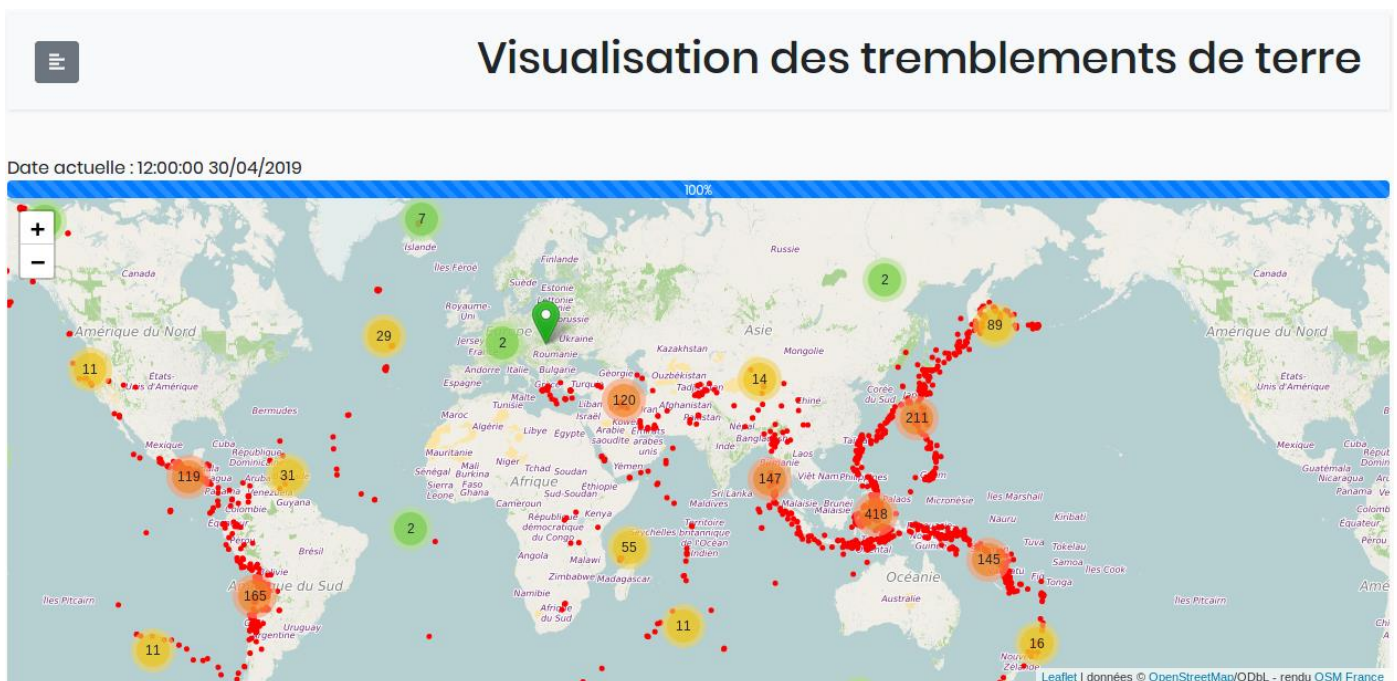


Rapport du mini projet

Web sémantique

Développer une application Mashup de visualisation des tremblements de terre



Réalisé par :

- ☐ Théo Delacoux
- ☐ Ludovik Tekam

Etudiants ingénieurs en 2ème année spécialité informatique
Université Paris 13 - Sup Galilée

2018-2019

TABLE DES MATIÈRE

Introduction.....	4
1. Note Importante	5
2. Prérequis	5
3. Mise en place	5
4. Fonctionnalités	6
a. Fonctionnalité 1 : Emplacement des tremblements.....	6
b. Fonctionnalité 2 : Les marqueurs et la magnitude du séisme.....	8
c. Fonctionnalité 4 : IHM personnalisable.....	10
d. Fonctionnalité 5 : API youtube et Google Freebase.....	15
e. Fonctionnalité Supplémentaire : Déplacement sur la Map.....	24
Conclusion.....	30
Source de documentation fournie.....	31
Source de documentation utile	31

Introduction

Suite au cours de web sémantique qui nous a été dispensé ce semestre, il nous a été demandé de créer une application **Mashup** dont le contenu provient de plusieurs sources d'information dont l'API Google Maps et l'API Youtube et ce en utilisant les bases de connaissances Dbpedia ou Freebase. L'objectif étant de visualiser des tremblements de terre qui ont eu lieu dans un certain laps de temps et avec une certaine magnitude.

1. Note Importante

Au cours de la phase de développement, nous avons dû utiliser de nouveaux outils. le tableaux ci-dessous est un résumé les différents changement effectués.

outils initialement indiqué dans le cahier de charge	outils finalement utilisé lors du développement
Google Maps API	Open Street Maps API (librairie leaflet JS)
FreeBase API	DBpedia + SPARQL
Youtube API	

2. Prérequis

La connaissance et la maîtrise des langages suivant est recommandée pour bien comprendre les termes et le code de développement présentés dans certaine partie de ce rapport.

- JavaScript
- SPARQL
- HTML
- CSS

3. Mise en place

Dans cette partie, on vous explique le minimum à faire pour commencer le développement, c'est à dire comment faire pour insérer une Map opérationnel.

Pour ce faire, il vous suffit de suivre le tutoriel suivant :

<https://nouvelle-techno.fr/actualites/2018/05/11/pas-a-pas-inserer-une-carte-openstreetmap-sur-votre-site>

4. Fonctionnalités

Dans cette partie, nous vous présenteront les fonctionnalités que nous avons implémentés ainsi que les méthodes et les raisonnements utilisés pour les faire.

Remarque: Les séismes étant en nombre quasi infinie, nous avons décidé de fixer le nombre maximal de séisme à afficher à **2000**. Cette limite peut bien sûr être modifier ou supprimer, pour cela, il suffira de modifier la fonction **createQuery(...)** (voire fonctionnalité 1) et de supprimer ou modifier la valeur du champ limit (**&limit=2000**)

a. Fonctionnalité 1 : Emplacement des tremblements

L'emplacement des tremblements de terre est visualisé sur une carte Google Maps.

Nous avons décidé d'utiliser **openstreetmap** à la place de Google Maps. Ce choix est motivé par le fait que nous sommes contre l'idée de devoir mettre notre code de carte bancaire dans un projet étudiant.

Pour ce faire nous avons procédé en deux étapes :

étape 1 :

On récupérer les emplacements des tremblements de terre via une Base de connaissance.

