jDev
Vos solutions numériques
open-sources et SIG

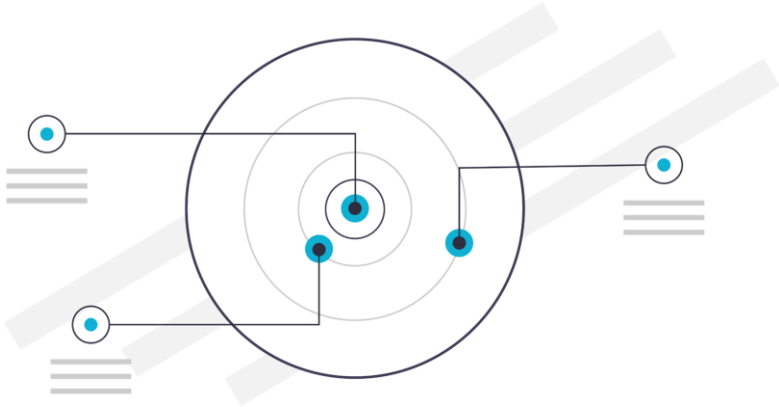
Formation

TP5
Custom layer & représentation

05/10/2021

TP5

Présentation



Objectifs :

> Savoir mettre en place un custom layer

> Manipuler un custom layer

> Connaître les différentes aides en ligne

> Représenter un custom layer

> Faire un cluster

TP5

Contexte

Vous disposez des application du répertoire /demo et /apps en guise d'exemple.

Tout le TP sera à réaliser dans le dossier /apps/entreprises.

Nous utiliserons l'éditeur de texte Visual Studio Code pour ouvrir, modifier, créer un custom layer.

Aidez-vous de la documentation Mviewer et des issues GitHub.

TP5

Mettre en place un custom layer

1. Créer un custom layer

Etape 1 : récupérer les données

- Ouvrez le répertoire /apps/entreprises
- Copier le fichier C:\xampp\htdocs\mviewer\customlayers\lycee.js
- Coller le fichier lycee.js et le renommer dans /apps/entreprises/customlayers/sirene.js
- Téléchargez le fichier https://gis.jdev.fr/mviewer/app/data/sirene_35.geojson
- Copier ce fichier et le coller dans un répertoire /apps/entreprises/data

TP5

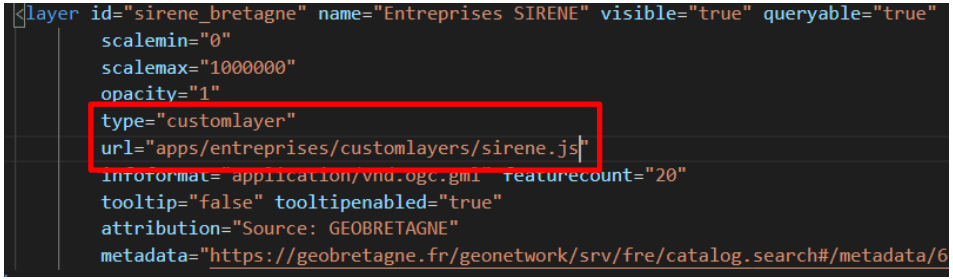
Mettre en place un custom layer

1. Créer un custom layer

Etape 2 : configurer le XML

Nous allons changer la couche « sirene_bretagne » pour utiliser un custom layer et non un WMS.

- Ouvrez /apps/entreprises/customlayers/sirene.js
- Ouvrez /apps/entreprises.xml
- Modifier l'URL et pointez vers apps/entreprises/customlayers/sirene.js
- Modifier le type wms par customlayer



jDev

TP5

Mettre en place un custom layer

1. Créer un custom layer

Etape 3 : créer la couche en JavaScript

Actuellement la carte ne s'affiche pas correctement. Il faut modifier / adapter le fichier sirene.js.

