

# Manual QA

Тестування API

# Manual QA

## План уроку

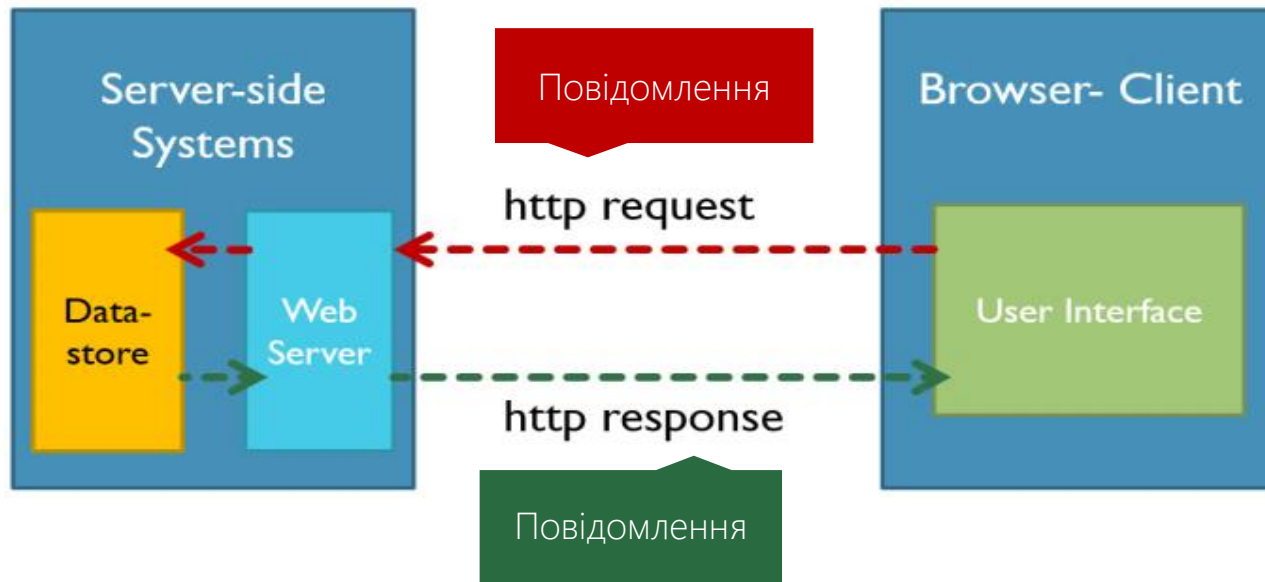
- Основна теорія
- Get, Post, Put, Delete etc.
- REST & SOAP
- JSON & XML
- Корисна тест-документація для API
- Swagger
- Postman
- SoapUI

