TD5

TD5

Exercice 1

1

mkdir td4

cd td4

2

git init

3

sudo apt-get update

sudo apt-get install python3-pip

4

sudo apt-get install virtualenv

5

virtualenv .env

6

source .env/bin/activate

7

pip list

8

git status

9

echo ".env/" > .gitignore

10

git status

11

git add .gitignore

git commit -m "Initial commit with .gitignore"

git status

EX2

import requests

nano derbyshire\_place\_ids.py

def get\_place\_ids():

url = "https://opendomesday.org/api/1.0/county/derbyshire/"

response = requests.get(url)

if response.status\_code == 200:

data = response.json()

places = data["places"]

place\_ids = [place["id"] for place in places]

return place\_ids

else:

print(f"Error: {response.status\_code}")

return None

if name == "main":

place\_ids = get\_place\_ids()

print("Place IDs in Derbyshire:")

for place\_id in place\_ids:

print(place\_id)

3

git add derbyshire\_place\_ids.py

git commit -m "Add script to fetch place IDs in Derbyshire"

EX3

nano manor\_ids.py

import requests

def get\_manor\_ids(place\_id):

    url = f"https://opendomesday.org/api/1.0/place/{place\_id}/"

    response = requests.get(url)     if response.status\_code == 200:

        data = response.json()

        manor\_ids = [manor['id'] for manor in data['manors']]

        return manor\_ids

    else:

        print(f"Error: {response.status\_code}")

        return None

python manor\_ids.py

python

import manor\_ids

place\_id="20086"

output=manor\_ids.get\_manor\_ids(place\_id)

print(output)

exit()

nano manor\_ids.py

import requests

def get\_manor\_ids(place\_id):

    url = f"https://opendomesday.org/api/1.0/place/{place\_id}/"

    response = requests.get(url)     if response.status\_code == 200:

        data = response.json()

        manor\_ids = [manor['id'] for manor in data['manors']]

        return manor\_ids

    else:

        print(f"Error: {response.status\_code}")

        return None if \_\_name\_\_ == '\_\_main\_\_':

    place\_id = "20086"

    print(get\_manor\_ids(place\_id))

python manor\_ids.py

git add manor\_ids.py

git commit -m "Add script to fetch manor IDs for a given place ID"

EX4

1

def get\_derbyshire\_place\_ids():

url = "https://opendomesday.org/api/1.0/county/DERBYSHIRE/places/"

response = requests.get(url)

if response.status\_code == 200:

data = response.json()

places = data["places"]

place\_ids = [place["id"] for place in places]

return place\_ids

else:

print(f"Error: {response.status\_code}")

return None

def get\_all\_manors\_in\_derbyshire():

place\_ids = get\_derbyshire\_place\_ids()

all\_manors = []

for place\_id in place\_ids:

manors = get\_manor\_ids(place\_id)

all\_manors.extend(manors)

return all\_manors

2

def get\_manor\_data(manor\_id):

url = f"https://opendomesday.org/api/1.0/manor/{manor\_id}/"

response = requests.get(url)

if response.status\_code == 200:

data = response.json()

geld\_paid = data["geld\_value\_1086"]

total\_ploughs = data["ploughs\_total\_1086"]

return (geld\_paid, total\_ploughs)

else:

print(f"Error: {response.status\_code}")

return None

3

import pandas as pd

def create\_dataframe(manor\_ids):

manor\_data = [get\_manor\_data(manor\_id) for manor\_id in manor\_ids]

df = pd.DataFrame(manor\_data, columns=["geld\_paid", "total\_ploughs"])

return df

4

def compute\_sums():

all\_manors = get\_all\_manors\_in\_derbyshire()

df = create\_dataframe(all\_manors)

total\_geld\_paid = df["geld\_paid"].sum()

total\_ploughs\_owned = df["total\_ploughs"].sum()

return total\_geld\_paid, total\_ploughs\_owned

6

git add manor\_ids.py

git commit -m "Add functions to fetch all manors in Derbyshire and compute sums of geld paid and total ploughs owned