Dans notre cas, vu que **FreeBase API** n'est plus disponible, nous avons décidé de récupérer les données directement à la source via une **url** au format **geojson** qui est une manière de représenter des données contenant des coordonnées géographiques au format json. Pour cela, nous nous sommes servis de la méthode **Ajax** de **JS**

Code implémenté :

- Cette fonction nous permet de créer une **url** en définissant les séismes qui seront récupérés

```
function createQuery(dateDebut,dateFin,minMag,maxMag){
    var q='https://earthquake.usgs.gov/fdsnws/event/1/';
    q
    += 'query?format=geojson&starttime='+dateDebut.toISOString()+'&endtime='+dateFin.toISOString()+'&min
    magnitude='+minMag+'&maxmagnitude='+maxMag+'&limit=2000';
    return q;
}
```

- Cette fonction nous permet de récupérer les séismes au format json, on ajoute chaque séisme récupéré dans un tableau de séisme qui sera affiché à l'aide de la fonction **afficherSeisme(.....)**

```
function recupererSeisme(dateDebut, dateFin, minMag, maxMag, deltaTime)
{
    var queryUrl = createQuery(dateDebut, dateFin, minMag, maxMag);
    $.ajax({
        dataType: "jsonp",
        url: queryUrl,
        cache: true,
        success: function( _data )
        {
            var results = _data.features;
            var taille = results.length;
            var tableSeisme = [];
            for(var i=0 ; i<taille ; i++)
            {
                var seisme = {
                    lat: results[i].geometry.coordinates[1],
                    long: results[i].geometry.coordinates[0],
                    place: results[i].properties.place,
                    mag: results[i].properties.mag,
                    time: results[i].properties.time,
                    date: new Date(results[i].properties.time)
                };
                tableSeisme.push(seisme);
            }
            afficherSeisme(dateDebut, dateFin, deltaTime, 0, tableSeisme)
        }
    });
}
```

étape 2 :

On affiche les tremblements de terre récupérés

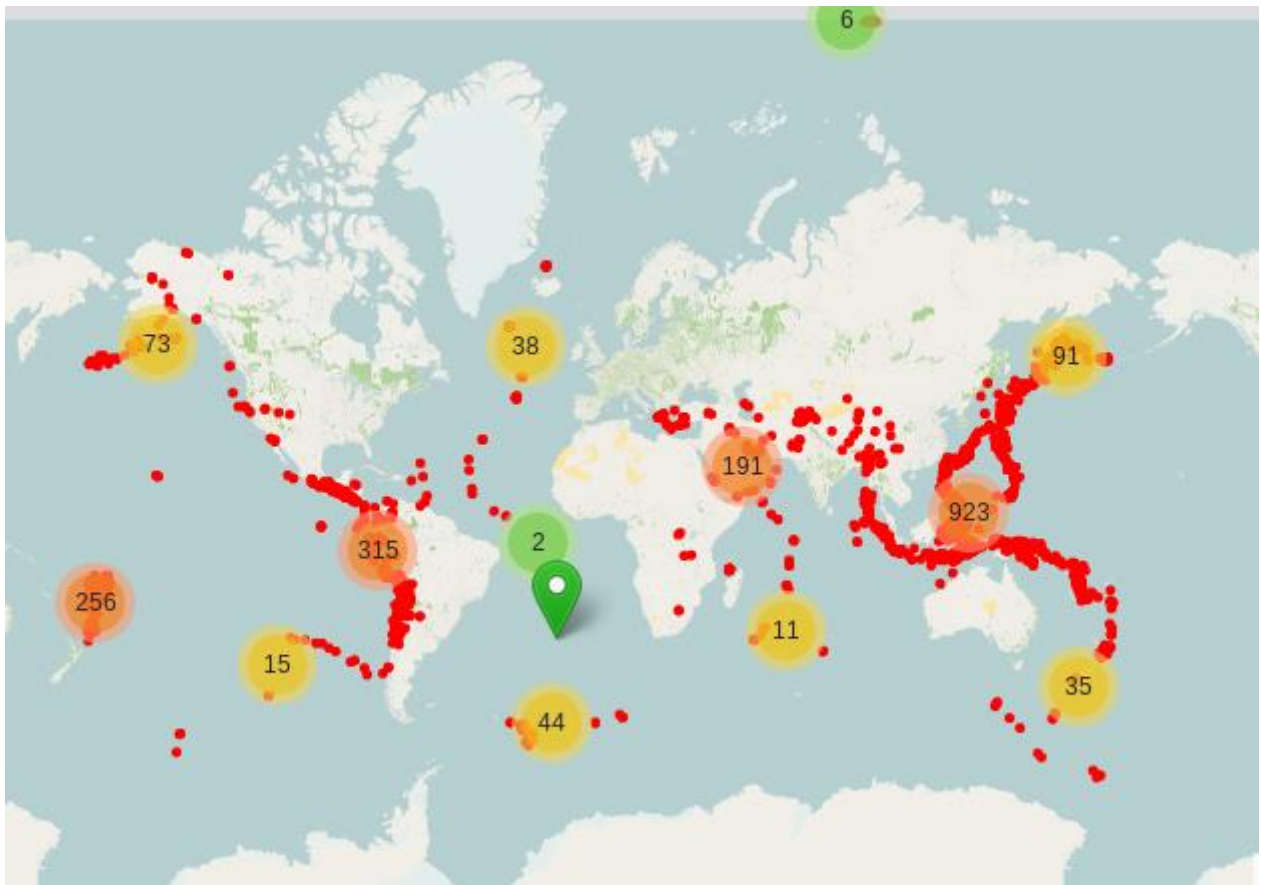
Code implémenté :

- Ici il suffit simplement de parcourir le tableau de séismes récupérés précédemment et de les afficher (voir fonction plus détaillée dans le code : fonction **afficherSeisme(..)**

```
while(tableSeisme.length > 0)
{
    seisme = tableSeisme.pop();
    if(seisme.time < currentTime)
    {
        addCircle(seisme.lat, seisme.long, seisme.mag, seisme.date, seisme.place,);
    }
}
```

```
}  
else  
{  
    //on ajoute le seisme dans le nouveau tableau  
    newTableSeisme.push(seisme);  
}  
}
```

Visualisation sur l'application :



b. Fonctionnalité 2 : Les marqueurs et la magnitude du séisme

Les marqueurs sont des cercles dont la taille dépend de la magnitude du séisme.

Pour implémenter cette fonctionnalité, nous avons utilisé une des méthodes de la librairie leaflet et avons simplement rajouté les bons paramètres se basant sur les données récupérer via la base de connaissance des tremblements de terre.

