

Compte rendu d'architectures distribuées TP1 et 2

Auteurs : Chaymae Ez-zahri, Merouan Oufquir

I- Introduction:

Code:

```
// Création de la carte
var map = new ol.Map({
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    })
  ],
  target: 'map',
  view: new ol.View({
    projection: 'EPSG:4326',
    center: [1.0687, 49.387],
    zoom: 18
  })
});

// ajouter du gestionnaire de clic
map.on("click", function(e) {
  // on récupère la position
  var lonlat = e.coordinate;
  var lat = lonlat[1].toFixed(4);
  var lng = lonlat[0].toFixed(4);
  // on l'affiche
  $("#mouse-position-click").html("You clicked near " + lonlat[1].toFixed(4) + " N, " +
lonlat[0].toFixed(4) + " E");
  // on requête le web service geoname pour retrouver la ville/ bâtiment qui correspond aux
coordonnées géographiques

/*$.get("http://api.geonames.org/findNearby?lat="+lat+"&lng="+lng+"&username=m1gil",function
(data){
  // on marshall / sérialise le XML information en chaînes de caractères
  // var xml = new XMLSerializer().serializeToString(data);
  var xml = new XMLSerializer().serializeToString(data);
  // on l'affiche
  $("#info").text(xml);
}); */

// TODO: appelez la version JSON de ce même web service

/*$.get("http://api.geonames.org/findNearbyJSON?lat="+lat+"&lng="+lng+"&username=m1gil",fu
nction(data){
  var json = JSON.stringify(data);
  $("#info").text(json);
}); */
```

```
});*/
```

```
$.get("https://api.wolframalpha.com/v2/query?input=rouen&format=plaintext&output=JSON&appid=m1gil",function(data){  
    var json = JSON.stringify(data);  
    $("#info").text(json);  
});  
  
});
```

a)

b)

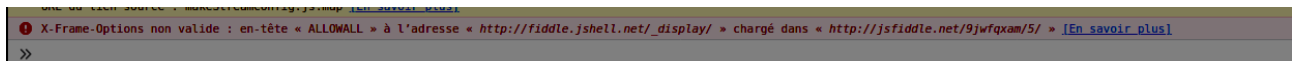
```
$.get("http://api.geonames.org/findNearbyJSON?lat="+lat+"&lng="+lng+"&username=m1gil",function(data){  
    var json = JSON.stringify(data);  
    $("#info").text(json);  
});
```

c)

```
$.get("https://api.wolframalpha.com/v2/query?input=rouen&format=plaintext&output=JSON&appid=m1gil",function(data){  
    var json = JSON.stringify(data);  
    $("#info").text(json);  
});
```

Cette requête affiche une erreur:

X-Frame-Options non valide : en-tête « ALLOWALL » à l'adresse «
http://fiddle.jshell.net/_display/ » chargé dans « http://jsfiddle.net/9jwfqam/5/ »



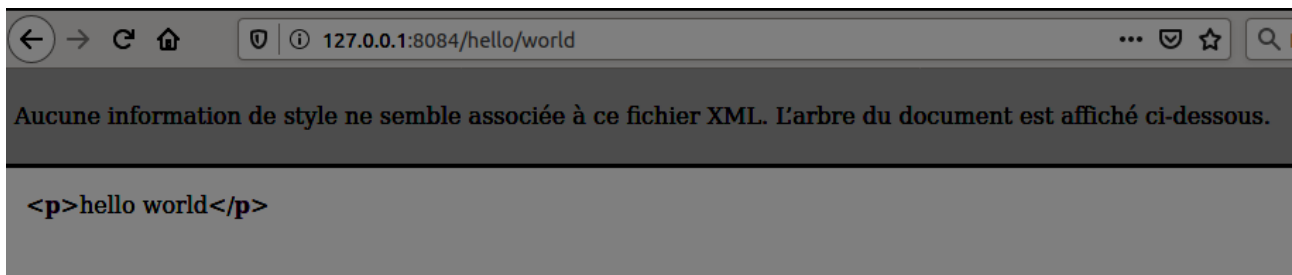
L'en-tête de cette réponse X-Frame-Options peut être utilisé afin d'indiquer si un navigateur devrait être autorisé à afficher une page au sein d'un élément <frame>, <iframe>, <embed> ou <object>.

