

# TEST2 : APPLICATION SERVEUR

Zayed Herma



**sysnav**  
NAVIGATION TECHNOLOGIES

## 1 Présentation du problème

Une application serveur récupère des trajectoires d'un véhicule sous la forme de fichiers **JSON** que l'on souhaite convertir au format **KML**, de deux manières en utilisant **python** et **php**.

## 2 Utilisation en ligne de commande

Les codes **php** et **python** sont dans le répertoire **test2**. Les fichiers **kml** sont dans le répertoire **test2/KML**. Vous pouvez les supprimer et les générer en utilisant l'une des méthodes suivantes :

### 2.1 Python

Pour convertir un fichier **json** en fichier **kml** en ligne de commande, le fichier **json** doit se trouver dans le répertoire **test2/trajectories**. Une fois entré au fichier **test2** à travers la ligne de commande, la conversion se fait en exécutant la ligne de commande **python test2.py trajectories/nomjsonfile.json**. Le fichier **kml** correspondant se trouvera ainsi dans le repértoire **kml** sous le nom **nomjsonfile.kml**.

L'exemple suivant illustre la commande précédente :

```
(base) zayed@MacBook-Pro-de-Zayed:~/test2% python test2.py trajectories/traj1.json
```

A la suite de cette commande de l'exemple, le fichier **kml** correspondant se trouvera dans le répertoire **test2/KML** sous le nom de **traj1.kml**.

### 2.2 Php

L'exécution en ligne de commande est la même qu'avec **python** en remplaçant **test.py** par **test.php** et la commande **python** par la commande **php**. Le résultat est alors identique avec celui obtenu avec **python**.

## 3 Présentation de la solution

### 3.1 bibliothèques et outils

#### — Python

- **Le module json** : Pour lire le contenu du fichier **json**.
- **Le module sys** : Pour lire les arguments donnés en ligne de commande.

#### — Php :

- **\$argv** : qui représente le tableau des arguments passés au script en ligne de commande.
- **json\_decode** : pour lire le fichier **json**.

### 3.2 Mise en œuvre et implémentation

L'idée est de parcourir le fichier **json** et de remplir le fichier **KML** par des balises type **KML** définissant notre trajectoire en utilisant la méthode **Write** pour **python** et **fwrite** pour **php** (voir **test2.py**, ligne 57 et **test2.php**, ligne 52).

## 4 Processus de validation des deux solutions

Le processus de validation est identique pour les deux solutions, et consiste à passer les fichiers **kml** dans google\_earth et voir si les résultats sont identiques avec ceux du **test1**.

### 4.0.1 Illustration

Après avoir passé les fichiers **kml** obtenus dans google\_earth, voici les résultats de visualisation obtenus pour les différentes trajectoires.