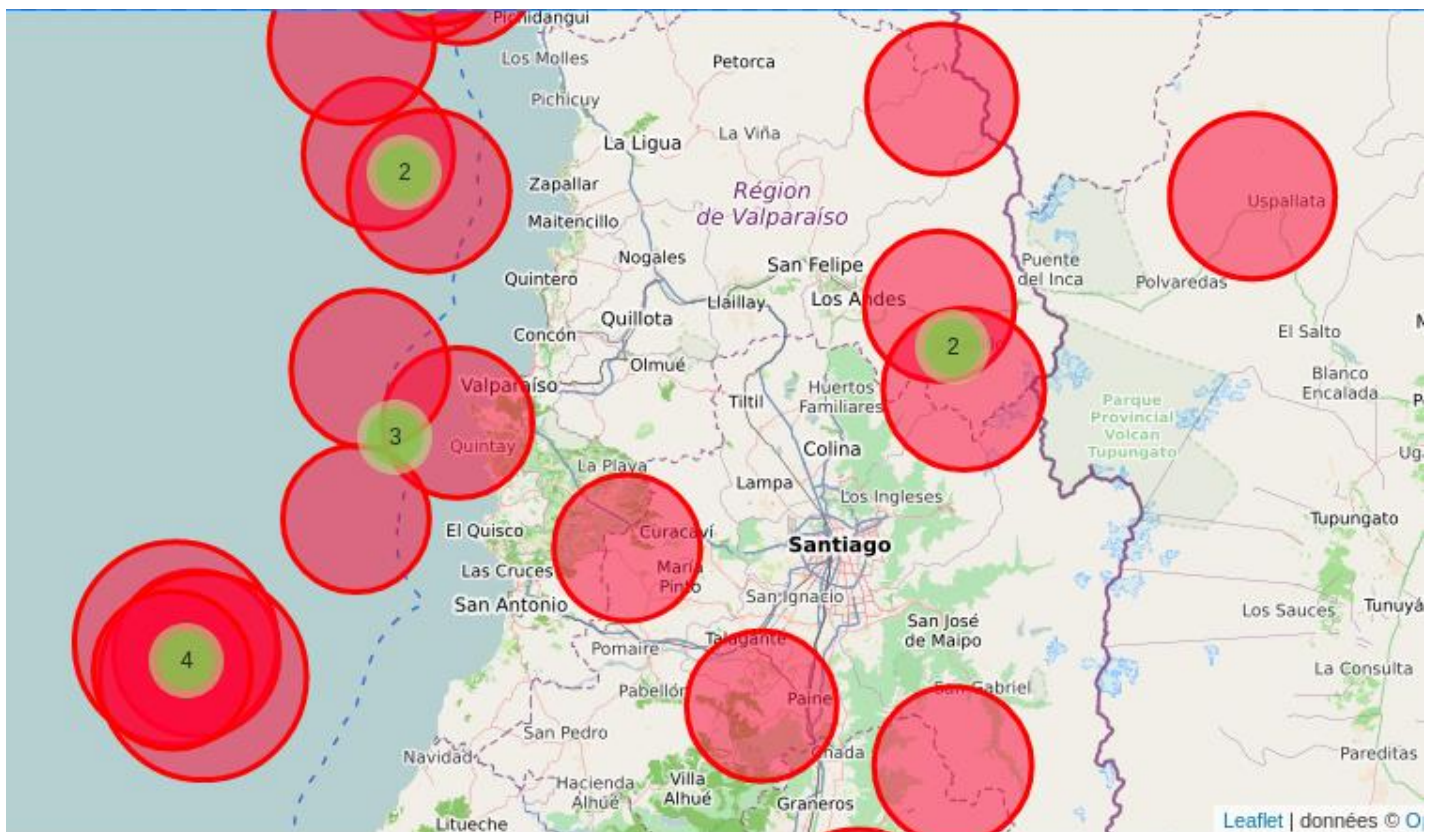
Code implémenté :

Remarque: nous avons volontairement multiplier la magnitude des séismes par 5000 afin de pouvoir distinguer la différence entre des séismes, car les valeurs d'origine sont assez petites et ne nous permettent pas de faire la différence. Bien sûr cela peut être modifié à votre guise, il suffit simplement de modifier le paramètre **radius** de la méthode **circle**.

```

var mycircle= L.circle([lat,long],{
  radius:5000*mag,
  color:"red",
  fillColor:"#f03",
  fillOpacity:0.5
});
mycircle.addTo(macarte);
  
```

Visualisation sur l'application :



c. Fonctionnalité 4 : IHM personnalisable

L'utilisateur peut choisir une fourchette de magnitude, et une fourchette de date (difficile, bonus) comme si il contrôlait la lecture d'un film, l'utilisateur peut cliquer sur un bouton play, et voir les marqueurs apparaître au fur et à mesure. L'utilisateur peut contrôler la vitesse de lecture, et doit pouvoir visualiser la date et l'heure au dessus de la carte.

Ici, le raisonnement est assez intuitif, toutefois pour bien implémenter les fonctionnalités demandées, nous avons procédé en trois étapes :

Étapes 1:

Création d'une interface ergonomique pour faciliter l'utilisation. Nous avons créé un formulaire dans un panneau déroulant pour gérer toutes les options de la simulation.

Nous sommes allés un peu plus loin en ajoutant un **side panel** qui globalement permet à l'utilisateur de masquer le formulaire afin d'avoir une meilleure visibilité de la map.

Code implémenté :

Code HTML

```
<p><br /><br />Option de la simulation</p>
  <li class="paddingLR">
    <form id="formOptionSeisme">
      <div class="form-group">
        <label for="DateDebutSeisme" class="col-form-label">Début de la simulation</label>
        <input id="formDateDebutSeisme" class="form-control" type="datetime-local" name="debut"
value="2019-02-14T12:00:00">
      </div>
      <div class="form-group">
        <label for="DateFinSeisme" class="col-form-label">Fin de la simulation</label>
        <input id="formDateFinSeisme" class="form-control" type="datetime-local" name="fin"
value="2019-04-30T12:00:00">
      </div>
      <div class="form-group">
        <label for="MagnitudeMin">Magnitude Minimal</label>
        <input id="formMagnitudeMin" type="number" step="any" class="form-control"
name="magnitudeMin" min="0" value="4">
      </div>
      <div class="form-group">
        <label for="MagnitudeMax">Magnitude Maximal</label>
        <input id="formMagnitudeMax" type="number" step="any" class="form-control"
name="magnitudeMax" value="10">
      </div>
    </form>
  </li>
```

```

</div>
<div class="form-group">
  <label for="VitesseSimulation" class="col-form-label">Vitesse de la simulation (temps passé
par sec)</label>
  <div class="meme-ligne">
    <div class="col-xs-3">
      <input id="formValeurVitesse" type="number" min="1" class="form-control" name="vitesse"
value="2" placeholder="1">
    </div>
    <div class="col-xs-5">
      <select id="formUniteVitesse" name="unite" class="custom-select">
        <option value="minutes">Minutes</option>
        <option value="heures">Heures</option>
        <option value="jours" selected>Jours</option>
        <option value="mois">Mois</option>
        <option value="annees">Années</option>
      </select>
    </div>
  </div>
</div>
<button type="submit" class="btn btn-secondary">Lancer la simulation</button>

