

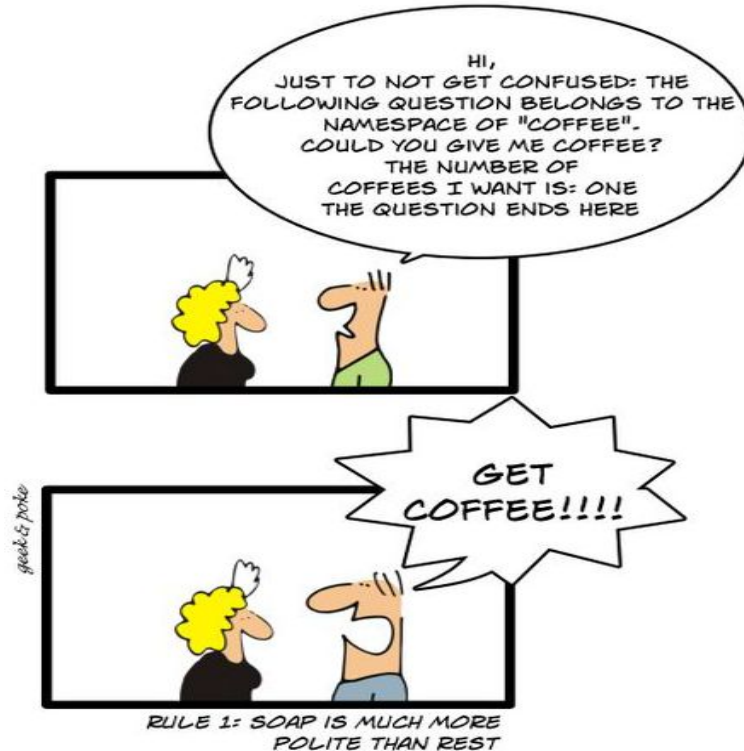
JAX-RS et Servlets

Introduction

Que connaissez-vous comme type de services web ?

Introduction

SERVICE CALLING MADE EASY



REST

Style d'architecture web reposant sur HTTP

Utilise les méthodes HTTP

Pas de format, ni de pré-requis

Servlets

Composant Java générant du contenu
dynamiquement

Spécification javax.servlet

Actuellement version 3.0

JAX-RS

La spécification JAX-RS décrit REST en Java

Il existe de multiples implémentations :

1. JBoss RESTEasy
2. Jersey

Servlets

Servlet

S'exécute côté serveur

Pas d'IHM

Génération de pages HTML

Servlet

Le conteneur de servlets (tomcat, ...) gère le cycle de vie des servlets

Mécanisme de cache, asynchrone ...

Modification direct de la réponse HTTP

Fonctionnement

- Un client fait une requête HTTP au serveur
- La requête est transmise au conteneur par le serveur
- Le conteneur décide à quelle Servlet transférer la requête (comment ?)
- Le conteneur envoie la réponse au client

Cycle de vie

- Création et initialisation de la servlet
- Gestion de requêtes
- Destruction et libération des ressources

Quand a lieu la création d'une servlet et pourquoi ?

Application Web

Une application web est un espace virtuel

Le fichier web.xml décrit toute la configuration

Doit être placé dans le répertoire WEB-INF

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="
http://java.sun.com/xml/ns/javaee"
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.
sun.com/xml/ns/javaee/web-app_2_5.xsd"
    version="2.5">
  <servlet>
    <servlet-name>appstats</servlet-name>
    <servlet-class>com.google.appengine.AppstatsServlet </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>appstats</servlet-name>
    <url-pattern>/appstats/*</url-pattern>
  </servlet-mapping>
</web-app>
```

Conclusion

Si vous voulez en savoir plus : http://jlguillaume.free.fr/www/documents/teaching/ntw1314/LI385_C1_servlets.pdf

Très verbeux

A ne pas utiliser directement pour le projet !

JBoss RESTEasy

Introduction

Implémentation complète de JAX-RS

Embarque un client

Gère l'asynchrone, compression, cache ...

JAX-RS vs Servlet

Pourquoi JAX-RS plutôt que Servlet ?

JAX-RS vs Servlet

Pourquoi JAX-RS plutôt que Servlet ?

- Tout est annotation
- Fait pour répondre à ce type de besoin
- Très simple d'utilisation

Example

