

TP10

WEB SERVICES AVEC REST (2)

A rendre pour la semaine suivant la séance d'exercices.

1 Installation du client REST pour navigateur RESTED

- Navigateur open source Firefox, se rendre sur cette page pour l'installer si nécessaire : <https://www.mozilla.org/fr/firefox/new/>
- RESTED : dans Firefox, ≡ > Modules Complémentaires, Extensions, Rechercher RESTED puis Ajouter à Firefox. L'icône </> doit apparaître en haut à droite de Firefox.

2 Envoyer une requête GET

Dans le précédent TP, nous avons mis en place un premier web service REST élémentaire consistant à retourner une partie du path avec un message de bienvenue.

L'adresse suivante dans le navigateur :

`http://localhost:8080/JerseyDemo/helloworld/<votre prénom>`

devrait retourner *Hello, <votre prénom> !*

- 1) Avec RESTED, envoyer la même requête. Donner la réponse, le header et le payload retourné.
- 2) Dans la même application HelloWorldService créé au TP précédent, ajouter maintenant un service permettant de récupérer un paramètre indiqué par l'utilisateur dans l'URL (Query). Par exemple :

`http://localhost:8080/JerseyDemo/helloworld/?name=<votre prénom>`

doit retourner *Bienvenue sur la page <votre prénom> !*

Indice : utiliser l'annotation `@QueryParam("name")` au lieu de `@PathParam("name")`

3 Envoyer une requête POST

- 3) Créer une classe Etudiant avec les attributs nom, adresse et faculté et un constructeur permettant d'initialiser une instance d'Etudiant. Ajouter une méthode toString() qui produit un String structuré de la manière suivante : « nom : » nom « , adresse : » adresse « , faculté : » faculté.
- 4) Dans l'application HelloWorldService, ajouter un service permettant de créer un nouvel étudiant à partir d'un payload structuré en XML envoyé par le client web à l'URL suivant :

`http://localhost:8080/JerseyDemo/etudiants`

Pour le formatage en XML, l'élément racine de la structure XML doit être le nom de la classe commençant par une minuscule. Les attributs sont représentés par des sous éléments de cet élément racine. Après instanciation de la classe Etudiant, faire afficher cette instance dans la console de TOMCAT via la déclaration : `System.out.println(etu)` ; où `etu` est la variable référençant l'instance d'Etudiant créée. Il faudra bien entendu analyser le contenu du payload en XML pour identifier les attributs de l'objet à instancier.

- 5) Avec RESTED, envoyer cette requête POST en composant à la main le payload en XML selon la structure choisie. Pensez à spécifier dans le header le format de données envoyé. Donner la réponse, le header et le payload retourné dans RESTED. Vérifier que l'objet s'affiche correctement dans la console de TOMCAT, conformément à la méthode `toString()`.

4 Travail Maison

- Les réponses à chacun des questions posées dans le TP avec une copie d'écran montrant les sorties console TOMCAT le cas échéant
- Votre projet complet et fonctionnel exporter depuis Eclipse au format .zip