```

Code JS pour le side panel

```

$(document).ready(function () {
  $("#sidebar").mCustomScrollbar({
    theme: "minimal"
  });

  $('#dismiss, .overlay').on('click', function () {
    // hide sidebar
    $('#sidebar').removeClass('active');
    // hide overlay
    $('.overlay').removeClass('active');
  });

  $('#sidebarCollapse').on('click', function () {
    // open sidebar
    $('#sidebar').addClass('active');
    // fade in the overlay
    $('.overlay').addClass('active');
    $('.collapse.in').toggleClass('in');
    $('a[aria-expanded=true]').attr('aria-expanded', 'false');
  });
});

```

Visualisation sur l'application :

Option de la simulation

Début de la simulation

Fin de la simulation

Magnitude Minimal

▲ ▼

Magnitude Maximal

▲ ▼

Vitesse de la simulation (temps passé par sec)

▲ ▼

Jours ▲ ▼

Lancer la simulation

Étapes 2:

On supprime les anciens marqueurs (cercles)

Code implémenté :

```
while(circles.length >0)
{
    var c = circles.pop();
    markerClusters.removeLayer(c);
}
```

```
c.removeFrom(macarte);
}
```

Étapes 3:

- On récupère les données fournies par le formulaire pour ensuite réaliser la requête nécessaire sur la base de données <https://earthquake.usgs.gov> pour obtenir les données des séismes correspondant à la simulation. Ensuite, on les ajoute simplement à la map à l'aide des fonctions présentées dans la partie fonctionnalité 1
- Nous avons également utilisé la fonction **setTimeout** de **JS** ce qui nous a permis de simuler une lecture
- Nous avons aussi rajouté une barre de chargement qui indique l'état de l'affichage en pourcentage

Code implémenté :

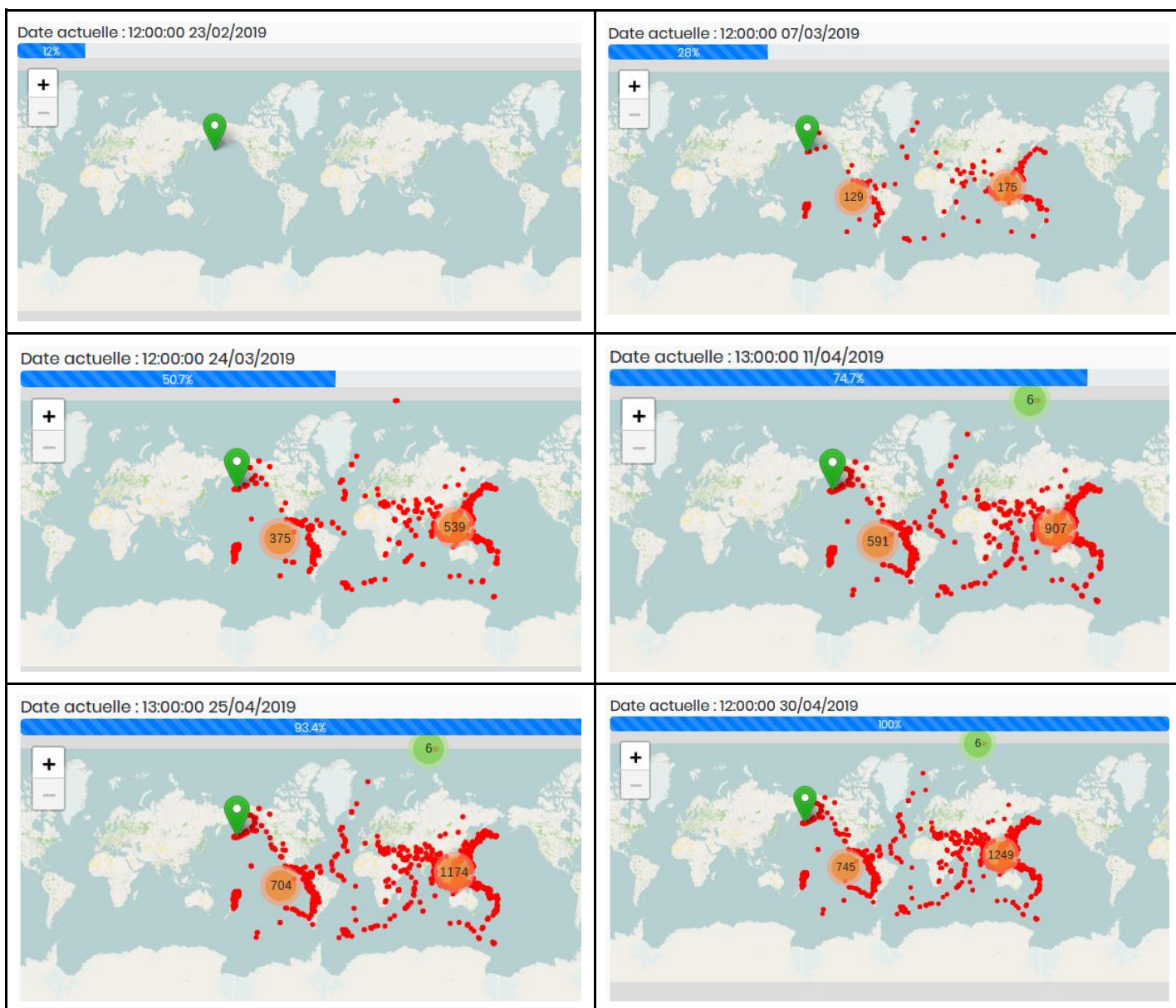
```
if(currentTime < dateFin.getTime())
{
    var htmlOutput = "<div>Date actuelle : "+currentDate.toLocaleTimeString()+"
"+currentDate.toLocaleDateString()+"</div>";
    htmlOutput += "<div class='progress'>";
    htmlOutput += "<div class='progress-bar progress-bar-striped active' role='progressbar'
aria-valuenow="+pourcent+" aria-valuemin='0' aria-valuemax='100' style='width:"+pourcent+"%>";
    htmlOutput += pourcent+"%";
    htmlOutput += "</div>";

    document.getElementById("info_simul").innerHTML = htmlOutput;

    setTimeout(function(){
        afficherSeisme(dateDebut, dateFin, deltaTime, time+deltaTime, newTableSeisme);
    },1000);
}
else
{
    var htmlOutput = "<div>Date actuelle : "+dateFin.toLocaleTimeString()+"
"+dateFin.toLocaleDateString()+"</div>";
    htmlOutput += "<div class='progress'>";
    htmlOutput += "<div class='progress-bar progress-bar-striped active' role='progressbar'
aria-valuenow='100' aria-valuemin='0' aria-valuemax='100' style='width:100%>";
    htmlOutput += "100%";
    htmlOutput += "</div>";

    document.getElementById("info_simul").innerHTML = htmlOutput;
}
```


