

# 10. Django REST framework & Vue Pinia.

## Контрольні запитання

- 10.01. Для чого потрібен BackEnd API?
- 10.02. Які існують основні способи реалізації BackEnd API?
- 10.03. Що таке серіалізатор в контексті Django REST framework?

## Завдання для аудиторної роботи

[A]10.01. Розширити Django-проект бібліотеки [A]07.01, описавши REST API та задіявши бібліотеку Pinia фреймворку Vue для використання REST API. Проект має бути розміщений у Git-репозиторії. Кожен пункт повинен бути окремим комітом в Git-репозиторії з прикріпленим тегом, який можна додати командою

```
git tag <tagname> <commit-hash>
```

Для того щоб вивантажити на віддалений репозиторій коміти разом з тегами, потрібно виконати команду

```
git push origin <main-branch> --tags
```

16. Оновити контейнери, додавши контейнер для Vue та зареєструвавши Django REST framework

- Реорганізувати серверну частину
- Зареєструвати Django REST framework в проекті, записавши в `settings.py`

```
INSTALLED_APPS = [  
    ...  
    "api.apps.ApiConfig",  
    "rest_framework",  
]
```

- Створити Vue-проект та додати Dockerfile для його контейнера

```
FROM node:22  
  
WORKDIR /frontend  
  
COPY package*.json ./  
RUN npm install  
  
COPY . .  
  
CMD ["npm", "run", "dev", "--", "--host"]
```

- Додати фреймворк Vue в окремий контейнер та вимкнути nginx.

```
frontend:
  build: ./frontend
  container_name: vue
  command: npm run dev -- --host
  volumes:
    - ./frontend:/frontend
    - /frontend/node_modules
  ports:
    - "5173:5173"
  environment:
    - CHOKIDAR_USEPOLLING=true
  depends_on:
    - web
```

- Забезпечити інсталяцію бібліотеки pinia та axios.
- Запустити Django контейнер на порту **8000**.
- Додати у **vite.config.js** опрацювання шляху **/api** для REST API

```
export default defineConfig({
  server: {
    port: 5173,
    proxy: {
      "/api": "http://host.docker.internal:8000"
    }
  }
});
```

## 17.Створити REST API для моделей Book та Author.

- Для відображення JSON в **settings.py** використати запис

```
REST_FRAMEWORK = {
  "DEFAULT_RENDERER_CLASSES": [
    "rest_framework.renderers.JSONRenderer",
    # "rest_framework.renderers.BrowsableAPIRenderer"
  ]
}
```

## 18.Описати сторінку, використовуючи Vue, для зручного перегляду списку книг, використовуючи Vue Pinia та Django REST framework.

## 19.Описати сторінку, використовуючи Vue, для зручного перегляду інформації про задану книгу, використовуючи Vue Pinia та Django REST framework.

## Завдання для самостійної роботи

- [B]10.01. Доповнити Django-проєкт блогу, який описано в задачі [B]07.01. Додати автентифікацію та інтерактивною взаємодією зареєстрованих користувачів з даними. Проєкт має бути розміщений у Git-репозиторії. Кожен пункт повинен бути окремим комітом в Git-репозиторії з прикріпленим тегом, який можна додати командою

```
git tag <tagname> <commit-hash>
```

Для того щоб вивантажити на віддалений репозиторій коміти разом з тегами, потрібно виконати команду

```
git push origin <main-branch> --tags
```

- 17.Оновити контейнери, додавни контейнер для Vue та зареєструвавши Django REST framework.
- 18.Створити REST API для моделей Article та Comment.
- 19.Описати сторінку для зручного перегляду списку статей, використовуючи Vue Pinia та Django REST framework.
- 20.Описати сторінку, використовуючи Vue, для зручного перегляду інформації про задану статтю, використовуючи Vue Pinia та Django REST framework.