FAM - Crida servei REST

Taula de continguts

1. Introducció

En aquest article, s'explica com fer crides a serveis REST des d'una aplicació tipus FAM.

Aquest tipus de serveis té una estructura molt més simple que la de serveis web (fins i tot es poden fer crides per url)

2. Crida des del FAM

Donat un servei REST publicat volem fer una crida des d'una aplicació FAM. Ho farem amb l'ajut de la llibreria Jersey [1]

2.1. Dependències

Primer de tot necessitem definir al pom.xml la dependència amb la llibreria del client de Jersey

```
<dependency>
    <groupId>com.sun.jersey</groupId>
    <artifactId>jersey-client</artifactId>
    <version>1.8</version>
</dependency>
```

Per problemes de llibreries s'ha d'afegir la llibreria client de weblogic

```
<dependency>
    <groupId>weblogic</groupId>
    <artifactId>wlfullclient</artifactId>
    <version>10.3.0.0</version>
```

```
    <scope>provided</scope>
</dependency>
```

2.2. Classe client

Un cop definida la dependència podem fer el client que farà les crides

```
Client client = Client.create();
WebResource wr = client.resource(url);
ClientResponse response = wr.accept("application/xml").get(ClientResponse.class);
```

A partir d'una url et fa la crida

2.3. Exemple repositori de plantilles

En aquest exemple (implementat al projecte eNoti) es tracta la resposta per validar l'estat i després es parseja per poder carregar la classe resposta i poder llegir la informació, com ara el contingut de la plantilla.

```
if (response.getStatus() != 200) {
    throw new RuntimeException("Failed : HTTP error code : " + response.getStatus());
}

PlantillaRestResposta output = response.getEntity(PlantillaRestResposta.class);
byte[] byteArray = output.getPlantilla();
```

La resposta real del repositori de plantilles és un xml com

```
<plantillaRestRespostaplantillaRestResposta>
    <nomPlantilla>TramesaGen</nomPlantilla>
    <plantilla>bytearrayplantilla</plantilla>
    <versio>1.1</versio>
</plantillaRestResposta>
```

Per poder parsejar aquesta resposta es defineix la classe PlantillaRestResposta com

```
package cat.diba.jee.enoti.ws.repositoriplantilles;

import javax.xml.bind.annotation.XmlRootElement;

@XmlRootElement
public class PlantillaRestResposta {
    private byte[] plantilla;
    private String versio;
    private String nomPlantilla;

    public PlantillaRestResposta() { }
    [... Getters i Setters ...]
```

```
}
```

La situació óptima seria que en fer un servei REST es donessin també les classes de resposta en un jar

2.4. Exemple validació VUS

L'exemple del VUS és anàleg, però la resposta és molt més complexa i s'han de fer multiples objectes relacionats per poder tractar-la.

Al projecte eNoti hi ha un integració feta.

3. Proves

Per poder provar aquests serveis, es pot fer des del navegador o amb l'ajuda de plugins que permeten definir la crida amb més detall (per exemple, paràmetres per post)

Per Chrome tenim el plugin Postman https://chrome.google.com/webstore/detail/postman-rest-client/fdmmgilgnpjigdojojpjoooidkmcomcm

Categories: Framework v3
Categories: Plataforma JEE
Etiquetes: FAM
Etiquetes: rest

URL d'origen: https://comunitatdstsc.diba.cat/wiki/fam-crida-servei-rest

Enllaços:
[1] https://jersey.java.net/