

# Rapport de projet

## Architectures Distribuées

### Création d'un service RESTFUL

**Professeurs :** Mr Arnaud Saval

Master 1 GIL

Mr Clément Caron

Année Universitaire 2018/2019

**Auteur :** Lounis RAHMANI

Université de Rouen

## Table des matières

I-	Introduction .....	3
II-	Initiation aux services web .....	4
	II-1- Difficultés rencontrés et solutions apportées .....	4
III-	Projet ZOO_MANAGER .....	5
	III-1- Fonctionnalités fournies et attendues.....	6
	III-2- Difficultés rencontrées .....	6
	III-3- Fonctionnalités ajoutées en amélioration .....	7
	III-4- Autres améliorations apportées .....	8
IV-	Conclusion .....	8

## I- Introduction :

Ce projet d'architectures distribuées, RESTFUL, avait pour but de nous introduire aux notions de service web ou on a effectué dans un premier temps la récupération des informations fournies par un service web géographique ou on a utilisé les services Geonames. Ces TPs aussi, nous ont permis d'utiliser les web service Wolfram|Alpha, ainsi que GraphHopper.

Petite définition des web service utilisés :

→ **GeoNames** C'est une base de données géographiques qui couvre tous les pays et contient plus de onze millions de noms de lieux qui peuvent être téléchargés gratuitement.



→ **Wolfram|Alpha** C'est un service qui se comporte comme un moteur de recherche en répondant sous différents formats aux requêtes formulées. Calcule des réponses de niveau expert en utilisant la percée de Wolfram algorithmes, base de connaissances et technologie de l'IA.



→ **GraphHopper** C'est un service fournissant une planification d'itinéraire, une navigation et une optimisation des itinéraires en fonction du trafic



## II- Initiation aux services web :

### TP1 : GeoNames et Wolfram|Alpha

Lien JSFiddle : [http://jsfiddle.net/Rahmani\\_lounis/6j74zanL/](http://jsfiddle.net/Rahmani_lounis/6j74zanL/)

Dans cette partie du TP, le prof nous a communiqué un lien ou on a fait un « Fork » de l'exemple, cet exemple consistait à récupérer la ville (ou le bâtiment) le plus proche en fonction de coordonnées géographiques (en utilisant GeoNames) ainsi que d'afficher des informations complémentaires sur cette dernière (en utilisant Wolfram|Alpha).



Exemple d'un clic effectué près de l'INSA

### II-1- Difficultés rencontrés et solutions apportées :

Donc, j'ai utilisé l'interface de test JsFiddle avec des parties HTML, JavaScript et CSS. J'ai eu à ma disposition un code de base qui permettait d'afficher uniquement la ville ou le bâtiment en choisissant une position sur la carte. Il récupérait les informations du service GeoNames sous format XML.

L'exercice consistait à modifier le format de sortie en JSON pour l'appel au service GeoNames puis de traiter les données JSON obtenues afin de l'afficher en dessous de la carte.

- La première difficulté était d'ajouter un marqueur sur le bâtiment renvoyé par GeoNames. En effet, OpenLayers est une bibliothèque que nous n'avions jamais utilisée auparavant, il n'était pas toujours simple de se documenter tant cette bibliothèque est complexe et dont l'API peut changer sensiblement selon les versions.
- Au bout de quelques essais, j'ai rencontré un second problème : `{"status":{"message":"the hourly limit of 1000 credits for m1gil has been exceeded. Please throttle your requests or use the commercial service."},"value":19}}`.

Indiquant que le nombre limite par heure d'appel de services sur cet api a été atteint sur le compte GeoNames fournit par le professeur car étant utilisé par l'ensemble du groupe de TP. J'ai dû créer un compte sur GeoNames pour pallier à ce problème et modifier le login sur l'URL utilisé pour l'appel de service.

- Une autre difficulté était d'obtenir des informations sur le bâtiment depuis Wolfram|Alpha. Je me suis rendu compte qu'il était impossible d'effectuer une requête multiorigines (Cross-Origin Request).

En effet, on ne peut pas effectuer de requêtes dites « Cross Domain », ou plus simplement, il est impossible d'effectuer une requête AJAX depuis un domaine vers un autre domaine.

**Solution :** Afin de pallier à ce problème, j'ai opté pour effectuer la requête côté serveur (et non côté client avec une simple requête AJAX).

### III- Projet ZOO\_MANAGER

#### TP2 : Zoo Manager

Ce TP (dont le code source est disponible sur GitLab) consistait à développer un web service portant sur la gestion d'un zoo. Plusieurs fonctionnalités étaient attendues, dont certaines portaient sur l'utilisation de web services tierces tels que Wolfram|Alpha ou GraphHopper.



```
<?xml version='1.0'>
<ns2:center xmlns:ns2="m1GIL:rest:tp">
  <cages>
    <capacity>25</capacity>
    <name>usa</name>
    <position>
      <latitude>49.305</latitude>
      <longitude>1.2157357</longitude>
    </position>
    <residents>
      <cage>usa</cage>
      <id>b00770ea-c332-416c-85f0-747a8b1539fa</id>
      <name>Dalmatien 1</name>
      <species>Dalmatien</species>
    </residents>
    <residents>
      <cage>usa</cage>
      <id>b6a6bcaa-0031-429b-9aa0-8caa33f2ccd5</id>
      <name>Dalmatien 2</name>
      <species>Dalmatien</species>
    </residents>
    <residents>
      <cage>usa</cage>
      <id>68289889-d2ad-44e1-8339-1fe0b0e2100f</id>
      <name>Dalmatien 3</name>
      <species>Dalmatien</species>
    </residents>
    <residents>
      <cage>usa</cage>
      <id>d4a3bcbc-7d7e-4719-84ec-cd0ec63763cd</id>
      <name>Dalmatien 4</name>
      <species>Dalmatien</species>
    </residents>
  </cages>
</center>
```

