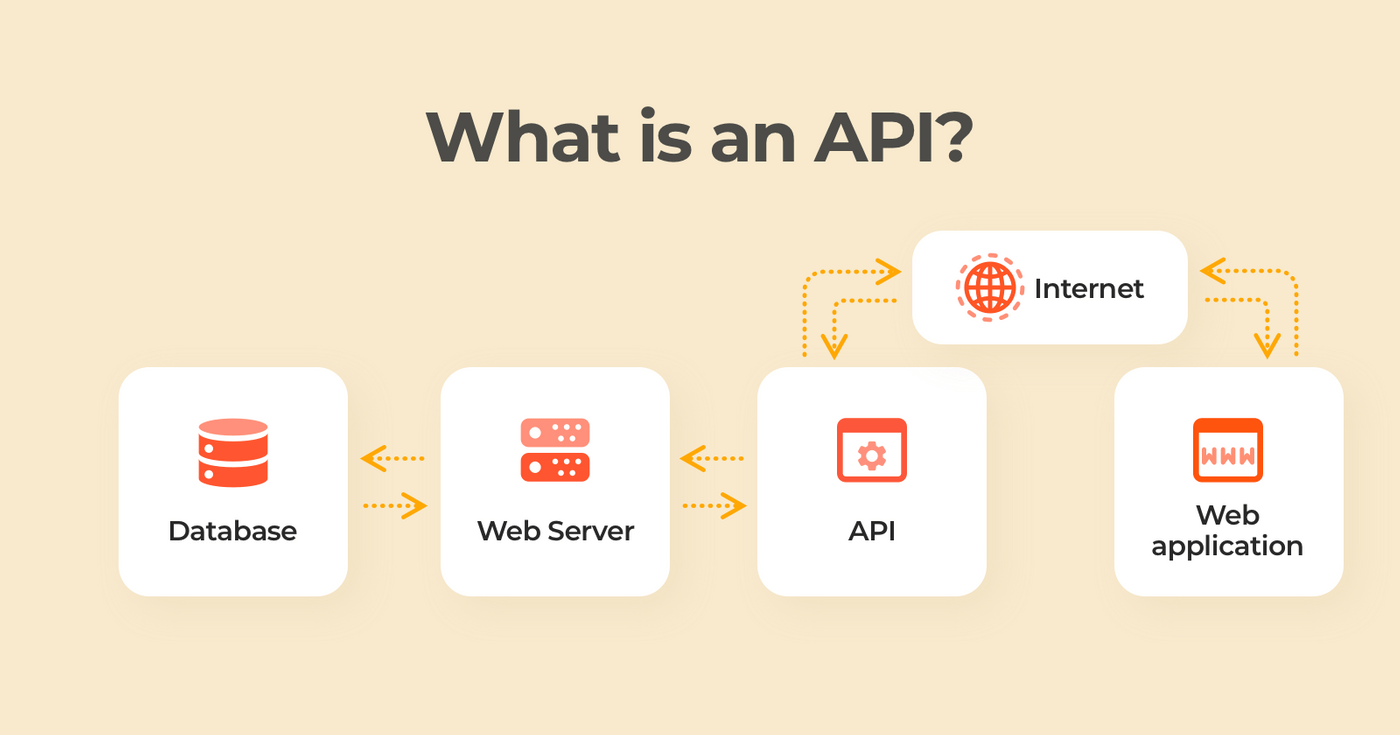
Introduction to API’s

**Application Programming Interface**

**By Eli Moser**



November 2021 (Ver. 1.000)

[Chapter 1: Introduction 3](#_Toc87459804)

[Chapter 2: Implementation 4](#_Toc87459805)

[2.1 Example 4](#_Toc87459806)

# Introduction

In recent years the concept of API’s is gaining popularity, you may have not noticed  
but they are literally everywhere.  
Have you ever wondered how Facebook is able to display your Instagram photos?   
How about the latest tweet from your favorite celebrity appearing in a news article?  
Or the latest price of a stock or crypto coin appearing in an article?  
Your friend sends you his live location on a map that appears in your chat app.  
  
How is all that possible?   
  
The answer: Using an ‘Application Programming Interface’ or **API**,

Basically using an API is a way for companies to provide information to other computers who can now access their information and services.  
A high percentage of the data on the internet is passed through API’s.

Let’s say you want an app or a website that provides the cheapest flights and hotels available (truly an original concept…).  
  
The way to get that information would be to manually check for the cheapest option in all the airlines and Hotels websites and post it in our app for each search - obviously this is very time consuming and not a good option when the app becomes popular (not scalable).

Here comes the API and allows us to connect to all the service providers we want, all we need is to send a request to the API and it will return all the information in a way we could automatically display in our app. Fantastic!

Many companies provide API’s so that customers could access their information, Google, Facebook, Tweeter and many other service providers have API’s and that’s how your Instagram photos could appear on Facebook.  
Your weather app on your phone is constantly being updated through an API as well. Cool stuff, now let’s see how it’s done in the real world.

# Implementation

API’s works with Requests and Responses

Request - consists of:

* URL (http://…) – The server address where the database is located,
* Method (GET, POST, PUT, DELETE).  
  GET - used to get a resource from a server.  
  POST - used to create a new resource on a server.  
  PUT - used to update a resource on a server.  
  DELETE - used to delete a resource from a server.
* List of headers - used to provide information to both the client and server. It can be used for many purposes, such as authentication and providing information about the body content.
* Body (data).

Response – the response from the server in a JSON format.

## Example

We will send a GET request to Skyscanner’s API to access available cheap flights.

<https://partners.api.skyscanner.net/apiservices/uk/us/anytime/anytime?apikey=prtl67493879>

We could see here the first part is the address:



By default the method is set to GET.  
The list of headers: the details of the flight locations and dates we are looking for, in our case UK-US from anytime to anytime:   
There is also an API key for authentication – it could be seen in the URL

The result is a long list of flights in a JSON format:



Press on the [Link](https://partners.api.skyscanner.net/apiservices/browsequotes/v1.0/FR/eur/en-US/uk/us/anytime/anytime?apikey=prtl6749387986743898559646983194) to see more. .