

Технології серверного програмного забезпечення

Лабораторна робота 2

Розробка базового REST API

Мета роботи: Розробити та протестувати базове REST API

Завдання:

- Розробити базовий REST API для веб застосунку
- Тестування REST API за допомогою postman

Практичне завдання:

Практичне завдання полягає в розробці базового REST API застосунку для обліку витрат. Застосунок повинен надавати користувачу такі можливості:

- Створення користувача
- Створення категорії витрат
- Створення запису про витрати
- Отримання списку категорій
- Отримання списку записів по певному користувачу
- Отримання списку записів в категорії для певного користувача

Приблизна структура даних, яка потрібна буде для виконання даної лабораторної:

Користувач
Id
Ім'я

Категорія
Id

Назва категорії

Запис

Id

Id користувача

Id категорії

Дата та час створення запису

Сума витрати

Шляхи ендпоінтів мають бути такими:

GET /user/<user_id>

DELETE /user/<user_id>

POST /user

GET /users

GET /category

POST /category

DELETE /category

GET /record/<record_id>

DELETE /record/<record_id>

POST /record

GET /record - повинен приймати параметрами user_id, category_id. Як окремо так і разом, і фільтрувати по ним. Без параметрів видає помилку

Вимоги до виконання:

- Виконане завдання повинно знаходитись у вигляді репозиторію в системі контролю версій(рекомендовано github, проте при великому бажанні можна і інші).
- Повинні бути структуровані коміти - репозиторій, в якому весь код залитий одним комітом з повідомленням “*Initial commit*” не годиться. Коміти повинні бути осмисленні з відповідними повідомленнями.
- В даній лабораторній використання бази даних не вітается - це мета роботи наступної лабораторної роботи, в цій лабораторній концентруємось на створенні функціоналу.
- В README файлі повинні міститись повні інструкції для запуску проекту локально.
- Проект повинен бути задеплоєний.
- Для успішної здачі лабораторної потрібно надати в classroom: посилання на репозиторій, посилання на задеплоєний проект та файл колекції postman, яким ви тестували застосунок.

Методичні рекомендації:

1. Розробіть логіку застосунку - початкові дані, робота з ними та ендпоінти які вимагає завдання. Не забувайте комітити етапи роботи!
2. Початкові дані можете зберігати просто у вигляді словника в пам'яті застосунку.
3. Приклад реалізації ендпоінтів для користувача:

```

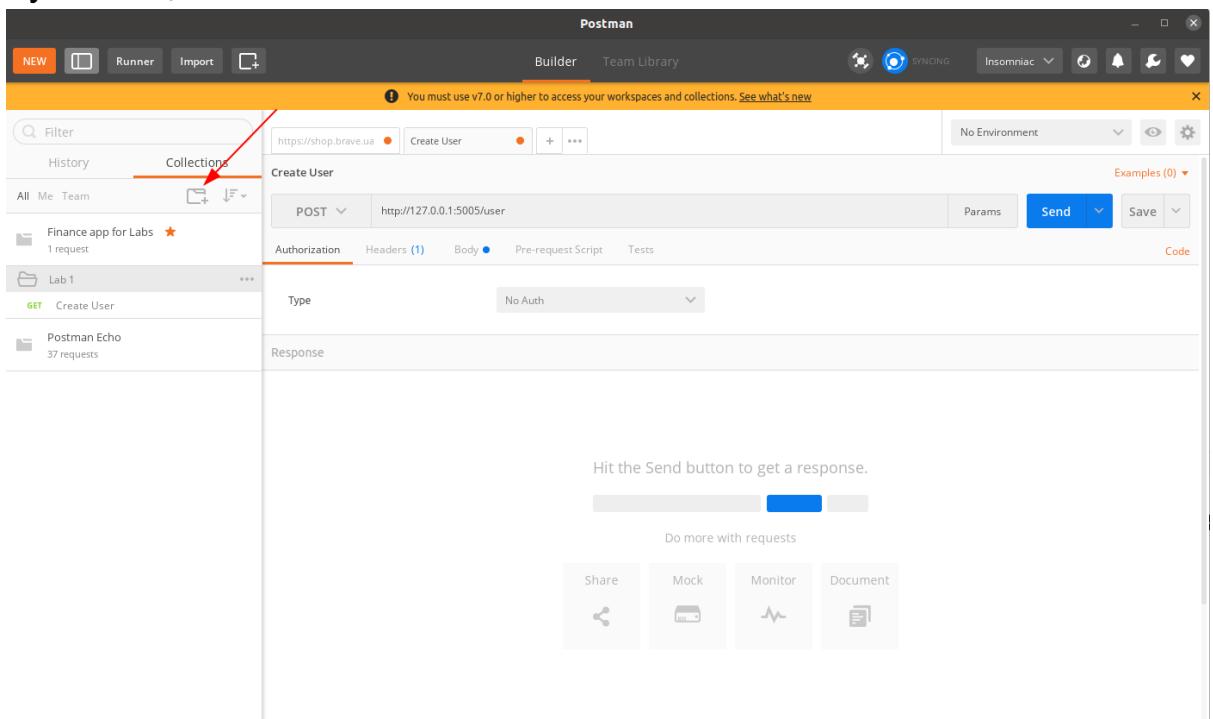
@app.post('/user')
def create_user():
    user_data = request.get_json()
    user_id = str(uuid.uuid4).hex
    user = {"id": user_id, **user_data}
    users[user_id] = user
    return user

new *
@app.get('/users')
def get_users():
    return list(users.values())

