# Основи API

Заняття 1

Тестування API

Лектор
Йосип Волощук

# Йосип Волощук

Lead Software Engineer in Test
у SoftServe

Лектор курсу

- Lead Software Engineer in Test у SoftServe
- використовую понад 7 мов програмування в роботі
- 8+ років досвіду в тестуванні, працює з API, Web, Mobile, Performance
- маю досвід впровадження тестування у продукти з високим трафіком: 1200 users/second
- брав участь у 25+ проєктах: від E-commerce до систем охорони здоров’я, проджект-менеджменту й краудфандингових платформ

# ПЛАН ЗАНЯТТЯ

- Що таке API
- Вебсервіси
- Види API: SOAP, REST, GraphQL. Їхні архітектури та основні компоненти

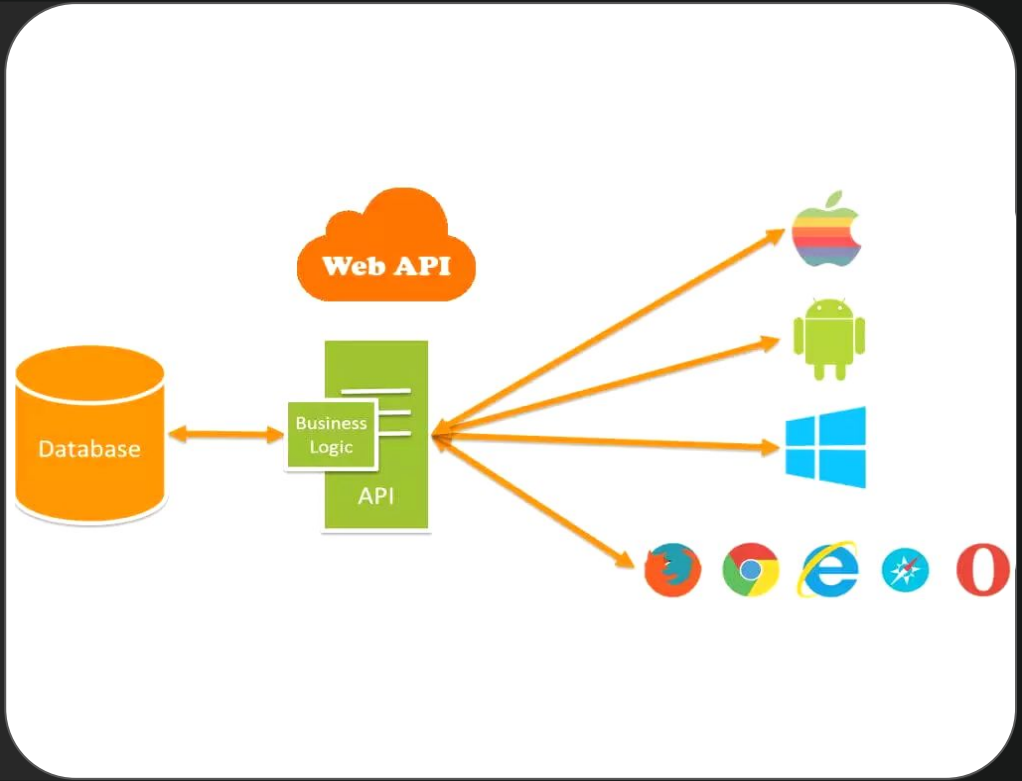
