

Learn to code — free 3,000-hour curriculum

APRIL 25, 2022 / [#REST API](#)

How to Use REST APIs – A Complete Beginner's Guide



Alex Husar

Application programming interfaces – or APIs – are an important programming concept to understand. And if you invest the time to learn more about these interfaces, it can help make your tasks more manageable.

One of the common types of APIs is a REST API. If you've ever considered getting data from another website, such as Twitter or GitHub, you've probably used this kind of API.

So why is understanding a REST API useful? How does it ensure modern business connectivity?

Before building or operating an API, or a REST API in particular, you should first learn what an API is. This article will walk you through the REST API principles, and how they grew into powerful applications.

Learn to code — free 3,000-hour curriculum

APIs represent a set of definitions and protocols. You need them for app development and integration as they facilitate data exchange between two pieces of software, like an information supplier (a server) and a user.

APIs specify the content available to the client making the call from the producer that's returning the response.

Programs use an API to communicate, retrieve information, or perform a function. APIs allow users to work with the system to return their desired result.

To put it simply, an API acts as a mediator between users (clients) and resources (servers).

When users make API requests or visit an online store, they expect a fast response. So you need to [optimize Magento TTFB \(Time To First Byte\)](#) or use other performance enhancement strategies that work best for your CMS.

The reasons to integrate an API include:

- streamlining resource and information sharing
- controlling who has access to what with the help of [authentication and defining the rights](#)
- safety and control
- no need to understand the software specifics

Learn to code — free 3,000-hour curriculum

Overview of REST APIs

.noD1WovETSrQdSalv8rh5WUVPxVjPN9_cvVokx7E4fZxzGyCY0_XMRA2cikjPkWIUDIX
Ξgxi8sEVwpVTeiHBNsqCULC-78QCE4dJ0_ieC1mQzn

RESTful refers to software architecture which stands for
“Representational State Transfer”. You may have heard of it in the

Learn to code — free 3,000-hour curriculum

These web services utilize a stateless protocol to make textual representations of their online resources available for reading and processing. A client performs well-known HTTP protocol-based activities like fetch, update, and delete.

REST was first established in 2000 with the goal of improving performance, scalability, and simplicity by enforcing specific limits on an API.

It has gained popularity because of the opportunity to cover various devices and applications. Below you will find some of the purposes of using REST APIs.

[Forum](#)

[Donate](#)

[Learn to code — free 3,000-hour curriculum](#)

Learn to code — free 3,000-hour curriculum

projects, such as:

- web development
- iOS apps
- IoT devices
- Windows Phone apps

As you won't have to stick to a specific client-side stack, you can build any infrastructure for your company.

2. Applications in the Cloud

REST API calls are ideal for cloud applications due to their statelessness. If something goes wrong, you can re-deploy stateless components, and they can grow to manage traffic shifts.

3. Cloud Computing

An API connection to a service requires controlling how the URL is decoded. That's why REST has become more useful in cloud services.

RESTful API architecture will become the norm in the future, thanks to cloud computing and microservices.

How do REST APIs Work?

Data (such as images, videos, and text) embody resources in REST. A client visits a specific URL and sends a server request to receive a response.

[Forum](#)

[Donate](#)

[Learn to code — free 3,000-hour curriculum](#)

[Forum](#)

[Donate](#)

[Learn to code — free 3,000-hour curriculum](#)

Learn to code — free 3,000-hour curriculum

- the **endpoint**, which is the URL with the structure `root-endpoint/?`
- the **method** with one of the five possible types (GET, POST, PUT, PATCH, DELETE)
- the **headers**, serving various functions, including authentication and providing information about the content of the body (you can use the `-H` or `--header` option to send HTTP headers)
- **data (or body)**, that's what you send to the server through the `-d` or `--data` option with POST, PUT, PATCH, or DELETE requests.

The HTTP requests allow you to operate with the database, such as:

- POST request to create records
- GET request to read or get a resource (a document or image, a collection of other resources) from the server
- PUT and PATCH requests to update records
- DELETE request to delete a resource from a server

These operations stand for four possible actions, known as CRUD: Create, Read, Update and Delete.