- En bas du fichier, modifier l'ID en utilisant celui défini dans la configuration.

```
<layer id="sirene_bretagne" na
       scalemin="0"
```

```
new CustomLayer("sirene_bretagne", layer, legend);
```

TP5

Mettre en place un custom layer

1. Créer un custom layer

Etape 3 : créer la couche en JavaScript

Actuellement la carte ne s'affiche pas correctement. Il faut modifier / adapter le fichier sirene.js.

- Modifier l'URL par la position du fichier geojson (ou vers l'URL du WFS si la couche est publiée par un serveur cartographique)

```
source: new ol.source.Vector({
    url: "apps/enteperises/data/sirene_35.geojson",
    format: new ol.format.GeoJSON()
}),
```

TP5

Mettre en place un custom layer

1. Créer un custom layer

Etape 3 : créer la couche en JavaScript

Actuellement la carte ne s'affiche pas correctement. Il faut modifier / adapter le fichier sirene.js.

- Modifier l'URL par la position du fichier geojson (ou vers l'URL du WFS si la couche est publiée par un serveur cartographique)

```
source: new ol.source.Vector({
    url: "apps/enteperises/data/sirene_35.geojson",
    format: new ol.format.GeoJSON()
}),
```

TP5

Mettre en place un custom layer

1. Créer un custom layer

Etape 4 : modifiez le style par défaut

- Supprimez les lignes 1 à 14
- Renommez la variable stylePrive par uniqStyle

```
1   let uniqStyle = [new ol.style.Style({
2       image: new ol.style.Circle({
3           fill: new ol.style.Fill({
4               color: 'rgba(99, 110, 114,1.0)'
```

TP5

Mettre en place un custom layer

1. Créer un custom layer

Etape 4 : modifiez le style par défaut

- Supprimez les lignes 1 à 14
- Renommez la variable stylePrive par uniqStyle

```
1   let uniqStyle = [new ol.style.Style({
2       image: new ol.style.Circle({
3           fill: new ol.style.Fill({
4               color: 'rgba(99, 110, 114,1.0)'
```

- Modifiez l'appel au style dans la source de la couche comme ceci :

```
style: function(feature, resolution) {
    return uniqStyle;
}
```

TP5

Mettre en place un custom layer

1. Créer un custom layer

Etape 5 : Visualiser

- Les données ne semblent pas visible
- Faites un zoom arrière (–) pour voir toute l'Europe

Que voyez-vous ?

TP5

Mettre en place un custom layer

2. Système de projection

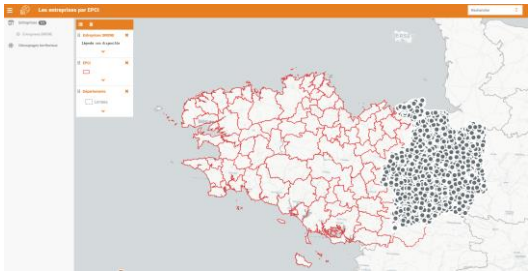
Etape 1 :

La projection ne semble en effet pas adéquate…

- Dans le fichier de config XML entre les balises <mapoption> changez le code pour passer en EPSG:2154

L'étendue visible par défaut n'est plus bonne… il faut alors en utiliser une autre

- Positionnez vous sur la carte pour voir toutes les données en Bretagne :



jDev

TP5

Mettre en place un custom layer

2. Système de projection

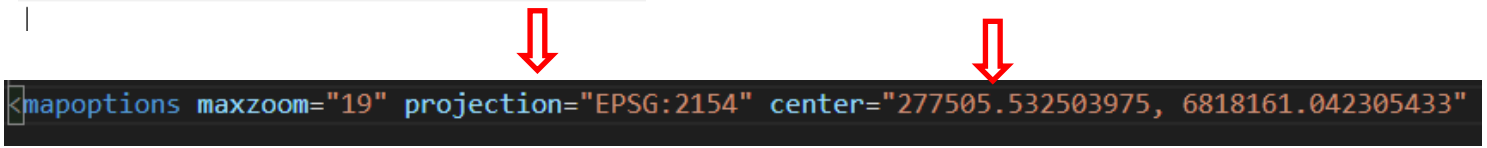
Etape 2

- Ouvrez la console du navigateur (F12)
- Dans l'onglet « console » coller ce code et appuyez sur « Entrée »

mviewer.getMap().getView().getCenter()

