

Stockage de données: les solutions simples - TP2

Hiba ALQASIR

Automne 2021

Objectif de la séance

L'objectif de cette séance est la prise en main des formats de fichiers de stockage de données GeoJSON et YAML.

A la fin de la séance, vous déposerez sur le portail une archive au format zip contenant tous vos fichiers, *i.e.* un fichier texte avec vos réponses (pas de code à l'intérieur de ce fichier), et tous les fichiers javascript, yaml, geojson ...

Attention, lorsque vous copiez d'un fichier pdf, supprimez les espaces supplémentaires causés par le copier-coller, car ils causent souvent des problèmes au moment de l'exécution, exemple : `'utf8 '` n'est pas la même chose que `'utf8'`, il y a un espace supplémentaire dans le premier, qui doit être supprimé.

1 Représentation des données sur une cartographie numérique

1. Cliquer sur [arrondissements-42-loire](#), et copier le code du fichier.
2. Aller sur le site [geojson.io](#) .
3. Coller le code dans la fenêtre de droite (`</>JSON`) comme indiqué dans la Figure 1.
4. Observer les trois polygones et supprimer les deux qui ne représentent pas la ville de Saint-Étienne (cliquer sur le polygone, puis `'Delete feature'`).
5. Changer la couleur de fond de la ville de Saint-Étienne en vert.
6. Sauvegarder le résultat, depuis le menu 'Save', puis 'GeoJSON' (voir Figure 2), le nom de fichier proposé par défaut est `map.geojson` (changer le en `sainté.geojson`).



Figure 1: geojson.io interface de base.

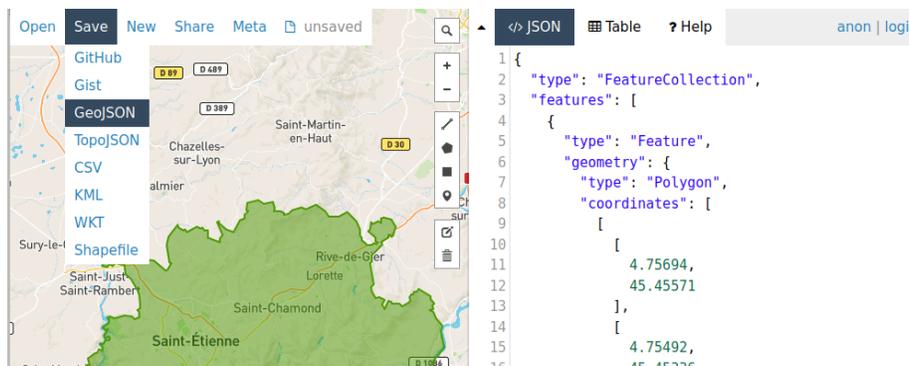


Figure 2: Sauvegarder un fichier au format geojson.

2 Créer une carte personnalisée au format GeoJSON

1. Aller sur le site geojson.io .
2. Zoomer sur la carte pour trouver Télécom Saint-Étienne, ou utiliser l'outil de recherche **Q**.
3. Créer une zone pour délimiter Télécom Saint-Étienne (voir Figure 3). Pour cela cliquer sur les outils de dessin sur le côté gauche pour dessiner des points, des polygones, des lignes et des rectangles.
4. Changez la couleur de fond de la zone en rouge.
5. Ajouter un champ **date** (cliquer sur le polygone, onglet 'Properties', puis 'Add row' ...).
6. Sauvegarder le résultat dans un fichier nommé **télécom.geojson**.



Figure 3: Télécom Saint-Etienne

3 Créer un fichier au format GeoJSON manuellement

A partir du fichier [station_information.json](#) de TP1 nous allons extraire des informations pour construire `station_information.geojson`, pour cela:

1. Créer un nouveau fichier et le nommer `station_information.geojson`.
2. Copier le code suivant et le coller dans le fichier `station_information.geojson`.

```
1 {
2   "type": "FeatureCollection",
3   "features": [
4     {
5       "type": "Feature",
6       "properties": {
7         "marker-color": "#f57900",
8         "marker-size": "large",
9         "marker-symbol": "bicycle",
10        "station": "Châteaureux"
11      },
12      "geometry": {
13        "type": "Point",
14        "coordinates": [
15          4.398855,
16          45.442972
17        ]
18      }
19    }
20  ]
21 }
22 }
```

