

# Etude de l'utilisation des parkings de la ville de Montpellier

---

A noter: J'ai finalement réalisé cette SAE seul, puisque Raid Neghouche a été absent à plusieurs séances c'est donc pour cela que j'ai préféré travailler seul pour pas me pénaliser

## **Lien du dépôt Github:**

<https://github.com/adrienvalles/SAE15>

---

## Préparation du mini projet

Pour ce faire, j'ai utilisé les données extraites du site **open data de Montpellier** qui m'ont permis d'analyser les données .

Avant de pouvoir réaliser le mini projet, j'ai eu une séance TP afin de me préparer au mieux au traitement des données du site opendata. Pour pouvoir récupérer ces données je me suis appuyé sur les premières questions du TP qui m'a beaucoup aidé, en demandant de tester le programme.

En effet ce programme va grâce à la fonction requests, envoyer une demande http vers le site en question afin d'extraire les données pour les afficher dans un fichier texte contenant toutes les informations pour tous les parkings voiture de la Ville de Montpellier

```
PAR=[ 'FR_MTP_ANTI', 'FR_MTP_COME', 'FR_MTP_CORU', 'FR_MTP_EURO', 'FR_MTP_FOCH', 'FR_MTP_GAMB', 'FR_MTP_GARE', 'FR_MTP_TRIA', 'FR_MTP_ARCT', 'FR_MTP_PITO', 'FR_MTP_CIRC', 'FR_MTP_SABI', 'FR_MTP_GARC', 'FR_MTP_SABL', 'FR_MTP_MOSS', 'FR_STJ_SJLC', 'FR_MTP_MEDC', 'FR_MTP_OCCI', 'FR_CAS_VICA', 'FR_MTP_GA109', 'FR_MTP_GA250', 'FR_CAS_CDGA', 'FR_MTP_ARCE', 'FR_MTP_POLY' ]

for i in PAR:

    x=requests.get(f'https://data.montpellier3m.fr/sites/default/files/ressources/{i}.xml')
    print(x.text)
```

---

## Analyse de données

***Ceci est un programme qui permet le suivi de l'occupation de tous les parkings de Montpellier et en affichant à la fin le pourcentage de places libres et de places occupées de toute la ville.***

```
import requests
```

```

from lxml import etree
PAR=
['FR_MTP_ANTI', 'FR_MTP_COME', 'FR_MTP_CORU', 'FR_MTP_EURO', 'FR_MTP_FOCH', 'FR_MTP_GAM
B', 'FR_MTP_GARE', 'FR_MTP_TRIA', 'FR_MTP_ARCT',
'FR_MTP_PITO', 'FR_MTP_CIRC', 'FR_MTP_SABI', 'FR_MTP_GARC', 'FR_MTP_SABL', 'FR_MTP_MOSS
', 'FR_STJ_SJLC', 'FR_MTP_MEDC', 'FR_MTP_OCCI', 'FR_CAS_VICA', 'FR_MTP_GA109', 'FR_MTP_G
A250', 'FR_CAS_CDGA', 'FR_MTP_ARCE', 'FR_MTP_POLY']

y=0
w=0
p=0
for i in PAR:

x=requests.get(f'https://data.montpellier3m.fr/sites/default/files/ressources/{i}.
xml')
    f1=open(f'{i}.txt',"w", encoding='utf8')
    f1.write(x.text)
    f1.close()
    tree = etree.parse(f"{i}.txt")    #retirer les donnees
    for user in tree.xpath("Name"):    #extraire les données de name
        print('Nom du parking :',user.text)
    for user1 in tree.xpath("Total"):
        print('Places totales :',user1.text)
    for user2 in tree.xpath("Free"):
        print('Nombre de places libres :',user2.text)
        x=int((user1.text))-int((user2.text))
        print('Nombres de places occupées:' ,x)
    y=int((user1.text))+ y
    w=int((user2.text))+w
    p=int((x))+ p
#print('Nombres de places occupées dans toute la ville:' ,x)
print('Nombre de Places totales de toute la ville:' ,y)
print('Nombres de places libres de toute la ville:' ,w)
print('Nombre de place occupées dans toute la ville:' ,p)
pour=int((w))*100/int(y)
opour=int((p))*100/int(y)
print('Pourcentage de places libres de toute la ville' ,round(pour,2) ,'%')
print('Pourcentage de places occupées de toute la ville' ,round(opour,2) ,'%')

```

J'ai par la suite pris l'exemple de 4 parkings afin de connaître leur pourcentages de places libres ainsi que leur taux d'occupation sur une journée pour voir l'évolution de chaque parkings et pour vérifier si ils ont la même fréquentation.

