



## News & Insights

### News & Insights Home

Innovation

IT Careers & Skills

Cloud

Cyber Security

Future of Work

### All Categories

Marketing

HR

Finance

### Community

[Ask question](#)

### Community Home

Spiceworks Originals

Cloud

Collaboration

Networking

Water Cooler

Windows

### All forums

How-Tos

Scripts

Vendors

Meetups

[Reviews](#)

[Online Events](#)

[Login](#) [Join](#)



[Login](#) [Join](#)

We encourage you to read our updated [PRIVACY POLICY](#). [x](#)



DevOps

# What Is an API (Application Programming Interface)? Meaning, Working, Types, Protocols, and Examples

*An application programming interface (API) is a mediator that enables applications to communicate with each other.*



Hossein Ashtari Technical Writer

Last Updated

***An application programming interface (API) is defined as a software mediator that allows enterprises leverage***

We encourage you to read our updated [PRIVACY POLICY](#).

×

***access to the functionality and data of their applications. This article covers the meaning, working, types, protocols, and examples of APIs.***

## Table of Contents

- [What Is an API \(Application Programming Interface\)?](#)
- [How Does an API Work?](#)
- [Types of APIs](#)
- [List of API Protocols](#)
- [Examples of API](#)

# What Is an API (Application Programming Interface)?

An application programming interface (API) is a ‘software mediator’ that enables applications to communicate with one another. APIs give internal and external developers access to a secure, documented interface that they can use to leverage the functionality and data of enterprise applications.

In the last ten years, numerous business models that rely heavily on APIs have been established. This has led to the creation of an ‘API economy,’ a term that refers to businesses using APIs and [microservices](#) to make their services and data more scalable, accessible, and ultimately profitable. As a result, many software applications that are a mainstay in our lives today rely on APIs to function.

We encourage you to read our updated [PRIVACY POLICY](#). 


APIs simplify the development of new tools and the management of existing ones. They also enhance collaboration by allowing enterprises to connect the thousand of otherwise disjointed applications they use daily. Organizations also use API integration for workflow automation—a reliable way to boost productivity.

However, that's not all. In today's fast-paced market, enterprises must keep innovating to stay relevant and profitable. APIs simplify the innovation process by giving organizations the flexibility to enhance their products with solutions created by external development teams. This lets them bring new offerings to the clientele and also allows them to access new markets.

Another benefit of using APIs is the added layer of [data security](#) between servers and enterprise data. One can further fortify this layer through signatures, tokens, and transport layer security (TLS) encryption. Developers can also implement API gateways for traffic authentication and management.

Companies in the API economy sometimes choose to offer their APIs for free temporarily. This attracts developers and makes the API an integral part of the applications that need its functionality. Once the company has established its presence in its target market, it can then choose to charge a fee for access to its API, thus opening a new revenue stream for itself. Stripe and AccuWeather are two examples of companies striking it rich in the API economy.

**Stripe**, a leading online payment processing company, started as an API with only seven lines of code. With time, it forged partnerships with leading enterprises, diversified into corporate cards and loans, and is today valued at approximately \$95 billion.

Similarly, **AccuWeather** released a self-service portal that offered numerous API packages for sale to developers. In just ten months and sold 11,000 packages. We encourage you to read our updated [PRIVACY POLICY](#). 

## What does API stand for?

API is an acronym for **Application Programming Interface**, a ‘software mediator’ that enables applications to communicate with each other.

---

See More: [What Is Serverless? Definition, Architecture, Examples, and Applications](#)

## How Does an API Work?

Just like a user interface (usually a graphical user interface or GUI) is built for humans to interact with applications, APIs are built for applications to interact with each other.

The application programming interface layer is responsible for data transfer and processing between a server and an application. APIs collate data from one application, format it for export, and transmit it to the destination application without compromising accuracy or security.

This process is unaffected even in cases where the feature sets of the requesting app have been updated. Additionally, developers without knowledge of the backend workings of a specific API can still integrate it with their software.

Here’s how an API works:

- **Step 1:** An API call (also known as a request) is placed by a client application. This call consists of headers, a request verb, and occasionally a request body.
- **Step 2:** The API’s uniform resource identifier (URI) is used to process this request for data retr

We encourage you to read our updated [PRIVACY POLICY](#). 

- **Step 3:** Once the API receives a valid request, it places a call to the external server or application.
- **Step 4:** The external server or application transmits the requested information back to the API. This is known as a response.
- **Step 5:** The API sends the information to the client application that originally requested it.


Simply put, APIs delink the requesting application from the architecture that provides the requested service. While using different web services can lead to changes in the data transfer process, the entire request and response system relies on APIs for completion.