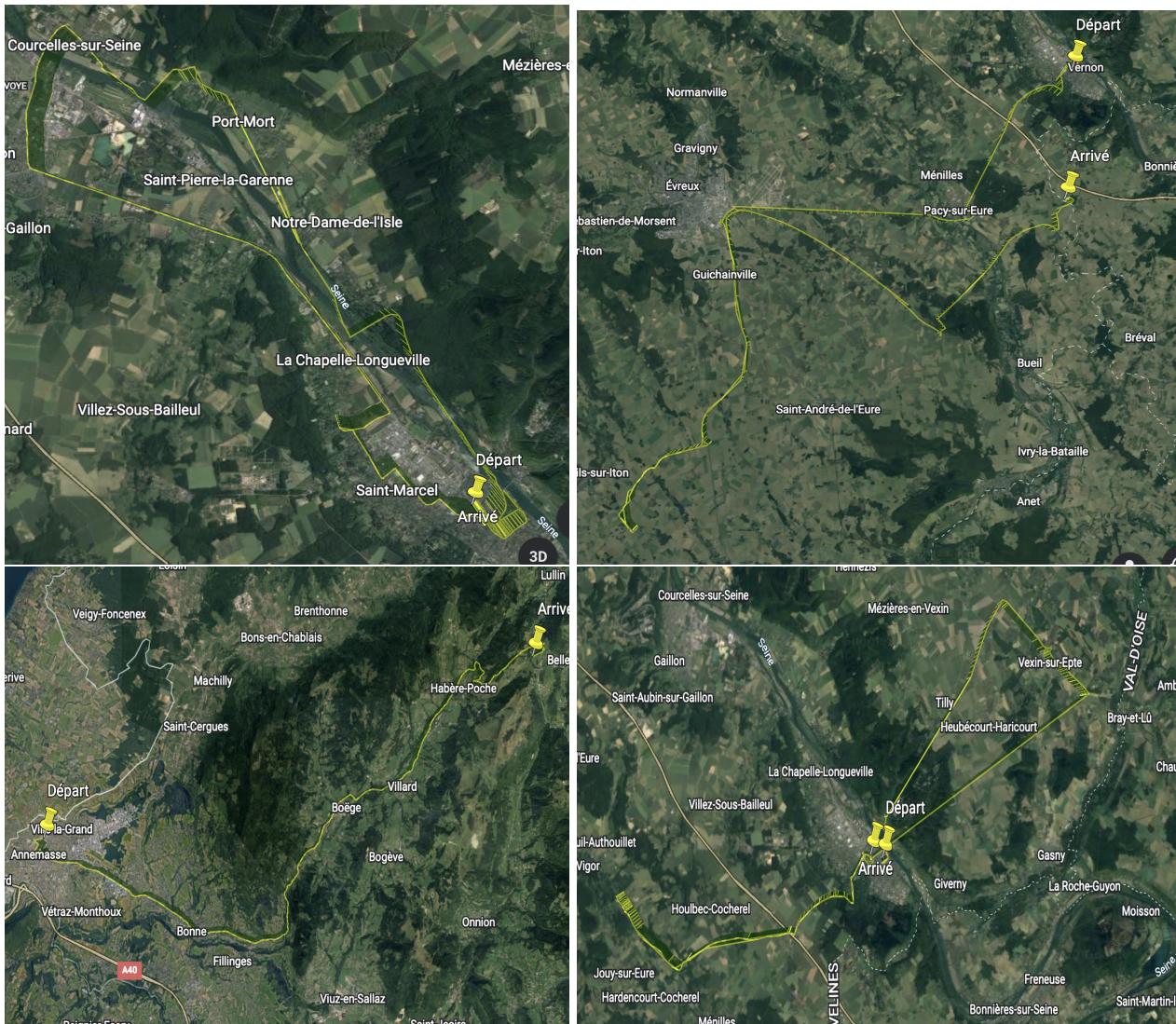


FIGURE 1 – traj1\_traj2\_traj3\_traj4

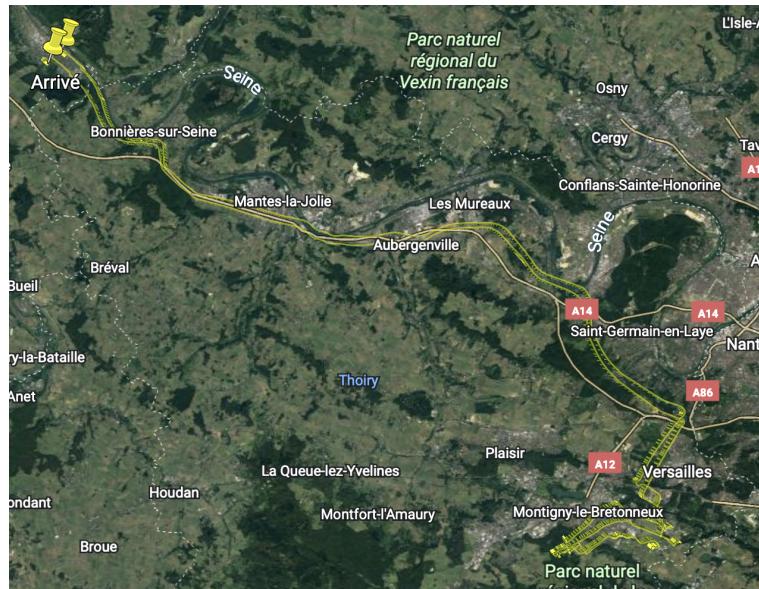


FIGURE 2 – traj5

## 5 Le temps passé sur ce test

Le temps passé sur ce test est environ 5 heures.