Les sites peuvent utiliser cet en-tête afin d'éviter les attaques de clickjacking pour s'assurer que leur contenu ne soit pas embarqués dans d'autres sites.

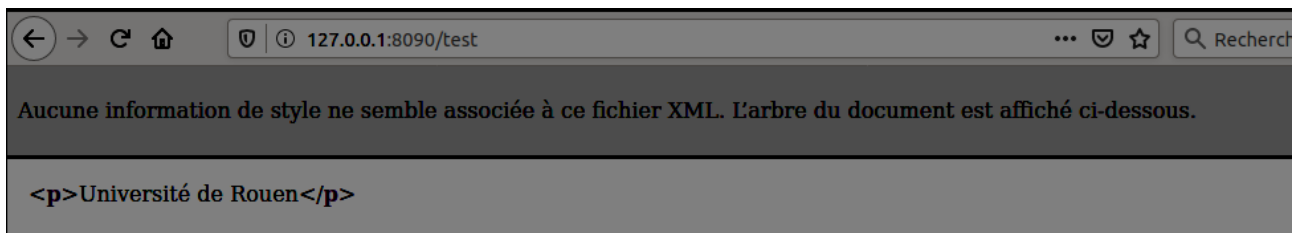
Il semble qu'ALLOWALL ait récemment été ajouté à autorisez tous les sites à utiliser le code en tant que code source.

2) Création d'un service :

Résultat obtenu :



a) Afficher « Université de Rouen » sur l'adresse : <http://127.0.0.1:8090/test>



b) Afficher « Réponse du service REST » à l'adresse : <http://127.0.0.1:8084/hello/world>

Erreur l'adresse est déjà utilisée

3) Création d'un service RESTFUL

Pour voir la partie de code complète, veuillez s'il vous plait voir le projet dans le gitlab.
J'ai essayé de faire un commentaire mentionnant le type et la fonction de chaque requête avant son code.

1-Pour la requête : DELETE /animals Supprime l'ensemble des animaux :

Partie code

```

294         .stream()
295         .filter(cage -> cage.getName().equals(animal.getCage()))
296         .findFirst()
297         .orElseThrow(() -> new HTTPException(404))
298         .getResidents()
299         .add(animal);
300     return new JAXBSource(this.jc, this.center);
301 } else if ("DELETE".equals(method)) {
302     // DELETE /animals Supprime l'ensemble des animaux
303     this.center.getCages().forEach(cage -> {
304         cage.setResidents(null);
305     });
306     return new JAXBSource(this.jc, this.center);
307 } else if ("PUT".equals(method)) {
308     //PUT /animals : Modifie l'ensemble des animaux
309
310     Animal animal = unmarshalAnimal(source);
311     //pour modifier l'ensemble des animaux on enleve tous les elemet de
312     // la collection des animaux
313     //puis on r  ajoute tous
314     this.center.getCages()
315         .forEach(cage -> {
316             cage.getResidents()
317                 .forEach(a -> {
318                     a.setName(animal.getName());

```

Test dans postman :

Avant la requete delete ;

GET http://127.0.0.1:8084/animals

Status: 200 OK Time: 223ms Size: 1.31 KB

```

1 <?xml version="1.0" ?>
2 <ns2:center xmlns:ns2="m1GIL:rest:tp">
3   <cages>
4     <capacity>25</capacity>
5     <name>usa</name>
6     <position>
7       <latitude>49.305</latitude>
8       <longitude>1.2157357</longitude>
9     </position>
10    <residents>
11      <cage>usa</cage>
12      <id>4b764da6-cbb1-4304-ab74-8185f8d0b2a8</id>
13      <name>Tic</name>

```

Après l'exécution de la requête delete :

DELETE http://127.0.0.1:8084/animals Send Save

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Code

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (3) Test Results Status: 200 OK Time: 15ms Size: 560 B Save Response