# Manual QA

Що ми вивчили раніше

# Manual QA

## Клієнт-Сервер схема

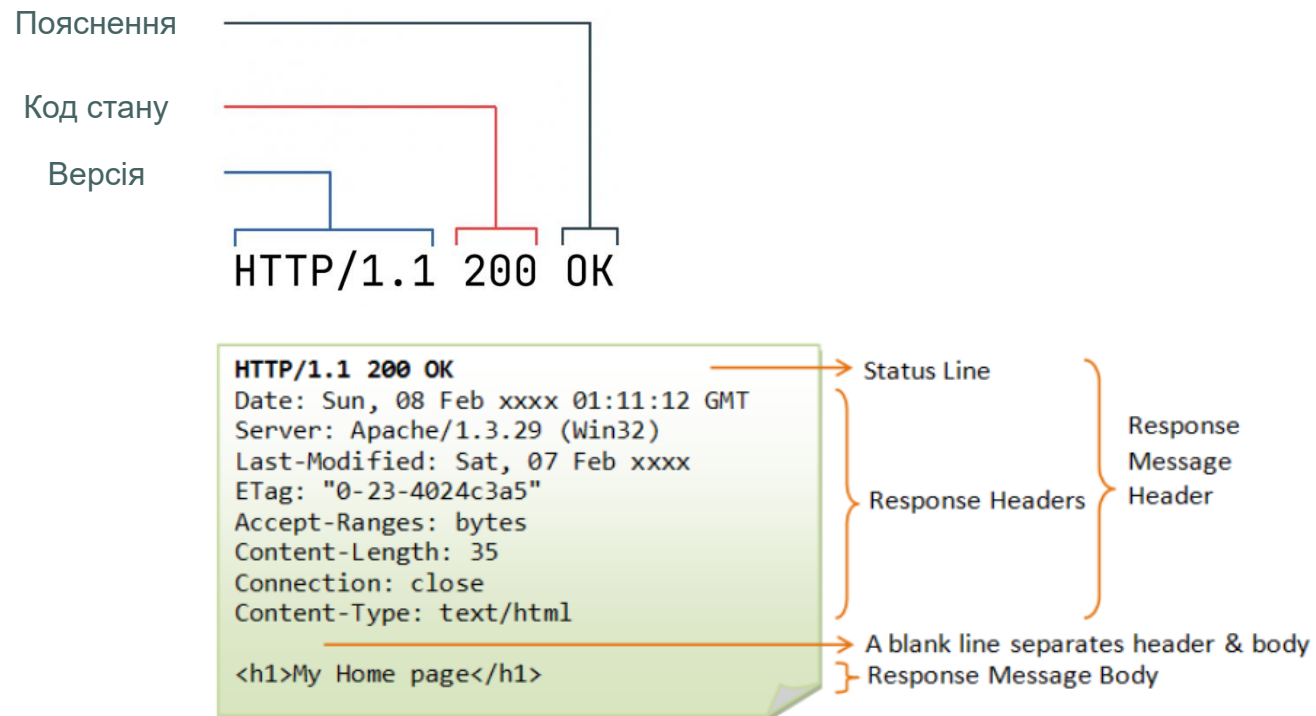


З чого починається серфінг в інтернеті?

1. Надсилання запиту в пошуковому рядку
2. Надсилання тіла запитутела запроса
3. Отримання відповіді
4. Обробка відповіді

# Manual QA

## Рядок статусу HTTP відповіді



## Manual QA

## HTTP статус коди



# Manual QA

## Всі HTTP запити

<b>GET</b>	Запитує інформацію
<b>HEAD</b>	Запитує заголовки ресурсу
<b>PUT</b>	Оновлює дані
<b>POST</b>	Створює дані
<b>PATCH</b>	PUT для фрагмента ресурсу
<b>TRACE</b>	Відстежує проміжні сервери
<b>OPTIONS</b>	Дізнається про можливості сервера
<b>DELETE</b>	Видаляє ресурс

