

# J2EE – TP2 : couche de présentation

L'ensemble des ressources nécessaires au TP est disponible dans dépôt GitHub suivant :

<https://github.com/jdassonvil/epsi>

Le corrigé est également disponible :

<https://github.com/jdassonvil/epsi-j2ee-tp1>

*Not e : un schéma et un jeu de données enrichi est fourni dans la cadre de ce deuxième TP.*

## 2.1 Mise en place d'un serveur Tomcat

1. Téléchargez Tomcat 8 sur le site d'Apache, décompressez l'archive (par exemple sous Windows dans C:\Programmes\Apache)
2. Dans Eclipse affichez la vue serveur (Window > Show View > Server) et créez un nouveau serveur Tomcat 8
3. Essayez de lancer le serveur
  - a. Si le port par défaut (8080) est déjà utilisé, changez-le pour un autre port (par ex. 9000).
4. Dans votre navigateur ou avec la commande unix `curl` accédez à votre serveur local.
  - a. Un message d'erreur de Tomcat doit s'afficher.

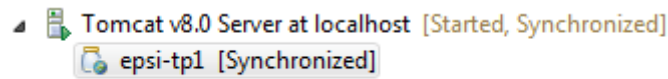
## 2.2 Import ou migration du projet

1. Si vous décider de continuer le TP depuis le projet initié lors de la première séance, vous devez convertir votre projet en projet web :
  - Sous Eclipse, dans les propriétés du projet, ajoutez le facet « Dynamic Web Module » à votre projet
    - Lors de la configuration accédez au paramètres avancés, le paramètre « Content Directory » doit pointer sur /src/main/webapp (convention MAVEN)
    - Sur certaines versions d'Eclipse l'opération peut échouée et doit être répétée une seconde fois
  - Dans les propriétés du projet, ajoutez les dépendances Maven à la liste « Deployment Assembly »
  - Si la conversion c'est bien passée, Eclipse reconnait votre application comme une application web déployable et vous pouvez à présent l'ajouter au serveur Tomcat.
2. Si vous décider de continuer à partir du corrigé :
  - Supprimez d'Eclipse une version antérieur du projet si existant
  - Mettez à jour votre dépôt vers le tag 2.0
  - Importez le projet : File > Import > Existing Maven Project
    - Eclipse configure automatiquement votre projet grâce au contenu du POM.xml

## 2.3 Premier Servlet up and running

1. Dans la vue serveur, associez votre application à l'instance Tomcat :

- Clic droit sur le serveur > Add and remove
- Vous devez obtenir le résultat suivant :



2. Dans le fichier POM.xml ajoutez la dépendance suivante

- Group id : javax.servlet
- Artifact id : javax.servlet-api
- Version : 3.1.0

3. Créez un package front

4. Dans ce package écrivez votre premier Servlet qui répondra un « Hello World » sur une requête GET

5. S'il n'existe pas déjà créez un fichier web.xml dans /src/main/webapp

6. Configurez le fichier web.xml pour répondre sur /hello

7. Testez votre API

## 2.4 Appels à la couche DAO depuis un Servlet

8. Ecrivez un servlet qui affichera la liste des artistes dans une réponse HTML

## 2.5 Les JSP

- Ajoutez au projet une JSP qui affichera un Hello World  
⇒ Dans /src/main/webapp
- Configurez le web.xml pour afficher cette page à la racine de votre application
- Mettez à jour la JSP pour afficher la liste des artistes comme effectué précédemment avec un Servlet

## 2.6 MVC

- A l'aide des JSP et des Servlet, reprenez votre code pour appliquer proprement le pattern MVC