...  
    RequestDispatcher rd = req.getRequestDispatcher("url");  
    rd.include(req,res);  
    ...  
}

# Mise en oeuvre dans une Servlet

## Forward

- methode (... req, ... res) {  
    ...  
    RequestDispatcher rd = req.getRequestDispatcher("url");  
    rd.forward(req,res);  
}

# *Mise en oeuvre dans une Servlet*

## *SendRedirect*

- methode (... req, ... res) {  
    ...  
    res.sendRedirect("url"),  
}



# Mise en oeuvre dans une JSP

## Include

- Syntaxes possibles

- `<jsp:include page="adresseRelative" />`

- `<jsp:include page="adresseRelative">`

- `<jsp:param name="nomParam" value="valeur"/>`

- ...

- `</jsp:include>`



# Mise en oeuvre dans une JSP

## Forward

- Syntaxes possibles

- `<jsp:forward page="adresseRelative" />`

- `<jsp:forward page="adresseRelative">`

- `<jsp:param name="nomParam" value="valeur"/>`

- ...

- `</jsp:include>`



# *Enchaînement de pages et passage d'information*

- HTML --> Servlet/JSP
  - URLs et GET : mettre un lien avec les paramètres dans le lien

```
<a href="adresse?param1=value1&param2=value2">texte</a>
```
  - Formulaires et GET/Post
- Servlet --> Servlet/JSP
  - SetAttribute
- HTML1 --> HTML2 --> Servlet
  - Attributs cachés
- Au cours d'une exécution

# Formulaire HTML & Traitement

## Généralités

- Déclaration d'un formulaire
  - Syntaxe de base :  

```
<form action="uri" method="get|post"> ... </form>
```
  - Autre attribut
    - Enctype (pour méthode post)
      - Valeur par défaut : *application/x-www-form-urlencoded*
      - Autre cas : *multipart/form-data* pour de l'*upload* de fichier
- Avantages get/post
  - Get : peut être utilisé pour un lien hors formulaire
  - Post : masque les données, permet l'*upload* de fichier...

# Formulaire HTML & Traitement

## Éléments d'un formulaire (1)

- Élément INPUT (1)
    - Syntaxe : `<input type="monType" autresAttributs/>`
    - Types :
      - TEXT : champ texte standard (1 ligne)
      - PASSWORD : champ texte caché
      - CHECKBOX : case à cocher (CC)
      - RADIO : bouton radio (BR)
      - SUBMIT : bouton entraînant l'action associée au formulaire
      - RESET : réinitialisation de tous les éléments du formulaire
      - FILE : pour l'upload d'un fichier
      - HIDDEN : champ caché (voir plus tard)
      - IMAGE : bouton submit sous la forme d'une image
      - BUTTON : un autre bouton que submit et reset
- 
-

# Formulaire HTML & Traitement

## Éléments d'un formulaire (2)

- Élément INPUT (2) : quelques autres attributs
    - Name : nom de l'élément
    - Value : valeur par défaut (sauf CC et BR) ; nom pour un bouton
    - Checked : valeur par défaut (CC et BR)
    - Disabled : invalide
    - Size : taille du champ (dépend du type)
    - Maxlength : longueur maximale du texte (Text)
    - Tabindex : ordre pour l'appui sur <TAB>
- 
-

# Formulaire HTML & Traitement

## Éléments d'un formulaire (3)

- Élément Button
    - Création d'un bouton
    - Plus riche que le `<input type="button"/>`
    - Syntaxe
      - `<button name="nom" value="texte" type="submit|reset|button"/>`
      - `<button name="nom" value="texte" type="submit|reset|button">`
      - ...
      - `</button>`
    - Plusieurs boutons avec des actions différentes dans un même formulaire : utilisation de l'attribut `onclick`
- 
-

# Formulaire HTML & Traitement

## Éléments d'un formulaire (4)

- Élément Select (1)
  - Création d'une comboBox
  - Syntaxe

```
<select name="nom" size="nbLignes">
```

*Options*

```
</select>
```

- Quelques autres attributs :
  - multiple (sélection multiple, simple par défaut)
  - disabled
  - tabIndex

# Formulaire HTML & Traitement

## Éléments d'un formulaire (5)

- Élément Select (2)
  - Options
    - Syntaxe de base  
`<option value="valeur">libellé</option>`
    - Autres attributs
      - Selected
      - Disabled
  - Groupes d'options
    - Syntaxe de base  
`<optgroup label="libellé">`  
*Options*  
`</optgroup>`
    - Autre attribut
      - Disabled

# Formulaire HTML & Traitement

## Eléments d'un formulaire (6)

- Élément TextArea

- Zone de texte de plusieurs lignes
- Syntaxe de base

```
<textarea name="nom" rows="nblignes" cols="nbcols">
```

Valeur d'initialisation

```
</textarea>
```

---

---



# *Formulaire HTML & Traitement*

## *Traitement : principe de base*

- Servlet
  - Suivant la méthode d'appel prévue, surcharger :
    - Soit `void doGet(HttpServletRequestRequest, HttpServletResponse)`
    - Soit `void doPost(HttpServletRequestRequest, HttpServletResponse)`
  - Dans les rares cas où les 2 méthodes doivent être rendues possibles, surcharger :
    - `void service(HttpServletRequestRequest, HttpServletResponse)`
- JSP
  - Rien de spécial à faire !

# *Formulaire HTML & Traitement*

## *Traitement : récupérer les données*

- Appliquer l'une des méthodes suivantes :
    - String `getParameter(String nom)`
    - Enumeration `getParameterNames()`
    - String[] `getParameterValues(String nom)`
    - Map `getParameterMap() : Map de <String, String[]>`
  - Dans une Servlet
    - Au paramètre de type `HttpServletRequest` de la méthode
  - Dans une JSP
    - A l'objet implicite `request`
- 
-

# *Authentication et Session*

- Moyens d'authentification
- Attributs de session
- Ecouteurs de session

