Université de Rouen, UFR Sciences et Techniques 1ere année Master Génie Informatique du Logiciel



Rapport TP3 EIP

Realisé par

BENAOUICHA Mohand-Said et BABAGHAYOU Khaled

8 mai 2017



Sommaire

Objectifs du projet	ე
Apache Camel	3
Question 1	4
Question 2	4
Question 3	5
Question 4	6

Projet Apprentissage Réseau de neurones pour « jeu des bâtons »



Objectifs du projet

Le but de ce TP est de se familiariser ave les EIP (Enterprise Intégration Patterns) les plus communément rencontrés. Le system des EIP repose sur l'échange de messages (constitués d'un corps « body » et d'entêtes « headers ») entre des points d'accès (endpoints) « producteurs » et « consommateurs » à travers des « routes ».

Les EIP sont une tentative de formalisation pour les problèmes d'intégrations qui est historiquement apparue avec un livre de Gregor Hohpe et Bobby Woolf.

Apache Camel

Le cœur du système CAMEL est un constructeur de moteur de routage, son objectif principal est de comment, à partir d'une source et de plusieurs destinations, le routage des messages est possible, sa philosophie repose sur le fait qu'il n'a pas d'apriori sur les données transportées. En d'autres termes, ceci implique que la conversion vers un format intermédiaire quelconque pour transporter les données n'est pas nécessaire.

C'est aussi un Framework d'intégration, implémentant les EIP (Entreprise Integration Pattern) définis par Gregor Hohpe & Bobby Woolf. Les EIP sont des patterns d'intégration qui nous permettent de résoudre ces problématiques d'intégration avec des solutions éprouvées.

On utilise Camel car il offre les caractéristiques suivant :

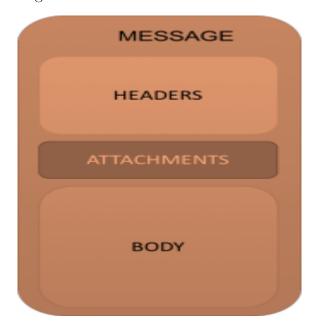
- Routage et médiation
- EIP (Entreprise Integration patterns)
- Librairies extensibles
- Architecture modulaire
- Conversion automatique de types

Il est Très léger, Testable, Facilement configurable, et la communauté qui l'utilise est très grande et active dans le web.





L'architecture d'un message Camel est définie ci-dessous



Question 1

On a utilisé la fonction

```
Scanner sc = new Scanner(System.in);
System.out.println("saisir le message :");
str = sc.nextLine();
if(str=="exit"){System.exit(0);}
```

Question 2

```
BeginWithW = "header";
if (str.startsWith("w")) {
          BeginWithW = "ecrire";
}
```



Question 3

On a implémenté tous les service basic effectuer dans les ancien TP dans notre serveur et tous les réponses seront reçues au format JSON

- Afficher tous les animaux
- Afficher un animal
- Ajouter un animal
- Supprimer un animal
- Changer un animal
- Afficher la position d'un l'animal en utilisat le service Geonames

3.A) La méthode pour rechercher un animal par son nom et affichez le résultat

3.B)

```
@RequestMapping(path = "/findAnimalPositionByName/{name}", method = GET,
produces = APPLICATION_JSON_VALUE)
public String getCagePosition(@PathVariable String name) throws
JAXBException, AnimalNotFoundException, CageNotFoundException {
    Cage cage = center.CageByName(center.findAnimalByName(name).getCage());
    try {
        String string = IOUtils.toString(new InputStreamReader(new
URL("http://api.geonames.org/findNearbyJSON?lat=" +
cage.getPosition().getLatitude() + "&Ing=" +
cage.getPosition().getLongitude() + "&username=mlgil").openStream()));
    return string;
} catch (MalformedURLException e) {
        throw new HTTPException(404);
} catch (IOException e) {
        throw new HTTPException(404);
}
```

Projet Apprentissage Réseau de neurones pour « jeu des bâtons »



Question 4

Au premier lieu on a eu du mal à instancier le service MyserviceController en même temps, donc on a dupliqué notre service en trois MyserviceController2 MyserviceController3, et on a lancé chaque service dans port séparé

- MyserviceController port 8084
- MyserviceController2 port 8085
- MyserviceController3 port 8086