**Теоретичне завдання**

**1. Що таке REST API?**

REST API − це архітектурний стиль для створення веб-сервісів, які дозволяють взаємодіяти з сервером за допомогою HTTP-запитів. У REST API, сервер надає доступ до ресурсів через уніфікований інтерфейс, що складається з HTTP методів, таких як GET, POST, PUT та DELETE, а також використовуючи стандартні формати даних, такі як JSON або XML.

REST API розробляється з декількох принципів, таких як: клієнт-серверна архітектура, відсутність стану, кешування: клієнти можуть кешувати відповіді сервера для зменшення кількості запитів, одноманітність інтерфейсу і шаровість. Ці принципи дозволяють створювати легкі та ефективні веб-сервіси, які можна використовувати для різних типів додатків та платформ.

**2. У чому різниця між REST і RESTfull?**

REST є архітектурним стилем, що включає в себе набір принципів, які визначають, як взаємодіяти з ресурсами в мережі. Ці принципи включають в себе використання HTTP-методів, таких як GET, POST, PUT і DELETE, для взаємодії з ресурсами, а також використання URI для ідентифікації ресурсів, які можна отримати або змінити.

RESTful є веб-сервісом, що дотримується принципів REST. Це означає, що веб-сервіс повинен використовувати HTTP-методи і URI для взаємодії з ресурсами. Інші характеристики RESTful веб-сервісів включають використання формату даних, який підтримується клієнтом та сервером, використання кешування для зменшення навантаження на сервер, і незалежність стану, тобто, клієнт повинен включати всю необхідну інформацію для взаємодії з сервером у кожному запиті, без необхідності зберігати стан на сервері.

**3. Які основні HTTP методи використовують для розробки WebAPI? Опишіть призначення кожного.**

**GET** − використовується для отримання даних з сервера. Клієнт надсилає запит на сервер і отримує відповідь від нього.

**POST** − використовується для створення нових записів на сервері. Клієнт надсилає дані на сервер, які використовуються для створення нового ресурсу.

**PUT** − використовується для оновлення існуючих записів на сервері. Клієнт надсилає дані на сервер, які використовуються для оновлення існуючого ресурсу.

**DELETE** − використовується для видалення існуючих записів на сервері. Клієнт надсилає запит на сервер для видалення ресурсу.

Також для розробки RESTful WebAPI використовуються:

**PATCH** − використовується для часткового оновлення існуючих записів на сервері. Клієнт надсилає дані на сервер, які використовуються для часткового оновлення існуючого ресурсу.

**HEAD** − використовується для отримання заголовків відповіді сервера, але без тіла відповіді.

**OPTIONS** − використовується для отримання списку підтримуваних сервером методів та параметрів конфігурації.

**4. У чому різниця між REST API та JSON API?**

REST API та JSON API − це два різних підходи до розробки веб-сервісів.

REST API − це стиль архітектури веб-сервісів, який використовує HTTP протокол для створення легких, масштабованих та розширюваних систем. REST API передає дані між клієнтом та сервером у вигляді ресурсів, використовуючи стандартні HTTP методи, такі як GET, POST, PUT та DELETE.

JSON API − це стандарт, який визначає підхід до створення веб-сервісів, який базується на REST архітектурі та використовує формат JSON для передачі даних між клієнтом та сервером. JSON API встановлює стандарти для того, як повинні виглядати запити та відповіді сервера, і включає рекомендації для того, як повинні бути представлені дані, включаючи метадані та структуру відносин між ресурсами.

Основна різниця між REST API та JSON API полягає в тому, що JSON API встановлює стандарти для того, як повинні бути представлені дані, в той час як REST API не обмежує формат передачі даних та не накладає жорстких правил щодо використання метаданих та структури відносин між ресурсами. Також JSON API має певні переваги в порівнянні з REST API, такі як підвищена читабельність, простота та стандартизація. Однак, REST API є більш гнучким та може бути більш пристосованим до конкретних потреб розробника.

**5. Які основні принципи для розробки REST API?**

**Клієнт-серверна архітектура** − сервіс повинен бути розділений на клієнтську та серверну частини, які можуть розвиватись незалежно одна від одної.

**Використання HTTP-протоколу** − REST API повинно використовувати HTTP-протокол для обміну даними між клієнтом та сервером.

**Безстанність** − сервер не зберігає жодної інформації про клієнта між запитами. Кожен запит вважається окремим запитом без стану.

**Кешування** − сервер повинен надавати можливість кешування відповідей на запити для покращення швидкодії та зменшення навантаження на сервер.

**Використання URI** − усі ресурси повинні бути доступні через унікальний URI.

Використання HTTP **методів** − REST API використовує HTTP методи для доступу до ресурсів: GET для отримання даних, POST для створення нового ресурсу, PUT для оновлення існуючого ресурсу та DELETE для видалення ресурсу.

**Передача даних у вигляді JSON або XML** − REST API повинно використовувати JSON або XML для передачі даних між клієнтом та сервером.

**Самоописність** − сервіс повинен надавати достатньо інформації для того, щоб клієнт міг використовувати його, не звертаючись до розробника.

**Глобальна доступність** − сервіс повинен бути доступним з будь-якого місця та будь-якого пристрою, що має доступ до Інтернету.