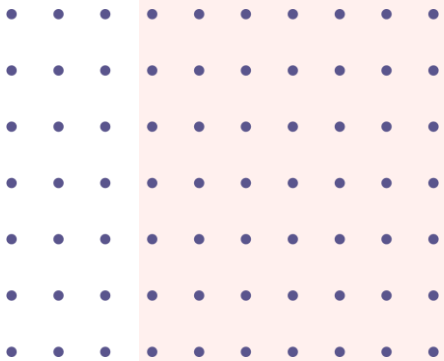


## Урок 19. Тестування API за допомогою SoapUI

# Структура заняття

1. Клієнт-серверна архітектура
2. API
3. REST vs SOAP
4. Протоколи передачі даних
5. JSON vs XML
6. Домашнє завдання



# REST vs SOAP

## REST (Representational State Transfer)

Архітектурний стиль взаємодії компонентів розподіленої програми у мережі.

REST - це набір правил того, як програмісту організувати написання коду серверного додатку, щоб усі системи легко обмінювалися даними та програму можна було масштабувати.

REST є узгодженим набором обмежень, що враховуються при проектуванні розподіленої гіпермедіа-системи. У певних випадках (інтернет-магазини, пошукові системи, інші системи, засновані на даних) це призводить до підвищення продуктивності та спрощення архітектури.

У широкому сенсі компоненти в REST взаємодіють на кшталт взаємодії клієнтів та серверів у Всесвітній павутині.

## SOAP (Simple Object Access Protocol)

Протокол обміну структурованими повідомленнями у розподіленому обчислювальному середовищі. Спочатку SOAP призначався переважно реалізації віддаленого виклику процедур (RPC). Наразі протокол використовується для обміну довільними повідомленнями у форматі XML, а не лише для виклику процедур.

SOAP може використовуватися з будь-яким протоколом прикладного рівня: SMTP, FTP, HTTP, HTTPS та ін. Проте його взаємодія з кожним із цих протоколів має свої особливості, які мають бути визначені окремо. Найчастіше SOAP використовується поверх HTTP.

SOAP є одним із стандартів, на яких базуються технології веб-служб.



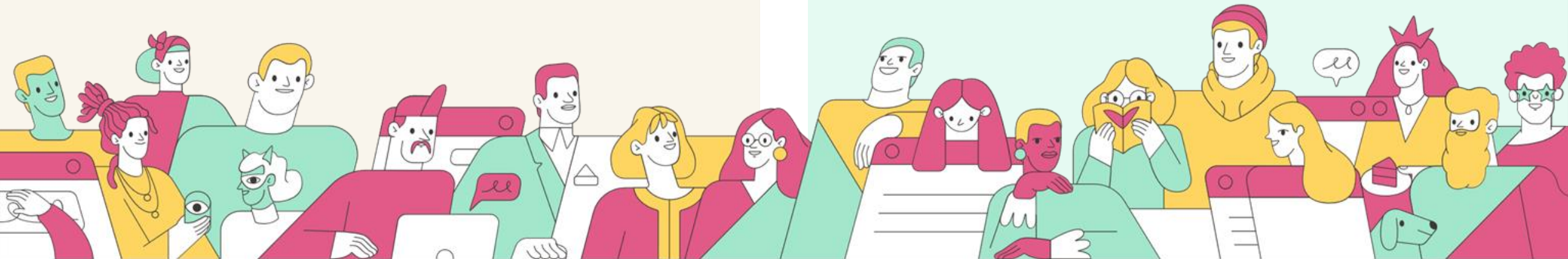
# JSON vs XML

## JSON (JavaScript Object Notation)

Текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передавання структурованої інформації через мережу.

## XML (eXtensible Markup Language)

Стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет. XML-документ складається із текстових знаків, і придатний до читання людиною.



# JSON vs XML

## JSON (JavaScript Object Notation)

JSON будується на двох структурах:

