

Ключові елементи API та робота з документацією

Заняття 2

Тестування API

Лектор
Йосип Волощук

The logo consists of the lowercase letters 'r' and 'd' separated by an underscore, all in a bold, sans-serif font. It is contained within a white, rounded rectangular shape that has a small triangular notch at the top right corner.

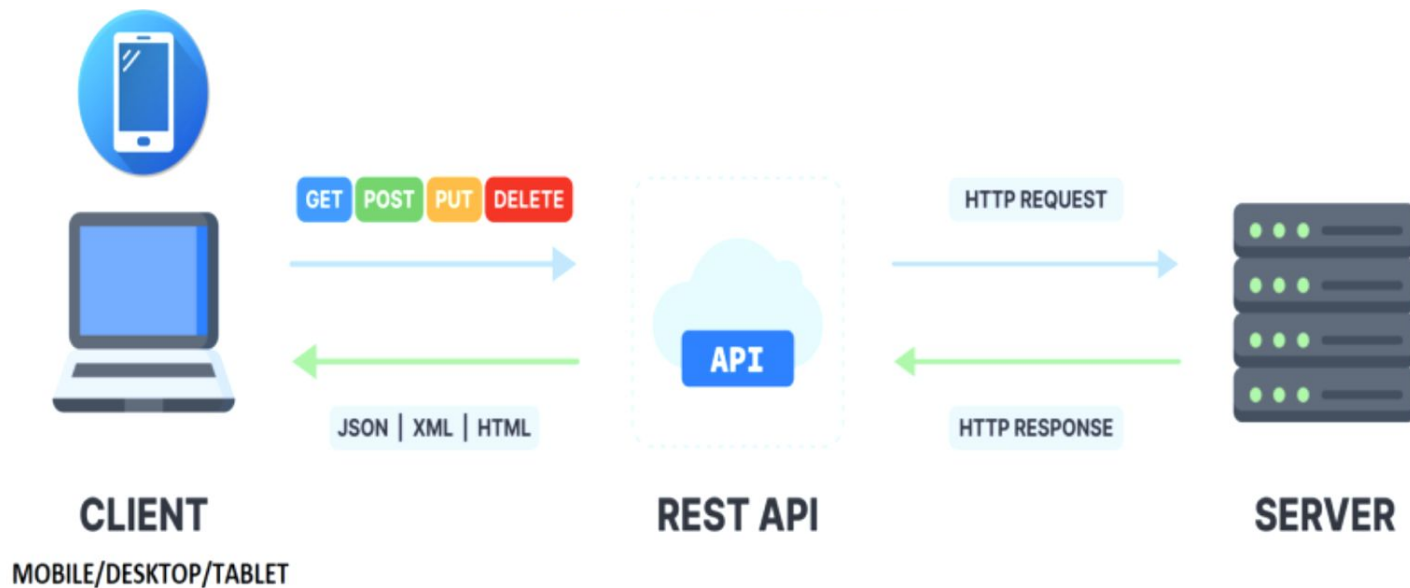
r_d

ПЛАН ЗАНЯТТЯ



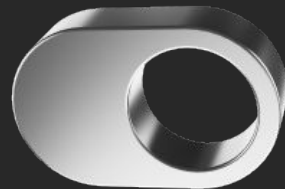
- Особливості тестування API
- CRUD — створення, читання, оновлення, видалення
- HTTP-методи в REST API
- Headers в REST API запиті
- CONTENT TYPE в REST API
- Статуси відповідей HTTP
- Розробка тестових випадків

REST API MODEL



ВСТУП

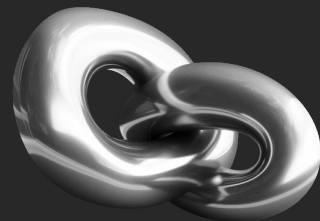
Тестування API відіграє важливу роль у забезпеченні якості програмного забезпечення. Розуміння основних концепцій тестових випадків і тест-дизайну допомагає покращити процес тестування та забезпечити ефективне виявлення помилок.



