



Documentation Installeur Désinstalleur

Projet d'intégration 2015

Ahmed El Moden

v1.0

Contenu

I-Serveur Python.....	3
1-Présentation	3
2-Architecture BDD	3
3-Bibliothèques utilisées	4
II-Installeur/Désinstalleur.....	4
1-Procédure d'installation.....	4
2-Procédure de désinstallation	4
III-Problèmes rencontrés.....	5

I-Serveur Python

1-Présentation

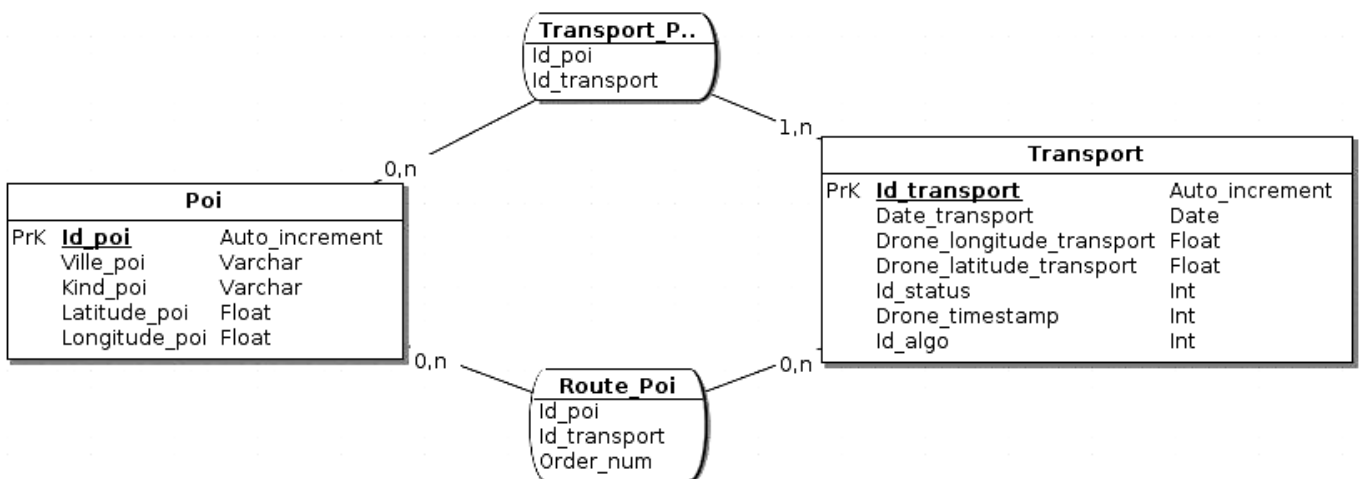
Le serveur python est un webservice qui doit répondre à des requêtes HTTP. Il gère l'enregistrement des informations relatives aux trajets et aux POI dans une base de données.

Pour ce faire la librairie Flask a été employée. Joint à cela, sqlite a été utilisé comme BDD pour des raisons de simplicité et pour avoir une persistance des informations.

Le serveur répond à plusieurs requêtes, la spécification concernant ces routes peut être consultée à cette adresse.

2-Architecture BDD

La base de données suit le modèle-relationnel suivant :



Il existe un nombre prédéfinis de Poi mais on peut en créer à volonté. Toutefois ils sont unique. La table Poi contient les informations sur le nom de la ville, les coordonnées GPS de cette dernière et le type de Poi (city, country, etc..).

Pour ajouter un transport, il faut rajouter une entrée dans la table Transport avec le status correspondant, l'algorithme demandé (Aléatoire, Alpha, Chemin le plus rapide).

La table Transport_Poi fait le lien entre un transports et des Poi de manière à avoir des Poi uniques mais qui peuvent appartenir à plusieurs Transports.

Un transport contient des Poi mais n'est pas ordonné : n'indique pas dans quel ordre on doit suivre les Poi. Pour remédier à cela, il y'a une table Route_Poi qui elle ressemble à la table Transport_Poi mais qui en plus rajoute l'ordre des Poi dans une colonne Order_num.

3-Bibliothèques utilisées

Concernant les bibliothèques utilisées, il a fallu utiliser sqlite qui est par défaut inclus dans python3 et permet de gérer une base de données au travers d'un fichier local.

Pour parser un fichier KML qui en essence est un fichier XML particulier, il a fallu utiliser une bibliothèque nommé lxml et qui permet de parcourir un fichier XML. Cette bibliothèque a la particularité d'accéder aux balises XML en utilisant la namespace.

II-Installeur/Désinstalleur

L'installateur permet de mettre en place le serveur python et le programme C avec les fichiers html dans un dossier static pour pouvoir fournir la page Web aux clients web.

Il a été choisi d'utiliser un script shell pour sa robustesse et sa flexibilité, tout en étant simple à créer.

1-Procédure d'installation

L'installateur vérifie et installe les paquets suivants dans l'ordre :

- **Wget** : Permet de télécharger des fichiers en ligne. Ici utilisé pour installer pip si nécessaire
- **Python** : Interpréteur python
- **Flask** : Webservice en python pour répondre aux requêtes HTTP
- **Lxml** : Bibliothèque pour traiter un fichier KML

L'installateur est accompagné d'une archive qu'il décompresse. Cette archive contient le serveur python et le programme C compilé, ainsi que tous les fichiers web.

2-Procédure de désinstallation

Le désinstalleur procède similairement. Il exécute les tâches suivantes dans l'ordre qui suit :

- Suppression du serveur python
- Suppression du serveur C
- Désinstallation de Flask
- Désinstallation de lxml

III-Problèmes rencontrés

- Parser un fichier KML en utilisant lxml : En effet pour lire correctement un fichier KML avec cette bibliothèque j'ai dû spécifier le namespace avant chaque balise.

C'est à dire qu'un fichier kml/xml comme le suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<kml xmlns="https://www.opengis.net/kml/2.2">
```

```
<kml>
```

```
    <example>
```

```
        <foo>
```

```
    </foo>
```

```
    </example>
```

```
</kml>
```

Au lieu d'accéder à la balise foo de la manière suivante :

kml/example/foo

Il faut ajouter le namespace (ici : <https://www.opengis.net/kml/2.2>) devant chaque balise :

```
{https://www.opengis.net/kml/2.2}kml/{https://www.opengis.net/kml/2.2}example/{https://www.opengis.net/kml/2.2}foo
```

- Exécuter le serveur proprement sans avoir une erreur de permission, pour cela il a fallu ajouter un chmod 755 au serveur et au programme C