

TP n°1

Toutes les ressources seront déclarées dans un package fr.tp.isima

Exercice 1 : Installer Eclipse !

Pour pouvoir travailler correctement tout au long du cours, vous devez suivre le tutoriel de configuration d'Eclipse et des différents composants que nous utiliserons.

Cela vous permettra également de reproduire la configuration chez vous, si vous êtes amené à utiliser un autre poste.

<https://github.com/bkuchcik/cours-jee-isima-2015/blob/master/documentations/conf-eclipse/Configurer%20eclipse.pdf>

Exercice 2 : Créer une servlet HelloWorld

La direction souhaite activement une application qui affiche hello world. Les utilisateurs demandent cette fonction depuis des mois et enfin le budget est débloqué.

1°) Réaliser une servlet HelloWorld. Déployer sur votre instance de Tomcat sous Eclipse et l'exécuter.

2°) Modifier Hello World pour afficher par un nom. « Hello « nom ». Vous passerez le paramètre en GET (?name=valeur) manuellement.

Exercice 3 : Créer une servlet ListWebContent

Le répertoire WebContent de votre projet Eclipse contient toutes les ressources qui seront déployés. Votre Tomcat travaille d'ailleurs sur une copie de ces fichiers (d'où la nécessité en cas de modification de synchroniser votre répertoire projet et votre répertoire de déploiement).

1°) Ecrire une Servlet ListWebContentServlet qui affiche le répertoire où est déployé le contenu de votre application.

Aide pour réaliser cette première question :

La classe HttpServlet possède une méthode getServletContext(). L'objet ServletContext contient, comme son nom l'indique, bon nombre d'informations sur le contexte d'exécution de l'application.

Sur le ServletContext, dont vous pouvez trouver la documentation ici (<https://docs.oracle.com/javaee/6/api/javax/servlet/ServletContext.html>), la méthode getRealPath() permet de retourner sous forme de String, le chemin d'exécution

2°) Utilisez l'api java.io.File pour construire un objet File à partir du chemin que vous avez trouvé dans le « ServletContext ». Utilisez la méthode listFiles() pour lister l'ensemble des répertoires et affichez les sur la sortie de la Servlet en utilisant un for.

3°) Il existe deux autres méthodes de File avec pour nom listFiles() :listFiles(FileFilter) et listFiles(FileNameFilter). Utilisez le FileNameFilter pour ne pas afficher les répertoires META-INF et WEB-INF.

Copiez le code suivant :

```
final File[] listOfDirectoriesAndFiles = webContent.listFiles(new FilenameFilter()
{
    @Override
    public boolean accept(File dir, String name) {
        return !"WEB-INF".equals(name) && !"META-INF".equals(name);
    }
});
```

Que se passe-t-il au moment où vous faites ctrl+s ?

4°) Pour les plus braves, vous pouvez tenter d'essayer de remplacer votre for par l'API stream de Java 8.

Utilisez Arrays.asList().map() pour passer d'une liste de File() à une liste de nom de fichier (String) puis forEach() pour afficher chaque item.

Que se passe-t-il si vous essayez d'utiliser un java.io.Writer directement dans le forEach() ?