# ЩО ТАКЕ API

# ЩО ТАКЕ API

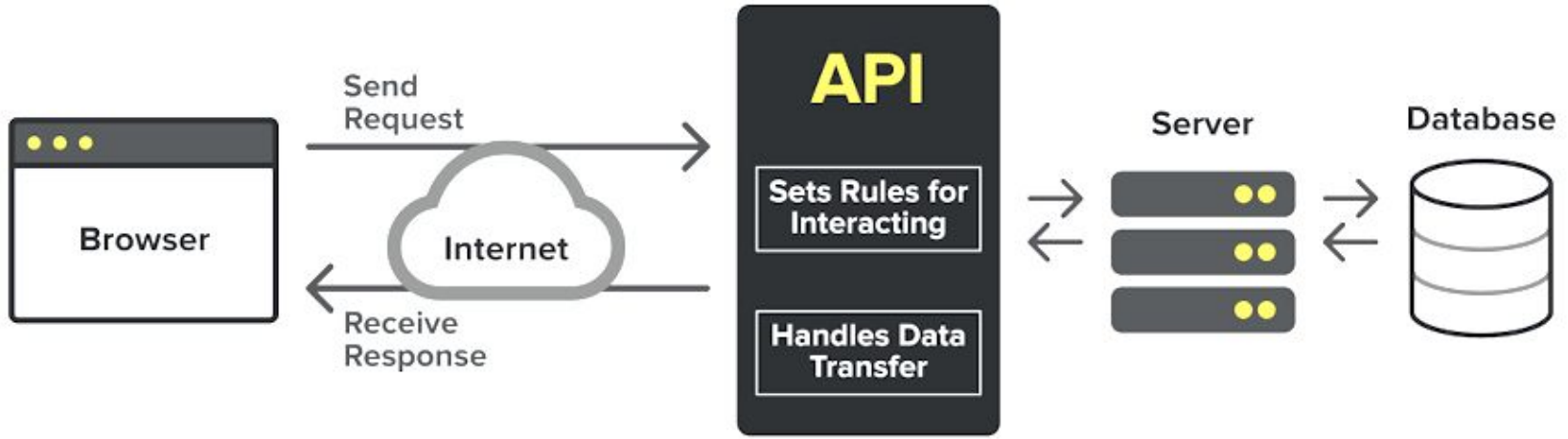
**API** — це зв’язок між різними системами або рівнями програми.

**Програми часто мають три рівні:**

- рівень даних (база даних)
- сервісний рівень (API)
- рівень презентації (UI)

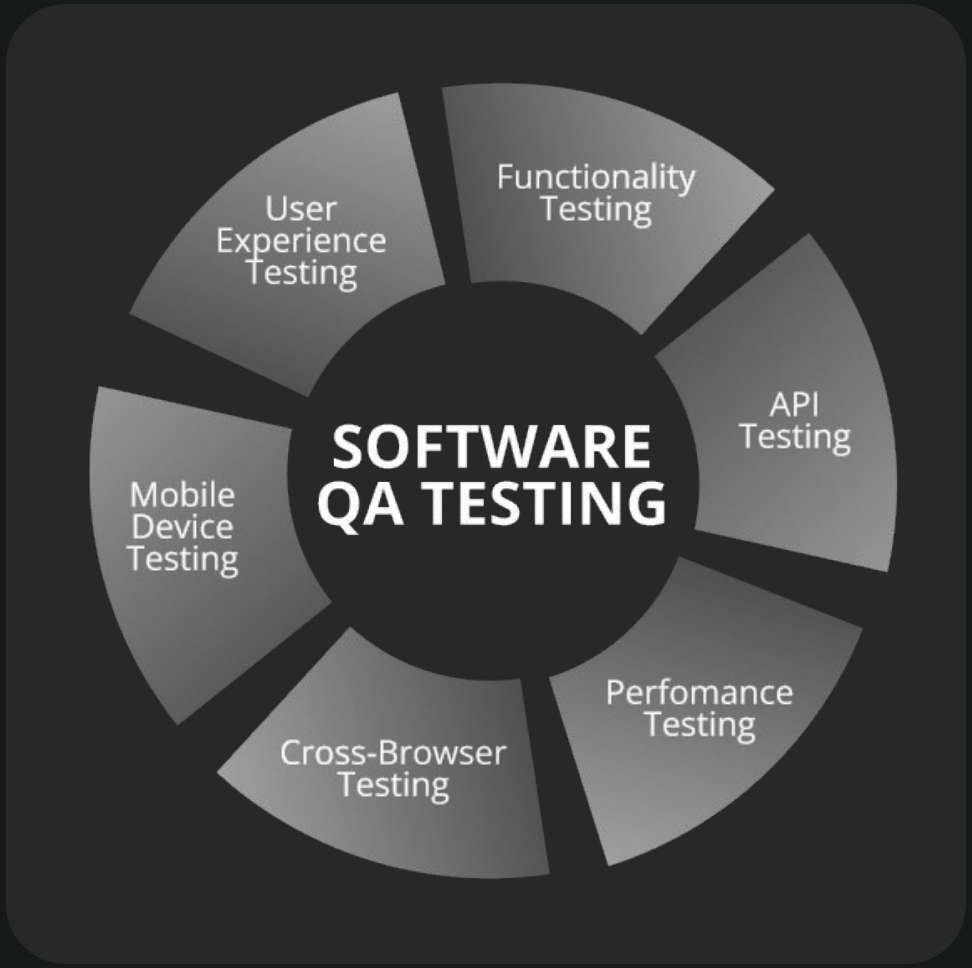


# ЯК ЦЕ ВСЕ ПРАЦЮЄ?



# ПЕРЕВАГИ API-ТЕСТУВАННЯ

- Раннє тестування дає змогу виявити помилки на ранніх етапах розробки програмного забезпечення.
- Просте обслуговування API спрощує процес тестування і робить його більш ефективним.
- Швидкість і охоплення тестування забезпечують швидке виявлення проблем та перевірку різних сценаріїв.

