Київський національний університет імені Тараса Шевченка Факультет комп'ютерних наук та кібернетики Кафедра системного аналізу та теорії прийняття рішень

Звіт з лабораторної роботи № 3 Реалізація простого RESTful API

Виконав студент групи К-23

Забровський В.Д.

Прийняв Махно М.Ф.

Варіант №6

Мета:

Створити простий веб-сервер, що надає RESTful API для керування певним ресурсом (за варіантом).

Індивідуальне завдання:

1. Розробка моделі даних:

Мій варінт полягає у розробці наступної моделі даних "Музичні треки": з полями: id, song_name, artist, album, release_year.

2. Реалізація НТТР-методів:

Реалізовані наступні методи:

- GET для отримання списку ресурсів або одного ресурсу за ID.
- POST для створення нового ресурсу.
- PUT для повної зміни ресурсу.
- РАТСН для часткової зміни ресурсу.
- DELETE для видалення ресурсу.

3. Розробка АРІ:

Для розробки програмного інтерфейсу було використано мову Python, зокрема фреймворк Flask(flask_restful), який широко застосовується для розробки вебзастосунків. Також було використане розширення flask_sqlalchemy для побудови бази даних для зберігання треків на базі SQLite.

4. Тестування АРІ:

Для тестування API створено test.py файл, в якому передбачено тестові функції get_track, post_track, patch_track, delete_track, put_track для тестування відповідних запитів за допомоги бібліотеки request

Коди файлів

main.py:

```
from flask sqlalchemy import SQLAlchemy
app = Flask( name )
api = Api(app)
app.config['SQLALCHEMY DATABASE URI'] = 'sqlite:///tracks.db'
db = SQLAlchemy(app)
parser = reqparse.RequestParser()
parser.add_argument("song_name", type=str, required=False)
parser.add_argument("artist", type=str, required=False)
parser.add_argument("album", type=str, required=False)
parser.add argument("release year", type=int, required=False)
     artist = db.Column(db.String(255))
     album = db.Column(db.String(255))
          self.artist = artist
          self.release year = release year
          db.session.delete(track)
```

```
"release year": new track.release year,
        track.release year = args["release year"]
def patch(self, track id):
def get(self):
            "release year": track.release year,
```

```
new track = TrackModel(
api.add resource(Track, "/api/tracks/<int:track id>")
    with app.app context():
```

test.py

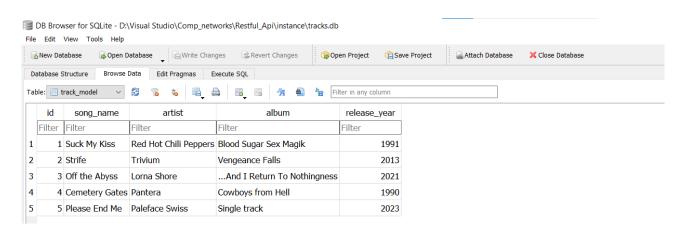
```
BASE URL = "http://127.0.0.1:3000/api/tracks"
def get track(track_id=None):
def post track(song_name, artist, album, release_year):
def put track(track id, song name=None, artist=None, album=None,
release year=None):
           data["artist"] = artist
        if release year:
```

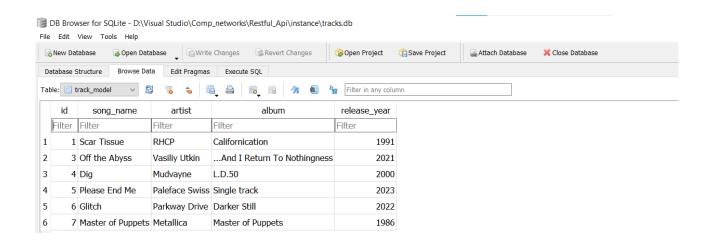
```
def patch track(track id, **kwargs):
```

```
except requests.exceptions.RequestException as e:
    print('Delete_track')
    print(f"An error occurred while making the request: {e}")
    print('-' * 20)

delete_track(2)
post_track("Glitch", 'Parkway Drive', 'Darker Still', 2022)
get_track()
put_track(1, 'Sir Psycho Sexy', 'Red Hot Chili Peppers', 'BSSM', 1991)
get_track()
patch_track(1, song_name="Scar Tissue", artist="RHCP", album='Californication')
get_track()
put_track(4, 'Dig', 'Mudvayne', 'L.D.50', 2000)
post_track("Master of Puppets", 'Metallica', 'Master of Puppets', 1986)
patch_track(3, artist='Vasiliy Utkin')
patch_track(4, album='Vulgar Display of Power', song_name='Walk',
release_year='dsdjsoiddjdid')
get_track(2)
get_track(-5)
```

Стан бази даних tracks.db після відповідних запитів:





Вивід таіп.ру у терміналі:

Put_track 4

```
MEMBER: This is a development server. Do not use it in a production deployment. Use a production BSGI server instead.

* Rounting on htts://IZA.0.1.13009

Press CTRACE to quit

* Restarting with stat

* Obbugger 19: 115-282-2449

**Obbugger 19: 115
```

```
137.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 133:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 130:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 130:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 130:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 130:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 130:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 130:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 130:03.5] "MEX / Jar/Freeze MITP/137 200 - 127.0.0.1 - [04/08/2783 130:03.5] "MEX / Jar/Freeze MITP/137 200 - [04/08/2783 130:03.5] "MEX / Jar/F
```