Pretty Raw Preview Visualize XML

```

1 <?xml version="1.0" ?>
2 <ns2:center xmlns:ns2="m1GIL:rest:tp">
3   <cages>
4     <capacity>25</capacity>
5     <name>usa</name>
6     <position>
7       <latitude>49.305</latitude>
8       <longitude>1.2157357</longitude>
9     </position>
10  </cages>
11  <cages>
12    <capacity>15</capacity>
13    <name>amazon</name>

```

Bootcamp Build Browse

2-Pour la requete : PUT /animals ?????? Modifie l'ensemble des animaux

Partie code :

```

} else if ("PUT".equals(method)) {
    ///PUT /animals : Modifie l'ensemble des animaux

    Animal animal = unmarshalAnimal(source);
    //pour modifier l'ensemble des animaux on enleve tous les elemet de
    // la collection des animaux
    //puis on rÃajoute tous
    this.center.getCages()
        .forEach(cage -> {
            cage.getResidents()
                .forEach(a -> {
                    a.setName(animal.getName());
                    a.setSpecies(animal.getSpecies());
                });
        });

    return new JAXBSource(this.jc, this.center);
}

```

3-Pour la requete : POST /animals/{animal id} Animal Crée l'animal identifié par {animal id}

Partie code :

```

else if ("POST".equals(method)) {
    //POST /animals/{animal_id} Animal Cr e lâ€™animal identi   par {animal_id}
    Animal animal = unmarshalAnimal(source);
    animal.setId(UUID.fromString(animal_id));

    this.center.getCages()
        .stream()
        .filter(cage -> cage.getName().equals(animal.getCage()))
        .findFirst()
        .orElseThrow(() -> new HTTPException(404))
        .getResidents()
        .add(animal);

    return new JAXBSource(this.jc, this.center);
}

```

4-Pour la requete : PUT /animals/{animal_id} Animal Modifie l'animal identifi   par {animal_id}

Partie code :

```

8     else if ("PUT".equals(method)) {
9         //PUT /animals/{animal_id} Animal Modifie lâ€™animal identi   par {animal_id}
10        try {
11            Animal origine = center.findAnimalById(UUID.fromString(animal_id));
12            Animal tempAnimal = unmarshalAnimal(source);
13            origine.setSpecies(tempAnimal.getSpecies());
14            origine.setCage(tempAnimal.getCage());
15            origine.setName(origine.getName());
16
17            this.center.getCages()
18                .stream()
19                .filter(cage -> cage.getName().equals(origine.getCage()))
20                .findFirst()
21                .orElseThrow(() -> new HTTPException(404))
22                .getResidents()
23                .add(origine);
24
25            return new JAXBSource(this.jc, origine);
26        } catch (AnimalNotFoundException e) {
27            throw new HTTPException(404);
28        }
29    }

```

5- Pour la requete : DELETE /animals/{animal_id} Supprime l'animal identifi   par {animal_id}

Partie code :

```

    }
    else if ("DELETE".equals(method)) {
        //DELETE /animals/{animal_id} Supprime l'animal identifié par {animal_id}
        try {
            Animal origine = center.findAnimalById(UUID.fromString(animal_id));

            this.center.getCages()
                .stream()
                .filter(cage -> cage.getName().equals(origine.getCage()))
                .findFirst()
                .orElseThrow(() -> new HTTPException(404))
                .getResidents()
                .remove(origine);

            return new JAXBSource(this.jc, center);
        } catch (AnimalNotFoundException e) {
            throw new HTTPException(404);
        }
    }
}