Learn to code — free 3,000-hour curriculum

The server sends the data to the client in one of the following formats:

- HTML
- JSON (which is the most common one thanks to its independence of computer languages and accessibility by humans and machines)
- XLT

Learn to code — free 3,000-hour curriculum

- plain text

Why Use a REST API?

Why should you prefer REST over other APIs, such as SOAP? There are numerous reasons, like scalability, flexibility, portability, and independence.

yJ2QDrpGbA-

RpzwhXOXr1yl9aGTvVHXeiuyBvFxsMtE5KQu2wRmNLwlCX7cNGOlP1TjRK-
P9VsBsFaGRNkxZw-QWvggxqXLYFtLg-THCIHzB-
5GJIMX6hGkY3DQnFh1YpzkHt2iE

Not relying on the project structure

A separate client and server operation means that developers aren't bound to any project parts. Thanks to adaptive REST APIs, they can

Learn to code — free 3,000-hour curriculum

Portability and adaptability

REST APIs work only when the data from one of the requests is successfully delivered. They allow you to migrate from one server to another and update the database at any moment.

Opportunity to scale the project in the future

As the client and server act independently, the coders may swiftly develop the product.

Features of the RESTful Architecture Style

Developers have to consider a rigid structure of some APIs, such as SOAP or XML-RPC. But REST APIs are different. They support a wide range of data types and may be written in practically any programming language.

The six REST architectural constraints are principles for designing the solution and are as follows:

Learn to code — free 3,000-hour curriculum

1. Uniform Interface (A Consistent User Interface)

This concept dictates that all API queries for the same resource, regardless of their origin, should be identical, that is, in one specific language. One uniform resource identification (URI) is associated with the same data, such as a user's name or email address.

Another uniform interface principle states that messages should be self-descriptive. They must be comprehensible for the server to determine how to handle it (for example, the type of request, mime types, and so on).

Learn to code — free 3,000-hour curriculum and server implementations. The thing is, they can be done independently and don't have to know about the other.

For example, the client has only the uniform resource identification (URI) of the requested resource and can't communicate with the server program any other way. On the other hand, the server shouldn't affect the client software. So it sends the essential data over HTTP.

What does this mean? You can modify the client code at any moment without impacting the server's operation.

The server code is in the same boat: changing the server's side won't affect the client's operation.

You can keep client and server programs both modular and independent as long as each side knows what message format to deliver to the other.

What do we achieve by separating the user interface problems from the data storage constraints? We improve the interface flexibility across platforms and boost scalability.

Furthermore, each component benefits from the separation because it can evolve independently. A REST interface assists different clients in:

- accessing the same REST endpoints
- executing identical activities
- receiving similar responses

Learn to code — free 3,000-hour curriculum

REST-based systems are stateless, meaning that the client state remains unknown to the server and vice versa. This constraint allows the server and the client to understand any sent message, even if they haven't seen the preceding ones.

To enforce this constraint of statelessness, you need to use resources rather than commands. These are the nouns of the web. Their purpose is to describe any object you may want to keep or communicate to other services.

You can control, change, and reuse components without affecting the system as a whole, so the benefits of this constraint include achieving:

- stability
- speed
- scalability of RESTful applications

Note that each request should include all the information required to complete it. Client applications have to save the session state since server apps shouldn't store any data linked with a client request.

4. Cacheable Data


REST requires caching client-side or server-side resources wherever possible. Data and response caching are critical in today's world because it results in better client-side performance.

How does it affect a user? Well-managed caching can reduce or eliminate some client-server interactions.

Learn to code — free 3,000-hour curriculum

connection.

5. Layered System Architecture

 DBk2dcqnTMZdz-dBA0sFDUe5cQu71VxMqG8pW-ux4rqNvkVcsixRNR_ZyuY1z6UeWWZ5NRV11FPIv8XYK86EGr2G-Nnb7O_njC9PER6a5TdmfpZ2qmRTI7f9P--S7QU50cYwD9EC

The RESTful layered design structure is the next constraint under discussion. This principle involves grouping different layers with specified functions.