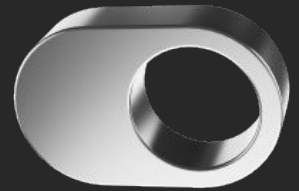


# ВЕБСЕРВІСИ

# ВЕБСЕРВІСИ

**Опис**

- **Вебсервіси** — це програмне забезпечення, яке надає змогу взаємодіяти з користувачами через мережу Інтернет.
- Ці сервіси можуть виконувати різні **функції:** від обробки даних до надання доступу до ресурсів через вебінтерфейс.



# ВЕБСЕРВІСИ.ТИПИ ВЕБСЕРВІСІВ

| SOAP (Simple Object Access Protocol) | REST (Representational State Transfer) | GraphQL |
| --- | --- | --- |
| протокол обміну структурованими інформаційними повідомленнями через мережу. | архітектурний стиль для розробки вебсервісів | мова запитів і маніпуляції даними з відкритим кодом для API і середовище виконання для обслуговування запитів з наявних даних. |

# ВИДИ API: SOAP, REST, GRAPHQL. ЇХНІ АРХІТЕКТУРИ ТА ОСНОВНІ КОМПОНЕНТИ

# SOAP

# ОСНОВНІ ХАРАКТЕРИСТИКИ SOAP

- SOAP є незалежним від платформи та мови
- SOAP використовує формат XML для представлення даних
- Простота інтеграції SOAP з іншими системами
- SOAP підтримує широкий спектр протоколів для обміну повідомленнями

# СТРУКТУРА SOAP ПОВІДОМЛЕННЯ

| Конверт | Заголовок | Тіло повідомлення |
| --- | --- | --- |
| Кожен SOAP повідомлення починається з елемента Envelope, який оточує всі інші елементи повідомлення. | Header-елемент може містити метадані про повідомлення, як-от ідентифікатори, маршрутизаційні дані або інші корисні відомості. | Body-елемент містить фактичний вміст повідомлення, який передається між відправником та отримувачем. |

# СТРУКТУРА SOAP ПОВІДОМЛЕННЯ

# SOAP: ПЕРЕВАГИ ТА НЕДОЛІКИ

## ПЕРЕВАГИ

- SOAP є надійним та безпечним протоколом для обміну даними.
- Легко інтегрується з різними мовами програмування, як-от Java або C++.
- Підтримує розширення даних XML для складних структур.

## НЕДОЛІКИ

- SOAP може бути менш продуктивним порівняно з іншими протоколами через великий обсяг даних XML.
- Вимагає великої пропускної здатності мережі через повільніший обмін даними.
- Складний у порівнянні з REST через більшу складність і обсяг коду.