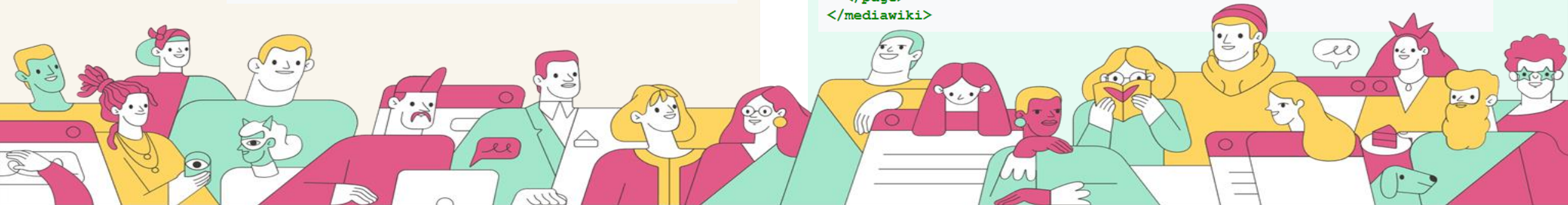
- Набір пар назва/значення. У різних мовах програмування це реалізовано як об'єкт, запис, структура, словник, хеш-таблиця, список із ключем або асоціативним масивом.
- Впорядкований список значень. У багатьох мовах це реалізовано як масив, вектор, список або послідовність.

```
{  
  "firstName": "Іван",  
  "lastName": "Коваленко",  
  "address": {  
    "streetAddress": "вул. Грушевського 14, кв.101",  
    "city": "Київ",  
    "postalCode": 21000  
  },  
}
```

## XML (eXtensible Markup Language)

Стандарт визначає набір базових лексичних та синтаксичних правил для побудови мови описання інформації шляхом застосування простих тегів.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<mediawiki xmlns="http://www.mediawiki.org/xml/export-0.3/" xml:lang="uk">  
  <page>  
    <title>Фукідід</title>  
    <id>1529</id>  
    <revision>  
      <id>4382</id>  
      <timestamp>2006-09-18T22:11:53Z</timestamp>  
      <minor />  
      <comment>Interwiki</comment>  
      <text xml:space="preserve">{{Wikipedia}}  
      *Історія — це філософія в прикладах.  
    </text>  
    </revision>  
  </page>  
</mediawiki>
```



# XML

`<?xml version="1.1" encoding="UTF-8" ?>` - XML пролог, необов'язковий (але якщо він є, то це повинен бути перший рядок документа XML). У документі можуть бути міжнародні символи, тому необхідно вказувати кодування, або зберегти XML файл у форматі UTF-8 (кодування XML документів за замовчуванням).

Документ повинен містити один елемент, який буде батьківським для всіх інших елементів – кореневий елемент.

`<composition>` - батьківський/кореневий елемент

`<ingredient>`Мука`</ingredient>` - дочірній елемент

`<ingredient>`Дрожжи`</ingredient>`

`<ingredient>`Тёплая вода`</ingredient>`

`</composition>` - закриття батьківського/кореневого елемента

У XML всі елементи повинні дотримуватися коректної вкладеності:

`<b><i>`Тёплая вода`</i></b>`



# XSD

**XML Schema Definition** – спосіб описання типу XML документу, як правило, визначається шляхом введення обмежень на структуру та зміст документів заданого типу на додаток до базових синтаксичних обмежень самого формату XML.

Опис типів даних та їх обмеження

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="country">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="country_name" type="xs:string"/>
        <xs:element name="population" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

# Для тестування SOAP

**SOAP-повідомлення (request)** – це звичайний XML-документ, що містить елементи:

- **Конверт (Envelope)** – обов'язковий елемент, що визначає початок та кінець повідомлення
- **Заголовок (Headers)** – необов'язковий елемент, містить будь-які необов'язкові атрибути, що використовуються при обробці повідомлення, або в проміжній точці.
- **Тіло (Body)** – обов'язковий елемент, містить дані XML, що містять повідомлення, яке надсилається
- **Несправність (Faultcode)** – необов'язковий елемент несправності, який надає інформацію про помилки, що виникають під час обробки повідомлення.



# SOAP-повідомлення (request)

```
<? xml version = "1.0" ?> <SOAP-ENV:Envelope xmlns:SOAP-ENV = "http://www.w3.org/2001/12/soap-
env" >

  <SOAP-ENV:Header>
    ...
  </SOAP-ENV:Header> <SOAP-ENV:Body>

    ...
    <SOAP-ENV:Fault>
      ...
    </SOAP-ENV:Fault>
    ...
  </SOAP-ENV:Body> </SOAP-ENV:Envelope>
```





# Для тестування SOAP

**Web Services Description Language, WSDL** – мова опису інтерфейсів вебсервісу заснована на XML, що описує функціональність вебсервісу і спосіб доступу до нього.

В WSDL документі описуються доступні методи (опції), а також опис параметрів для кожного окремого запиту та відповіді.



Домашнє завдання