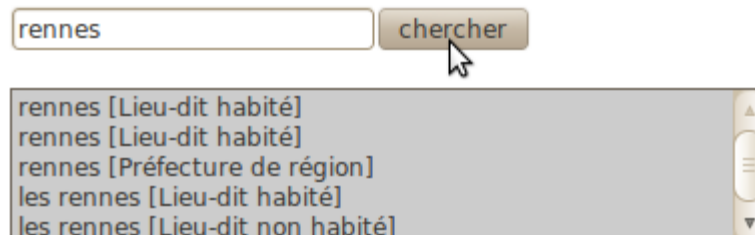


TP OpenLS 2

Recherche « à la main »

Le but de ce TP est de créer un formulaire permettant d'effectuer la recherche par lieu:



Rappels:

- Avoir un contrat qui supporte PositionOfInterest : recherche par lieu;
- La JSDoc est disponible ici : <http://api.ign.fr/tech-docs-js/2.1.2/jsdoc/>

Etapes détaillées :

- 1) Créer une couche OpenLS
 - **NB**: Cette couche contient un attribut queriedAdressees qui est pour l'instant vide : []. On verra dans la suite du TP que le résultat de chaque géocodage sera rajouté à ce tableau.
- 2) Ajouter la couche créée à la carte en utilisant la méthode addLayer() de la classe Geoportal.Map
- 3) Lors du clic sur le bouton 'Rechercher', la méthode myGeocode() est exécutée. Elle applique à la couche OpenLS la méthode GEOCODE() de la classe Geoportal.Layer.OpenLS.LocationUtilityService_:
 - Prend en entrée les paramètres suivants :
 - un tableau contenant les lieux recherchés qui sont des instances de la classe Geoportal.OLS.Address avec en paramètres du constructeur :
 - le countryCode ;
 - un objet contenant les options liées à cette adresse dont { 'name' : 'nom de lieu saisi' }
 - un objet contenant les événements suivants:
 - 'onSuccess' : a pour valeur le nom de la fonction appelée quand la requête est exécutée avec succès
 - 'onFailure' : a pour valeur le nom de la fonction appelée si la requête renvoie une erreur

- 'scopeOn' : contexte d'exécution des fonctions associées aux évènements 'onSuccess' et 'onFailure'. Dans notre cas, cet objet sera vide: {}
 - Renvoie un objet XMLHttpRequest dont l'attribut responseXML contient le résultat de la recherche.
- 4) Écrire la fonction associée à l'évènement 'onSuccess' qui va effectuer les actions suivantes :
- Récupérer le tableau features correspondant au résultat du géocodage via l'attribut queriedAddresses de la couche OpenLS
 - var features= toponymLayer.queriedAddresses[0].features ;
 - Ajouter ces features à la couche OpenLS en lui appliquant la méthode addFeatures()
 - Afficher dans la console le tableau features
 - Traiter éventuellement le résultat de la recherche en le mettant dans un select (la fonction est détaillée dans le code source du tp)
- 5) Modifier le style de l'affichage des résultats (en affichant par exemple des points d'un diamètre croissant en fonction de la vraisemblance du résultat) :
- Créer une variable styleMap de type OpenLayers.Layer.StyleMap qui va définir les styles suivants en fonction de l'état des feature :
 - « default » :

```
new OpenLayers.Style({
    pointRadius : "${getRadius}",
    fillColor : "black"
},{
    context : {
        getRadius: function(feat) {
            return feat.attributes.geocodeMatchCode.accuracy*10 ;
        }
    }
})
```

- « select »

```
new OpenLayers.Style({
    pointRadius : "${getRadius}",
    fillColor : "red"
```

```
    }, {  
      context : {  
        getRadius: function(feat) {  
          return feat.attributes.geocodeMatchCode.accuracy*10 ;  
        }  
      }  
    }  
  })
```

b) Rajouter cette variable à la construction de la couche OpenLS avec l'option styleMap.

6) Rendre la liste des réponses active pour que lorsqu'on sélectionne une réponse dans la liste alors le point sur la carte soit sélectionné :

a) Compléter le code de la fonction selectResult :

1. Récupérer l'index de l'élément sélectionné dans la liste
2. Faire passer l'état du feature correspondant à « select » en utilisant la méthode select de la propriété selectCntrl de la couche OpenLS :

```
geocodeLayer.selectCntrl.select(feature) ;
```

3. Gérer la désélection du précédent feature à l'aide de la méthode unselect.