```

Валідацію параметрів окрім вказаної для GET /record шляху в цій лабораторній роботі можна не реалізовувати, це буде мета наступної роботи.

4. Для тестування ендпоїнтів, які ви розробляєте, можна сразу зробити колекцію в postman. Для цього встановіть програму(доступна на сайті <https://www.postman.com/>) та створіть нову колекцію - Collections -> New collection



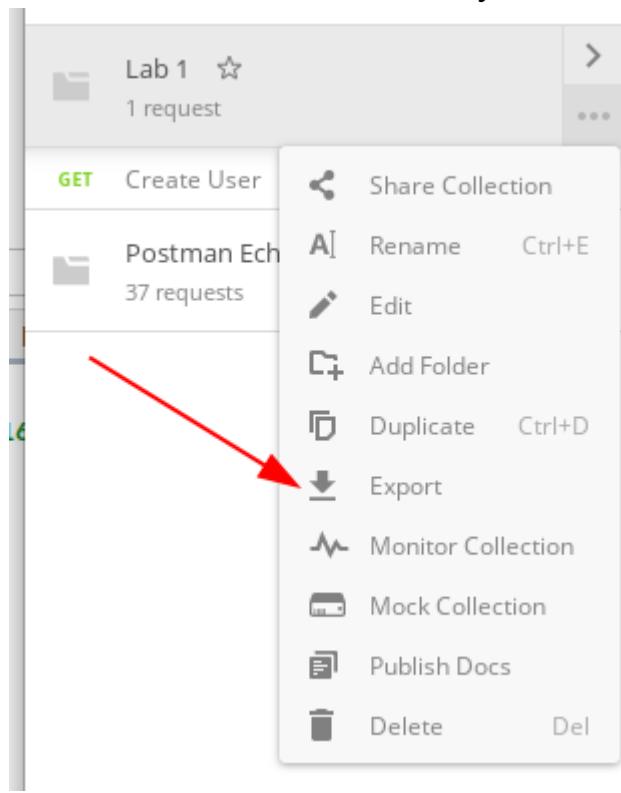
5. В колекції створіть середовище для локальної розробки де вкажіть домен який відповідає тому де запущений застосунок

локально. Одразу використовуйте цю змінну як показано на другому скріншоті

The screenshot shows two main parts of the Postman interface. At the top is a modal window titled "MANAGE ENVIRONMENTS" with a tab bar for "Manage Environments" and "Environment Templates". The "Manage Environments" tab is selected, showing an "Edit Environment" section for "Local". A table lists environment variables: "domain" with value "127.0.0.1:5005" and a new row "New key" with "Value" empty. There are "Cancel" and "Update" buttons at the bottom. Below the modal is the main Postman window. It displays a collection named "Create User" with one POST request. The request URL is "http://{{domain}}/user". The "Authorization" tab is selected, showing "Headers (1)" and "Body (1)". The "Body" tab is active, containing the JSON payload: {"name": "test", "email": "test@brave.ua", "password": "1234567890"}.