## The commonly cited ‘waitstaff’ example

Think of yourself as a patron visiting a pizzeria. You’ll ask someone from the waitstaff to bring you a pizza. The waitstaff employee shares your order with the kitchen staff and even brings you your pizza once it is prepared.

In this example, you are the client application, the waitstaff employee is the API, your order for pizza is the API call, and the kitchen and its staff are the servers. Just like an API, the waitstaff is an intermediary between the kitchen (server) and the patron (client application).

Adding to this example with a suitable metaphor for API implementation: As a pizzeria patron, you don’t need to understand the workings of the kitchen or how to prepare a pizza. You only need to understand how to place an order for a pizza.

Let’s conclude the example by understanding how API security works. The kitchen staff would only take orders from someone they recognize as a waitstaff employee. Similarly, API calls normally rely on authorization credentials to allow the server to recognize legitimate requests. We encourage you to read our updated [PRIVACY POLICY](#). 

server. Cookies, HTTP headers, and query string parameters are also used to bolster data security further.

See More: [What Is CI/CD? Definition, Process, Benefits, and Best Practices for 2022](#)

## Types of APIs

Most application programming interfaces today are web-based. These APIs allow external stakeholders to access the functionality and data of an application over the internet.




We encourage you to read our updated [PRIVACY POLICY](#). **×**

## Types of APIs

The four key types of APIs are:


### 1. Public APIs

Public APIs are open source and disseminated for general use. This is why they are also referred to as open APIs. These application programming interfaces have specific API endpoints that are accessed using the HTTP protocol. We encourage you to read our updated [PRIVACY POLICY](#). 



Open APIs allow users to request information from any enterprise that provides the interface. This type of API is a key component of smartphone applications. It also used to integrate popular services with websites easily. Google Maps API is an example of a popular public API.

## 2. Private APIs

Unlike open APIs that are accessible by the public at large, private APIs exist within a [software vendor's](#)  system framework. They are also known as closed or internal APIs and are often proprietary. These interfaces aim to bolster communication and boost productivity.

Enterprises leverage closed APIs to privately transmit data among internal business applications such as enterprise resource planning (ERP), financial systems, or customer relationship management (CRM). Private APIs are normally not revealed to external users.

## 3. Partner APIs

As the name suggests, partner APIs allow two different companies to enter into an exclusive data-sharing agreement. Using this type of application programming interface, vendors gain access to the data streams of partner companies. In return the company granting access to its data receives added services or system features.

Developers can normally access these partner interfaces in self-service mode using an open API dev portal. However, they would still be required to go through an onboarding process and enter login credentials to gain access to partner APIs. This type of API is a critical component of strategic business partnerships in the API economy.



We encourage you to read our updated [PRIVACY POLICY](#). 

## 4. Composite APIs

Composite APIs combine different service or data APIs. This variant of the application programming interface enables dev teams to access multiple endpoints by raising a single call. Composite APIs are often seen in microservices architectures, where data from more than one source is frequently needed to complete a given task.

Composite interfaces compile multiple calls sequentially and create a single API request. This request is transmitted to the server, which, in turn, sends back one response. The distinction between composite APIs and batch APIs is the lack of a sequence in the latter.

For instance, an ecommerce platform might use a composite API to create an order by a new customer. By doing so, only a single request would need to be raised to create a new customer profile, generate an order for the new customer profile, add an item to the new order, and revise the order status.

See More: [CI/CD vs. DevOps: Understanding 8 Key Differences](#)

## List of API Protocols

Developers rely on specific collections of requests and techniques to integrate application programming interfaces with their software. The specifications that regulate the application and management of such requests are called API protocols. These protocols guide developers on API usage standards, the types of data used, the correct commands to be given, and many other facets of API-related development.



We encourage you to read our updated [PRIVACY POLICY](#). **×**

## List of API Protocols

Popularly used API protocols are:

# 1. Simple object access protocol (SOAP)

The SOAP API protocol uses extensible markup language (XML) to power API communications. With its first appearance in June 1998, SOAP is the oldest protocol still in use today.

We encourage you to read our updated [PRIVACY POLICY](#). 

This protocol primarily uses XML files to transmit data over an HTTP or HTTPS connection. However, it is also flexible enough to transmit data over other protocols such as simple mail transport protocol (SMTP), transmission control protocol (TCP), and user data protocol (UDP).

XML-encoded SOAP messages use the format defined below:

- **Envelope:** The core element of the message. It ‘envelopes’ the message by placing tags at the start and the end.
- **Header (optional):** It defines specific additional message requirements, such as authentication.
- **Body:** The request or response is included here.
- **Fault (optional):** Information about errors that might arise during the execution of the API call or response is highlighted here, along with information on how one can address these errors.

