Герасимчук В.В., магістр,
Єфремов Ю.М., к.т.н., доцент кафедри,
Житомирський державний технологічний університет

ПРИКЛАДНИЙ ПРОГРАМНИЙ ІНТЕРФЕЙС (API), ЯК ІНСТРУМЕНТ РОЗРОБКИ

В даному дослідженні представлено результати проведеного аналізу прикладного програмного інтерфейсу (API) та обґрунтовано його використання в своїх проектах, незважаючи на розмір проекту та особистих навиків розробника.

По-перше, використання прикладного програмного інтерфейсу (API) дозволяє використовувати наявну логіку, яку вам не потрібно писати. Деякі речі, які ви просто не можете розробити самі! Щоб зекономити дорогоцінний час як розробника, важливо мати уявлення про те, як виглядає API.

По-друге, багато проблем з розробкою, з якими ви зіткнетесь, вже були вирішені іншими розробниками в межах прикладного програмного інтерфейсу (API). Це існуючі рішення – це FaaS, бібліотеки, веб-сервіси, SDK, API-вміст. Незалежно від форми, яку вони вживають, вам, швидше за все, потрібен API для взаємодії з ними.

Дефініція поняття API може бути представлена наступним чином: в комп'ютерному програмуванні, інтерфейс прикладного програмування (API) – це набір визначень підпрограм, протоколів та інструментів для створення прикладного програмного забезпечення.

Загалом, це сукупність чітко визначених методів зв’язку між різними програмними компонентами. Хороший API полегшує розробку комп’ютерної програми, надаючи всі будівельні блоки, які потім складаються програмістом .

Простіше кажучи, API оголошує інтерфейс для взаємодії з його логікою без необхідності знати, що «відбувається під капотом». Це визначення стосується будь-якої мови, протоколу чи середовища, в якому ви працюєте.

Єдина вимога полягає в тому, що це відбувається на програмному рівні. Щоб пролити світло на API-інтерфейс, давайте перерахуємо чим він не являється:

- API не обов'язково є зовнішньою службою. Наприклад, ви можете включити бібліотеки безпосередньо в своє рішення або використовувати їх через API.
- API - це не просто інтерфейс. Це як специфікація/формат, так і реалізація.
- API не GUI (графічний інтерфейс користувача). API не взаємодіє на графічному рівні. Він виключно працює на програмному прошарку. Це може бути або через мову програмування, або через протокол зв'язку.

Усі API не створюються рівними. Незважаючи на те, що в основному вони поєднують однакову мету, деякі досягають цього шляху краще, ніж інші. Метою API є полегшити ваше життя як розробника.

Поєднанням особливостей / функцій і виявлення цих функцій через кінцеві точки. Це, як правило, шаблони URL, які використовуються для спілкування з API. Ці кінцеві точки є єдиним способом взаємодії з будь-яким API. Кожна кінцева точка матиме певний формат для своїх запитів і відповідей. Зазвичай цей формат визначається у документації API. Кінцеві точки можуть бути простими функціями. Або вони можуть складатися з багатьох функцій, які можуть викликати інші інтерфейси API тощо.

Вирішальний момент полягає в тому, що логіка цих функцій повністю абстрагована. Вам не потрібно знати про те, що відбувається всередині них, щоб їх використовувати.

Поки ви використовуєте правильний формат, ви зможете їх споживати. Тут «споживати» є привабливим способом сказати «використовувати частини їх для вашої програми».

Суть полягає в тому, що API, такий же як і будь-який інший інтерфейс. Для порівняння, перемикач світла вмикає світло, навіть якщо ви не знаєте як працює електроенергія.

Висновки:

- API прискорить вашу швидкість та розширить обсяг вашої розробки.
- API не обов'язково пов'язаний з веб-екосистемою.
- Завжди перевіряйте документи API, який ви хочете використовувати.
- Завжди шукайте існуючі інструменти (API або інше) у своїй екосистемі, перш ніж почати кодувати.