```

6-Pour la requete : GET /find/byName/{name} Recherche d'un animal par son nom

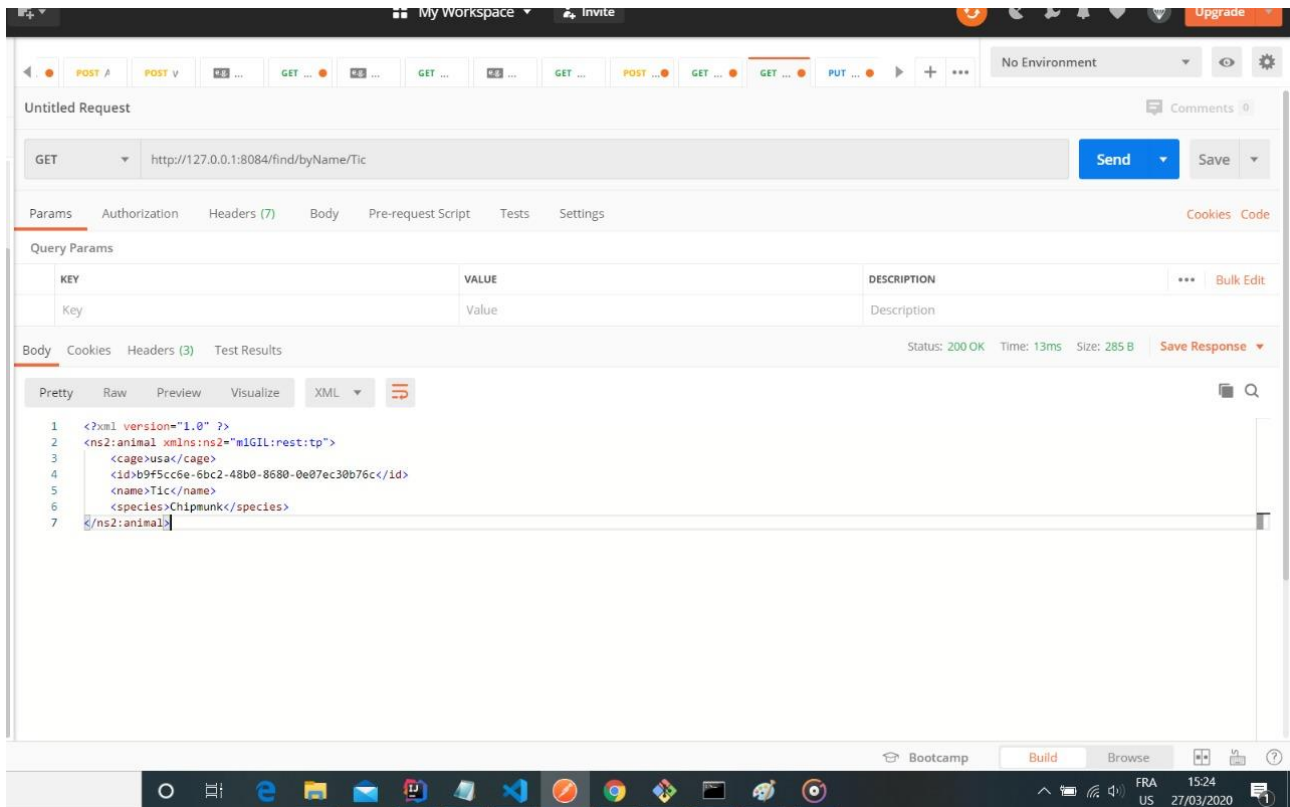
Partie code :

```

10 //GET /find/byName/{name} Recherche d'un animal par son nom
11 private Source findAnimalByName(String method, Source source, String path_part) throws JAXBException {
12     if(method.equals("GET")){
13         Animal animal = this.center.getCages()
14             .stream()
15             .map(r->r.getResidents())
16             .flatMap(r->r.stream())
17             .filter(r->r.getName().equals(path_part))
18             .findFirst()
19             .orElseThrow(() -> new HTTPException(404));
20
21         return new JAXBSource(this.jc, animal);
22     }
23     else {
24         throw new JAXBException("method not supported");
25     }
26 }

```

Tester par postman le fonctionnement de la requête :



7-Pour la requete : GET /find/at/{position} Recherche d'un animal par position

Partie code :

```

157
158 // GET /find/at/{position} Recherche d'un animal par position
159 private Source findAnimalByPosition(String method, Source source, String query) throws JAXBException {
160     if(method.equals("GET")){
161         Map<String,Double> queryParam = this.getValueQuery(query);
162         Position temp = new Position(queryParam.get("lat"),queryParam.get("lng"));
163
164         List<Animal> animals = this.center.getCages()
165             .stream()
166             .filter(cage -> cage.getPosition().equals(temp))
167             .map(cage -> cage.getResidents())
168             .flatMap(r->r.stream())
169             .collect(Collectors.toList());
170
171         return new JAXBSource(this.jc, new ListAnimals(animals));
172     }
173     else {
174         throw new JAXBException("method not supported");
175     }
176 }
177