***Voici le programme qui a calculé le nombres de places libres et par la suite en ajoutant une fonction , calculer un pourcentage et en déduire le taux d'occupation de chacun des 4 parkings.***

```

"""php import requests #librairie HTTP import time from lxml import etree

```

```
parkings=
["FR_MTP_ANTI","FR_MTP_COME","FR_MTP_CORU","FR_MTP_EURO","FR_MTP_FOCH","FR_MTP_GAMB","FR_MT
P_GARE","FR_MTP_TRIA","FR_MTP_ARCT","FR_MTP_PITO",
"FR_MTP_CIRC","FR_MTP_SABI","FR_MTP_GARC","FR_MTP_MOSS","FR_STJ_SJLC","FR_MTP_MEDC","FR_MTP_OC
CI","FR_CAS_VICA","FR_MTP_GA109", "FR_MTP_GA250","FR_CAS_CDGA","FR_MTP_ARCE"] #Parkings de
Montpellier
```

```
url="https://data.montpellier3m.fr/dataset/disponibilite-des-places-dans-les-parkings-de-montpellier-
mediterranee-metropole" #lien du site open data montpellier
```

```
num=17
```

```
for j in range(num):#Nombre de fois où la boucle va se répéter for i in parkings:
response=requests.get("https://data.montpellier3m.fr/sites/default/files/ressources/"+i+".xml") #Récupérer
tout les liens de chaque parking de montpellier f1=open(i,"w", encoding='utf8') #Ouvre un fichier dans lequel
toutes données seront sauvegardées et écrasées à chaque mise à jour des données f1.write(response.text)
f1.close() f2=open("resultats.text","a",encoding='utf8') #Ouvre un fichier dans lequel les données seront
sauvegardées, et non écrasées à chaque tree = etree.parse(i) for user in tree.xpath("Date_time"):
time=user.text f2.write('\n') #('\n') : sert à revenir à la ligne f2.write('Date:') f2.write(user.text) # Ecriture de la
date dans le fichier f2.write('\n')
```

```
    for user in tree.xpath("Name"):
        print('Nom du parking :',user.text) #Ici Nous avons utilisé un print qui
me permet de voir directement dans la console si le code marche, au lieu d'aller
vérifier les fichiers à chaque fois que le programme est en marche
        f2.write('Parking :')
        f2.write(user.text) #Afficher le nom du parking sur le fichier
        f2.write('\n')
        #Afficher le nom du parking

    for user in tree.xpath("Total"):
        f2.write('Nombre total de places:')
        f2.write(user.text) #Afficher le nombre de total de places du parking
        f2.write('\n')
        total=int(user.text) #Valeur nombre de places totales afin de calculer le
pourcentage

    for user in tree.xpath("Free"):
        f2.write('Nombre de places libres :')
        f2.write(user.text) #Afficher le nombre de places libres sur le fichier
        f2.write('\n')
        free=int(user.text) #Valeur places libres afin de calculer le pourcentage
        pourcentage_l=round((free*100)/total,2)
        f2.write('Le pourcentage de places libres est de :')
        f2.write(str(pourcentage_l)) #Affichage du pourcentage de places lbres
        f2.write("%")
        f2.write('\n')
        f2.write('\n')
        pourcentage_o=100-(pourcentage_l*100)/total
```

```

        f2.write('Le pourcentage de places occupées est de :')
        f2.write(str(pourcentage_o)) #Affichage du taux d'occupation
        f2.write("%")
        f2.write('\n')
        f2.write('\n')

f2.write('\n')
f2.write('\n') #Sauter des lignes afin de rendre le fichier lisible
f2.write('\n')
f2.write(' Programme en pause') # Me permet de me repérer sur le fichier puisqu'il
ya beaucoup de données qui ont été récupérées
f2.write('\n')
f2.write('\n')
f2.write('\n')

time.sleep(3600)# fonction qui permet d'endormir le programme selon une durée
déterminée

```

<br>  
<br>  
J'ai voulu ensuite analyser toutes les données pour tous les parkings de Montpellier en calculant le taux d'occupation total de la ville  
Pour cela j'ai ajouter dans une liste tous les noms de parkings puis j'ai fais une fonction qui affiche le taux d'occupation des parkings de Monteppllier.

\*\*\*Voici mon programme avec les différentes fonctions commentées et leur signification\*\*\*

```

` `` `php

```

```

import requests
from lxml import etree
import time

```

```

parkings=
['FR_MTP_ANTI', 'FR_MTP_COME', 'FR_MTP_CORU', 'FR_MTP_EURO', 'FR_MTP_FOCH', 'FR_MTP_GAM
B', 'FR_MTP_GARE',

'FR_MTP_TRIA', 'FR_MTP_ARCT', 'FR_MTP_PITO', 'FR_MTP_CIRC', 'FR_MTP_SABI', 'FR_MTP_GARC
', 'FR_MTP_SABL',

'FR_MTP_MOSS', 'FR_STJ_SJLC', 'FR_MTP_MEDC', 'FR_MTP_OCCI', 'FR_CAS_VICA', 'FR_MTP_GA10
9', 'FR_MTP_GA250',