Ce code représente une seule station ‘Châteaureux’, les coordonnées sont extraites du fichier **station_information.json** (voir Figure 4), attention, le premier est la longitude et le second la latitude.



```
JSON  Données brutes  En-têtes
Enregistrer Copier Tout réduire Tout développer Filtre le JSON
last_updated: 1635511574
ttl: 60
data:
  stations:
    0:
      station_id: "101"
      name: "Châteaureux"
      lat: 45.442972
      lon: 4.398855
      capacity: 32
    1:
      station_id: "102"
      name: "Chaléassiere"
      lat: 45.453601
      lon: 4.383062
      capacity: 16
```

Figure 4: station_information.json

3. Pour ajouter une nouvelle station, copier les lignes de 4 à 19 et les coller à la ligne 20, remplacer le nom station par ‘Chaléassiere’ et les coordonnées par (4.383062, 45.453601). Ajouter une virgule après l’accolade de la ligne 19 pour séparer les deux stations.
4. Aller sur le site geojson.io.
5. Ouvrir le fichier **station_information.geojson** depuis le menu ‘Open’, puis ‘File’. Reporter et commenter le résultat.
6. Refaire 3 pour quatre autres stations au choix, puis refaire 4 et 5.

4 Créer un fichier au format GeoJSON avec JavaScript

Le code JavaScript suivant permet d'analyser des données au format GeoJSON:

```
1 const fs = require('fs')
2
3 data = {
4   "type": "Feature",
5   "properties": {
6     "marker-color": "#4e9a06",
7     "marker-size": "medium",
8     "marker-symbol": "star",
9     "lieu": "Stade Geoffroy-Guichard"
10  },
11  "geometry": {
12    "type": "Point",
13    "coordinates": [
14      4.390416741371154, // long
15      45.46084929602109 // lat
16    ]
17  }
18 }
19 console.log("les clefs du dictionnaire :")
20 for(key in data)
21   console.log(key)
22
23 console.log("type de données de data :"+ data["type"])
```

1. Copier le code dans un fichier 'file_formats6.js'.
2. Lancer l'exécution de ce programme dans VS Code.
3. Reporter et commenter le résultat de l'exécution de ce programme.
4. Ajouter au programme des instructions:
 - pour afficher le lieu.
 - pour afficher les coordonnées.
 - pour sauvegarder l'objet **data** au fichier nommé **stade.geojson**.
5. Aller sur le site geojson.io.
6. Ouvrir le fichier **stade.geojson** depuis le menu 'Open', puis 'File' ...
7. Changer le marker-symbol en 'soccer', enregistrer dans un nouveau fichier **stade2.geojson**, comparer à **stade.geojson**, quelle ligne a été modifiée?

5 YAML

1. JSON2YAML
 - (a) Copier le contenu du fichier **stade2.geojson**.

- (b) Aller sur le site [json2yaml](https://json2yaml.com/).
- (c) Coller le contenu du fichier dans la fenêtre de gauche (JSON).
- (d) Que s'est-il passé? est ce que c'est un format valide? pourquoi?
- (e) Copier le code de la fenêtre de droite (YAML) dans un fichier et le nommer **stade2.yml**.

2. Le code JavaScript suivant permet d'analyser des données au format YAML:

```

1  const fs = require('fs')
2
3  fs.readFile('./stade.yml', 'utf8', (err, data) => {
4    if (err) {
5      console.error(err)
6      return
7    }
8
9    var rows = data.split('\n');
10   for(var i = 1; i < rows.length; i++){
11     rows[i] = rows[i].trim();
12
13     if( rows[i].startsWith('-') )
14       console.log(rows[i]);
15     else
16     {
17       var row = rows[i].split(':');
18
19       if (row[1]) // in case the line contains value
20         console.log('key: ' + row[0] + ', value: ' + row[1]);
21       else // in case the line doesn't contain value
22         console.log('key: ' + row[0] );
23     }
24   }
25 });

```

- (a) Copier le code dans un fichier 'file_formats7.js'.
- (b) Remplacer `./stade2.yml` par le chemin complète vers le fichier, par exemple `/home/user/Bureau/stade2.yml`. Ou placer 'file_formats7.js' et 'stade2.yml' dans le même répertoire, sans changer le code.
- (c) Indiquer le fonctionnement de l'instruction `trim()`¹.
- (d) Pourquoi la boucle `for` commence à 1 et non à 0, qu'y a-t-il dans `rows[0]` ?
- (e) Lancer l'exécution de ce programme dans VS Code.
- (f) Reporter et commenter le résultat de l'exécution de ce programme.

¹Vous pourrez vous reporter au site internet javascript.info qui est un guide générale sur le JavaScript. Ou au site internet nodejs.dev qui est un tutoriel Node.js.