

## **ПРИНЦИП РОБОТИ WEB API ТА ЙОГО ЗАСТОСУВАННЯ**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Web API - це інтерфейс програмування застосувань, який дає можливість взаємодіяти з веб-сервером з боку клієнта. Веб-сервер може надавати різні функції через API, такі як: доступ до бази даних, відправка та отримання даних, доступ до ресурсів та інше. Зазвичай Web API надаються у вигляді HTTP-сервісів та можуть використовувати формати передачі даних, такі як JSON або XML. Web API може бути використаний у різних веб-додатках та сервісах, таких як: соціальні мережі, електронна комерція, фінансові додатки, додатки для моніторингу та інше. Наприклад, соціальна мережа може надавати API для отримання даних користувачів, відправки повідомлень, створення та редагування постів. Електронний магазин може надавати API для оплати замовлень, створення каталогів товарів та іншого. Веб-розробники можуть використовувати Web API для створення власних додатків та сервісів, що взаємодіють з різними веб-серверами та отримують різні функціональні можливості. Це дозволяє розробникам створювати веб-додатки зі складною функціональністю, без необхідності розробляти всі функції самостійно. Web API може бути розроблений на різних мовах програмування, таких як JavaScript, PHP, Python та інші.

Клієнтська сторона веб-додатка може взаємодіяти з Web API через AJAX-запити або за допомогою бібліотек та фреймворків, таких як: jQuery, AngularJS, React та інші.

Для того щоб використовувати Web API, розробники повинні зрозуміти, як працює API, які функціональні можливості доступні через API, як взаємодіяти з API та які формати даних підтримуються. Для цього можна використовувати документацію API, яку надає веб-сервер, або використовувати спеціальні сервіси, такі як Swagger, які автоматично генерують документацію на основі коду API. Одним з найбільш поширених форматів даних для передачі через Web API є JSON.

JSON є легким та простим у використанні форматом даних, який підтримується більшістю мов програмування та додатків. Для передачі даних через Web API можна використовувати HTTP-запити, такі як: GET, POST, PUT та DELETE. Для забезпечення безпеки Web API можуть використовувати різні методи аутентифікації та авторизації, такі як OAuth, JWT та інші. Ці методи забезпечують контроль доступу до API та зменшують ризик зловживання з боку зловмисників.

Web API можна використовувати у різних сферах, наприклад:

1. Соціальні мережі: Facebook, Twitter, Instagram та інші соціальні мережі надають Web API для розробників. Завдяки цьому розробники можуть інтегрувати функціональність соціальних мереж в свої додатки та сервіси.

2. Фінансові сервіси: PayPal, Stripe та інші фінансові сервіси надають Web API для операцій з оплатою та грошовими переказами.

3. Карти та геодані: Google Maps, Bing Maps та інші сервіси надають Web API для отримання геоданих та роботи з картами.

4. Хмарні сервіси: Amazon Web Services, Microsoft Azure та інші хмарні сервіси надають Web API для керування інфраструктурою в хмарі та роботи зі зберіганням даних.

5. Інтернет-магазини: Amazon, eBay та інші інтернет-магазини надають Web API для роботи зі списками товарів та замовлень.

6. Інтернет-транспорт: Uber, Lyft та інші сервіси надають Web API для замовлення транспорту та отримання геоданих.

Web API можуть бути синхронними та асинхронними. Синхронні API надсилають відповідь тільки після того, як отримають запит. Асинхронні API надсилають відповідь, як тільки отримають запит, але необхідно використовувати механізм, щоб отримати відповідь після завершення операції.

Можна виділити різні підходи до реалізації Web API. Одним з підходів є REST (Representational State Transfer) - архітектурний стиль, який використовує HTTP-запити та формат даних JSON для передачі даних між сервером та клієнтом. Другим підходом є SOAP (Simple Object Access Protocol) - протокол обміну повідомленнями, який використовує XML для передачі даних між сервером та клієнтом. У сучасній веб-розробці REST-архітектура є досить популярним підходом до реалізації Web API. REST-сервіси передають запити та відповіді у форматі JSON, що робить їх більш простими для розуміння та використання в порівнянні з SOAP-сервісами.

Одним з важливих аспектів Web API є документація, яка надається розробникам. Документація містить опис методів, що надаються сервісом, параметри запитів та формат відповідей. Зазвичай, документація надається в форматі Swagger або OpenAPI, що дозволяє автоматично генерувати код на різних мовах програмування для взаємодії з сервісом.

Також важливо забезпечити безпеку Web API. Це можна зробити за допомогою автентифікації та авторизації запитів. Автентифікація полягає у перевірці ідентичності користувача, який надсилає запит, за допомогою токенів або паролів. Авторизація визначає, чи має користувач право на виконання запиту, і базується на ролях та дозволах.

З Web API пов'язані різні технології та інструменти, які можуть забезпечити його ефективну реалізацію. Наприклад, бібліотеки для розробки Web API на різних мовах програмування, такі як Flask та Django в Python, ExpressJS в JavaScript, Ruby on Rails в Ruby, або ASP.NET в C#, можуть спростувати процес створення та підтримки Web API.

Окрім того, існує ряд інструментів для тестування Web API. Наприклад, Postman дозволяє відправляти запити до сервісу та перевіряти відповіді на запити. Інші інструменти, такі як Swagger UI та OpenAPI, надають можливість тестувати Web API та переглядати його документацію.

У світі Web API існує також ряд стандартів, які регулюють взаємодію між клієнтом та сервером. Наприклад, HTTP (Hypertext Transfer Protocol) визначає, як клієнт і сервер взаємодіють між собою через мережу Інтернет. Інші стандарти, такі як JSON (JavaScript Object Notation), XML (Extensible Markup Language) та HTML (Hypertext Markup Language), визначають формати даних, які можуть передаватись через Web API.

Загалом, Web API - це важлива технологія, яка забезпечує ефективний спосіб обміну даними між різними сервісами та додатками в Інтернеті. Зростання популярності цієї технології свідчить про те, що вона має великий потенціал для використання в різних галузях та проектах веб-розробки.

#### Список літератури

1. [Leonard R., Mike A., Sam R. RESTful Web APIs: 2013p.406c. URL: https://sd.blackball.lv/library/RESTful\\_Web\\_APIs\\_\(2013\).pdf](https://sd.blackball.lv/library/RESTful_Web_APIs_(2013).pdf)
2. [Brenda J. Designing Web APIs Building APIs That Developers Love: 2018,232c URL : https://www.academia.edu/43452338/Designing\\_Web\\_APIs\\_BUILDING\\_APIS\\_THAT\\_DEVELOPERS\\_LOVE](https://www.academia.edu/43452338/Designing_Web_APIs_BUILDING_APIS_THAT_DEVELOPERS_LOVE)
3. [Sam N. Building Microservices Designing Fine-Grained Systems: 2015 ,280c URL:https://github.com/rootusercop/Free-DevOps-Books-1/blob/master/book/Building%20Microservices%20-%20Designing%20Fine-Grained%20Systems.pdf.](https://github.com/rootusercop/Free-DevOps-Books-1/blob/master/book/Building%20Microservices%20-%20Designing%20Fine-Grained%20Systems.pdf)
4. [Литвинюк В.С., Січко Т.В. Особливості створення веб-додатків та веб-сайтів за допомогою технології BOOTSTRAP 4. Прикладні інформаційні технології: матеріали всеукр. наук.-практ. конф., м. Вінниця, 2020. С. 111-113.](#)