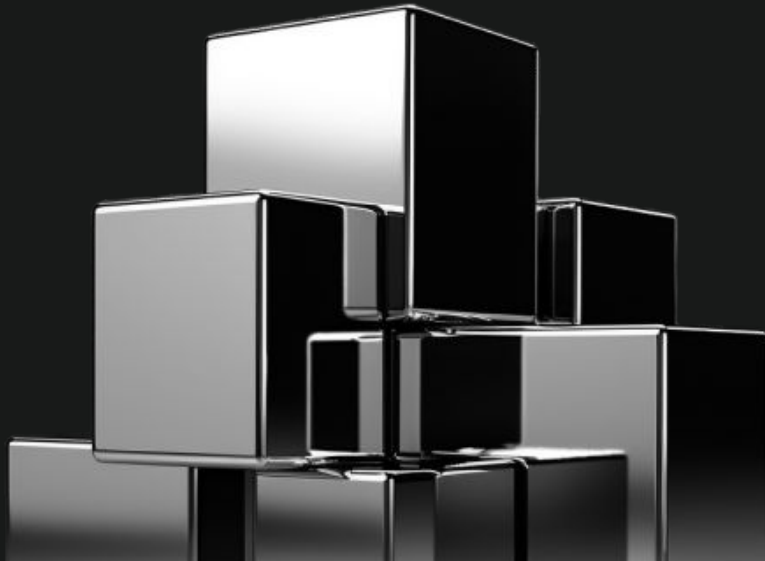
ОСОБЛИВОСТІ ТЕСТУВАННЯ API

Тестування API відрізняється від тестування інтерфейсів користувача чи інших компонентів програмного забезпечення тим, що воно **фокусується на взаємодії** з програмним інтерфейсом без відтворення графічного інтерфейсу.

Під час тестування API важливо перевіряти правильність передачі даних та відповідність вимогам специфікації API.

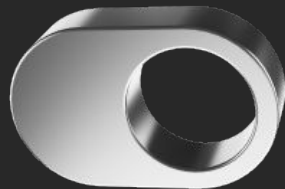


CRUD



CRUD — СТВОРЕННЯ, ЧИТАННЯ, ОНОВЛЕННЯ, ВИДАЛЕННЯ

- ▶ Створення (CREATE) — додає нові дані до системи.
- ▶ Читання (READ) — отримання даних із системи.
- ▶ Оновлення (UPDATE) — зміна наявних даних.
- ▶ Видалення (DELETE) — видалення даних із системи.



МЕТОДИ HTTP В REST API



МЕТОДИ HTTP В REST API

GET

Отримання даних із сервера.
Використовують для запиту ресурсів у REST API.

POST

Надсилання даних на сервер для створення нового ресурсу.
Застосовують для створення ресурсів у REST API.

PUT

Оновлення наявного ресурсу на сервері.
Використовують для зміни інформації про ресурс у REST API.

DELETE

Видалення наявного ресурсу на сервері.

HEADERS В REST API ЗАПИТІ



HEADERS В REST API ЗАПИТІ

1

Заголовки (Headers) у REST API запитах містять інформацію про тип даних, мову, автентифікацію тощо.

2

Headers допомагають серверу розуміти, як обробляти запит та яких даних очікувати у відповіді.

3

Деякі заголовки можуть бути обов'язковими для правильної обробки запиту або вказувати на потребу автентифікації.

ПАРАМЕТРИ В REST API

- ▶ Параметри в REST API використовують для передачі даних за виклику ресурсів.
- ▶ Їх можна передати через URL-адресу запиту або в тілі запиту у вигляді JSON або XML.
- ▶ Параметри можуть бути обов'язковими або необов'язковими залежно від API.
- ▶ Важливо дотримуватися документації API щодо типів параметрів, формату передачі та обробки помилок.