```

Tester par postman :

GET http://127.0.0.1:8084/find/at?lat=49.305&lng=1.2157357

Params Authorization Headers (7) Body Pre-request Script Tests Settings

Query Params

KEY	VALUE
<input checked="" type="checkbox"/> lat	49.305
<input checked="" type="checkbox"/> lng	1.2157357
Key	Value

Body Cookies Headers (3) Test Results

Pretty Raw Preview Visualize XML

```

1 <?xml version="1.0" ?>
2 <ns2:listAnimals xmlns:ns2="m1GIL:rest:tp">
3   <list>
4     <cage>usa</cage>
5     <id>ddb707bd-90fe-4ede-8737-48a158c60da7</id>
6     <name>Tic</name>
7     <species>Chipmunk</species>
8   </list>
9   <list>
10    <cage>usa</cage>
11    <id>a9d5f3df-5668-4cd0-8e06-e7176efb3ea6</id>
12    <name>Tac</name>
13    <species>Chipmunk</species>
14  </list>
15 </ns2:listAnimals>

```

8-Pour la requete : GET /find/near/{position} Recherche des animaux près d'une position

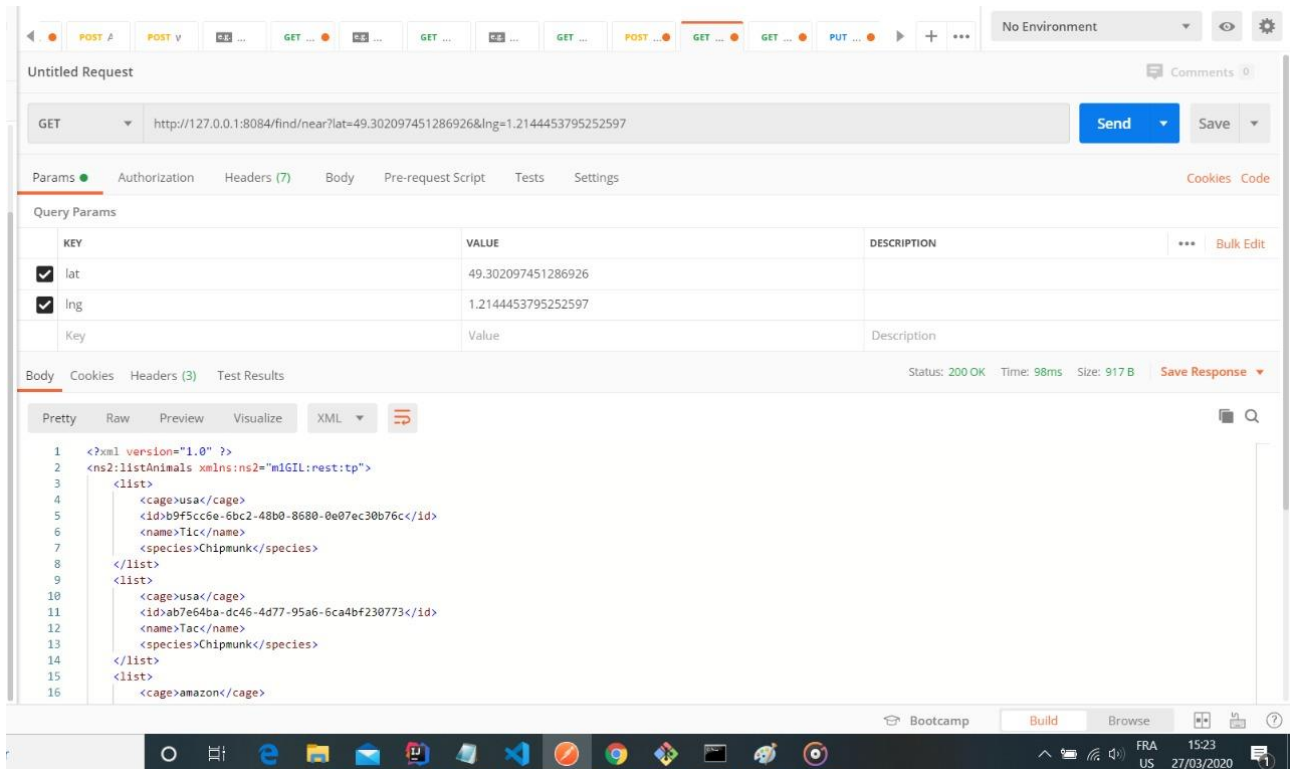
Partie code :

```

35
36 //GET /find/near/{position} Recherche des animaux près d'une position
37 private Source findNearest(String method, Source source, String query) throws JAXBException {
38     if(method.equals("GET")){
