**Міністерство освіти і науки України**

**Івано-Франківський національний технічний університет нафти і газу**

Кафедра програмного забезпечення автоматизованих систем

**ЛАБОРАТОРНА РОБОТА №5**

Тема:«Розроблення RESTful API для веб-додатку “Список книг”**»**

Виконав студент

групи ПІт-15-3 **Зьола І. П.**

Перевірив:

**Броновський І.В.**

Івано-Франківськ

2016

**Мета:** Розробити RESTful JSON API для веб-додатку “Список книг”.

**Хід роботи**

REST (скор. англ. Representational State Transfer, «передача репрезентативного стану») — підхід до архітектури мережевих протоколів, які забезпечують доступ до інформаційних ресурсів. Був описаний і популяризований у 2000 році Роєм Філдінгом, одним із творців протоколу HTTP. Відповідно, в основі REST закладено принципи функціонування Всесвітньої павутини і, зокрема, можливості HTTP протоколу. Філдинг розробив REST стиль паралельно разом із HTTP 1.1 базуючись на попередньому протоколі HTTP 1.0.

Дані повинні передаватися у вигляді невеликої кількості стандартних форматів (наприклад HTML, XML, JSON). REST протокол (як і HTTP) повинен підтримувати кешування, не повинен залежати від мережевого прошарку, не повинен зберігати інформацію про стан між парами «запит-відповідь». Стверджується, що такий підхід забезпечує масштабування системи і дозволяє їй еволюціонувати з новими вимогами.

Для серіалізації об’єктів було використано gem active\_model\_serializers. Він дозволяє описувати які поля необхідно серіаізувати, і як саме їх серіалізувати.

**Код класу api/v1/BooksСontroller:**

class Api::V1::BooksController < Api::V1::BaseController

def index

books = Book.includes(:genres)

.search(params[:keyword]).filter(params[:filter]).approved

render json: books, each\_serializer: Api::V1::BookSerializer

end

def show

book = Book.find(params[:id])

render(json: Api::V1::BookSerializer.new(book).to\_json)

end

def create

book = Book.new(book\_params)

if book.save

render json: Api::V1::BookSerializer.new(book).to\_json, status: :created

else

render json: book.errors, status: :unprocessable\_entity

end

end

private

def book\_params

params.require(:book).permit(:title, :author, :description, :amazon\_id,

{ genre\_ids: [] }, :rating)

end

end

**Код класу api/v1/GenresСontroller:**

class Api::V1::GenresController < Api::V1::BaseController

def index

genres = Genre.all

render json: genres, each\_serializer: Api::V1::GenreSerializer

end

end

**Код класу api/v1/BookSerializer:**

class Api::V1::BookSerializer < ActiveModel::Serializer

attributes :id, :title, :author, :description, :rating, :approved, :genres

end

**Код класу api/v1/GenreSerializer:**

class Api::V1::GenreSerializer < ActiveModel::Serializer

attributes :id, :name

end

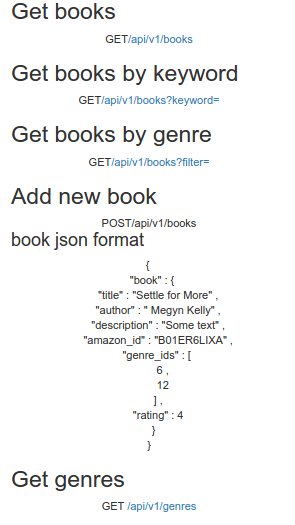


Рис. 1 - Формат запитів до API.

**Висновок:** На даній лабораторній роботі було розроблено JSON API для веб-додатку “Список книг”.