6. Запустіть ваш застосунок та перевірте що ваш ендпоїнт з минулої лабораторної працює коректно
7. Тепер залишилось додати новий environment в колекцію postman - наприклад production і записати домен, на якому задеплоєний застосунок та перевірити його роботу за допомогою колекції.

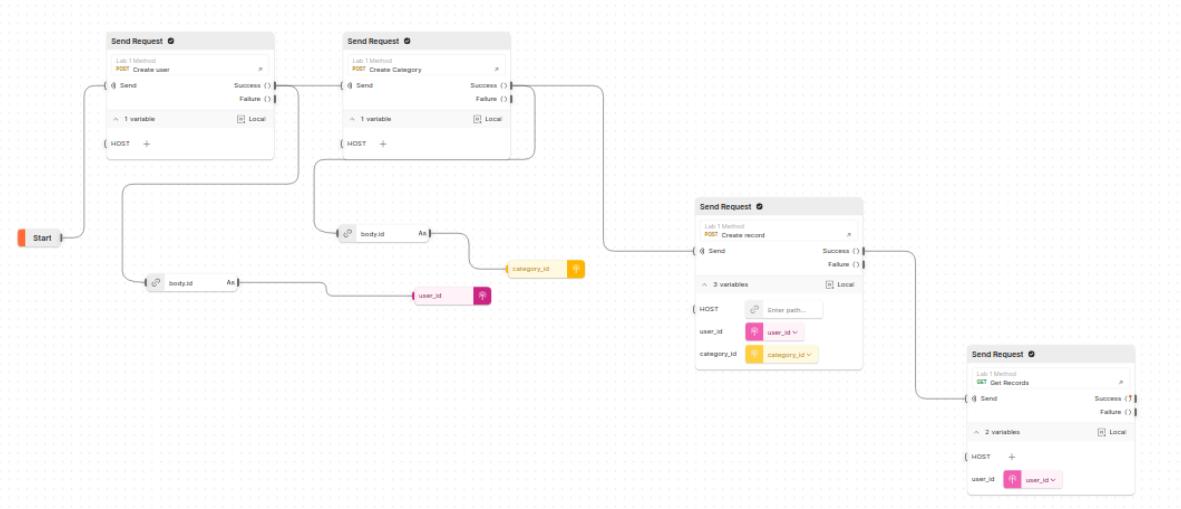
8. Щоб експортувати колекцію потрібно вибрати три крапи на логотипі колекції та натиснути кнопку Export



9. Також для зручності Вам потрібно створити postman flow для тестування ваших запитів, інструкції для flow можна знайти тут - <https://learning.postman.com/docs/postman-flows/concepts/blocks-and-connections/>

10. Також, рекомендую переглянути це відео для того, щоб розуміти як передавати параметри з відповіді в наступний запит - https://youtu.be/5ILpcE8_4RI?si=Tp929rFWQY4r8aMz

11. Flow має виглядати приблизно так:



Критерії оцінювання:

Всього можна отримати 20 балів за лабораторну

Розподіл балів за лабораторну 2:

10 балів - всі зазначені в методичці ендпоїнти наявні і коректно працюють

4 бали - є postman колекція з відповідними environment

4 бали - є postman flow з яким можна прогнати флоу застосунку

2 бали - коректна робота з git: повідомлення комітів та адекватна кількість комітів

Також на розсуд перевіряючого може бути доставлено 1-2 бали за цікаві рішення.