

# SER : Labo 3, GEOJSON - KML

Da Silva Michaël, Rieder Thomas

## Classe java

---

Pour ce laboratoire, nous avons pensé à créer les deux Classes Java suivantes.

### GeoJsonReader

Cette classe nous permet de lire un fichier en format `geojson` et ainsi nous retourner un tableau de type `JSONArray` du niveau le plus haut nous intéressant.

Elle possède les fonctions suivantes :

- `parseGeoJson()` : prend en paramètre un fichier `geojson` et retourne la listes des pays avec leur coordonnées dans un `JSONArray`

### KMLWriter

Cette classe nous permet de lire plus en profondeur les données contenu dans un `JSONArray` venant d'un fichier en format `geojson` puis dans les traiter pour finalement les écrire dans un fichier au format `kml`.

Elle possède les fonctions suivantes :

- `generateKML()` : point d'entrée permettant la création d'un fichier KML, elle prend en paramètre un `JSONArray` contenant la liste des pays et un `String` correspondant au chemin vers le fichier à écrire. Cette fonction appelle les autres fonction permettant de traiter ces données puis fini par écrire le fichier KML.
- `parseCountryObject()` : Est appelée pour chaque pays se trouvant dans le `JSONArray` de la fonction `generateKML()` pour traiter les données correspondants au pays.
- `transformPolygon()` : Est appelée pour chaque Polygon de chacun des pays pour ainsi transformer les données lues en données au format `kml`.
- `writeKMLFile()` : Cette fonction va assembler toutes les données ayant été traitées par les autres fonction afin de créer un fichier valide au format `kml`.

## Difficultés rencontrées

---

Dans ce laboratoire, la plus grande difficulté était de comprendre le format `kml` afin de créer un fichier valide et compréhensible par Google Earth. Nous y sommes parvenu en lisant la documentation **KML Reference** fournit dans la donnée du laboratoire. <https://developers.google.com/kml/documentation/kmlreference>

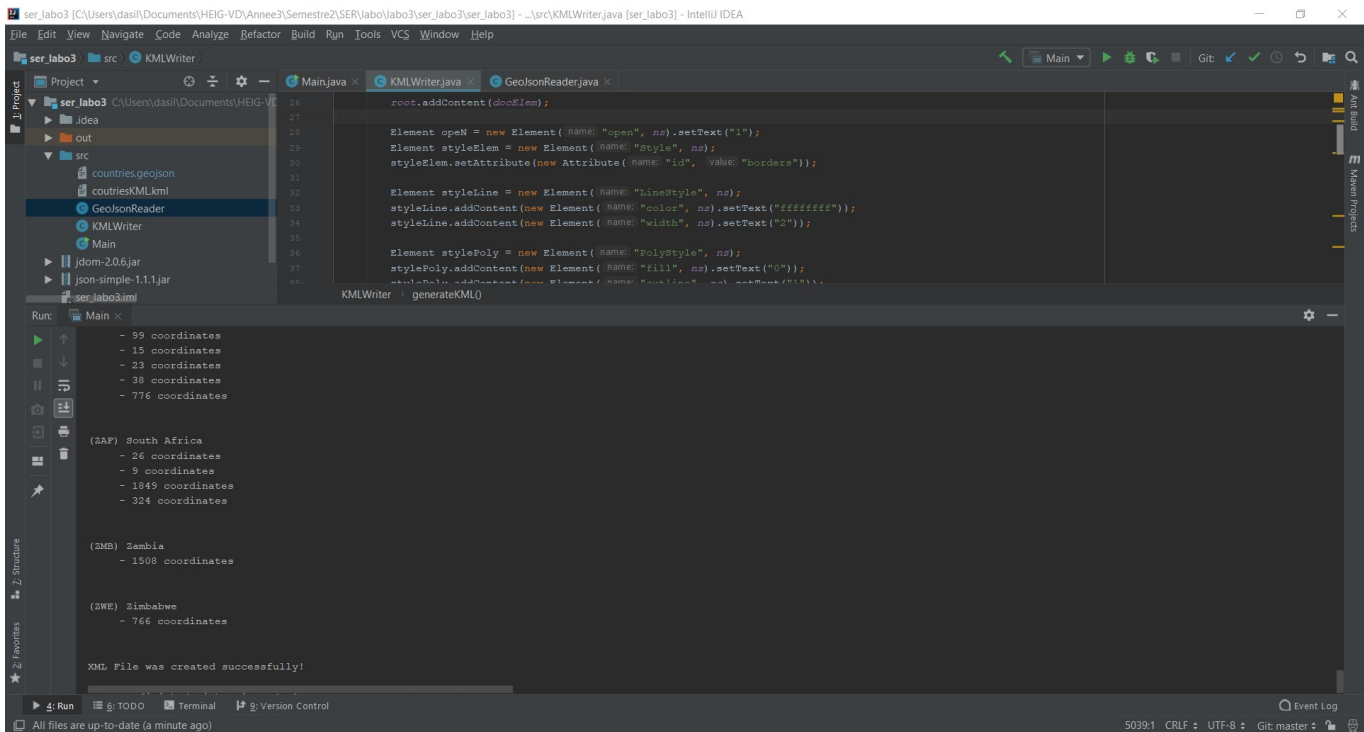
## Problèmes connus

---

Nous sommes parvenus à rendre un programme sans aucuns problème connus restant !

