

Прикладний програмний інтерфейс

Прикладний програмний інтерфейс (інтерфейс програмування застосунків, інтерфейс прикладного програмування, API) (англ. Application Programming Interface, API) — набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Спрощено — це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного забезпечення. API може бути для веббазованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Щоразу, коли ви користуєтеся програмою пошуку поїздок (наприклад, BlaBlaCar), дивитесь прогноз погоди або надсилаєте мобільний платіж ви використовуєте API. Ці програми підключаються до Інтернету та надсилають дані на сервер. Потім сервер отримує ці дані, інтерпретує їх, виконує необхідні дії та надсилає назад на ваш телефон. Потім програма інтерпретує ці дані та надає вам потрібну інформацію у зрозумілому вигляді.

Прикладний програмний інтерфейс (API)

це готові конструкції мови програмування, що дозволяють розробнику будувати складну функціональність із меншими зусиллями. Вони “приховують” складніший код від програміста, забезпечуючи простоту використання.

Для кращого розуміння розглянемо аналогію із домашніми електромережами. Коли ви хочете використовувати якийсь електроприлад, ви просто підключаєте його до розетки і все працює. Ви не намагаєтеся підключити дроти безпосередньо до джерела струму – робити це марно і, якщо ви не електрик, складно та небезпечно.

Термін «API» зазвичай використовується для опису інтерфейсів підключення до програми. Проте з роками сучасний API набув деяких унікальних характеристик, які справді змінили технологічний простір. По-перше, сучасні API відповідають певним стандартам (зазвичай HTTP і REST), які дозволяють API бути зручними для розробників.

REST API

Більшість бізнес-додатків повинні взаємодіяти з іншими внутрішніми та сторонніми програмами для виконання різних завдань. Наприклад, щоб генерувати щомісячні платіжні відомості, ваша внутрішня бухгалтерська система повинна обмінюватися даними з банківською системою вашого клієнта, щоб автоматизувати виставлення рахунків та взаємодіяти з внутрішнім додатком для обліку робочого часу. RESTful API підтримують такий обмін інформацією, оскільки вони дотримуються безпечних, надійних та ефективних стандартів програмної взаємодії.

Інтерфейс прикладного програмування (API) визначає правила, які необхідно дотримуватися для зв'язку з іншими програмними системами. Розробники впроваджують або створюють API-інтерфейси, щоб інші програми могли програмно взаємодіяти з їхніми програмами. Наприклад, програма з таблицею робочого часу містить API, який запитує повне ім'я співробітника та діапазон дат. Отримавши цю інформацію, інтерфейс внутрішньо обробляє таблицю робочого часу

співробітника та повертає кількість годин, відпрацьованих за вказаний період.

Таким чином, мережний API функціонує як шлюз між **клієнтами** та **ресурсами** в Інтернеті.

Клієнти

Клієнти – це користувачі, які бажають отримати доступ до інформації в Інтернеті. Клієнтом може бути людина чи програмна система, що використовує API. Наприклад, розробники можуть створювати програми, які отримують доступ до даних погоди з метеосистеми. Також можна отримати доступ до цих даних з браузера, відвідавши веб-сайт з інформацією про погоду.

Ресурси

Ресурси – це інформація, яку різні програми надають своїм клієнтам. Ресурси можуть бути зображеннями, відео, текстом, числами чи даними будь-якого типу. Комп'ютер, який надає ресурси клієнту, також називається сервером. API дозволяє організаціям спільно використовувати ресурси та надає веб-служби, забезпечуючи безпеку, контроль та автентифікацію. Крім того, API допомагає визначити, які клієнти можуть отримати доступ до певних внутрішніх ресурсів.

Що таке REST?

Representational State Transfer (REST) – це програмна архітектура, яка визначає умови роботи API. Спочатку REST створювалася як керівництво для управління взаємодіями у складній мережі, такій як Інтернет. Архітектуру на основі REST можна використовувати для підтримки високопродуктивного та надійного зв'язку в необхідному масштабі. Її можна легко впроваджувати та модифікувати, забезпечуючи прозорість та кросплатформну переносимість будь-якої системи API.

Розробники можуть створювати API з використанням кількох архітектур. API-інтерфейси, які відповідають архітектурному стилю REST, називаються REST API. Веб-служби, що реалізують архітектуру REST, називаються веб-службами RESTful. Як правило, термін RESTful API відноситься до мережевих RESTful API. Однак REST API та RESTful API є взаємозамінними термінами.

Нижче наведено деякі принципи архітектурного стилю REST:

Єдиний інтерфейс

Єдиний інтерфейс є конструктивною основою будь-якого веб-сервісу RESTful. Це свідчить про те, що сервер передає інформацію у стандартному форматі. Наприклад, сервер може зберігати дані у вигляді тексту, але надсилати їх у форматі подання HTML.

Відсутність збереження стану

В архітектурі REST відсутність збереження стану відноситься до методу зв'язку, при якому сервер виконує кожен запит клієнта незалежно від усіх попередніх запитів. Клієнти можуть звертатися до ресурсів у будь-якому порядку, і кожен запит або ізольований від інших запитів,

або його стан не зберігається. Це конструктивне обмеження REST API передбачає, що сервер може щоразу повністю зрозуміти та виконати запит.

Багаторівнева система

У багаторівневій системній архітектурі клієнт може підключатися до інших авторизованих посередників між клієнтом та сервером і, як і раніше, отримувати відповіді від сервера. Сервери також можуть надсилати запити іншим серверам. Ви можете спроектувати свою веб-службу RESTful для роботи на декількох серверах з кількома рівнями (безпекою, додатками та бізнес-логікою), які спільно виконують клієнтські запити. Ці рівні залишаються невидимими для клієнта.

Місткість кешу

Веб-служби RESTful підтримують кешування, тобто процес збереження деяких відповідей на клієнта або посередника для скорочення часу відповіді сервера. Наприклад, ви заходите на веб-сайт із загальним зображенням верхнього та нижнього колонтитулів на кожній сторінці. Щоразу, коли ви відвідуєте нову сторінку веб-сайту, сервер повинен повторно надсилати ті самі зображення. Щоб уникнути цього, клієнт кешує або зберігає ці зображення після першої відповіді, а потім використовує зображення з кешу. Веб-служби RESTful керують кешуванням за допомогою відповідей API, які визначають себе як кешовані або некашовані.

Код за запитом

В архітектурному стилі сервери REST можуть тимчасово розширювати або налаштовувати функціональні можливості клієнта, передаючи код програмного забезпечення. Наприклад, коли ви заповнюєте реєстраційну форму на будь-якому веб-сайті, ваш браузер відразу ж виділяє всі допущені помилки (наприклад, неправильні номери телефонів). Це відбувається завдяки коду, надісланому сервером.

Джерела

- [Прикладний програмний інтерфейс](#)
- [REST](#)
- [What is an API? \(Application Programming Interface\) | MuleSoft](#)
- [Introduction to web APIs - Learn web development | MDN](#)
- [What is RESTful API? - RESTful API Beginner's Guide - AWS](#)

From:

<https://library.vpuhlukhiv.com.ua/> - **Вікі Глухівського ВПУ**

Permanent link:

<https://library.vpuhlukhiv.com.ua/subjects:basic:informatika:web:api>

Last update: **11.01.2023 00:24**

