

## **ВИКОРИСТАННЯ DJANGO REST FRAMEWORK ДЛЯ СТВОРЕННЯ API: ПЕРЕВАГИ ТА МОЖЛИВОСТІ**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

В епоху стрімкого розвитку технологій та постійного прагнення до ефективності в сфері веброзробки використання потужних інструментів стає ключовим елементом успіху. Один із таких інструментів, що здобуває все більше популярності у спільноті розробників, – Django REST Framework (DRF). Вбудований в фреймворк Django, цей інструмент дає змогу легко створювати вебсервери та API, сприяючи швидкій розробці та принципам чистого та практичного дизайну.

Django – це високорівневий вебфреймворк, написаний мовою програмування Python, який сприяє швидкій розробці та чистому, практичному дизайну вебдодатків. Фреймворк використовує архітектурний шаблон Model-View-Controller (MVC) і підтримує принципи DRY (Don't Repeat Yourself) і CoC (Convention over Configuration).

Django REST Framework (DRF) – це розширення для фреймворку вебдодатків Django, яке надає потужні інструменти для створення вебсерверів та API. DRF допомагає легко створювати API, використовуючи стандарти RESTful, і надає широкий вибір можливостей для ефективної розробки вебдодатків.

DRF продовжує вирізнятися своєю гнучкістю, широким функціоналом і зручним інструментарієм для розробки високоефективних вебсервісів та API. З погляду тенденцій розробки програмного забезпечення акцент на мікросервісній архітектурі та розподілених системах підкреслює значимість добре розроблених і зручних інструментів для створення API, зокрема DRF. Актуальність теми підтверджується постійним попитом на фахівців, які володіють навичками роботи з Django REST Framework, і високим рівнем зацікавленості спільноти веброзробників у вдосконаленні та розширенні можливостей DRF для розвитку сучасних та ефективних вебдодатків [1].

Django REST Framework допомагає створювати API, використовуючи простий та ефективний підхід. Загалом у нього є багато переваг та можливостей, наприклад [2]:

- зручна робота з Django: DRF інтегрується плавно з фреймворком Django, розширюючи його можливості для швидкого створення API;
- автоматична документація: DRF автоматично генерує документацію для API, що спрощує розробку, тестування і розгортання;

- легкість серіалізації та десеріалізації: має потужний інструментарій для перетворення об'єктів Django в формати даних, як-от JSON, і навпаки;
- гнучка автентифікація та авторизація: надає готові рішення для автентифікації та авторизації, як-от токени, OAuth, базова автентифікація;
- розширюваність та кастомізація: легко розширюється та кастомізується з допомогою власних переглядів, серіалізаторів та інших компонентів;
- використання Django ORM: дає змогу використовувати Django ORM для взаємодії з базою даних, що робить роботу з даними простішою та зручною;
- вбудована підтримка переглядів: містить великий набір вбудованих переглядів для різноманітних сценаріїв використання, що прискорює розробку;
- відмінна підтримка автентифікації користувача: надає можливості роботи зі стандартними моделями користувача Django та автоматичну обробку сесій;
- маршрутизація: вбудована система маршрутизації для визначення, які URL відповідають різним переглядам;
- фільтрація і пошук: підтримка фільтрації та пошуку для легкості отримання конкретних наборів даних;
- робота з файлами і зображеннями: можливість обробляти завантаження файлів і роботи зображень;
- підтримка форматів даних: підтримка різних форматів введення / виведення даних, як-от JSON, XML, YAML;
- вбудовані засоби валідації: засоби для валідації та перевірки правильності даних у запитах.

Загалом DRF робить створення API простим та ефективним, забезпечуючи готові рішення для багатьох ключових аспектів розробки. Стабільність Django та спільноти навколо нього найімовірно зросла з моменту його першого випуску [3]. Офіційна документація та навчальні посібники фреймворку є одними з найкращих у своєму роді. І з кожною новою версією Django продовжує додавати нові можливості. Отже, використання Django REST Framework не лише допомагає швидко створювати високоякісні API, але і стає ключовим фактором у вдалих та масштабованих веб-розробках, де ефективність і надійність важливі як ніколи.

### Список використаних джерел

1. Фреймворки, які полегшують веб-розробку на Python: вебсайт. URL: <https://avivi.academy/blogs/python-framework/> (дата звернення: 26.11.2023).
2. Вкладені API за допомогою Django REST Framework: вебсайт. URL: <https://devzone.org.ua/post/vkladeni-api-za-dopomogoiu-django-rest-framework> (дата звернення: 26.11.2023).
3. Плюси і мінуси Django: вебсайт. URL: <https://blog.ukrnames.com/web-master/plyusi-i-minusi-django> (дата звернення: 26.11.2023).

4. Павлов Д. Л., Січко Т. В. Принцип роботи Web API та його застосування. *Прикладні інформаційні технології: матеріали всеукр. наук.-практ. конф.*, м. Вінниця, 2023. С. 166–168.

#### **УДК 004.021:656.02**

*Поліщук А. М., здобувачка 3 курсу спеціальності 122 Комп'ютерні науки, Січко Т. В., канд. техн. наук, доцент, доцент кафедри інформаційних технологій*

### **ОПТИМІЗАЦІЯ МАРШРУТИЗАЦІЇ ТРАНСПОРТНИХ МЕРЕЖ**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Транспортна мережа відіграє важливу роль у переміщенні товарів, послуг та людей. Незалежно від розмірів та складності мережі, оптимізація маршрутизації стає ключовим завданням, спрямованим на покращення ефективності та зменшення витрат. Ця проблема особливо актуальна в умовах зростаючого руху, коли міста та транспортні мережі стикаються з викликами, як-от затори, забруднення довкілля та пальне.

Дослідження та оптимізація маршрутизації стають необхідністю для подолання цих проблем. Точне та науково обґрунтоване планування маршрутів транспортних засобів може значно полегшити пересування та скоротити час подорожі. Оптимізована маршрутизація також може допомогти зменшити витрати на пальне, що важливо для економії ресурсів та збереження навколишнього середовища.

Отже, оптимізація маршрутизації у транспортних мережах є актуальною та важливою темою з практичним значенням і великим потенціалом для досліджень у галузі операційного дослідження та інженерії транспорту. У цьому контексті обговорюється важливість оптимізації маршрутизації та підходи до її вирішення.

Останні дослідження у сфері оптимізації маршрутизації вказують на зростання інтересу до використання штучного інтелекту та машинного навчання. Зокрема, використання цих методів дає змогу враховувати історичні дані про трафік, прогнозувати потік трафіка та враховувати різноманітні фактори, як-от погодні умови та події на дорогах. Це веде до підвищення точності та ефективності процесу маршрутизації.

Також варто зазначити зростання ролі геоінформаційних систем (ГІС) у цьому контексті. Використання ГІС дає змогу не лише визначати оптимальні маршрути, а й враховувати географічні обмеження, як-от особливості маршруту