

Upload de fichiers

- Struts permet de gérer facilement l'upload de fichiers au travers des formulaires web



Upload de fichiers

■ Prérequis

- Il faut disposer des librairies suivantes
 - commons-io.jar
 - commons-fileupload.jar

Upload de fichiers

■ Formulaire web d'envoi de fichier (dans la JSP) :

- L'attribut enctype du formulaire doit valoir "multipart/form-data"
- Le formulaire doit contenir au moins un composant de formulaire permettant la sélection d'un fichier <html:file/>

```
<html:form action="/upload.do" enctype="multipart/form-data">  
<html:file property="file"/>  
<html:submit />  
</html:form>
```

Upload de fichiers

■ L'ActionForm :

- L'attribut de l'ActionForm contenant le fichier, doit être de type org.apache.struts.upload.FormFile

```
public class UploadActionForm extends ActionForm {  
  
    private FormFile myFile;  
  
    public FormFile getMyFile() {  
        return myFile;  
    }  
    public void setMyFile(FormFile myFile) {  
        this.myFile = myFile;  
    }  
  
}
```

Upload de fichiers

■ L'Action :

- Dans l'Action, il suffit de récupérer le FormFile en passant par l'ActionForm que l'on vient de définir.

```
public class UploadAction extends Action {  
  
    public ActionForward execute(  
        ActionMapping mapping,  
        ActionForm form,  
        HttpServletRequest request,  
        HttpServletResponse response  
    ) throws Exception  
    {  
  
        UploadActionForm uploadForm = (UploadActionForm) form;  
        FormFile uploadedFile = uploadForm.getMyFile();  
  
        return mapping.findForward("success");  
    }  
}
```

Upload de fichiers

■ Et maintenant ?

- A partir de là, le fichier uploadé est stocké
 - En mémoire
 - Ou sous la forme d'un fichier temporaire dont le nom est généré automatiquement et portant l'extension .tmp
- Nous y reviendrons plus en détail par la suite

Upload de fichiers

■ Et maintenant ?

- On dispose d'une instance de `FormFile` représentant ce fichier qui permet de récupérer :
 - Le nom d'origine : `getFileName()`
 - Le type de contenu : `getContentType()`
 - La taille en octets : `getFileSize()`
 - Le fichier en forme binaire dans son intégralité (`byte[]`) : `getFileData()`
 - Un flux binaire sur le fichier (`InputStream`) : `getInputStream()`
- Il est donc de la responsabilité du développeur d'enregistrer une version définitive du fichier (en base, en système de fichier ou autre) à partir de sa représentation binaire.

Upload de fichiers

- Configuration des paramètres d'upload au niveau de l'application se fait grâce aux attributs de la balise `<controller>` dans `struts-config.xml` :
 - tempDir : permet de spécifier un répertoire pour stocker le fichier de manière temporaire dans le système de fichier du serveur
 - maxFileSize : la taille maximum des fichiers que l'on peut uploader (à suffixer par K,M ou G)
 - memFileSize : la taille maximum d'un fichier à partir de laquelle on passe d'un stockage mémoire à un stockage sous la forme d'un fichier temporaire (à suffixer par K,M ou G)

```
<controller memFileSize="256K" maxFileSize="20M" tempDir="/tmp"/>
```


Upload de fichiers

- A noter qu'il est également possible de faire de l'upload de fichier avec un DynaActionForm
 - Il suffira de spécifier le type FormFile à l'attribut qui devra contenir le fichier

```
<form-bean name='myForm' type='org.apache.struts.action.DynaActionForm'>  
  <form-property name='myFile' type='org.apache.struts.upload.FormFile' />  
</form-bean>
```