Visualisation sur l'application :



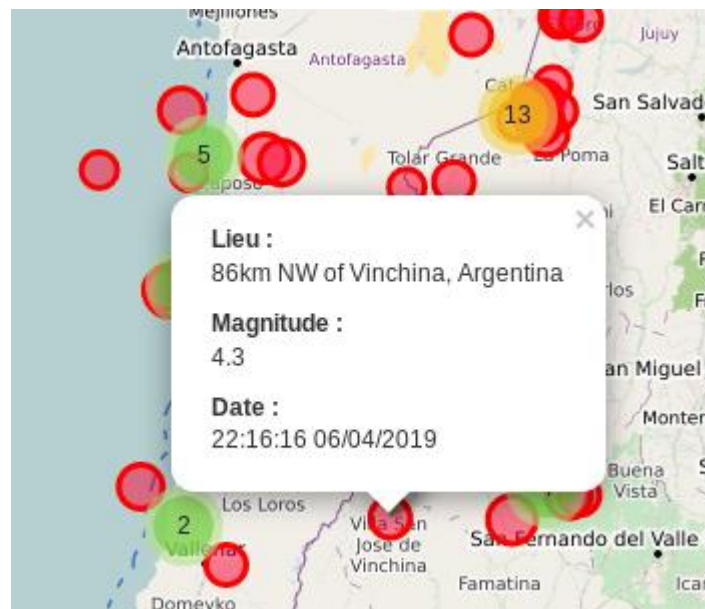
Lorsque l'utilisateur clique sur un marqueur, une info-bulle s'ouvre et donne les caractéristiques principales du tremblement de terre.

Tout comme pour l'ajout de marqueur sur la map, nous avons simplement utilisé une des méthodes de la librairie leaflet et avons rajouté les bons paramètres en se basant sur les données récupérer via la base de connaissance des tremblements de terre.

Code implémenté :

```
//gestion des informations popup du seisme
var popup = "<dl><dt>Lieu : </dt>"
  + "<dd>" + lieu + "</dd>"
  + "<dt>Magnitude : </dt>"
  + "<dd>" + mag + "</dd>"
  + "<dt>Date : </dt>"
  + "<dd>" + date.toLocaleTimeString() + " " + date.toLocaleDateString() + "</dd></dl>";
mycircle.bindPopup(popup);
```

Visualisation sur l'application :



d. Fonctionnalité 5 : API youtube et Google Freebase

Sous la carte Google Maps, l'utilisateur peut voir une liste de vidéo youtube qui sont en lien avec l'emplacement du tremblement de terre sélectionné

Code implémenté :

```
//lorsque l'on clique sur un seisme on affiche les informations du lieu
mycircle.on('click', function(){
    afficherInfoCoord(lat,long);
});
```

vous utiliserez en particulier la méthode `search.list` de l'API youtube, et (au mieux) lecteurs vidéo youtube entre la carte Google Maps et la liste de vidéo, on affiche:

- *quelques caractéristiques et une description succincte du lieu,*

ici, nous avons rédigé des requêtes **SPARQL** que nous avons exécutées sur **dbpedia.org** pour récupérer les informations sur un lieu sélectionné. Le lieu sélectionné est symbolisé par un marqueur vert.

Code implémenté :

Requête SPARQL :

```
var url = "http://dbpedia.org/sparql";
var query = [
    "PREFIX dbo: <http://dbpedia.org/ontology/>",
    "PREFIX dbr: <http://dbpedia.org/resource/>",
    "PREFIX dbp: <http://dbpedia.org/property/>",
    "PREFIX foaf: <http://xmlns.com/foaf/0.1/>",
    "PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>",
    "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>",
    "PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>",
    "SELECT ?cityName ?info ?language ?population ?countryName ?area WHERE {",
    "?city a dbpedia-owl:PopulatedPlace;",
    "        ?city rdfs:label ?cityName;",
    "        ?city geo:lat ?lat;",
    "        ?city geo:long ?long;",
    "        ?city dbo:abstract ?info.",
    "BIND("+long+" AS ?longbis).",
    "BIND("+lat+" AS ?latbis).",
    "BIND(lang(?info) AS ?language).",
    "BIND(bif:st_distance(",
    "        ?city bif:st_point(?long,?lat),",
    "        ?city bif:st_point(?longbis,?latbis)",
    ") AS ?distance).",
    "FILTER(bif:st_intersects( bif:st_point (?long, ?lat), bif:st_point (?longbis, ?latbis), 10))",
    "OPTIONAL {?city dbo:populationTotal ?population.}",
    "OPTIONAL {?city dbo:country ?country. ?country rdfs:label ?countryName.",
    "FILTER(langMatches(lang(?countryName),'FR'))}",
    "}
```



```
"OPTIONAL {?city dbo:areaTotal ?area.}",  
"OPTIONAL {?city dbo:area ?area.}",  
"}",  
"ORDER BY ?distance"  
].join(" ");
```