## Parsing GeoJson

Voici une partie des logs que nous pouvons voir lors du parsing du fichier `geojson` et la création du fichier `km1` par notre programme Java :



The screenshot shows the IntelliJ IDEA IDE with the `KMLWriter.java` file open. The code defines a `generateKML()` method that creates a KML document with a `Document` root, an `Open` element, a `Style` element with a `color` attribute, and a `Placemark` element with a `LineStyle` and a `Style` element. The `Run` console shows the output of the `generateKML()` method, indicating that the XML file was created successfully.

```
26 root.addContent(docElem);
27
28 Element open = new Element(RNAME "open", ns).setText("1");
29 Element styleElem = new Element(RNAME "Style", ns);
30 styleElem.setAttribute(new Attribute(RNAME "id", ns, "borders"));
31
32 Element styleLine = new Element(RNAME "LineStyle", ns);
33 styleLine.addContent(new Element(RNAME "color", ns).setText("EEEEEE"));
34 styleLine.addContent(new Element(RNAME "width", ns).setText("2"));
35
36 Element stylePoly = new Element(RNAME "PolyStyle", ns);
37 stylePoly.addContent(new Element(RNAME "fill", ns).setText("0"));
38
39 KMLWriter.generateKML()
```

Run: Main x

- 99 coordinates
- 15 coordinates
- 23 coordinates
- 38 coordinates
- 776 coordinates

(ZAF) South Africa

- 26 coordinates
- 9 coordinates
- 1849 coordinates
- 324 coordinates

(ZMB) Zambia

- 1508 coordinates

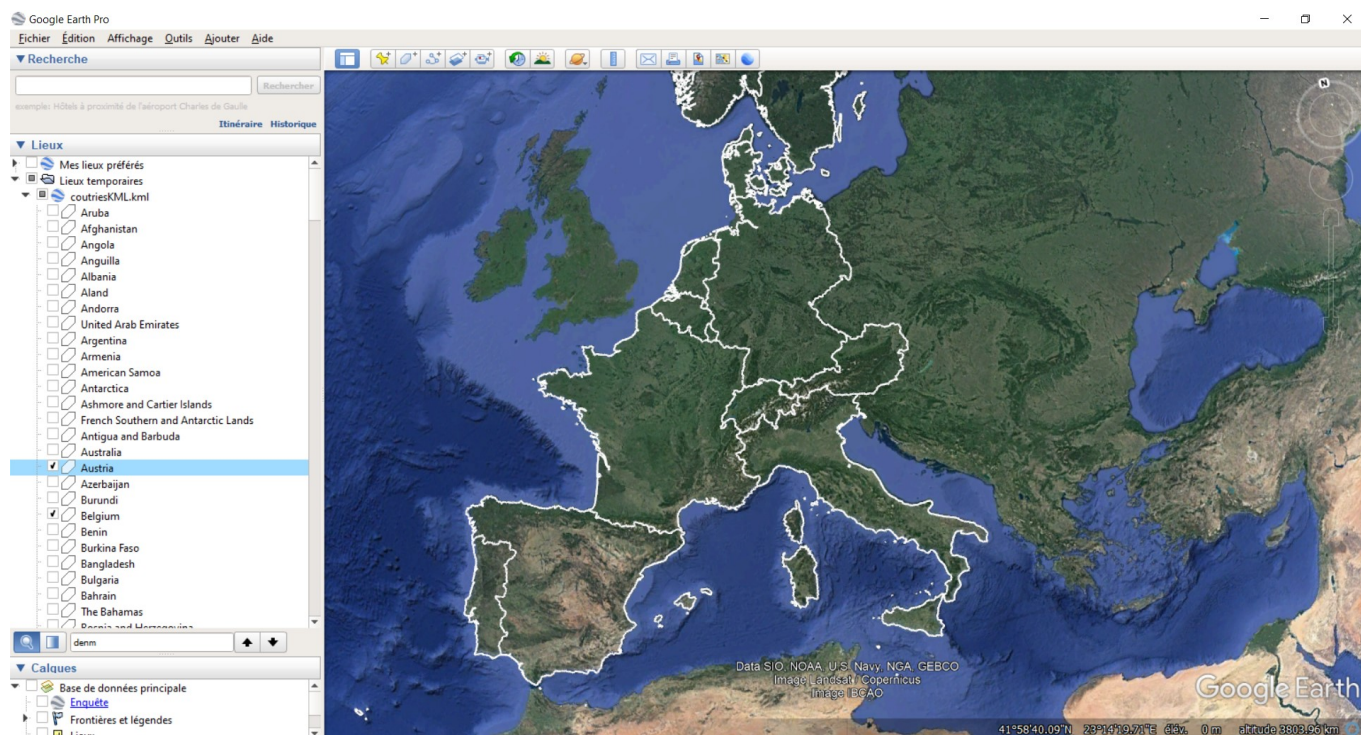
(ZWE) Zimbabwe

- 766 coordinates

XML File was created successfully!

## Test dans Google Earth

Voici une démonstration après l'import de notre fichier dans Google Earth Pro avec la sélection demandée.



## Apprentissage

Dans ce laboratoire, nous avons dû prendre connaissance de deux nouveaux format :

- GeoJson
- KML

## Conclusion

Le fait de réaliser un projet dont nous pouvons voir un résultat concret, nous a fait prendre conscience à quel point ces types de fichiers peuvent être utiles lors de la manipulation de données en grandes quantités.

Nous avons réussi à nous partager le travail équitablement ce qui a fortement contribué à la réalisation de ce laboratoire sans rencontrer de problèmes en restant très efficaces.