```

```

'FR_CAS_CDGA', 'FR_MTP_ARCE', 'FR_MTP_POLY']

Toville=0 #Le nombre total des parkings voituresde toute la ville
FrVille=0 #Le nombre total des parkings voitures libres de toute la ville
Nom=input("Nom du fichier:")# indiquer le nom du fichier
periode=int(input("Période(min):"))
periode=periode*60 # cela indique la durée de l'acquisition du programme en
minutes . Par exemple si je tape période= 2 , la durée sera de 120min c'est à dire
2 heures car ici on multiplie par 60
duree=int(input("durée(sec):"))
t=60 # t c'est le nombre de fois que le programme va se répéter .

for p in range(t):
    for i in parkings: #la liste "parkings" contient les noms des fichiers de
chaque zone. On fait un boucle pour récupérer des données de chaque zone.
#Récupérer les données et les mettre dans un nouveau fichier

data=requests.get(f"https://data.montpellier3m.fr/sites/default/files/ressources/{
i}.xml")
    f1=open(f"{i}.txt","w",encoding='utf8') #Cette fonction va ouvrir un
fichier texte conteant les informations de tous les parkings de Montpellier
    f1.write(data.text)
    f1.close()
    #trier ces données et choisir le nombre des places libres et des places
totales.
    tree=tree.parse(f"{i}.txt")
    a=0
    b=0
    for user in tree.xpath("Total"):
        total=int(user.text)
        a=a+total
    for user in tree.xpath("Free"):
        libre=int(user.text)
        b=b+libre
    # ajouter des nombrse dans la valeur du "nombre total". C'est pour obtenir
la somme de tous les parkings et celle de tous les parkings libres
    Toville=Toville+a
    FrVille=FrVille+b
    PVoiture=FrVille/Toville # la valeur : Parkings libres/ Parkings total
    PVoiture=1-PVoiture # pour obtenir le taux d'occupation des voitures
    temps=time.time()
    temps=time.ctime(temps)
    #fonction qui va permettre d'afficher la date a côté de chaque données

    # Ici ça va permettre d'ouvrir un fichier avec le champ "nom" que l'on
souhaite donné et que l'on a spécifié au début avec la fonction f2.write qui va
permettre d'écrire dans
    # ce fichier qui stockera les données avec la date et le taux d'occupation
    f2=open(f"{Nom}.txt","a",encoding='utf8')
    f2.write(f"{temps} Le Taux d'occupation des parkings de la ville de
Montpellier est de: {round(PVoiture*100, 2)}%")
    f2.write('\n')

```

```
f2.close()
time.sleep(duree) #fonction qui suspend l'exécution en fonction du nombre de
secondes que j'ai attribué à ma fonction "duree"
```

- Pour la partie vélos j'ai réussi à extraire les données des informations des différentes stations ainsi que leur statut dans 2 fichiers texte différents mais je n'ai pas pu réussir à les analyser en calculant leur taux d'occupation ou encore leur moyenne car à la base cela devrait être mon binôme qui s'en occupé, j'ai donc assuré aussi cette partie en faisant ce que je pouvais faire seul.

***Voici donc mon sympathique programme qui récupère les données json des stations vélos de Montpellier***

```
import json
import requests
import time

# Les liens vers les données JSON des stations
status_url = "https://montpellier-fr-smooove.klervi.net/gbfs/en/station_status.json"
info_url = "https://montpellier-fr-smooove.klervi.net/gbfs/en/station_information.json"

period=int(input("Période(min):"))
period=period*60
duration=int(input("duration(sec):"))

# Récupère les données JSON des stations
response = requests.get(status_url)
status_data = response.json()

response = requests.get(info_url)
info_data = response.json()

# Stocke les données dans des fichiers différents
with open("status_data.txt", "w") as outfile:
    json.dump(status_data, outfile)

with open("info_data.txt", "w") as outfile:
    json.dump(info_data, outfile)
```

```
# Calcule le taux d'occupation des stations
total_stations = len(status_data)
occupied_stations = 0
for station in status_data:
    bikes_available = station["num_bikes_available"]
    station_id = station["station_id"]
    capacity = info_data[station_id]["capacity"]
    if bikes_available < capacity:
        occupied_stations += 1
occupancy_rate = occupied_stations / total_stations

# Affiche le taux d'occupation
print("Taux d'occupation des stations: {:.2f}%".format(occupancy_rate * 100))

time.sleep(duration)
```

---

## Traitement des données

Pour traiter les différentes données j'ai choisi d'utiliser "Gnuplot" qui est un programme de ligne de commande et d'interface graphique qui peut générer des tracés en deux et trois dimensions de fonctions, de données et d'ajustements de données. J'ai trouvé cette interface pratique et très simple d'utilisation

Voici ci-dessous les différentes lignes de commandes qui m'ont permis de tracer une courbe de données qui prend en paramètre le taux d'occupation des parkings voitures au cours du temps

```
set terminal png size 800,600
set output 'image.png'
set key inside bottom right
set autoscale
set xlabel 'temps (en heure)'
set xdata time
set timefmt "%H:%M:%S"
set format x "%H:%M"
set ylabel 'Taux d occupation des parkings voitures'
set title 'Evolution du taux d occupation des parkings de Montpellier'
plot "Data_voitures.txt" using 1:2 with linepoints
```