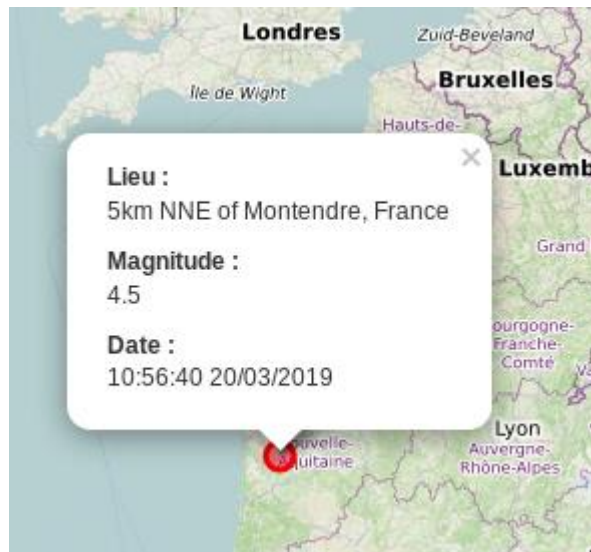
code ajax pour la récupération des données :

```
var queryUrl = url+"?query="+ encodeURIComponent(query) +"&format=json";  
$.ajax({  
  dataType: "jsonp",  
  url: queryUrl,  
  success: function( _data ) {  
    var results = _data.results.bindings;  
    if(typeof results[0] !== 'undefined')  
    {  
      var city = results[0].cityName.value;  
      // var parts = res.split('/');  
      // var city = parts.pop() || parts.pop();  
  
      //on recupere le nom de la ville  
      document.getElementById('place').innerHTML = city;  
  
      //on recupere des infos sur la ville  
      var infoCity = "";  
      //on recupere le nombre d'habitant  
      if(typeof results[0].population !== 'undefined')  
      {  
        infoCity += "<h4>Population : "+results[0].population.value+" habitants</h4>";  
      }  
  
      //on recupere la superficie  
      if(typeof results[0].area !== 'undefined')  
      {  
        infoCity += "<h4>Superficie : "+results[0].area.value/1000000+"km²</h4>";  
      }  
  
      //on recupere la description du lieu  
      var infoFinded = 0;  
      for(var i in results)  
      {  
        if(results[i].language.value == "fr")  
        {  
          infoCity += "<h4>Description : </h4>";  
          infoCity += results[i].info.value;  
          infoFinded = 1;  
          break;  
        }  
      }  
    }  
  }  
});
```

```
    }  
  }  
  if(infoFinded == 0)  
  {  
    for(var i in results)  
    {  
      if(results[i].language.value == "en")  
      {  
        infoCity += "<h4>Description : </h4>";  
        infoCity += results[i].info.value;  
        infoFinded = 1;  
        break;  
      }  
    }  
  }  
  
  //on ajoute toutes les informations générales sur la Page  
  document.getElementById('infoPlace').innerHTML = infoCity;  
  
  //on recupere le pays  
  var country;  
  if(typeof results[0].countryName !== 'undefined')  
  {  
    country = results[0].countryName.value;  
  }  
  else  
    country = city;  
  recupererPersonnalites(country);  
}  
});
```

Visualisation sur l'application :

séisme sélectionné :



Quelque information sur le lieu sélectionné :



Lieu : Expiremont

Informations générales :

Population : 120 habitants

Superficie : 5.66km²

Description :

... Expiremont est une commune du sud-ouest de la France située dans le département de la Charente-Maritime, en région Nouvelle-Aquitaine. Ses habitants sont appelés les Expiremontois et les Expiremontoises.

- *si le lieu est un pays, et pour les 10 premières personnalités connues qui sont nées à cet endroit, vous affichez une description succincte de cette personne.*

Remarque : avant de faire la requête, il faut bien s'assurer que le nom du pays n'as pas d'accent

Code implémenté :

suppression des accents

```
//on supprime les accents  
country = country.normalize('NFD').replace(/[\u0300-\u036f]/g, "");
```

Requête SPARQL :

```
var url = "http://dbpedia.org/sparql";  
var query = [  
    "PREFIX bif: <bif:>",  
    "PREFIX dbo: <http://dbpedia.org/ontology/>",  
    "PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>",  
    "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>",  
    "select distinct ?name ?bio where {",  
        "?cityPage rdfs:label ?country.",  
        "?pers dbo:birthPlace ?cityPage;",  
        "rdfs:label ?name;",  
        "dbo:abstract ?bio.",  
        "?country bif:contains '\"'+country+'\"'.",  
    "FILTER(langMatches(lang(?name),\"FR\") && langMatches(lang(?bio),\"FR\"))",  
    "OPTIONAL{?cityPage dbo:populationTotal ?population}",  
    "}",  
    "ORDER BY DESC (?population)",  
    "LIMIT 10"  
].join(" ");
```

code ajax pour la récupération des données :

```
var queryUrl = url+"?query="+ encodeURIComponent(query) +"&format=json";  
$.ajax({  
    dataType: "jsonp",  
    url: queryUrl,  
    success: function( _data ) {  
        var results = _data.results.bindings;  
        var taille = results.length;  
        var infoPeople = "";  
        for(var i=0 ; i<taille ; i++)  
        {  
            infoPeople += "<div><h3>"+results[i].name.value+"</h3>";  
            infoPeople += "<div>"+results[i].bio.value+"</div></div>";  
        }  
    }  
});
```

```
document.getElementById('infoPeople').innerHTML = infoPeople;  
}  
});
```

Visualisation sur l'application :

On visualise les personnalités connues du lieu du séisme sélectionné précédemment

Personnalités connues (France) :

Adalbert de Magdebourg

Adalbert, premier archevêque de Magdebourg et apôtre des Slaves et des Russes, est né vers 910, probablement en Lorraine, et mort le 20 juin 981.

Adrien-Marie Legendre

Adrien-Marie Legendre, né le 18 septembre 1752 à Paris et mort le 10 janvier 1833 à Auteuil, est un mathématicien français.

Dominique Aury

Anne Cécile Desclos, dite Dominique Aury alias Pauline Réage, est une femme de lettres française du XXe siècle. Première jeune fille admise en khâgne et pionnière du journalisme féminin, elle est l'auteur d'essais, de préfaces et de poèmes. Un quart de siècle durant, elle a été l'adjointe de la direction de la seconde NRF, la première femme à jouer, au sein de la prestigieuse maison Gallimard, plus qu'un rôle d'influence dans le monde de l'édition française et, conseillère du ministre de l'Éducation, à être reconnue professionnellement pour cela. Découvreuse de talents et membre de multiples jurys, elle a insufflé par ses choix et ses critiques une orientation résolument moderniste à la littérature de l'après-guerre. Cependant elle s'est attachée, en tant que directrice de collection et spécialiste de la littérature baroque et religieuse, à une exigence renouvelée de rigueur classique. Ses traductions ont contribué fortement à l'introduction d'auteurs anglais modernes dans le contexte des lettres françaises. Compagne clandestine de Jean Paulhan jusqu'au-delà de la mort de celui-ci, elle s'est révélée en 1994, à l'âge de quatre-vingt-six ans, être l'auteur d'Histoire d'O, un des textes français les plus traduits. Publié en 1954 sous le pseudonyme de Pauline Réage, cet unique roman et sa suite parue en 1969, amputés d'une troisième partie qui en délivrait le sens mystique, composent un manifeste érotique voulu comme une réponse aux fantasmes sadiens des hommes qui fait d'elle la créatrice d'un nouveau genre, la littérature libertine féminine.