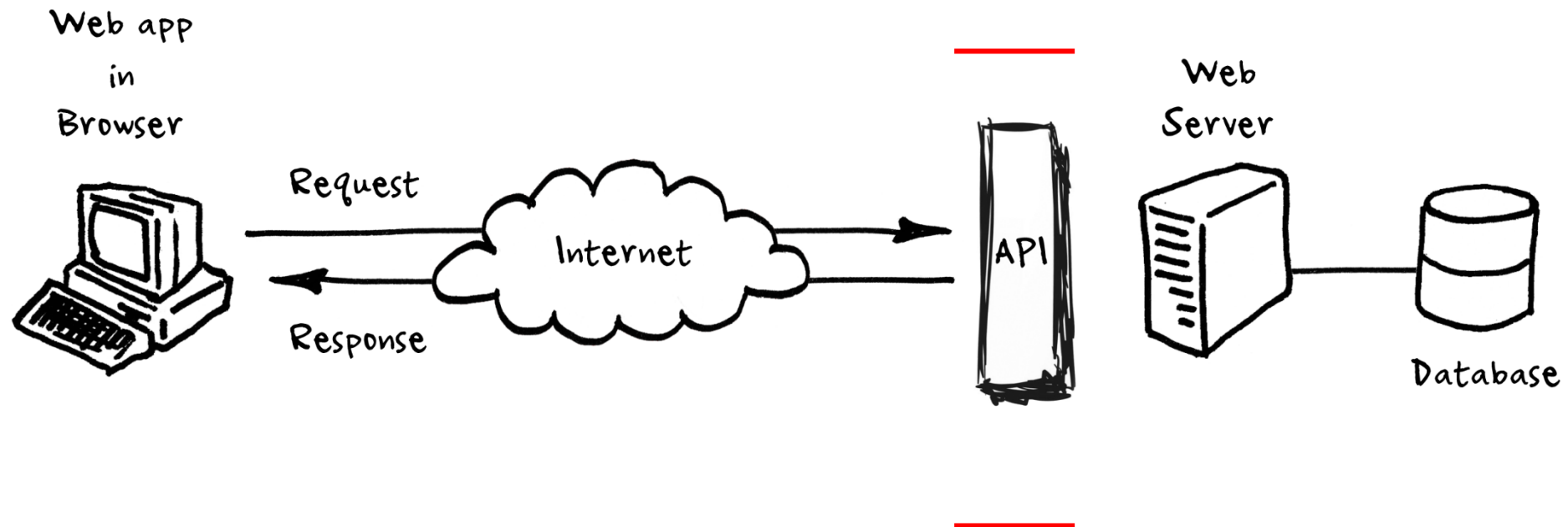
# Manual QA

## Основна теорія



# Manual QA

## Що таке API



# Manual QA

## Що таке API?

API – Application Programming Interface

Interface – це спосіб взаємодії

UI – User Interface

# Manual QA

## Інтерфейс у житті



# Manual QA

## Визначення API

API – це специфікація можливих взаємодій з компонентом ПЗ.

API – представляє собою спосіб, яким програми можуть взаємодіяти одна з одною.

# Manual QA

## Як може виглядати API

GET	/pet/{petId}	Find pet by ID
PUT	/pet	Update an existing pet
DELETE	/pet/{petId}	Deletes a pet
POST	/pet/{petId}/uploadImage	uploads an image

# Manual QA

## Ілюстрація запиту до API

Метод

The screenshot displays a REST client interface with the following components:

- Method:** GET (indicated by a red circle 1).
- URL:** https://jsonplaceholder.typicode.com/users (indicated by a red circle 2).
- Action:** Send (indicated by a red circle 3).
- Status:** 200 OK (indicated by a red circle 4).
