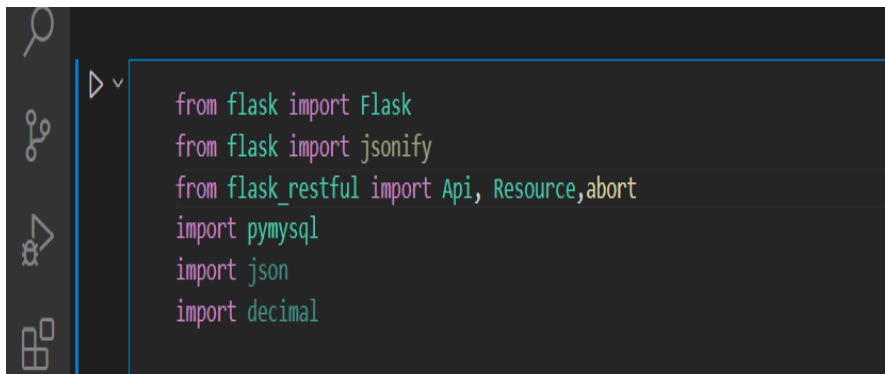


RAPPORT DU TEST API_CRUD

nous avons connecter notre api a une base de donnée Customer qui va afficher avec la méthode get les noms ,les prénoms, et les mails, les id et les stores dans notre api nous allons effectuer d

Différent opérations CRUD (Read, Upload, Delete) et vérifier si notre api retourne exactement les résultats désirer

1- IMPORTATION DES LIBRAIRIES



```
from flask import Flask
from flask import jsonify
from flask_restful import Api, Resource, abort
import pymysql
import json
import decimal
```

2-CONNECTION A LA BASE DE DONNEE



```
app = Flask(__name__)
api = Api(app)

conn = pymysql.connect(host='localhost',
                        user='root',
                        password='coucou02',
                        database='sakila',
                        charset='utf8mb4',
                        cursorclass=pymysql.cursors.DictCursor)

cursor = conn.cursor()
```