Catherine Deneuve

Catherine Dorléac, dite Catherine Deneuve, est une actrice française, née le 22 octobre 1943 dans le 17e arrondissement de Paris. Considérée comme l'une des plus grandes actrices françaises de sa génération et de la seconde moitié du XXe siècle, elle a été l'égérie de nombreux réalisateurs comme Jacques Demy.

- **vous utiliserez Freebase, en particulier les méthodes search et topic de l'API Google Freebase**

Freebase n'étant pas disponible nous avons opter pour une solution simple et pratique.
Nous ajoutons un lien vers la vidéo qui correspond au lieu du séisme sélectionné.

Code implémenté :

```
//on ajoute un lien vers une video youtube en relation avec le lieu  
infoCity += "<h4>Vidéo : <a href=\"http://www.youtube.com/embed?listType=search&list="+city+"\"
```

target="_blank"><i class="fab fa-youtube"></i></h4>;

Visualisation sur l'application :

Lieu : Expiremont

Informations générales :

Population : 120 habitants

Superficie : 5.66km²

Vidéo : 

Description :

- **voyez par exemple ce que Freebase sait sur: la nouvelle calédonie:**
<http://www.freebase.com/m/05c17>

Nous avons implémenté un petit lecteur de vidéos Youtube sur le site, et nous avons mis une vidéo expliquant l'origine des séismes.

Code implémenté :

```
//youtube API
//charge le lecteur de maniere asynchrone
var tag = document.createElement('script');

tag.src = "https://www.youtube.com/iframe_api";
var firstScriptTag = document.getElementsByTagName('script')[0];
firstScriptTag.parentNode.insertBefore(tag, firstScriptTag);

//creation du lecteur
```

```
var player;
function onYouTubeIframeAPIReady() {
  player = new YT.Player('player', {
    height: '360',
    width: '640',
    videoId: 'eMotO_iyo-I',
    events: {
      'onReady': onPlayerReady,
      'onStateChange': onPlayerStateChange
    }
  });
}

//appelle quand le lecteur est pret
function onPlayerReady(event) {
  event.target.playVideo();
}

//fonction appelee quand l'etat du lecteur change
var done = false;
function onPlayerStateChange(event) {
  if (event.data == YT.PlayerState.PLAYING && !done) {
    setTimeout(stopVideo, 6000);
    done = true;
  }
}
function stopVideo() {
  player.stopVideo();
}
```


Visualisation sur l'application :



e. Fonctionnalité Supplémentaire : Déplacement sur la Map

Nous avons jugé qu'il serait intéressant de rajouter cette fonctionnalité de placement. Cela permet à l'utilisateur de se déplacer vers un lieux précis en utilisant soit les coordonnées (latitude et longitude) soit en entrant simplement le nom de la ville et/ou le nom du pays dans lequel il souhaite se rendre.

Cette fonctionnalité peut s'avérer très pratique lorsqu'on souhaite visualiser les tremblements de terre a proximité d'un lieu précis par exemple.

Remarque :

Il est parfois nécessaire d'indiquer le pays où se trouve la ville lorsque cette ville à un homonyme. C'est le cas par exemple de Brest. Parfois la requête prend du temps à être traité par Dbpedia dans le cas où l'on indique le pays et la ville.

Code implémenté :

Code HTML

```
<p>Rechercher un lieu</p>
<li class="paddingLR">

<ul class="nav nav-pills mb-3" id="pills-tab" role="tablist">
```



```

<li class="nav-item">
  <a class="nav-link active" id="pills-lieu-tab" data-toggle="pill" href="#pills-lieu" role="tab"
aria-controls="pills-lieu" aria-selected="true">Lieu</a>
</li>
<li class="nav-item">
  <a class="nav-link" id="pills-coord-tab" data-toggle="pill" href="#pills-coord" role="tab" aria-
controls="pills-coord" aria-selected="false">Coordonnées</a>
</li>
</ul>
<div class="tab-content" id="pills-tabContent">

  <div class="tab-pane fade show active" id="pills-lieu" role="tabpanel" aria-labelledby="pills-lieu-
tab">
    <form id="formCityCountry">
      <div class="form-group">
        <label for="City">Ville</label>
        <input id="formCity" type="text" class="form-control" name="city" placeholder="Ermont">
      </div>
      <div class="form-group">
        <label for="Country">Pays</label>
        <input id="formCountry" type="text" class="form-control" name="country"
placeholder="France">
      </div>
      <button type="submit" class="btn btn-secondary">S'y rendre</button>
    </form>
  </div>

  <div class="tab-pane fade" id="pills-coord" role="tabpanel" aria-labelledby="pills-coord-tab">
    <form id="formLatLong">
      <div class="form-group">
        <label for="Latitude">Latitude</label>
        <input id="formLatitude" type="number" step="any" class="form-control" name="latitude"
placeholder="48.8566101">
      </div>
      <div class="form-group">
        <label for="Longitude">Longitude</label>
        <input id="formLongitude" type="number" step="any" class="form-control" name="longitude"
placeholder="2.3514992">
      </div>
      <button type="submit" class="btn btn-secondary">S'y rendre</button>
    </form>
  </div>

</div>
</li>

