



Compte rendu TP1 & 2

TP REST

Projet réalisé par

- ❖ Zebouchi Mohammed TP 4
 - ❖ Belkhous Redha Nabil TP 3
- 

Sommaire

1. Partie 1

2. Partie 2

3. Partie 3

1

Partie 1

Partie 1

Adresse du jsfiddle :

 <http://jsfiddle.net/cwwwgoek/3/>

- ❖ Remplacer le web service appelé par sa version JSON

Pour faire l'appel avec l'utilisation du JSON, nous avons utilisé le code ci-dessous :

```
$.get("http://api.geonames.org/findNearbyJSON?lat="+lat+"&lng="+lng+"&username=mlgil",function(data){  
  
    $("#info").text(JSON.stringify(data));  
});
```

- ❖ Ajout du web service wolfram|alpha

Pour communiquer avec l'API wolfram|alpha nous avons créé un compte qui nous a permis d'ajouter une application dans leur service web avec un APPID.

Nous avons utilisé l'APPID dans le lien pour communiquer en ajoutant les informations de la ville recherchée.

Lors de l'utilisation de cette API une exception est survenue car la plateforme de wolfram|alpha ne permet de « cross domaine » (ce problème est détecté dans le cas où on veut faire une requête sur un autre domaine qui n'est pas celui qu'on utilise)

```
$.get("http://api.wolframalpha.com/v2/query?appid=P6LU2H-AJ42KYYX9X&input="+data.geonames[0].toponymName,function(data1){  
  
    var xml = new XMLSerializer().serializeToString(data1);  
    $("#info").text(JSON.stringify(data)+"<br/> Call API <br/>" +xml);  
});
```

2

Partie 2

Partie 2

```
public Source invoke(Source source) {  
  
    String replyElement = new String(SendMessage);  
    StreamSource replay = new StreamSource(new StringReader(replyElement));  
    return replay;  
}  
  
public static void main(String[] args) {  
    Endpoint e = Endpoint.create(HTTPBinding.HTTP_BINDING, new  
    REST("<p>Université de Rouen</p>"));  
    e.publish("http://127.0.0.1:8084/hello/world");  
}
```

Ce programme permet de lancer un service web qui va nous renvoyer ce qui a été donné à l'instanciation de l'objet REST.

Il s'agit de faire écouter un port précis en déployant un service dans ce dernier.

Partie 3

Au court de cette partie nous avons mis en place un service CRUD complet.

Nous avons implémenté :

- Une fonction GET qui permet de récupérer la ou les villes à partir d'un nom.
- Une fonction PUT qui permet d'ajouter une ville.
- Une fonction DELETE qui permet de supprimer une ville.
- Une fonction POST qui permet de récupérer une ville précise ou les villes qui sont proche à partir d'une position.
- Une gestion des exceptions.

Nous avons mis à votre disposition la Javadoc pour comprendre notre code source.