# REST API

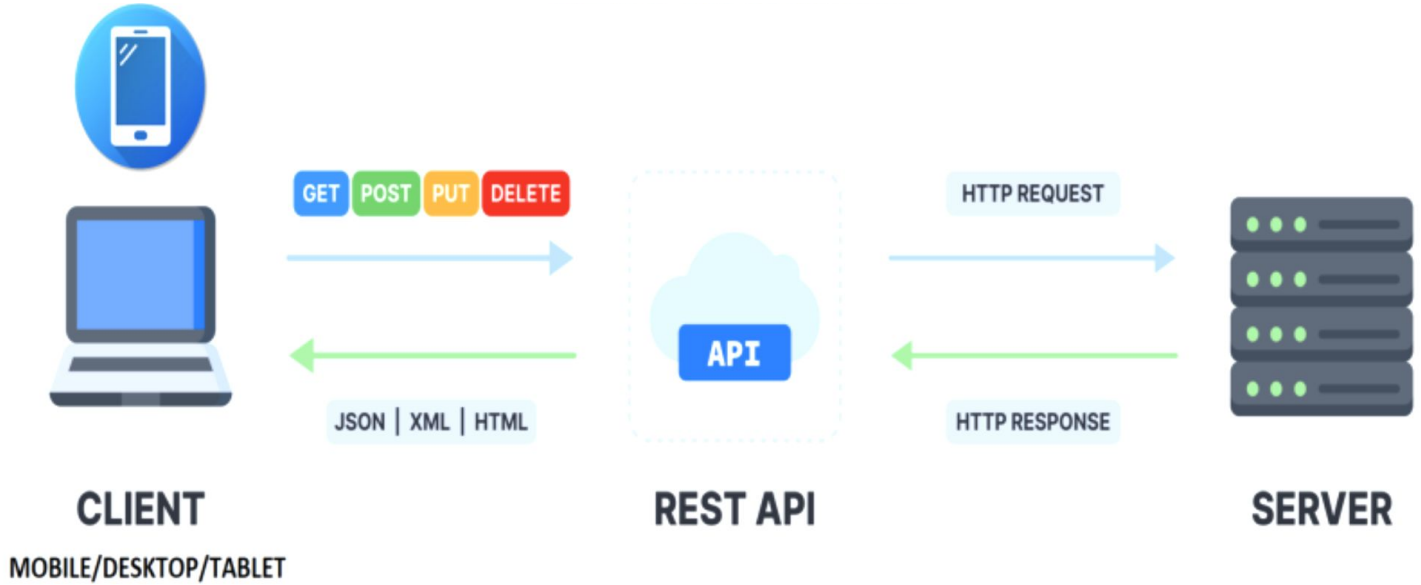
# ОСНОВНІ ПРИНЦИПИ REST API

- REST (Representational State Transfer) — це архітектурний стиль для розроблення вебзастосунків.
- REST використовує HTTP-протокол для комунікації між клієнтом і сервером.
- У REST кожен ресурс (наприклад, користувач або товар) має унікальну адресу URL та може бути доступним за допомогою різних методів (GET, POST, PUT, DELETE).

# СТРУКТУРА ПОВІДОМЛЕННЯ REST

| Метод | URL | Заголовки | Тіло |
| --- | --- | --- | --- |
| вказує на тип операції, наприклад, GET, POST, PUT, DELETE. | адреса ресурсу, до якого звертається запит. | містять метадані про запит або відповідь. | дані, які передають у відповіді або запиті. |

# REST API MODEL



# ПРИКЛАДИ ВИКОРИСТАННЯ REST API

### ВЕБРОЗРОБКА

REST API використовують у веброзробці для обміну даними.

### ЗАСТОСУНКИ

У мобільних застосунках REST API дозволяє отримувати дані із сервера.

### ІНТЕГРАЦІЯ

Інтеграція систем використовує REST API для сполучення різних сервісів.

### ФІНАНСИ

У фінансовому секторі REST API допомагає виконувати транзакції та отримувати фінансові дані.

### МЕДИЦИНА

У медичній сфері REST API можуть використовувати для обміну медичною інформацією між системами.

### КОМЕРЦІЯ

Електронна комерція використовує REST API для обміну даними про замовлення та оплату.

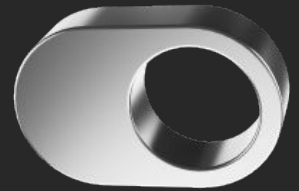
# GRAPHQL

# ВСТУП ДО GRAPHQL

- GraphQL — це мова запитів до сервера, яка дає змогу клієнтам запитувати лише потрібні дані.
- Переваги GraphQL полягають у зменшенні кількості запитів до сервера, гнучкості та структурованості запитів.
- Історія розробки GraphQL пов’язана з Facebook і була випущена як відкрите програмне забезпечення у 2015 році.

# ОСНОВНІ КОНЦЕПЦІЇ GRAPHQL

- Визначення GraphQL як мови запитів до API, яка дає змогу клієнтам обирати саме ті дані, які їм потрібні.
- Типи даних у GraphQL охоплюють Scalar (скалярні), Object (об'єктні), Query (запити) та Mutation (мутації).
- Запити в GraphQL є гнучкими та дають змогу клієнтам отримувати лише ті дані, які їм потрібні.
- Мутації в GraphQL використовують для зміни даних на сервері та оновлення стану системи.



# ПЕРЕВАГИ GRAPHQL НАД REST

## Переваги GraphQL

- Гнучкість у виборі даних для отримання лише потрібної інформації.
- Швидкість виконання запитів завдяки одному запиту на сервер, що повертає потрібні дані.
- Легкість використання завдяки автоматичній побудові схеми запитів.

## Переваги REST

- Неефективність у великому масштабі через можливість отримання зайвої інформації.
- Обмежена швидкість у разі потреби великої кількості окремих запитів.
- Складність у розумінні та використанні через розділення даних на кілька ресурсів.