```

Code JS : requête SPARQL et code AJAX de récupération au format json

```
function formCityCountry(e)
{
    e.preventDefault();
    var city = document.getElementById('formCity').value;
    var country = document.getElementById('formCountry').value;

    //on supprime les accents
    city = city.normalize('NFD').replace(/[\u0300-\u036f]/g, "");
    country = country.normalize('NFD').replace(/[\u0300-\u036f]/g, "");

    var url = "http://dbpedia.org/sparql";
    var query;
    if(city != "" && country != "")
    {
        query = [
            "PREFIX bif: <bif:>",
            "PREFIX dbo: <http://dbpedia.org/ontology/>",
            "PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>",
            "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>",
            "PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>",
            "select distinct ?lat ?long where {",
            "    ?cityPage rdfs:label ?city;",
            "    dbo:country ?countryPage;",
            "    geo:lat ?lat;",
            "    geo:long ?long.",
            "    ?countryPage rdfs:label ?country.",
            "    ?city bif:contains \"\""+city+"\".",
            "    ?country bif:contains \"\""+country+"\".",
            "OPTIONAL{?cityPage dbo:populationTotal ?population}",
            "}",
            "ORDER BY DESC (?population)",
            "LIMIT 1"
        ].join(" ");

        var queryUrl = url+"?query="+ encodeURIComponent(query) +"&format=json";

        $.ajax({
            dataType: "jsonp",
            url: queryUrl,
            success: function( _data ) {
                var results = _data.results.bindings;
                var lat = results[0].lat.value;
                var long = results[0].long.value;

                //on desactive le panneau lateral
                $('#sidebar').removeClass('active');
                $('.overlay').removeClass('active');
```

```

        afficherInfoCoord(lat, long);
        macarte.flyTo([lat,long],12);
    }

});
}
else if(city != "")
{
    query = [
        "PREFIX bif: <bif:>",
        "PREFIX dbo: <http://dbpedia.org/ontology/>",
        "PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>",
        "PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>",
        "PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>",
        "select distinct ?lat ?long where {",
        "  ?cityPage rdfs:label ?city;",
        "  geo:lat ?lat;",
        "  geo:long ?long.",
        "  ?city bif:contains \""+city+"\".",
        "OPTIONAL{?cityPage dbo:populationTotal ?population}",
        "}",
        "ORDER BY DESC (?population)",
        "LIMIT 1"
    ].join(" ");

    var queryUrl = url+"?query="+ encodeURIComponent(query) +"&format=json";

    $.ajax({
        dataType: "jsonp",
        url: queryUrl,
        success: function( _data ) {
            var results = _data.results.bindings;
            var lat = results[0].lat.value;
            var long = results[0].long.value;

            //on desactive le panneau lateral
            $('#sidebar').removeClass('active');
            $('.overlay').removeClass('active');

            afficherInfoCoord(lat, long);
            macarte.flyTo([lat,long],12);
        }
    });
}
else if(country != "")
{
    query = [
        "PREFIX bif: <bif:>",
        "PREFIX dbo: <http://dbpedia.org/ontology/>",
        "PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>",

```

```

"PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>",
"PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>",
"select distinct ?lat ?long where {",
  "?cityPage rdfs:label ?country;",
  "geo:lat ?lat;",
  "geo:long ?long.",
  "?country bif:contains \"\""+country+"\"".",
"OPTIONAL{?cityPage dbo:populationTotal ?population}",
"}",
"ORDER BY DESC (?population)",
"LIMIT 1"
].join(" ");

var queryUrl = url+"?query="+ encodeURIComponent(query) +"&format=json";

$.ajax({
  dataType: "jsonp",
  url: queryUrl,
  success: function( _data ) {
    var results = _data.results.bindings;
    var lat = results[0].lat.value;
    var long = results[0].long.value;

    //on desactive le panneau lateral
    $('#sidebar').removeClass('active');
    $('.overlay').removeClass('active');

    afficherInfoCoord(lat, long);
    macarte.flyTo([lat,long],5);
  }
});
}
}

```

Visualisation sur l'application :

Option de recherche de séisme

Rechercher un lieu

Lieu

Coordonnées

Ville

Ermont

Pays

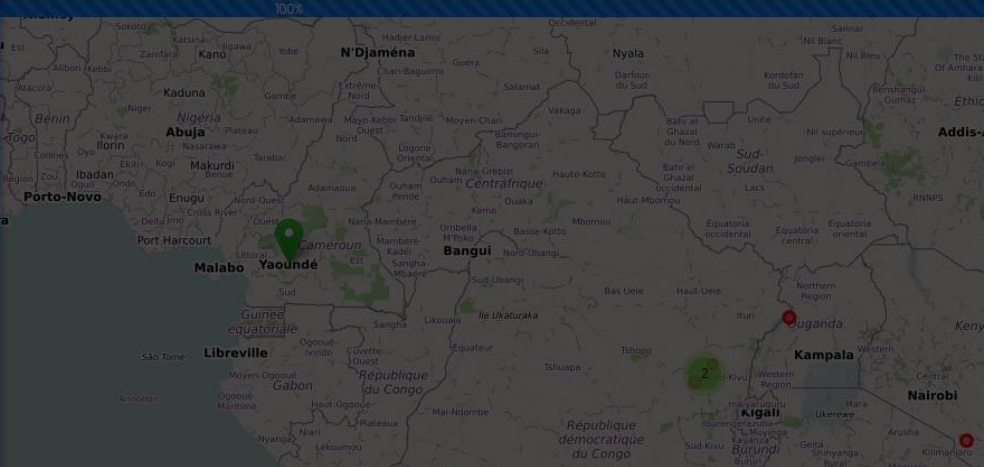
Cameroun

S'y rendre

Option de la simulation

Début de la simulation

Visualisation des tremblements de terre



Conclusion

Comme dans tous projets, nous avons eu l'occasion d'apprendre de nouveaux concepts et notions et d'approfondir ceux qui nous ont été enseignés. Ce projet nous a notamment permis d'utiliser différentes APIs et de les intégrer à notre code JavaScript. Nous avons ainsi pu faire des requêtes sur des données issus d'une page web, ce qui nous a permis de récupérer des coordonnées, une description et bien d'autres informations relatives au lieu ou séismes sélectionnés. En répondant aux différentes demandes du cahier des charges, on a pu se rendre compte que ce projet pourrait être utilisé par le future pour répondre à d'autres besoins de la vie courante et non pas se limiter à la visualisation de tremblement de terre.

Source de documentation fournie

- a source de données: <http://earthquake.usgs.gov/earthquak...month.geojsonp>
- Description des sources de données: <http://earthquake.usgs.gov/earthquak....0/geojson.php>
- pour commencer avec Google maps: <https://developers.google.com/maps/d...cript/tutorial>
- La base de votre application: <https://developers.google.com/maps/t...ng/earthquakes>
- des exemples de code avec Google maps (les info windows vous intéressent):
<https://developers.google.com/maps/d...ript/examples/>
- l'API youtube: <https://developers.google.com/youtube/v3/>
- des exemples de code avec Youtube
API: https://developers.google.com/youtube/v3/code_samples
- l'explorateur d'API Google: <http://developers.google.com/apis-explorer/#p/>
- l'API freebase: <https://developers.google.com/freebase/>

Source de documentation utile

<https://www.youtube.com/watch?v=Zxf1mnP5zcw&t=1327s>

Débuter avec OpenStreetMap : <https://nouvelle-techno.fr/actualites/2018/05/11/pas-a-pas-inserer-une-carte-openstreetmap-sur-votre-site>
<https://leafletjs.com/examples/quick-start/>

API Documentation - Earthquake Catalog
<https://earthquake.usgs.gov/fdsnws/event/1/#parameters>