SOAP is popular for the flexibility of its transmission channel. However, it relies heavily on XML, thus requiring rigid formatting rules. Also, XML is difficult to debug, making SOAP less popular for modern-day API requirements.

## 2. Representational state transfer (REST)

REST protocols are more flexible than SOAP. This is primarily achieved by removing the dependency on XML; REST prominently transmits data in JSON. However, it can transfer data in multiple other formats, including [Python](#), HTML, and even media files and plain text.

An API that uses the REST protocol is known as a RESTful API. These APIs use a client-server structure. Unlike SOAP, REST can exchange HTTP and HTTPS. We encourage you to read our updated [PRIVACY POLICY](#).

REST requests typically include these key components:

- **HTTP method:** This component outlines the four basic processes that a resource can be subjected to—POST (create a resource), GET (retrieve a resource), PUT (update a resource), and DELETE (delete a resource).
- **Endpoint:** The uniform resource identifier that locates the resource on the internet is a part of this component. URLs are the most common type of URI.
- **Header:** Data related to the server and the client is stored in this component. Like in SOAP, one can also use REST headers to store authentication measures such as API keys, server IP addresses, and the response format.
- **Body:** This component contains additional information for the server, such as data that needs to be added or replaced.

Compared to SOAP, REST implementation is flexible, scalable, and lightweight. A key feature of RESTful APIs is stateless communication. This forbids the storage of client data between GET requests.

Caching is another key feature of REST. This enables web browsers to store the request response received locally and access it periodically to enhance efficiency.

Finally, GET requests are required to be disconnected and distinct. With REST, a unique URL is assigned to every operation. This allows the server to follow preset instructions for executing a received request.

### 3. Google remote procedure call (gRPC)

Developed by Google and released for public use in 2015, gRPC is an open-source remote procedure call (RPC) architecture that can operate in numerous environments.

We encourage you to read our updated [PRIVACY POLICY](#). 

The gRPC transport layer primarily relies on HTTP. The ability for developers to specify custom functions that allow for flexible inter-service communication is a significant feature of gRPC. This API protocol also offers extra features such as timeouts, authentication, and flow control.

In the gRPC protocol, data is transmitted in protocol buffers, a platform and language-agnostic mechanism that allows for data to be structured intuitively. This mechanism defines the service and then the data structures that the service will use. Compiling is taken care of by protoc, the protocol buffer compiler.

The output of this process is a comprehensive class containing the user's defined data types and basic set methods in the chosen development language. Users can implement in-depth API operations using this class.

## 4. JavaScript object notation–remote procedure call (JSON-RPC)

JSON-RPC is a stateless and lightweight API protocol that communicates between web services using request objects and response objects. Introduced shortly after the turn of the millennium, JSON-RPC leverages JavaScript Object Notation (JSON) to allow API communications' simple, albeit limited, execution.

This protocol defines requests that can take care of all functionalities within its narrow scope. JSON-RPC has the potential to outperform REST in cases where one can apply it.

## 5. Graph query language (GraphQL)

Released in 2015, GraphQL is a database query language, and a server-side runtime for APIs developed at Facebook. By design, this protocol prioritizes giving users the exact data requested

We encourage you to read our updated [PRIVACY POLICY](#). 

supports the creation of fast and flexible APIs, including composite APIs. GraphQL can be used as an alternative for REST.

## 6. Apache Thrift

Also developed at Facebook, Thrift is a lightweight, language-agnostic software stack. This API protocol supports HTTP transmission, along with binary transport formats. Thrift is capable of clean abstractions and implementations for data serialization and transport and application-level processing. Its primary objective is point-to-point RPC implementation.

Apache can support 28 programming languages, allowing programs written in any of these languages to communicate with each other and request remote services using APIs.

See More: [Top 8 DevOps Strategies to Test in 2021](#) 

## Examples of API

The API economy is booming in 2022, with enterprises from across industry verticals leveraging application programming interfaces to allow secure and controlled access to their software resources.

A few popular examples of APIs are:

### 1. Twitter bots

Twitter bots are automated accounts on the microblogging website Twitter. These accounts use the Twitter API to automatically carry out software-defined operations such as tweet

We encourage you to read our updated [PRIVACY POLICY](#). 

Bot creators use the Twitter API to set up their bots to do simple actions or be notified when a specific event occurs on the platform. For instance, one can use the Twitter API to configure a bot to recognize when a new account follows it. Using the Twitter API, one can then program the bot to reply to that follower.

## 2. Sign in using Google/Facebook

Several websites and applications give users an option to sign in by using their Google or Facebook accounts. Unlike what a layperson might presume, these buttons do not allow the website to access users' Google or Facebook data.

Rather, the service provider's API [verifies the user's identity](#). Every time the website or application is accessed, it uses Facebook or Google's API to check whether the user is already logged in.