- Response Body:** A JSON array of user objects (indicated by a red circle 5).

The response body is shown in the following JSON format:

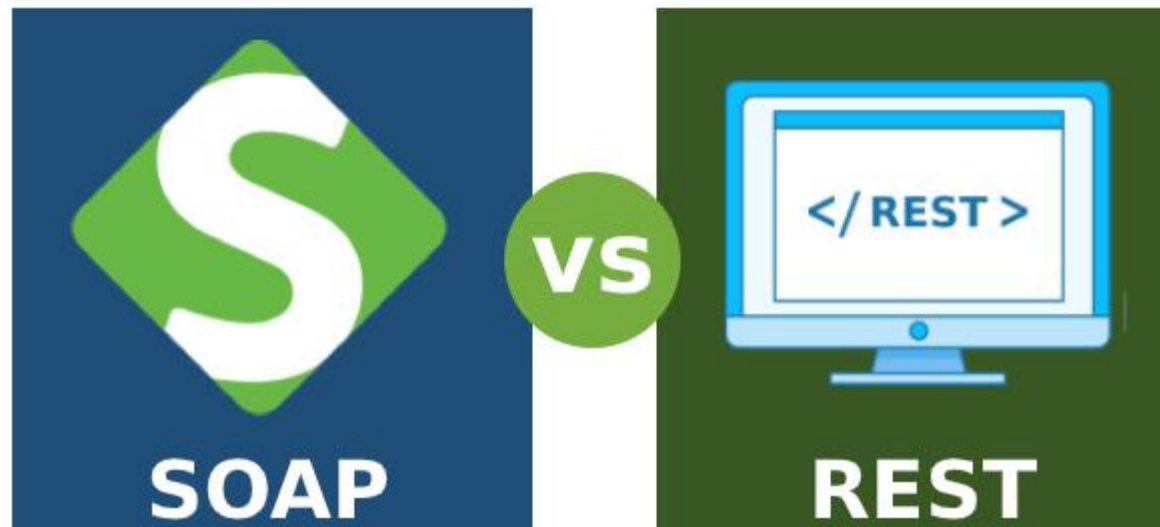
```
[{"id": 1, "name": "Leanne Graham", "username": "Bret", "email": "Sincere@april.biz", "address": {"street": "Kulas Light", "suite": "Apt. 556", "city": "Gwenborough", "zipcode": "92998-3874", "geo": {"lat": "-37.3159", "lng": "81.1496"}}, "phone": "1-770-736-8031 x56442", "website": "hildegard.org", "company": {"name": "Romaguera-Crona", "catchPhrase": "Multi-layered client server neural-net", "bs": "harness real-time e-markets"}}, {"id": 2, "name": "Ervin Howell", "username": "Bret", "email": "Sincere@april.biz", "address": {"street": "Kulas Light", "suite": "Apt. 556", "city": "Gwenborough", "zipcode": "92998-3874", "geo": {"lat": "-37.3159", "lng": "81.1496"}}, "phone": "1-770-736-8031 x56442", "website": "hildegard.org", "company": {"name": "Romaguera-Crona", "catchPhrase": "Multi-layered client server neural-net", "bs": "harness real-time e-markets"}}]
```