```
@Path("/test")  
public class Endpoint {
```

```
    @GET
```

```
    @Produces("text/plain")
```

```
    public Response hello() {
```

```
        return Response.ok("Hello World").build();
```

```
    }
```

```
}
```

@Path

Peut se mettre sur une classe / méthode

Définit le chemin d'accès à la ressource

Exemple : `@Path("/users/fetch")`

Paramètre

Plusieurs moyens de passer des paramètres :

- @PathParam
- @QueryParam
- @FormParam
- @HeaderParam
- @CookieParam

@PathParam

Permet de récupérer un paramètre défini dans l'URL

```
@Path("/mobile/play/{modelId}")  
public Response play(@PathParam("modelId") String modelId)  
{  
    return Response.ok(modelId).build();  
}
```

La requête sera /mobile/play/test

@QueryParam

Permet de récupérer un paramètre de l'URL

```
@Path("/mobile/play")  
public Response play(@QueryParam("modelId") String  
modelId) {  
    return Response.ok(modelId).build();  
}
```

La requête sera /mobile/play?modelId=test

@DefaultValue

Permet de donner une valeur par défaut si le paramètre n'est pas spécifié.

```
@Path("/mobile/play")
public Response play(@QueryParam("modelId")
@DefaultValue("test")
String modelId) {
    return Response.ok(modelId).build();
}
```

Que renvoie la requête /mobile/play ?

@Produces

Spécifie les types mimes que renvoie le service web

Propre à la classe ou à une méthode

@Consumes

Spécifie les types mimes acceptés par le service web

Propre à la classe ou à une méthode

@GZIP

Permet de GZIP la réponse d'une requête

```
@GZIP
```

```
public Response drop(@GZIP MultipartFormDataInput input) throws IOException
```

@Cache

Uniquement sur les requêtes GET

Équivalent du Cache-Control HTTP

Ne fonctionne que sur un code 200

Déploiement

Avoir un web.xml bien configuré

Préparer une classe Application

Générer un fichier war

web.xml

```
<web-app>
  <display-name>test</display-name>
  <context-param>
    <param-name>resteasy.scan</param-name>
    <param-value>>false</param-value>
  </context-param>
  <servlet>
    <servlet-name>Resteasy</servlet-name>    <servlet-class> org.jboss.resteasy.plugins.
server.servlet.HttpServletDispatcher </servlet-class>
    <init-param>
      <param-name>javax.ws.rs.Application</param-name>
      <param-value>com.test.App</param-value>
    </init-param>
  </servlet>

  <servlet-mapping>
    <servlet-name>Resteasy</servlet-name>
    <url-pattern> /*</url-pattern>
  </servlet-mapping>
</web-app>
```

Application

Classe Java décrivant les ressources utilisées

Singletons : Ne sont instanciés qu'une fois

Classes : Différents scopes

Application

```
public class App extends Application {
```

```
    @Override
```

```
    public Set<Class<?>> getClasses() {
```

```
        Set<Class<?>> clazz = new HashSet<>();
```

```
        clazz.add(Endpoint.class);
```

```
        return clazz;
```

```
    }
```

```
}
```


JSON

Gestion automatique

Les bibliothèques principales : jackson, gson

Ne faites pas cela vous même !

JSON

```
@POST
```

```
@Produces(MediaType.APPLICATION_JSON)
```

```
public Product create() {
```

```
    return new Product("test", 5);
```

```
}
```

```
@GET
```

```
@Produces(MediaType.APPLICATION_JSON)
```

```
public Product get(String name) {
```

```
    Product p = productService.getByName(name);
```

```
    return p;
```

```
}
```

XML

Automatique avec l'API JAXB

Déjà intégrée à Java

Uniquement des annotations

Gestion des erreurs

Dès lors qu'une exception est levée, celle-ci sera catchée

Définit la façon dont on gère les exceptions

Exemple

```
@Provider  
public class ApiErrorMapper implements ExceptionMapper<ApiError> {
```

```
    @Override  
    public Response toResponse(ApiError apiError) {  
        return Response.status(apiError.getStatus())  
            .entity(apiError)  
            .type(MediaType.APPLICATION_JSON)  
            .build();  
    }
```

```
}
```

Conclusion

JAX-RS est très simple d'utilisation

Très utilisé en entreprise

Vous devrez l'utiliser pour vos projets