STRUTS 2.x

Premiers pas

plan

- Première Installation
- Configuration
- La notion de « Package »
- Constantes de configuration
- Inclusion des fichiers de configuration

Première installation

Première installation

- Pour une utilisation minimale, Struts 2.x n'a besoin que de quelques bibliothèques (à placer dans `WEB-INF/lib`) :
  - struts2-core-2.x.x.jar
  - javassist-3.8.0.GA.jar
    - Nécessaire à partir de Struts en version 2.2.1

Première installation

- La librairie `struts2-core.jar` nécessite d'autres bibliothèques tierces externes pour son propre fonctionnement :
  - xwork-core
  - ognl
  - commons-io
  - commons-fileupload
  - freemarker
- Leurs versions dépendent de celle de struts2-core. Voir http://struts.apache.org/ pour connaître les détails.

Configuration

- Struts 2 respecte donc le principe **MVC2** en positionnant un filtre unique en façade de toute requête Http, c'est le « **Filtre** » `StrutsPrepareAndExecuteFilter`.
- Ce filtre doit bien évidemment être déclaré dans le descripteur de déploiement de l'application.

Configuration

Configuration

- Habituellement, ce filtre sera « mappé » derrière le pattern « **/*** » indiquant que toutes les URLs de l'application seront traitées par Struts.

```
<filter>
        <filter-name>struts2</filter-name>
        <filter-class>
                org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
        </filter-class>
</filter>
<filter-mapping>
        <filter-name>struts2</filter-name>
        <url-pattern>/*</url-pattern>
</filter-mapping>
```

Déclaration du filtre

Mapping des URL « intercetptées » par le filtre

Configuration

- Il est bien évidemment possible de placer le filtre derrière un pattern d'URLs plus restreint.
  - Attention dans ce cas : Les pages faisant appel aux taglib Struts 2 lèveront une exception si l'utilisateur y accède sans passer par le filtre.
- Struts se configure essentiellement par le biais d'un fichier xml. Ce fichier doit se nommer **struts.xml** et être placé à la racine du classpath (Généralement la racine des sources Java).

Configuration

- Le fichier de configuration de Struts est un fichier xml dont la racine est `<struts>` . Exemple :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
        "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
        "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
        <package name="formation-struts2" namespace="/admin"
extends="struts-default">
                <action name="index">
                        <result>/index.jsp</result>
                </action>
        </package>
</struts>
```

La notion de « package »

Package

- Dans l'exemple précédent on trouve :
  - Une balise `<package>` permettant de regrouper des éléments Struts (actions, intercepteurs,...) et permet de subdiviser la configuration d'actions en module logique.
    - Lors de la création d'un « package », il est possible d'hériter d'un autre et même d'en réécrire certaines parties si nécessaire.
    - On choisit généralement de faire hériter notre package d'un autre (`extends`) afin de bénéficier de comportement standards. Très souvent on choisit d'hériter du package « struts-default » défini dans le fichier `struts-default.xml` présent dans la librairie `struts2-core.jar`.
      - Ce dernier propose notamment des intercepteurs par défaut, gérant par exemple la validation des formulaires

Package

- Dans l'exemple précédent on trouve :
  - Un package sera identifié par l'attribut `namespace` et associé à un préfixe d'url qui va être utilisé pour éviter les conflits dans les noms d'actions.
    - Le « / » devant la valeur du namespace ramènera le namespace par rapport à la racine du contexte web

  - Ce namespace n'a pas obligatoirement de valeur :
    - `namespace="/"` correspondra à la racine du contexte web
    - `namespace=""` signifie « tout namespace non identifié » (catch-all)
    - Pas d'attribut namespace du tout équivaut à `namespace=""`

Package

- La balise `<action>` permet de déclarer une action à instancier puis à exploiter suivant la requête Http demandée.
- Lorsqu'il s'agit simplement de « rediriger » la requête vers une vue (comme le montre l'exemple précédent), il n'est pas nécessaire de déclarer d'action spécifique. Dans l'exemple, on indiquait que l'URL `index.action` « redirigeait » vers la vue `index.jsp`.
  - L'extension `.action` est un standard Struts 2 implicite là ou `.do` l'était pour Struts 1.
    - D'ailleurs la même URL sans l'extension donne dans la plupart des cas le même résultat (attention tout de même à certains effets de bord ici ou là).

TP n°3

- Voir Sujet

Les constantes de configuration

Constantes

- Enfin, il est possible de préciser certaines propriétés qui viennent modifier le comportement global de Struts.
- Ces dernières sont contenues dans le fichier `default.properties` (inclus dans struts2-core.jar).
- Leur modification s'effectue :
  - Soit au travers d'une balise `<constant>` en spécifiant le **name** de la propriété et sa **value**.
  - Soit en ajoutant à la racine des sources un fichier `struts.properties` incluant ces nouvelles valeurs.

Constantes

- L'exemple suivant permet d'obtenir des traces plus complète en phase de développement.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
        "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
        "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
        <constant name="struts.devMode" value="true"/>
        <package ..>
...
```

- Pour de plus ample détails, voir : struts.apache.org/2.2.3.1/docs/strutsproperties.html

Inclusion de fichiers de configuration

Inclusion

Inclusion

- Comme c'est le cas dans tous les framework à base de fichier XML, la configuration est divisible dans plusieurs fichiers, le fichier struts.xml reste maître, les autres lui étant inclus grâce à la balise `<include>` :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
        "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
        "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
        <constant name="struts.devMode" value="true"/>
        <include file="frontoffice.xml">
        <include file="backoffice.xml">
...
```