Vous obtenez alors la valeur du paramètre center à changer dans la balise <mapoptions> du fichier entreprises.xml.

```
mviewer.getMap().getView().getCenter()
▶ (2) [277505.532503975, 6818161.042305433]
```



- Changer le niveau de zoom si besoin

TP5

Représenter un custom layer

3. Changer le style

Nous avons déjà abordé le style simple et un style basé sur un attribut (lycée public / privé) en début de formation.
Nous allons voir comment réalisé un style plus complexe selon le niveau de zoom et selon un attribut.

Etape 1

- Désactivez toutes les couches et activez uniquement les communes
- Cliquez sur la commune de Rennes et récupérez son code postal (35238)

Etape 2

- Ouvrir le fichier /customlayers/sirene.js
- Rajoutez une condition pour n'afficher que les établissements avec un code postal 35238
- Rechargez la carte pour observez l'affichage des établissements

Nous voyons que le style peut nous permettre d'afficher des entités selon un attribut (filtre).

TP5

Représenter un custom layer

3. Changer le style

Nous avons déjà abordé le style simple et un style basé sur un attribut (lycée public / privé) en début de formation.
Nous allons voir comment réalisé un style plus complexe selon le niveau de zoom et selon un attribut.

Etape 1

- Désactivez toutes les couches et activez uniquement les communes
- Cliquez sur la commune de Rennes et récupérez son code postal (35238)

```
style: function(feature, resolution) {
    if (feature.get('codepostal') == 35000) {
        return uniqStyle;
    }
}
```

Etape 2

- Ouvrir le fichier /customlayers/sirene.js
- Rajoutez une condition pour n'afficher que les établissements avec un code postal 35238
- Rechargez la carte pour observez l'affichage des établissements

Nous voyons que le style peut nous permettre d'afficher des entités selon un attribut (filtre).

TP5

Représenter un custom layer

3. Changer le style

Nous avons déjà abordé le style simple et un style basé sur un attribut (lycée public / privé) en début de formation.
Nous allons voir comment réalisé un style plus complexe selon le niveau de zoom et selon un attribut.

Etape 2

- Ouvrir le fichier /customlayers/sirene.js
- Rajoutez une condition pour n'afficher que les établissements avec un code postal 35000

```
style: function(feature, resolution) {
    if (feature.get('codepostal') == 35000) {
        return uniqStyle;
    }
}
```

- Rechargez la carte pour observez l'affichage des établissements

Nous voyons que le style peut nous permettre d'afficher des entités selon un attribut (filtre).

TP5

Représenter un custom layer

3. Changer le style

Etape 3

- Modifiez les propriétés du style pour afficher en bleu les établissements avec le code postal 35000
- Renommez la variable « uniqStyle » en « blueStyle » et appliquez les modifications nécessaire dans la fonction style de la source de la couche

```
style: function(feature, resolution) {
    if (feature.get('codepostal') == 35000) {
        return blueStyle;
    }
}
```

- Maintenant, copier le style « uniqStyle » pour créer un second style appelé « greyStyle »
- Modifiez « blueStyle » pour afficher du bleu et modifiez « greyStyle » pour afficher du gris

TP5

Représenter un custom layer

3. Changer le style

Etape 3

- Enfin, sous le commentaire, rajoutez la condition pour utiliser « greyStyle » pour les autres établissements en dehors du CP 35000

```
let layer = new ol.layer.Vector({
        source: new ol.source.Vector({
            url: "apps/entreprises/data/sirene_35.geojson",
            format: new ol.format.GeoJSON()
        }),
        style: function(feature, resolution) {
            if (feature.get('codepostal') == 35000) {
                return blueStyle;
            }
            // ici rajouter une condition pour les autres établissements...
        }
});
```

