

Architecture distribuée EIP

par Geoffrey Spaur

8 Mai 2017

Contents

1	Présentation	3
2	Problèmes Rencontrés	3
2.1	JSON	3
2.2	Multi instances	3

1 Présentation

Le projet a pour but de présenter les EIP afin d'intégrer les services web.

2 Problèmes Rencontrés

2.1 JSON

Le service Zoo Manager utilisé répondait sous format *JSON*. Or avec la configuration de base que nous avons, notre EIP ne comprenait que le *XML*. Nous avons donc du rajouter une dépendance à notre projet:

```
<!-- https://mvnrepository.com/artifact/org.apache.camel/camel-jackson -->
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-jackson</artifactId>
  <version>2.18.3</version>
</dependency>
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-xstream</artifactId>
  <version>2.9.2</version>
</dependency>
```

Pour pouvoir utiliser le *Unmarshalling*:

```
...
    .to("http://127.0.0.1:8080/animals/findByName/")
    .unmarshal().json(JsonLibrary.Jackson)
    .log("${body}")
...
```

2.2 Multi instances

Pour pouvoir instancier plusieurs serveurs, j'ai du changer les ports de ces dernier pour éviter les conflits:

```
System.getProperties().put("server.port", "8081");
```

Afin d'éviter l'agrégation suggéré dans le sujet, j'ai travaillé sur les exceptions. Ainsi, si un animal, n'était pas trouvé, notre *EIP* demander au service suivant jusqu'à le trouver.

```
.doTry()
  .to("http://127.0.0.1:8080/animals/findByName/")
  .log("${body}")
.doCatch(HttpOperationFailedException.class)
  .doTry()
    .to("http://127.0.0.1:8081/animals/findByName/")
    .log("${body}")
  .doCatch(HttpOperationFailedException.class)
    .log("Animal introuvable")
  .end()
.end()
```