ФОРМАТИ ДАНИХ

{ } JSON

- Легко зрозуміти людині
- Підтримує більшість мов програмування
- Ідеальний для передачі структурованих даних

☑ XML

- Дозволяє валідацію даних за допомогою схем
- Підтримує різні стилі даних
- Може бути складнішою в розумінні та обробці

CONTENT TYPE B

REST API



CONTENT TYPE В REST API

ЗАГАЛЬНЕ ВИКОРИСТАННЯ

- `application/json`: стандартний формат для обміну даними
- `application/xml`: використовують для обміну даними у форматі XML
- `text/html`: застосовують для надсилання HTML-документів

СПЕЦІАЛІЗОВАНІ ТИПИ

- `application/x-www-form-urlencoded`: для відправлення даних форм у вебформатах
- `multipart/form-data`: використовують для надсилання файлів
- `application/pdf`: для відправлення документів у форматі PDF

СТАТУСИ ВІДПОВІДЕЙ HTTP



СТАТУСИ ВІДПОВІДЕЙ HTTP

ПРИКЛАДИ ВИКОРИСТАННЯ

200 OK: успішний запит, повернення даних користувачу.

404 Not Found: ресурс не знайдено, повідомлення про помилку.

503 Service Unavailable: сервер тимчасово недоступний.

ЗНАЧЕННЯ СТАТУСІВ

100–199: інформаційні повідомлення, процес запиту триває.

200–299: успішні відповіді, запит було коректно оброблено.

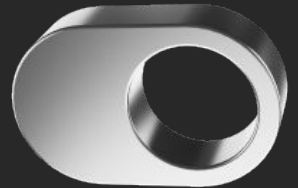
300–399: перенаправлення, потрібні додаткові дії для завершення запиту.

400–499: помилки клієнта, неправильні запити або відсутність доступу.

500–599: помилки сервера, неможливість обробити запит.

ВІДПОВІДІ REST API

- ▶ Status code відповіді вказує на результат виконання запиту.
- ▶ Status codes допомагають ідентифікувати проблеми або успішні операції.
- ▶ Найпоширеніші Status codes: 200 (Успішно), 404 (Не знайдено), 500 (Внутрішня помилка сервера).



РОЗРОБКА ТЕСТОВИХ СЦЕНАРІЇВ

ІДЕНТИФІКАЦІЯ ЦІЛЕЙ

Визначення конкретних цілей тестування API-запитів, які охоплюють функціональні та нефункціональні вимоги.

Формулювання списку цілей тестування

АНАЛІЗ API-ЗАПИТІВ

Дослідження та розуміння API-запитів, їхніх параметрів та очікуваних результатів для кожного сценарію тестування.

Опис сценаріїв тестування із вказівкою параметрів та результатів

ПІДГОТОВКА ТЕСТОВИХ ДАНИХ

Підготовка вхідних даних для виконання тестів API-запитів з урахуванням різних варіантів вхідних умов.

Набір тестових даних для кожного сценарію

КРИТЕРІЇ УСПІХУ

Встановлення об'єктивних критеріїв для оцінки успішності виконання тестів та відповідності API-запитів вимогам.

Список критеріїв успіху для кожного тестового сценарію

ПРОБЛЕМИ ТА ВИКЛИКИ

ТИПОВІ ПРОБЛЕМИ

- Недостатнє покриття тестами різних сценаріїв API.
- Складність синхронізації між розробниками й тестувальниками.
- Висока залежність від зовнішніх систем та сервісів.
- Труднощі з автоматизацією тестування через змінність API.

ВИКЛИКИ

- Забезпечення точності й актуальності тестових даних.
- Управління версіями API та сумісністю змін.
- Виявлення та аналіз причин помилок у відповідях API.
- Оптимізація процесу тестування для швидкого впровадження змін.

Q&A

???



ЗАВЖДИ Є КУДИ
ЗРОСТАТИ