TP5

Mettre en place un custom layer

3. Changer le style

Etape 3

- Sous le commentaire, rajoutez la condition pour utiliser « greyStyle » pour les autres établissements en dehors du CP 35000

```
let layer = new ol.layer.Vector({
        source: new ol.source.Vector({
            url: "apps/entreprises/data/sirene_35.geojson",
            format: new ol.format.GeoJSON()
        }),
        style: function(feature, resolution) {
            if (feature.get('codepostal') == 35000) {
                return blueStyle;
            }
            // ici rajouter une condition pour les autres établissements...
        }
});
```

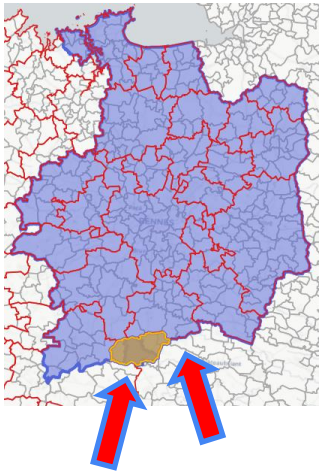
- Commentez les lignes qui commencent par « legend.items » et raffraîchissez la carte

Représenter un custom layer

3. Changer le style

Etape 4 : utiliser une icone pour le style

Nous souhaitons maintenant afficher une icône pour les établissements de l'EPCI « CC du Pays du Grand Fougeray » et l'EPCI « CC Au Pays de la Roche Aux Fées »



Représenter un custom layer

3. Changer le style

Etape 4 : utiliser une icone pour le style

- Récupérez le SVG dans le répertoire du TP5 /svg/store.svg et copier le dans /apps/entreprises/img
- Observez cet exemple : https://openlayers.org/en/latest/examples/icon.html

Les lignes 19 à 20 permettent d'utiliser une image pour représenter un point.

- Copiez ces lignes :

```
const iconStyle = new ol.style.Style({
  image: new ol.style.Icon({
    anchor: [0.5, 46],
    anchorXUnits: 'fraction',
    anchorYUnits: 'pixels',
    src: 'data/icon.png',
  }),
});
```

TP5

Représenter un custom layer

3. Changer le style

Etape 4 : utiliser une icone pour le style

- Collez ces lignes au dessus de la ligne qui commence par « let layer » en changeant l'URL (src) et en ajoutant une couleur :

```
const iconStyle = [new ol.style.Style({
    image: new ol.style.Icon({
      anchor: [0.5, 46],
      anchorXUnits: 'fraction',
      anchorYUnits: 'pixels',
      color: "red",
      src: 'apps/entreprises/img/store.svg',
    }),
  })];
```

TP5

Représenter un custom layer

3. Changer le style

Etape 4 : utiliser une icone pour le style

Nous allons utiliser ce style uniquement si le code postal est compris dans cette [liste] :

[32549, 35124, 35098]

Une liste commence et se termine par des crochets [a,b,c]

- Pour cela on rajoute une condition else if et on utilise une méthode .includes :

```
style: function(feature, resolution) {
    // Si Commune de Rennes avec CP 35000
    if (feature.get('codepostal') == 35000) {
        return blueStyle;
    // Sinon Si EPCI avec CP 35390 ou CP 35240
    } else if(["35390", "35240"].includes(feature.get("codepostal"))) {
        return iconStyle;
    }
    // Pour tous les autres points qui ont d'autres CP
    return greyStyle;
}
```

[1,2].includes(1) ➔ VRAI

[1,3,4].includes(2) ➔ FAUX

jDev

TP5

Représenter un custom layer

3. Changer le style

Etape 4 : utiliser une icone pour le style

- Faites un refresh de la page

- Modifiez la taille de l'icône en vous aidant de la documentation pour trouver la bonne propriété
https://openlayers.org/en/latest/apidoc/module-ol_style_Icon-Icon.html