Тіло відповіді

Статус код

# Manual QA

## Rest та SOAP



# Manual QA

## Rest та SOAP

	REST	SOAP
Значення	Representational State Transfer — «передача стану подання»	Simple Object Access Protocol — простий протокол доступу до об'єктів
Дизайн	Архітектурний стиль із вільними рекомендаціями	Стандартний протокол із певними правилами
Підхід	Data driven - <code>http://petsmanager/pets</code>	Function driven - <code>http://petsmanager/getPets</code>
Стан	За природою, не зберігає стан (stateless)	За замовчуванням немає, але може бути реалізований із збереженням
Безпека	SSL та HTTPS	WS-Security c SSL
Продуктивність	Вимагає небагато ресурсів	Потребує більше зусиль
Формат повідомлень	XML, JSON, plain text, HTML etc.	Only XML
Протокол передачі	Only HTTP	SMTP, HTTP, UDP etc.
Рекомендується для	Громадські API для Інтернет-сервісів, соц. мережі, мобільні сервіси	Фінансові сервіси, телекомунікації, програми enterprise рівня



# Manual QA

## Rest та SOAP. Недоліки



### REST

- Працює лише поверх протоколу HTTP
- Важко забезпечити авторизацію та безпеку

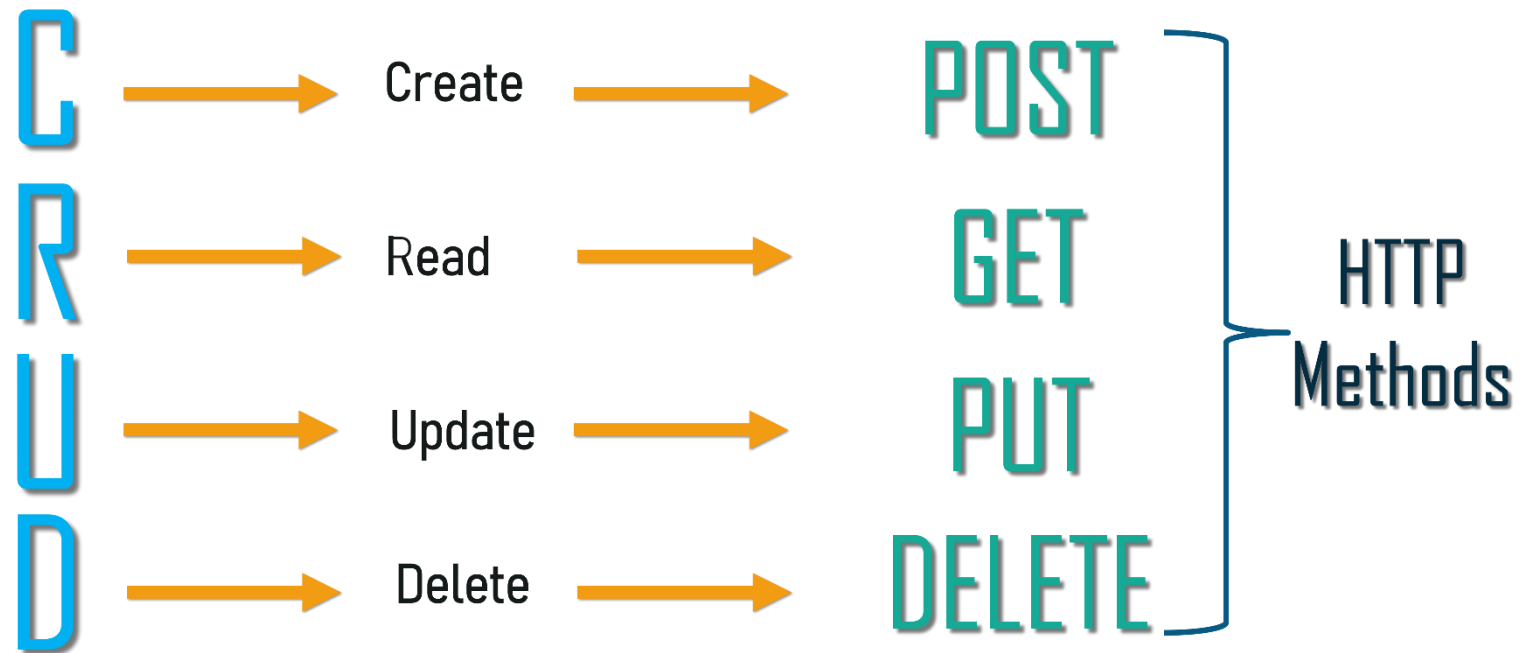


### SOAP

- Витрачає багато трафіку на обмін метаданими
- Важко реалізувати
- Використовує тільки XML

# Manual QA

## Основні методи REST



# Manual QA

## JSON та XML

### XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

### JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

# Manual QA

## Тест документація для API

GET	/pet/{petId}	Find pet by ID
PUT	/pet	Update an existing pet
DELETE	/pet/{petId}	Deletes a pet
POST	/pet/{petId}/uploadImage	uploads an image

## Матриця покриття

Endpoint	Method	Test Cases, #	Automated, #	Auto Coverage, %	Status
/pet	GET	10	10	100	
	PUT	5	2	40	
	DELETE	3	0	0	
	POST	2	0	0	

# Manual QA

## Тест документація для API



GET	/pet/{petId}	Find pet by ID
PUT	/pet	Update an existing pet
DELETE	/pet/{petId}	Deletes a pet
POST	/pet/{petId}/uploadImage	uploads an image

# Manual QA

## Тест документація для API



*Swagger* – це фреймворк та специфікація для визначення REST APIs у форматі, **дружнім до користувача та комп'ютера**.

# Manual QA

## Swagger

GET /pet/{petId} Find pet by ID

Returns a single pet

Parameters Cancel

Name	Description
<b>petid</b> * required integer(\$int64) (path)	ID of pet to return

12

Execute Clear

Responses Response content type application/json

Curly

```
curl -X GET "https://petstore.swagger.io/v2/pet/12" -H "accept: application/json"
```

Request URL

```
https://petstore.swagger.io/v2/pet/12
```

Server response

Code	Details
200	<p>Response body</p> <pre>{   "id": 12,   "category": {     "id": 1,     "name": "Dog"   },   "name": "doggie",   "photoUrls": [],   "tags": [],   "status": "available" }</pre> <span>Download</span>