The REST API layers have their responsibilities and come in hierarchical order. For example, one layer may be responsible for storing data on the server, the second for deploying the APIs on another server, and the third for authenticating requests in another server.

Learn to code — free 3,000-hour curriculum

What does it mean when each layer performs its function before transferring the data to the next? It improves the API's overall security and flexibility because adding, altering, or removing APIs doesn't affect other interface components.

6. On-Demand Coding (Non-obligatory)

The most common scenario of using REST APIs is to deliver static resource representations in XML or JSON.

However, this architectural style allows users to download and run code in the form of Java applets or scripts (such as JavaScript). For example, clients can retrieve the rendering code for UI widgets by calling your API.

Challenges You Should Expect When Using REST APIs

When you've understood REST API design and architectural constraints, you should know the issues to expect while employing this architectural style:

Learn to code — free 3,000-hour curriculum

Agreement on REST endpoints

APIs should remain consistent regardless of the URL construction. But with the growth of possible combinations of methods, it's harder to maintain uniformity in large codebases.

Versioning as a feature of REST APIs

APIs require regular updating or versioning to prevent issues with compatibility. However, old endpoints remain operational, which increases the workload.

A lot of authentication methods

Learn to code — free 3,000-hour curriculum

they can do with these variables.

But the 20 different authorization methods that exist can make your initial API call difficult. That's why developers don't proceed with the project due to the initial difficulties.

REST API security vulnerabilities

Although RESTful APIs have a layered structure, there still may be some security concerns. For example, if an application isn't secure enough due to a lack of encryption, it can expose sensitive data.

Or a hacker may send thousands of API requests per second, causing a DDoS attack or other misuses of the API service to crash your server.

Excessive data collection and requests

A server may return a request with all the data, which may be unnecessary. Or you might need to run multiple queries to get the needed information.

Wrapping Up

There's no surprise that APIs are predicted to streamline web-based communications in the future. Their purpose is to allow any web apps to interact and share data.

For example, they assist growing online businesses in developing robust and inventive systems. As the API architecture evolves, it

Learn to code — free 3,000-hour curriculum

So in this article you learned the basics of what you need to know about using REST APIs.



Alex Husar

Chief technology officer at Onilab with 8+ years of experience in developing PWAs, Magento migration, and Salesforce development.

If you read this far, tweet to the author to show them you care.

[Tweet a thanks](#)

Learn to code for free. freeCodeCamp's open source curriculum has helped more than 40,000 people get jobs as developers.

[Get started](#)

freeCodeCamp is a donor-supported tax-exempt 501(c)(3) charity organization (United States Federal Tax Identification Number: 82-0779546)

Our mission: to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. We also have thousands of freeCodeCamp study groups around the world.

Donations to freeCodeCamp go toward our education initiatives, and help pay for servers, services, and staff.

You can [make a tax-deductible donation here](#).

Learn to code — free 3,000-hour curriculum

[strftime\(\) in Python](#)[Python Lowercase](#)[Check for Null in JS](#)[Run Python Script](#)[Command Line Tutorial](#)[indexOf in Python](#)[Check Empty String JS](#)[KeyError in Python](#)[Make an Arrow in HTML](#)[K8s vs Docker Swarm](#)[Make a List in Python](#)[Multiline String JS](#)[How to Read JSON in JS](#)[Concatenate in Excel](#)[Python Print Same Line](#)[Python List.append\(\)](#)[Get Current Date in JS](#)[Population vs Sample](#)[Reverse a String in JS](#)[With Statement in Python](#)[List Index Out of Range](#)[Loop Through Array in JS](#)[Callbacks in JavaScript](#)[Check if Undefined in JS](#)[Insert into Array in JS](#)[Sort Alphabetically in JS](#)[JavaScript Random Number](#)[Python Dictionary Methods](#)

Our Charity

[About](#) [Alumni Network](#) [Open Source](#) [Shop](#) [Support](#) [Sponsors](#) [Academic Honesty](#)[Code of Conduct](#) [Privacy Policy](#) [Terms of Service](#) [Copyright Policy](#)