39         Map<String,Double> queryParam = this.getValueQuery(query);
40         Position temp = new Position(queryParam.get("lat"),queryParam.get("lng"));
41
42
43         List<Animal> animals = this.center.getCages()
44             .stream()
45             .filter(cage -> distance(cage.getPosition().getLatitude(),cage.getPosition().getLongitude())
46             .map(cage -> cage.getResidents())
47             .flatMap(r->r.stream())
48             .collect(Collectors.toList());
49
50
51         return new JAXBSource(this.jc, new ListAnimals(animals));
52     }
53     else {
54         throw new JAXBException("method not supported");
55     }
56 }
57
58 // GET /find/at/{position} Recherche d'un animal par position

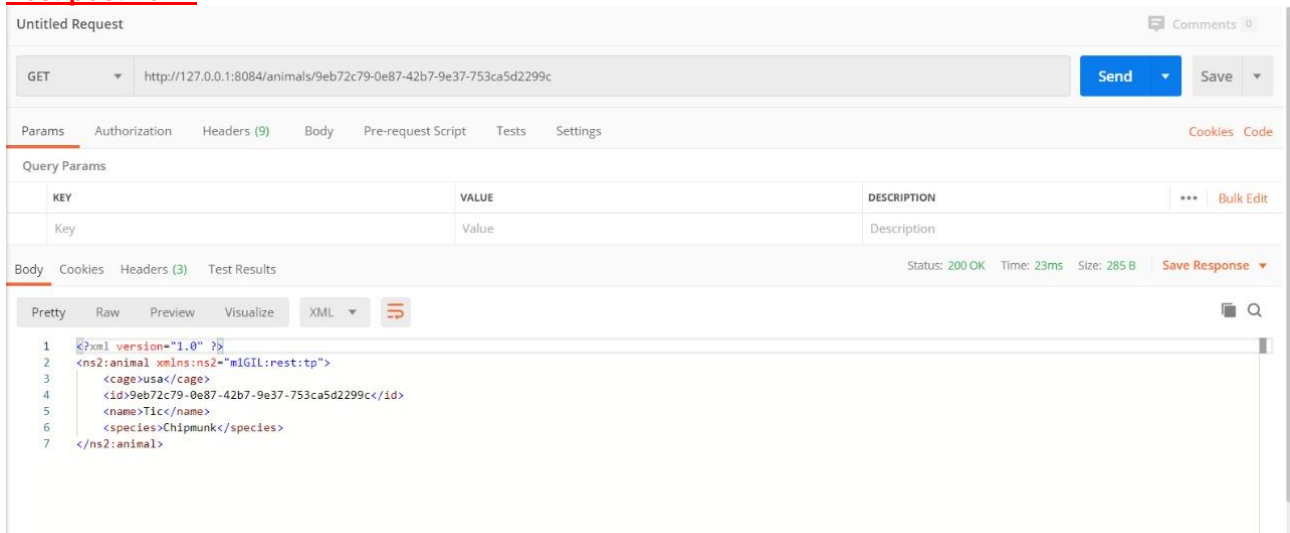
```

Test par postman :



9-Pour la requete : GET /animals/{animal id}/ wolf Récupération des info. Wolfram d'un animal

Test postman :



10-Pour la requete : GET /center/journey/from/{position} Récupération des info. Du trajet depuis une position GPS jusqu'à votre centre en utilisant le service G raphhoppe r.

Partie code :

Test postman :