Other large internet service providers such as Twitter and Apple also provide this functionality on some websites using APIs.

## 3. Weather forecasts

Many frequently visited websites display the local weather on their homepage using APIs. It is also easy to look up the weather worldwide by searching for '[place name] + weather' on Google. Doing so displays a dedicated weather snippet on top of the other results.

The data shown in this snippet is sourced from IBM's The Weather Company. Simply put, Google relies on The Weather Company's API to receive the latest weather data in a format that can be easily adapted to the widget on its results screen. The link to Weather.com (owner of the API) is also visible at the bottom corner of the widget.



We encourage you to read our updated [PRIVACY POLICY](#). **×**



## 4. Travel booking

Booking aggregator websites for flights and hotels collate thousands of options to give customers the best deals. APIs play a major role in displaying the availability and pricing of flights and hotels in real-time. Once a user completes a booking on the website, APIs are used to confirm the transaction with the service provider.

APIs have considerably simplified operations for the travel and tourism sector by making computers exchange data swiftly and autonomously. Without APIs, booking services would need to manually coordinate with customers, hotels, and airlines, which is a long, tedious process full of loopholes.


## 5. Payment methods

Ecommerce platforms allow customers to pay for their orders using third-party payment applications such as PayPal. Once again, API connections are used to provide these payment services without the user having to leave the ecommerce website or worry about their data falling into the wrong hands.

For instance, when a customer clicks on the 'Pay with PayPal' button, an 'order' request is communicated from the platform to the PayPal API. This API securely transmits data on the amount to be billed and other sensitive information. When the user [authenticates themselves](#) and their order through pop-ups, the API returns to the application with a payment confirmation.




See More: [DevOps Roadmap: 7-Step Complete Guide](#)

## Takeaway

Application programming interfaces (APIs) allow enterprise users to give third-party developers access to their systems. We encourage you to read our updated [PRIVACY POLICY](#). 

or security. Companies can choose who is granted access and how to use the software.

Different types and protocols are available for APIs. Any software-first company with an appropriate use case can leverage APIs and join the lucrative ‘API economy’ to enhance its profitability.

*Did this article help you gain a thorough understanding of application programming interfaces (APIs)? Do let us know on [LinkedIn](#) , [Twitter](#) , or [Facebook](#) !*

## MORE ON DEVOPS

- [What Is DevOps? Definition, Goals, Methodology, and Best Practices](#)
- [What Is DevOps Lifecycle? Definition, Key Components, and Management Best Practices](#)
- [Scrum vs. DevOps: Understanding the Key Differences](#)
- [DevOps vs. Agile Methodology: Key Differences and Similarities](#)
- [Top 10 DevOps Automation Tools in 2021](#)

[API](#)[DevOps](#)

Share This Article:

We encourage you to read our updated [PRIVACY POLICY](#).

×

**Join Spiceworks** to get the latest Tech News & Insights newsletter delivered to your inbox!

[Join Spiceworks](#)

## Hossein Ashtari

Technical Writer



Interested in cutting-edge tech from a young age, Hossein is passionate about staying up to date on the latest technologies in the market and writes about them regularly. He has worked with leaders in the cloud and IT domains, including Amazon—creating and analyzing content, and even helping set up and run tech content properties from scratch. When he's not working, you're likely to find him reading or gaming!

Do you still have questions? Head over to the Spiceworks Community to find answers.

[Take me to Community](#)

We encourage you to read our updated **PRIVACY POLICY**. [×](#)



## A new OnePlus Nord is coming

The newest OnePlus Nord is finally ready and you're not going to want to miss the launch.

AD OnePlus

[LEARN MORE](#)

### Popular Articles

[ChatGPT vs. Bing vs. Google Bard: Choosing the Most Helpful AI](#)

[Data as Currency: Consumers Willing to Pay for Personalized Experiences](#)

[The Impact of Digital Transformation on Workplace Trends](#)



We encourage you to read our updated [PRIVACY POLICY](#). ×

### Table of Contents

- [What Is an API \(Application Programming Interface\)?](#)
- [How Does an API Work?](#)
- [Types of APIs](#)
- [List of API Protocols](#)

## Recommended Reads



We encourage you to read our updated **PRIVACY POLICY**. **×**



IT Careers & Skills

## Versatility Redefined: The Rise of the Full-stack Developer

DevOps

## Why Automation Is Doomed Without Process Orchestration



DevOps

## How To Incentivize Open Source Maintainers for More Security

Networking

## Three Things They Didn't Tell You About SD-WAN

- [Sitemap](#)[Privacy Policy](#)
- [Terms of Use](#)
- [Guidelines](#)[Statement](#)
- [Accessibility](#)
- [Do Not Sell My Personal Information](#)