# АРХІТЕКТУРА GRAPHQL

| Схеми | Розв’язувачі | Типи |
| --- | --- | --- |
| У GraphQL, схема визначає всі запити, які можна виконати, та які дані можна отримати. | Розв’язувачі виконують запити та повертають дані, які відповідають на запити клієнта. | Типи в GraphQL визначають структуру даних, які можна запитати та повернути, включно зі скалярними, об’єктними та іншими типами. |

# ТИПИ ДАНИХ У GRAPHQL

| Скалярні типи (Scalar) | Типи об’єктів (Object) | Запити (Query) | Мутації (Mutation) |
| --- | --- | --- | --- |
| Прості типи даних, як-от рядок або число, які представляють скалярні значення. | Складні типи даних, що містять поля, які можуть бути іншими об’єктами або скалярними значеннями. | Операції для отримання даних із сервера. Запити визначають, які дані клієнт хоче отримати. | Операції для зміни даних на сервері. Мутації використовують для створення, оновлення або видалення даних на сервері. |

# МУТАЦІЇ ТА ПІДПИСКИ

- Мутації використовують для зміни даних у GraphQL, дозволяючи вносити модифікації до серверних даних.
- Підписки дають змогу клієнтам підписуватися на події та отримувати реактивні оновлення у реальному часі.
- Механізми мутацій та підписок у GraphQL дозволяють ефективно управляти даними та спрощують роботу з асинхронними операціями.
- Інтеграція мутацій та підписок у GraphQL дозволяє покращити продуктивність та функціональність вебзастосунків.

# МАЙБУТНЄ GRAPHQL

## Вплив на розробку

- Зручність у використанні та підтримка різних платформ.
- Покращення продуктивності розробників через ефективні запити.
- Розширення можливостей вебзастосунків завдяки гнучкості GraphQL.

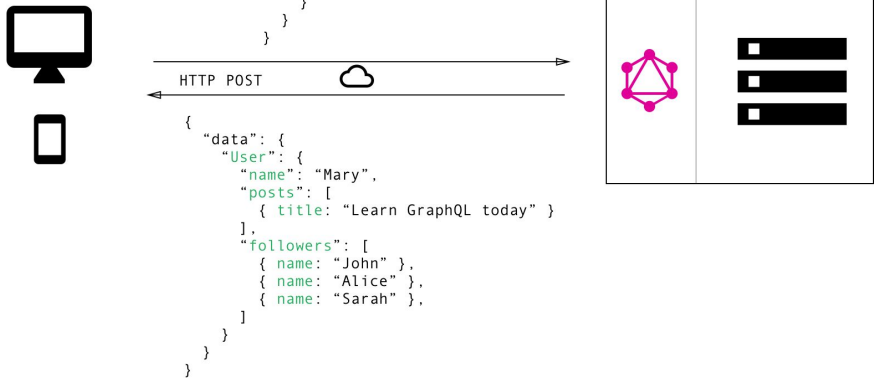
## Майбутні інновації

- Інтеграція з новітніми технологіями, як-от AI та машинне навчання.
- Розвиток інструментів для спрощення роботи з великими даними.
- Підвищення безпеки застосунків через вдосконалені механізми авторизації.

# СТРУКТУРА ПОВІДОМЛЕННЯ GRAPHQL

| Корінь | Поля | Аргументи | Зворотний тип |
| --- | --- | --- | --- |
| початкова операція запиту, що вказує на об’єкт, з якого починається запит. | інформація, яку потрібно отримати для обробки запиту. | параметри, які передаються в поля для фільтрації або сортування даних. | структура, яка визначає дані, що повернуться як відповідь на запит. |

# СТРУКТУРА ПОВІДОМЛЕННЯ GRAPHQL

# ПІДСУМУЄМО

- **API** — це інтерфейс програмування застосунків, який дає змогу взаємодіяти між програмами та передавати дані.
- **Переваги API** охоплюють забезпечення можливості інтеграції різних систем, автоматизацію процесів, а також підвищення продуктивності та швидкості роботи.
- **WEB SERVICES** — це сервіси, що надають функціональність через мережу Інтернет, дозволяючи взаємодіяти із застосунками та обмінюватися даними.
- **WEB SERVICES можна використати** для розширення функціонала, покращення доступу до інформації та спрощення обміну даними між різними платформами.

# Q&A

r_d

ЗАВЖДИ Є КУДИ ЗРОСТАТИ