Table des matières

[1. Introduction 2](#intro)

[2. API-Velo 2](#_Toc42177118)

[3. API-TISSEO 3](#_Toc42177119)

[4. API-Carrefours\_Feux\_Sonores 5](#_Toc42177120)

[5. API-LaPoste 6](#_Toc42177121)

## Introduction

Nous avons utilisé 4 API afin de récupérer des données.

Ces API portent sur les réseaux API-Velo, API-Tisseo, API- Carrefours\_Feux\_Sonores et API-LaPoste. Parmi ceux-ci, les API Vélo et Tisseo sont en temps réel tandis que les deux autres sont renouvelés occasionnellement.

On a codé tous les scripts avec Python qui nous ont permis d’obtenir des données sur deux formats : csv ou json, avec la possibilité de choisir le format désiré. Sur nos scripts, le « timer » permet d’exécuter une tâche dans une durée prédéfinie, et s’arrête après une durée d’exécution. Par exemple, on peut paramétrer l’API Velo pour qu’elle s’exécute toutes les 5 minutes, qu’elle enregistre des données en format csv et qu’elle s’arrête après une heure.

Vous trouverez les détails de manipulation de nos scripts par la suite.   
Nous utilisons Vscode 2017 comme IDE.

## API-Velo

1. Utilisation

Il faut changer le code dans le rectangle rouge aux lignes 40 et 48 en précisant le chemin d’accès où sont conservées les données de l’API dans votre machine.





Ici, il faut changer le code dans le rectangle rouge de la ligne 57, afin de définir la fréquence d’exécution, en secondes.



Il est possible de choisir le format de donnée souhaité, dans le rectangle rouge de la ligne 60, avec comme formats « csv » ou « json ».



Enfin, dans le rectangle rouge de la ligne 63, vous pouvez définir le temps après lequel le script s’arrête.



1. Exemple des données

Une image contenant ordinateur

Description générée automatiquement

## API-TISSEO

Cet algorithme contient deux sous-algorithmes, «  horaires.py » et « LigneStation.py ».

1. LigneStation.py

« horaires.py » permet de récupérer toutes les stations de tramway et bus à Toulouse en format json, afin de faciliter son utilisation.

Il faut changer le code dans le rectangle rouge à la ligne 68 en précisant le chemin d’accès où sont conservées les données de l’API dans votre machine.



Exemple des données

Une image contenant texte, assis, eau, tenant

Description générée automatiquement

1. horaires.py
2. Utilisation

Il faut changer le code dans le rectangle rouge à la ligne 11 en précisant le chemin d’accès où sont conservées les données de LigneStation.py dans votre machine.



Nous avons testé avec 2 stations. Ainsi, si vous souhaitez obtenir des données de toutes les stations, il faut changer en ’len(allstops)’



Ici, il faut changer le code dans le rectangle rouge de la ligne 84, afin de définir la fréquence d’exécution, en secondes.



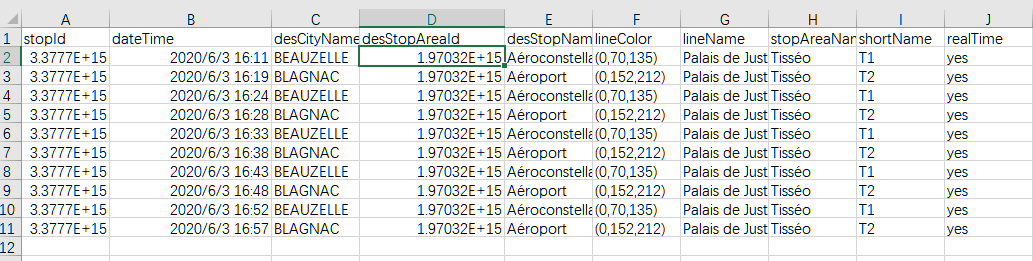
Il faut choisir le format de donnée souhaité, dans le rectangle rouge, avec comme formats « csv » ou « json ».



dans le rectangle rouge de la ligne 90, vous pouvez définir le temps après lequel le script s’arrête.



1. Example des données



## API-Carrefours\_Feux\_Sonores

1. Utilisation

Il faut changer le code dans le rectangle rouge aux lignes 36 et 43 en précisant le chemin d’accès où sont conservées les données de l’API dans votre machine.





Ici, il faut changer le code dans le rectangle rouge de la ligne 51, afin de définir la fréquence d’exécution, en secondes.



Il faut choisir le format de donnée souhaité, dans le rectangle rouge, avec comme formats « csv » ou « json ».



Dans le rectangle rouge de la ligne 90, vous pouvez définir le temps après lequel le script s’arrête.



1. Example des données

Une image contenant texte

Description générée automatiquement

## API-LaPoste

1. Utilisation

Il faut changer le code dans le rectangle rouge aux lignes 47 et 54 en précisant le chemin d’accès où sont conservées les données de l’API dans votre machine.





Ici, il faut changer le code dans le rectangle rouge de la ligne 61, afin de définir la fréquence d’exécution, en secondes.



Il est possible de choisir le format de donnée souhaité, dans le rectangle rouge de la ligne 64, avec comme formats « csv » ou « json ».



Dans le rectangle rouge de la ligne 67, vous pouvez définir le temps après lequel le script s’arrête.



1. Example des données

Une image contenant capture d’écran

Description générée automatiquement