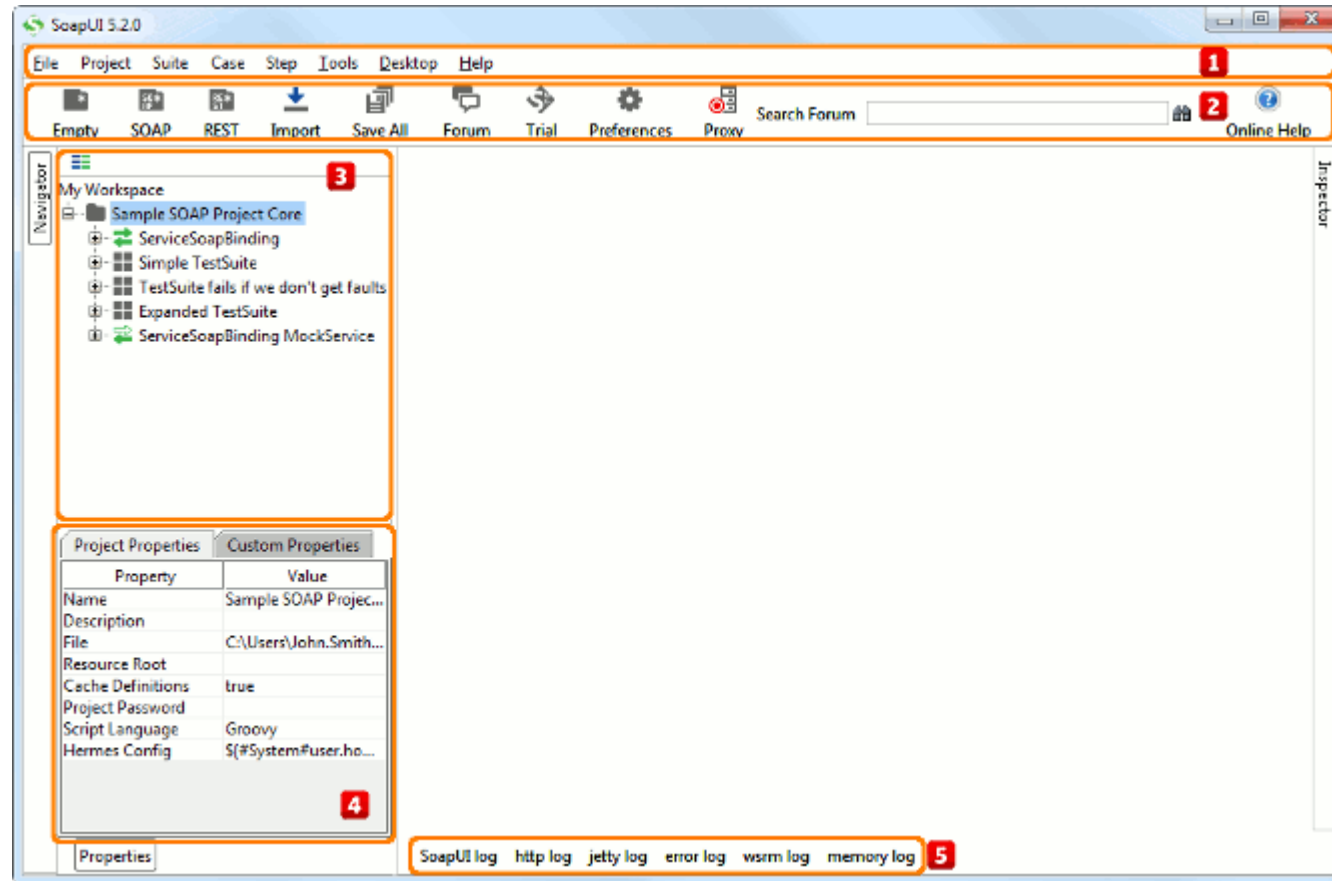
# Postman





# Manual QA

## SoapUI



# Manual QA

## SoapUI vs Postman

CHARACTERISTICS	 SoapUI	 POSTMAN
REST API Testing	Yes	Yes
SOAP API Testing	Yes	No
Automated assertion generation	Yes	Yes
Test script reusability	High	None
Data-driven support	Yes (*)	JSON, CVS
BDD Cucumber support	No	No
Environment handling	Yes (*)	Yes
Test reports	Yes (*)	Simple JSON, HTML formats
Scripting languages	Groovy	JavaScript
Web UI Testing	No	No
Mobile app testing	No	No
Test execution analytics	No	No

(\*) Only supported in the commercial edition.

# Manual QA

Що ми сьогодні вивчали

# Manual QA

## План уроку

- Основна теорія
- Get, Post, Put, Delete etc.
- REST & SOAP
- JSON & XML
- Корисна тест-документація для API
- Swagger
- Postman
- SoapUI

# Manual QA

## Підсумки

Що одне, найголовніше, ви дізнались сьогодні?

# Manual QA

Дякую за увагу! До нових зустрічей!

# Інформаційний відеосервіс для розробників програмного забезпечення