### III-1- Fonctionnalités fournies et attendues :

#### Fonctionnalités fournies :

- *Retourne l'ensemble des animaux du centre*
- *Ajoute un animal dans votre centre*
- *Retourne l'animal identifié par {animal\_id}*

#### Fonctionnalités attendues :

- Modifie l'ensemble des animaux
- Supprime l'ensemble des animaux
- Crée l'animal identifié par {animal\_id}
- Modifie l'animal identifié par {animal\_id}
- Supprime l'animal identifié par {animal\_id}
- Recherche d'un animal par son nom
- Recherche d'un animal par position
- Recherche des animaux près d'une position
- Récupération des info. Wolfram d'un animal
- Récupération des info. Du trajet depuis une position GPS jusqu'à votre centre en utilisant le service G raphhopper.

### III-2- Difficultés rencontrées :

#### 1- Modification de l'ensemble des animaux

Cette fonctionnalité est ambiguë car il n'y a aucune précision donnée sur la modification, on ne connaît ni les informations à modifier pour chaque animal et ni par quoi les remplacer.

**Solution adaptée :** Notre choix a été de demander à JAXB de nous fournir un objet Unmarshall Animal puis de modifier toutes les informations hormis l'identifiant de chaque animal du centre par les informations de l'animal fourni.

#### 2- Recherche d'un animal par sa position

Cette fonctionnalité n'est pas assez claire car les animaux n'ont pas de position, on ne connaît que la position de leurs cages. Et une cage peut contenir plusieurs animaux.

**Solution adaptée :** Notre choix a été d'envoyer un animal au hasard dans la cage dont la position est indiquée.

### 3- Recherche des animaux près d'une position

La difficulté sur cette fonctionnalité était le calcul de la distance entre deux points à partir de leurs longitudes et de leurs latitudes.

**Solution adaptée :** On a pu trouver une formule de calcul en faisant des recherches puis on l'a appliqué en Java. Connaissant la distance si elle est inférieure à 100km, on considère que les deux points sont proches.

### III-3- Fonctionnalités ajoutées en amélioration :

MÉTHODE	URL	BODY	DESCRIPTION
POST	/cages	Cage	Ajoute une cage dans le centre
PUT	/cages	Cage	Modifie toutes les cages du centre
DELETE	/cages		Supprime toutes les cages du centre
GET	/cages/{cage_id}		Retourne la cage dont l'identifiant est {cage_id}
PUT	/cages/{cage_id}	Cage	Modifie la cage dont l'identifiant est {cage_id}
DELETE	/cages/{cage_id}		Supprime la cage dont l'identifiant est {cage_id}
DELETE	/cages/clear/{cage_id}		Supprime tous les animaux de la cage dont l'identifiant de cage {cage_id}
GET	/find/allAt/{position}		Retourne tous les animaux à la position indiquée

- Certaines de ces fonctionnalités sont ajoutées pour pallier aux ambiguïtés rencontrées sur quelques fonctionnalités demandées (plusieurs choix d'implémentation).
- D'autres sont ajoutées car elles sont demandées implicitement dans le scénario de test fourni dans l'énoncé du projet. Par exemple : « Ajoutez un animal ... à Rouen dont la position est ... ». Cette cage n'existant pas au départ, on est obligé de l'ajouter avant de créer l'animal.
- Et il y'en d'autres qui sont ajoutées dans le seul d'améliorer le projet.

### III-4- Autres améliorations apportées :

#### 1- Implémentation des messages d'erreurs

Au lieu de simplement lever une HTTPException pour chaque problème rencontré, nous avons décidé de gérer toutes les erreurs relatives aux opérations du model de manière transparente pour l'utilisateur.

**Solution adaptée :** Pour cela, j'ai créé une classe RestException annotée de @XmlRootElement, permettant à JAX-WS de le traiter comme un élément à part entière et de le sérialiser en XML. Cela permet donc d'avertir l'utilisateur de façon concise sur la raison de l'échec de sa requête

#### 2- Identifiant en UUID

Les identifiants devant être transformés en UUID, pour prévenir les exceptions non gérés préalablement lors des conversions, on vérifie que la chaîne de caractère fournie respecte le format d'un UUID dont le pattern est :

```
^[0-9a-fA-F]{8}\\-[0-9a-fA-F]{4}\\-[0-9a-fA-F]{4}\\-[0-9a-fA-F]{4}\\-[0-9a-fA-F]{12}$
```

#### 3- Position (Long latitude, Long longitude)

Les positions devant être aussi transformées en objet de type Position, pour prévenir une quelconque erreur dans le découpage et des exceptions lors de la conversion des latitudes et longitudes en Long, on vérifie que la chaîne de caractère fournie respecte le format d'une position dont le pattern est :

```
^lat=\\d*\\.\\d*&lng=\\d*\\.\\d*$
```

### IV- Conclusion :

Ce TP fut très instructif, le développement d'un service web était une première pour nous. Nous avons pris aussi beaucoup de plaisir à développer ce service en Java, un langage que nous apprécions tout particulièrement et que nous pratiquons beaucoup moins cette année.