STITUT SUPERIEUR DES ETUDES TECHNOLOGIQUES
DE SFAX

Département Technologies de Informatique

Atelier SOA -TP10
Client Web JAVA pour un service web REST

Objectifs

Consommer un Service Web REST avec un client Web JAVA

1. **Récupérer les données d'une JSP**
   - A partir d'un formulaire
   - A partir d'un lien hypertexte
2. **Appeler un service à travers une servlet**
   - Réaliser le mapping URL/Objet
   - Réaliser le mapping Objet/JSON

A. Créer un projet web (webApp) avec Maven

1. Créer un projet **MAVEN (maven-archetype-webapp)** dans **IntelliJ IDEA** nommé **jax-rs-client-jsp** (sous le dossier **workspace)** ayant les caractéristiques suivantes :
   - groupId : **ws.rest**
   - artifactId : **jax-rs-client-jsp**
   - version : **1.0-SNAPSHOT**

2. Configurer ce projet pour supporter le développement des services web REST (dépendances de **servlet**, de **JERSEY** et de **Gson** et la dépendance au projet « **jax-rs-arc** ») et déclarer le plugin « **tomcat7** » pour exécuter l'application web avec le port « **9997** »

B. Opérations d'ajout et consultation

3. Créer, sous le dossier « **main/webapp** » un fichier « **formPersonne.jsp** » qui permet de définir un formulaire http présentant trois champs textes:
   - **Id** nommé « **id** »
   - **Nom** nommé « **nom** »
   - **Age** nommé « **age** »

La soumission du formulaire redirige la page vers la servlet « **FormPersonneAction** » avec la méthode « **POST** ».

4. Créer, sous le dossier « **main** », un dossier « **java** ».
5. Rendre le dossier « **java** » comme un dossier « **Sources Root** ».
6. Créer, sous « **java** » un package « **ws.rest.servlet** ».
7. Créer, dans le package « **ws.rest.servlet** », une servlet « **FormPersonneAction** ». Cette servlet réalise les opérations suivantes :

   a) Récupère les paramètres du formulaire « **formPersonne.jsp** »
   b) Crée une instance de la classe « **Personne** » avec les paramètres reçus du formulaire.
   c) Configure une connexion avec la ressource « /**person** » du projet « **jax-rs-arc** »
   d) Appeler le path « /**add** » avec la méthode « **POST** » en envoyant comme paramètre l'objet « **Personne** » déjà instancié.
   e) Appeler le path « /**getAll** » avec la méthode « **GET** » pour récupérer la liste des personnes existantes au format « **JSON** ».
   f) Convertir le tableau JSON en une liste d'objets « **Personne** » en utilisant « **Gson** » de Google
   g) Placer cette liste comme un attribut dans la requête http.
   h) Rediriger l'affichage vers la page « **listPersonnes.jsp** »

8. Créer, sous le dossier « **main/webapp** » un fichier « **listPersonnes.jsp** » qui permet d'afficher sous forme tabulaire la liste des personnes existantes :
   - Récupérer la liste des personnes (définie comme attribut dans l'objet « **request** »
   - Parcourir la liste pour afficher l'ensemble des personnes
   - A chaque personne sont associés deux liens hypertextes :
     - ✓ **Editer** : pour mettre à jour la personne en question. Ce lien redirige la requête http vers une servlet « **ListPersonnesAction** » avec deux paramètres :
       1. **action** ayant la valeur : « **editer** »
       2. **id** : la valeur de l'id de la personne en question
     - ✓ **Supprimer** : pour supprimer la personne en question. Ce lien redirige la requête http vers une servlet « **ListPersonnesAction** » avec deux paramètres :
       1. **action** ayant la valeur : « **supprimer** »
       2. **id** : la valeur de l'id de la personne en question

C. Opération de suppression

9. Créer, dans le package « **ws.rest.servlet** », une servlet « **ListPersonnesAction** ». Cette servlet réalise les opérations suivantes :

   a) Récupère les paramètres suivants :

      - **action** : qui détermine le type de traitement à réaliser (editer ou supprimer)
      - **id** : id de la personne sélectionnée

   b) S'il s'agit de l'action « supprimer », réaliser les opérations suivantes :

- Configurer une connexion avec la ressource « /**person** » du projet « **jax-rs-arc** ».
- Appeler le path « /**{id)/delete** » avec la méthode « **GET** » en envoyant comme paramètre l'id de la personne à supprimer.
- Rediriger la requête en mode « **GET** » vers la servlet « /**FormPersonneAction** » pour afficher la liste des personnes restantes.

D. Opération de mise à jour (édition)

10. Reprendre la servlet « **ListPersonnesAction** » pour traiter le cas de l'action « **editer** » :

- Appeler le path « /**{id}/get** » avec la méthode « **GET** » en envoyant comme paramètre l'id de la personne à éditer pour récupérer comme réponse la personne au format « **JSON** ».
- Convertir l'objet « **JSON »** en un objet « **Personne** » (JAVA) en utilisant « **Gson** » de Google.
- Placer cet objet comme un attribut dans la requête http.
- Rediriger l'affichage vers la page « **formPersonne.jsp** ».

11. Reprendre le formulaire « **formPersonne.jsp** » pour réaliser les opérations suivantes :

- Ajouter un script JSP pour récupérer l'attribut présentant l'objet « **Personne** » à éditer.
- Si cet objet est existant, placer ses attributs id , nom et age respectivement dans des variables **id**, **nom** et **age**.
- Ces variables sont utilisées comme valeurs pour les champs convenables du formulaire. (Ces valeurs ont comme valeurs par défaut des chaines de caractères vides en cas d'absence d'objet « **Personne** » à éditer càd en mode ajout)
- En cas de présence d'objet « Personne » à éditer, rendre le champ « id » en mode lecture seule.

- Lors de la soumission du formulaire, et pour distinguer entre le mode « **ajout** » et le mode « **edition** », ajouter dans le formulaire un champ « **hidden** » qui prend comme valeur :
  - ✓ **ajout** : en cas de saisie de nouvelle personne
  - ✓ **edit** : en cas de saisie d'une personne existante

12. Reprendre la servlet « **FormPersonneAction** » pour réaliser les opérations suivantes :

- Récupérer le paramètre « **mode** ».
- Si le mode est « **edit** »,
- Appeler le path « /**update** » avec la méthode « **PUT** » en envoyant comme paramètre l'objet « **Personne** » déjà instancié (à la base des paramètres reçus du formulaire)
- Appeler le path « /**getAll** » avec la méthode « **GET** » pour récupérer la liste des personnes existantes au format « **JSON** ».
- Convertir le tableau JSON en une liste d'objets « **Personne** » en utilisant « **Gson** » de Google
- Placer cette liste comme un attribut dans la requête http.
- Rediriger l'affichage vers la page « **listPersonnes.jsp** ».