TP5

Représenter un custom layer

3. Changer le style

Etape 4 : utiliser une icone pour le style

- Faites un refresh de la page

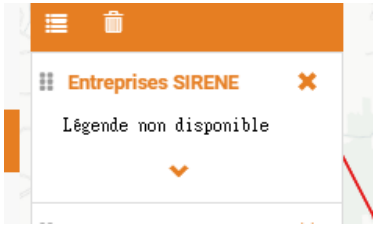
- Modifiez la taille de l'icône en vous aidant de la documentation pour trouver la bonne propriété
https://openlayers.org/en/latest/apidoc/module-ol_style_Icon-Icon.html

```
const iconStyle = [new ol.style.Style({
    image: new ol.style.Icon({
      anchor: [0.5, 46],
      anchorXUnits: 'fraction',
      anchorYUnits: 'pixels',
      color: "black",
      scale: 0.1,
      src: 'apps/entreprises/img/store.svg',
    }),
  })];
```

TP5

Représenter un custom layer

3. Changer le style



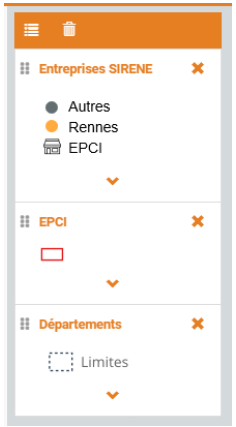
Etape 5 : Ajouter une légende

Notre couche n'a pas de légende...

- Copier l'image du répertoire /TP5/img/etab.png vers /apps/entreprises/img/

- Dans le fichier entreprises.xml ajouter l'URL de la légende

```
tooltip="false" tooltipenabled="true"
legendurl="apps/entreprises/img/etab.png"
attribution="Source: GEOBRETAGNE"
```



jDev

TP5

Représenter un custom layer

4. Filtre CQL

Etape 1 :
La couche met beaucoup de temps à s'afficher.
Nous allons donc restreindre les données pour ne voir que les établissements avec les CP 35390 et 35240

- Sur la page du catalogue, récupérez l'URL du WFS :
https://geobretagne.fr/geonetwork/srv/fre/catalog.search#/metadata/6bf9d921-b8eb-4e5f-bbb5-546f7c40dab3

- Ouvrez le fichier sirene.js et remplacez l'URL du fichier par l'URL du WFS

- Rajouter dans l'URL :

```
&CQL_FILTER=codepostal in ('35390','35240')
```

- Rechargez la carte et constatez

TP5

Représenter un custom layer

4. Filtre CQL

Etape 2 :

- Trouvez une autre icône ou bien faites une icône par vos propres moyens

Représentation en cluster

5. Affichage en cluster

Etape 1 : observer l'existant

- Retrouvez dans le répertoire /mviewer un exemple de custom layer en cluster et le config.xml associé

TP5

Représentation en cluster

5. Affichage en cluster

Etape 2 : copier l'existant

- Ouvrez le fichier sirene.js
- Créez une variable l'URL et une autre pour le CQL

```
let cql = "CQL_FILTER=codepostal in ('35390','35240')";
let url = "https://ows.region-bretagne.fr/geoserver/rb/wfs?SERVICE=WFS&REQUEST=GetCapabilities&request=GetFeature&service=WFS&version=1.1.0&typeName=sirene_bretagne&outputFormat=application/json";
```

- Commentez la définition de la couche

```
/*let layer = new ol.layer.Vector({
        source: new ol.source.Vector({
            //url: "apps/entreprises/data/sirene_35.geojson",
            url: "https://ows.region-bretagne.fr/geoserver/rb/wfs?SERVICE=WFS&REQ
            format: new ol.format.GeoJSON()
        }),
        style: function(feature, resolution) {
            // Si Commune de Rennes avec CP 35000
            if (feature.get('codepostal') == 35000) {
                return blueStyle;
            // Sinon Si EPCI avec CP 35390 ou CP 35240
            } else if(["35390", "35240"].includes(feature.get("codepostal"))) {
                return iconStyle;
            }
            // Pour tous les autres points qui ont d'autres CP
            return greyStyle;
        }
});*/
```

