

TP 1 Servlets

Matthieu Foucault
matthieu.foucault@labri.fr
<http://www.labri.fr/~mfoucaul>

18 février 2013

Q1.1 Prérequis

Pour ce TP, il ne sera pas nécessaire d'utiliser les machines virtuelles hébergées sur morpheus (cependant, rien ne vous empêche de le faire si vous avez réussi à démarrer des applications graphiques via ssh -Y).

Comme vous avez pu le constater à vos dépendis lors du TP précédent, coder et déployer des servlets sans IDE digne de ce nom est relativement pénible. Pour ce TP, nous allons donc utiliser Eclipse IDE for Java EE Developers, à télécharger sur le site eclipse.org.

Nous allons mettre de côté les `HttpServlet`s pour nous intéresser à Jersey, une api qui permet de coder des servlets plus efficacement, en utilisant les annotations java.

La classe correspondant au servlet `helloWorld` se trouve dans le fichier `teaching/enseirb/pg219/TP2/MyHelloServlet.java`, sur mon site perso.

Q1.2 Déploiement avec Eclipse

1. Dans eclipse, créez un nouveau projet de type Dynamic Web Application, nommé 'PhoneBook', en utilisant un runtime Apache Tomcat 7.0 (téléchargez Tomcat si nécessaire).
2. Dans les propriétés du projet, ajoutez le facet JAX-RS, afin de gérer les services REST. (Properties -> Project Facets -> JAX-RS). Modifiez la configuration afin d'avoir les mêmes informations que dans la Figure 1.

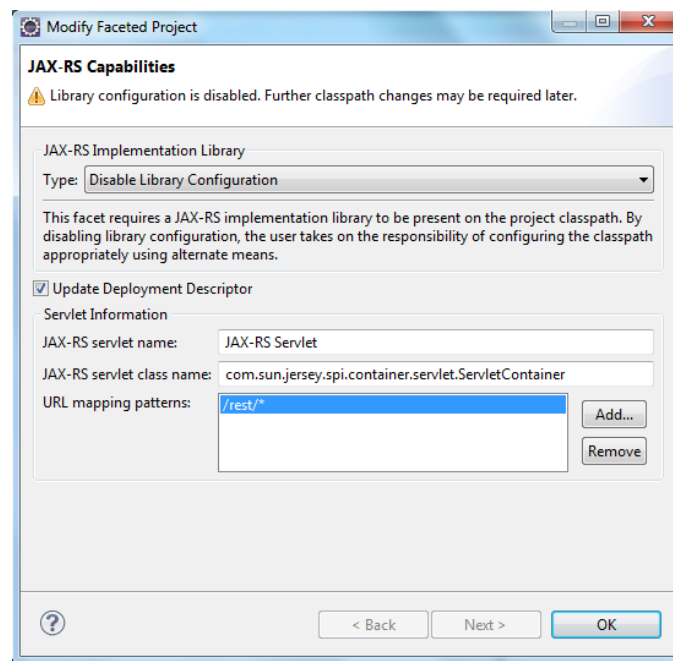


FIGURE 1 – Configuration du servlet

3. Ajoutez les .jar nécessaires dans le fichier `WEB-INF/lib` de votre projet. Il sont disponibles dans le dossier `teaching/enseirb/pg219/TP2` sur mon site perso.
4. Dans `Java Ressources/src`, ajoutez le fichier `MyHelloServlet.java`.
5. Faites un clic droit sur votre serveur (dans la vue `servers` d'Eclipse), puis "Add and Remove..." et ajoutez la ressource `PhoneBook`.
6. Démarrez le serveur, votre servlet `HelloWorld` est accessible à l'adresse `http://localhost:8080/PhoneBook/rest/helloWorld`.

7. Vous pouvez ajouter une page `index.html` dans le dossier `WebContent`. Créer un simple formulaire appelant la méthode `POST` de `MyHelloServlet`.

Cette page sera accessible à l'adresse `http://localhost:8080/PhoneBook/`

Q1.3 Manipulation de contenu JSON

Comme vous avez pu le deviner avec le nom du projet, le but de ce TP va être de créer un annuaire téléphonique. Une base de donnée `mongodb` est installée sur `http://mafoucault.rmorpheus.enseirb.fr`, et accessible via un serveur `Rest`, dont la documentation est disponible ici : `https://sites.google.com/site/mongodbjavarestserver/`

L'énoncé du TP contient les commandes suffisantes pour implémenter les fonctionnalités demandées.

Ajoutez la classe `Entries.java` a votre projet. Cette classe contient une fonction qui traite les requêtes `GET`, et qui renvoie la liste des entrées de la table `/test/phoneBook` de la base `mongodb`. Le résultat affiché est du `JSON` non formaté, ce qui n'est pas approprié pour un utilisateur courant.

Nous allons donc parser ce `JSON` pour afficher une page `html` contenant la liste des entrées de la table `/test/phoneBook`

La manipulation de contenu `JSON` se fait via la bibliothèque `Jackson`. Ajoutez les `.jar` contenus dans `libs-jackson.zip` dans le dossier `WEB-INF/lib`.

Utilisez un `ObjectMapper` pour récupérer les différents champs de la chaîne de caractères `JSON`, puis affichez ceux ci dans la réponse. Un résultat du type `"Nom : Toto, Adresse : 42 Fake St."` est suffisant.

La première action à exécuter est:

```
ObjectMapper mapper = new ObjectMapper();
ArrayNode entries = (ArrayNode) mapper.readTree(response);
```

Q1.4 Création d'objet JSON

Afin de ne pas tous écrire dans la même table, un numéro est inscrit en haut à gauche de votre énoncé. Si vous avez le numéro **X**, remplacez *phoneBook* par *phoneBookX* dans les requêtes vers le serveur où se trouve la base de données.

Afin d'ajouter un objet dans la base de données, faut envoyer une requête de type `PUT` vers l'URL `"http://mafoucault.rmorpheus.enseirb.fr/write/test/phoneBook"`, avec pour contenu un `Objet JSON` à insérer dans la table.

Exemple: `resource.entity("{\"name\":\"titi\"\", MediaType.APPLICATION_JSON).put();`

Bien sur, créer la chaîne de caractères à la main n'est pas idéal. Vous devrez utiliser l'`API Streaming` de `Jackson` pour serialiser l'objet `JSON` dans un `StringWriter`. (cf. `http://wiki.fasterxml.com/JacksonInFiveMinutes`).

Q1.5 Jackson Data Binding

Créez une classe *Contact* contenant les différents champs sauvegardés dans l'annuaire (nom, adresse, etc...)

Modifiez les services permettant de lister et d'ajouter les entrées dans l'annuaire afin d'utiliser le `DataBinding` de `Jackson`.

Un exemple se trouve ici : `http://wiki.fasterxml.com/JacksonInFiveMinutes` , dans le chapitre `Full Data Binding (POJO) Example`.