CREATION DE LA CLASSE DES METHODES

```
class Crud(Resource):

    def get(self):

        cursor.execute('select customer_id,store_id,first_name,last_name,email,active FROM customer')
        film = cursor.fetchall()

        print(film)
        if film is None:
            abort(404)
        return film

    def delete(self):
        cursor.execute('ALTER TABLE film DROP COLUMN store_id,create_date,last_update')
        film = cursor.fetchall()
        if film is None:
            abort(404)
        return film

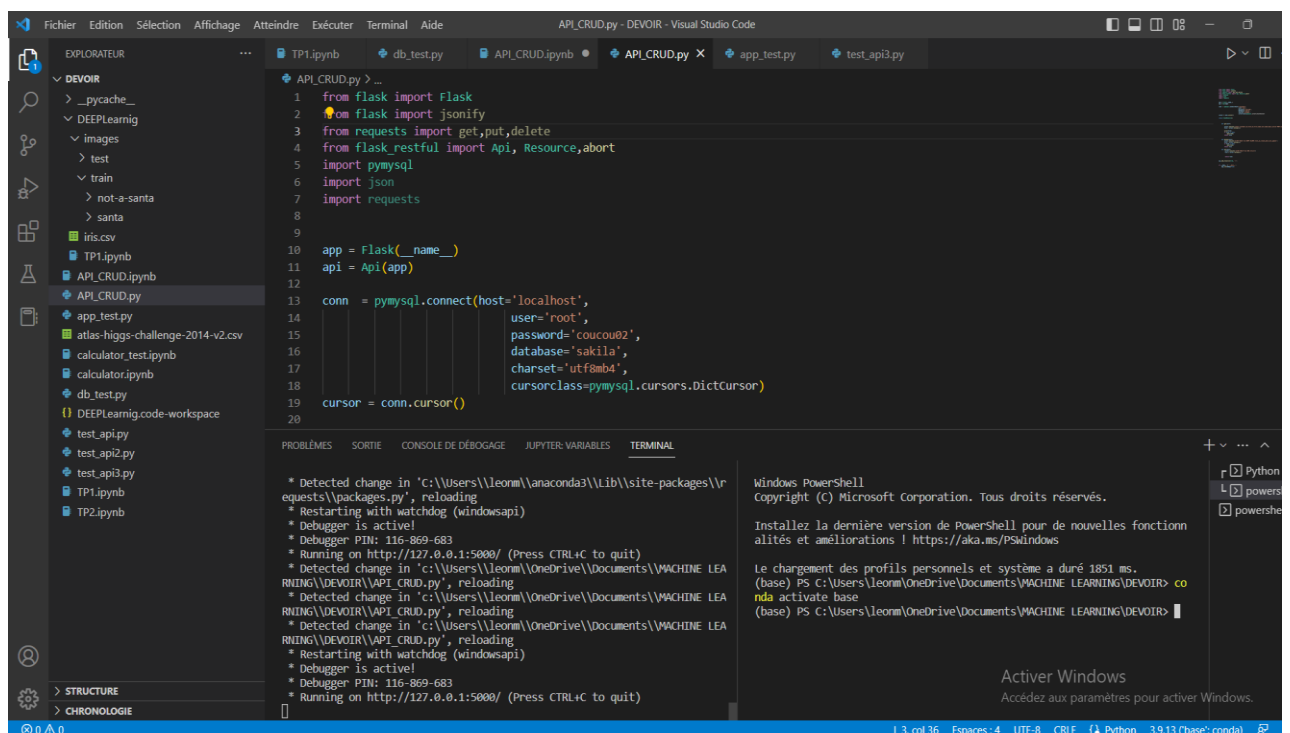
    def put(self):
        cursor.execute('ALTER TABLE film ADD article')
        film = cursor.fetchall()

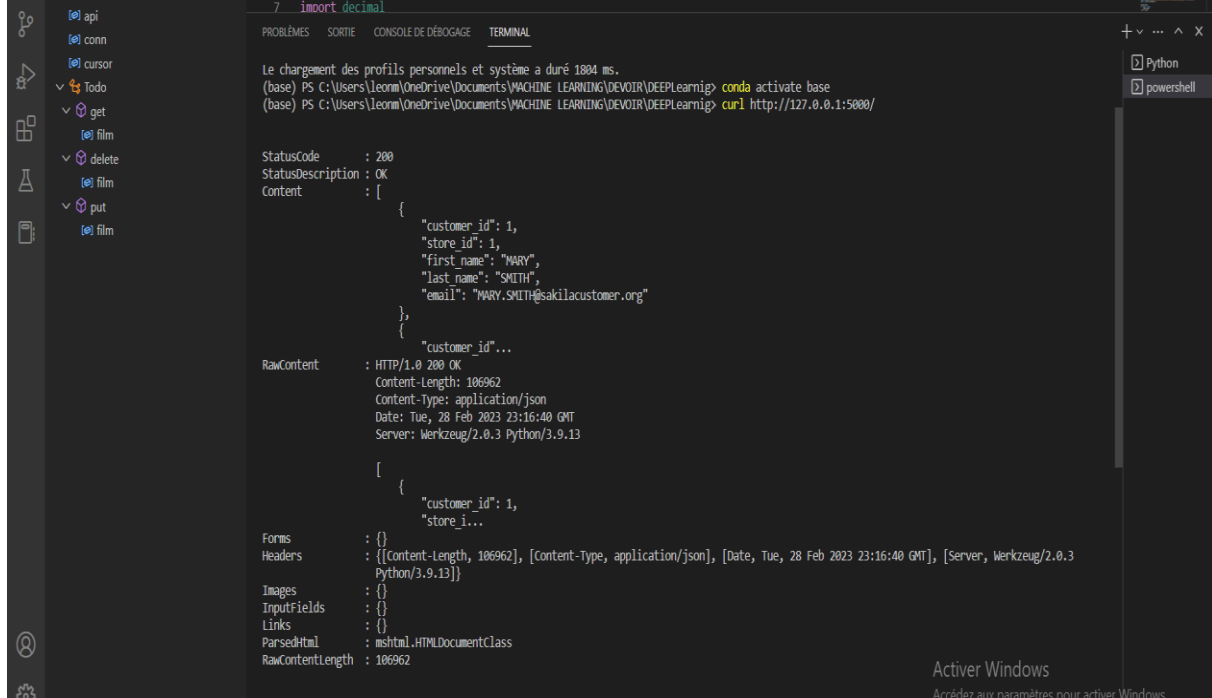
        return film

api.add_resource(Crud, '/')

if __name__ == '__main__':
    app.run(debug=True)
```

TEST DE L'API





RESULTAT SUR LE LOCALHOST