jDev

Représentation en cluster

5. Affichage en cluster

Etape 3 : copier l'existant

- Rajoutez cette nouvelle définition de couche à partir du code du custom layer du fichier cluster.js

```
let layer = new ol.layer.Vector({
    source: new ol.source.Cluster({
        geometryFunction: function(feature) {
            return new ol.geom.Point(ol.extent.getCenter(feature.getGeometry().getExtent()));
        },
        distance: 50,
        source: new ol.source.Vector({
            url: "https://geobretagne.fr/geoserver/dreal_b/wfs?service=WFS&version=1.0.0&request=GetFeature&type
Names=dreal_b:projets-environnement-diffusion&outputFormat=application/json&srsName=EPSG:4326&bbox=-6,47,0,49",
            format: new ol.format.GeoJSON()
        })
    }),
    style: clusterStyle

});
```

Représentation en cluster

5. Affichage en cluster

Etape 4 : copier l'existant

- Modifiez l'URL de la couche en utilisant la variable url

```
source: new ol.source.Vector({
    url: url + "&" + cql,
    format: new ol.format.GeoJSON()
})
```

5. Affichage en cluster

Etape 5 : copier l'existant

- Récupérez le style clusterStyle dans le fichier cluster.js et collez le dans le fichier sirene.js en haut du fichier

```
var clusterStyle = function(feature) {
       var size = feature.get('features').length;
       var max_radius = 40;
       var max_value = 500;
       var radius = 10 + Math.sqrt(size)*(max_radius / Math.sqrt(max_value));
       var radius2 = radius *80 /100 ;
       if (size == 1) {
           return uniqueStyle;
       } else {
           return manyStyle(radius, radius2, size);
       }
   };
```

Qu'est-ce qui manque à nos cluster pour que ce code fonctionne ?

TP5

Représentation en cluster

5. Affichage en cluster

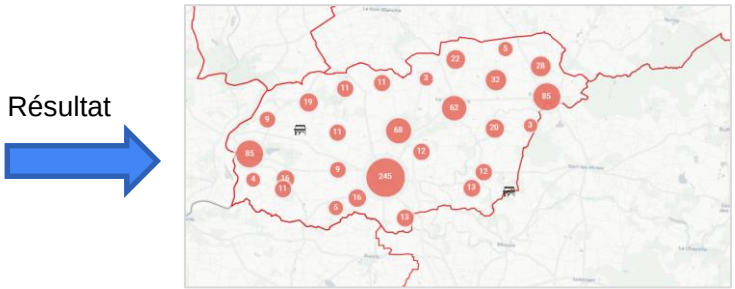
Etape 6 : copier l'existant

Il nous manque les styles manyStyle et uniqStyle.

- Remplacez uniqStyle par iconStyle ➔ optionnel ➔ C'est le style à afficher quand il y a qu'une seule feature dans un cluster
- Récupérez manyStyle dans le fichier cluster.js et copier / coller le style manyStyle vers notre fichier sirene.js

```
var clusterStyle = function(feature) {
       var size = feature.get('features').length;
       var max_radius = 40;
       var max_value = 500;
       var radius = 10 + Math.sqrt(size)*(max_radius / Math.sqrt(max_value));
       var radius2 = radius *80 /100 ;
       if (size == 1) {
           return iconStyle;
       } else {
           return manyStyle(radius, radius2, size);
       }
   };
```

Résultat



Représentation en cluster et en style simple

5. Bonus

A partir du résultat précédent, modifiez le style à afficher pour une seule feature afin d'avoir un style différent pour chaque code postal : 35000, 35240, 35390.

- Adaptez le CQL
- Adaptez les propriétés de style

CONTACT

Pour toute question, n'hésitez pas à nous contacter

pierre.jego@jdev.fr
gaetan.bruel@jdev.fr
agathe.adam@jdev.fr

CRÉDITS