

Ce TP vise à aborder l'utilisation et le développement des ressources de type REST. Dans un premier temps, nous allons utiliser la commande `curl` comme client afin de manipuler une ressource «todo list». Dans un deuxième temps, vous allez re-développer cette ressource en Java suivant le standard JAX-RS.

1. REST, côté client

- (a) Téléchargez sur Moodle l'archive `todo.zip` et décompressez-la. Placez vous ensuite dans le répertoire `todo_REST` et lancez la commande `run.sh`¹. Cette commande téléchargera les dépendances nécessaires et lancera le serveur REST correspondant à la *todo list*.
- (b) Afin de vérifier le fonctionnement de notre serveur, nous allons consulter la liste des tâches à réaliser (qui devrait être vide, soit []). Pour cela, dans un deuxième terminal, utilisez la commande `curl -i http://localhost:8080/tasks`. Cette commande envoie une requête HTTP GET à la ressource `tasks`. L'option `-i` permet l'affichage des en-tête HTTP, ce qui vous permet de voir les codes de retour ainsi que les types de données retournés.
- (c) Une tâche est définie par un nom et une description, la ressource lui attribue un identifiant automatiquement. En vous aidant de la page de manuel de `curl`, créez les tâches suivantes :
 - nom : tâche 1, description : tester les opérations CRUD
 - nom : tâche 2, description : développer en JAX-RSPour chaque création, vérifiez le code de retour HTTP et notez l'URI de la nouvelle ressource.
- (d) Demandez à la ressource la liste des tâches existantes. Demandez ensuite le détail de la tâche 2.
- (e) Modifiez la description de la tâche 2 en “recenser les URI et opérations HTTP utilisables”. Vérifiez ensuite que la modification a été prise en compte.
- (f) Pour l'instant les réponses de la ressource étaient représentées en JSON. Demandez lui de vous donner la liste des tâches en XML.
- (g) Vous pouvez maintenant détruire la tâche 1. Vérifiez ensuite qu'elle a bien été détruite.
- (h) La ressource REST a été développée en `node.js`, c'est à dire du *Javascript* qui s'exécute côté serveur. Consultez le code source et pour chaque opération HTTP, recensez les URI utilisables.

2. Développement de la ressource `tasks`

Nous allons maintenant développer la même ressource en Java avec JAX-RS. Vous devrez gérer les mêmes opérations et URI que la ressource fournie en `node.js`. Pour vous guider dans votre développement, vous pourrez suivre et vous inspirer du tutoriel fourni sur Moodle (`tutorielRest.zip`).

1. Il sera peut-être nécessaire de rajouter le droit d'exécution à ce fichier.

- (a) Télécharger l'archive `tutorielRest.zip` sur Moodle et décompressez-la. Ouvrez ensuite le fichier `tutoriel.html` avec votre navigateur. Ce tutoriel détaille les étapes de mise en place et de développement d'une ressource REST en Java.
- (b) En vous inspirant du tutoriel, développez la ressource *todo list* reproduisant celle fournie en `node.js`.