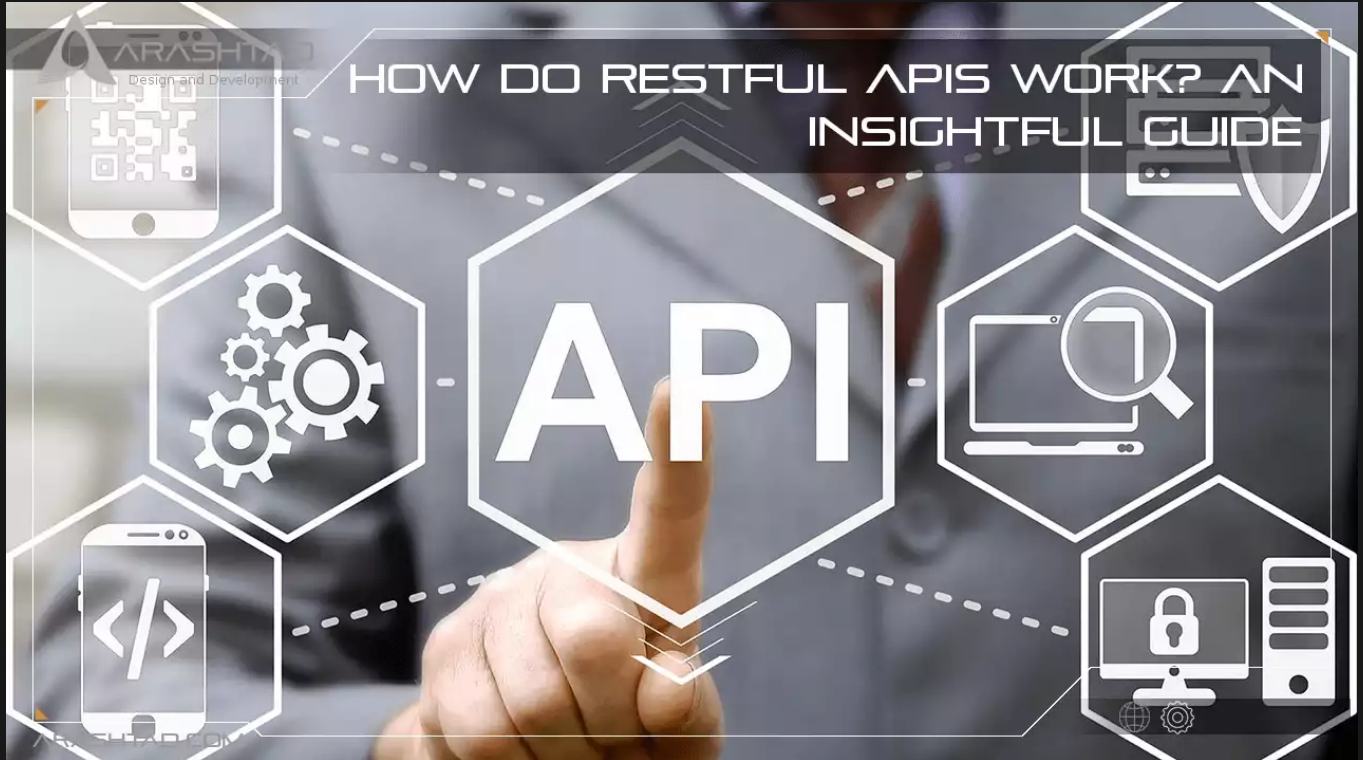


How do RESTful APIs work? An Insightful Guide

No comments



RESTful APIs are the kind of APIs that follow the constraints of a REST architectural style. We have covered a full article on the background architecture of this kind of API, its pros and cons, and its principles. This article focuses mainly on the way the RESTful APIs work, client requests, and server responses in these APIs. If you want to get familiar with the way these APIs work and then use different frameworks, follow along with the article to get familiar with the backbone architecture of a RESTful API.

How do RESTful APIs work?

Just the same as browsing the internet, in a RESTful API, The client contacts the server by using the API when it requires a resource. The API developers explain how the client should use the REST API in the server application API documentation. The process that happens in RESTful API, goes like this. First, The client sends a request to the server and writes the request in a format that has been mentioned in the documentation, so that the server can understand. Then, the server confirms if the client has the right to make a request by authenticating him or her. Afterward, the server receives and processes the request internally. And finally, it returns a response containing the request

acceptance or rejection and if the request has been accepted, it also returns what the client has requested.

What does the client request look like in a RESTful API?

All the RESTful API requests need to contain the following three main components:

1. URI
2. Method
3. HTTP Headers

What is URI? And how is it used in a client request?

URI is the acronym for Unique Resource Identifier. The server identifies each resource with a URI. In REST services, the server typically performs resource identification by using a URL (an acronym for Uniform Resource Locator). The URL specifies the path to the resource. A URL is similar to the website address that you enter into your browser to visit any webpage. The URL is also called the request endpoint and clearly specifies to the server what the client requires.

Method

RESTful APIs are often implemented using the HTTP (Hypertext Transfer Protocol) methods. These methods do a certain action on the server by telling it what it needs to do to the server. The followings are the four HTTP common methods that is also used in a RESTful API:

1. GET:

Clients use GET to access resources that are located at the specified URL on the server. They can cache GET requests and send parameters in the RESTful API request to instruct the server to filter data before sending. By using the GET request, the client queries the necessary items from a database.

2. POST:

Clients use the POST request to send data to the database. They include the data representation with the request. Sending the same POST request multiple times has the side effect of creating the same resource multiple times.

3. PUT:

Clients use PUT to update existing resources on the server. Unlike POST, sending the same PUT request multiple times in a RESTful web service gives the same result and it does not create new resources.

4. DELETE:

Clients use the DELETE request to remove the resource. A DELETE request can change the server state. However, if the user does not have appropriate authentication, the request fails.

HTTP headers

Request headers are the metadata exchanged between the client and server. For instance, the request header indicates the format of the request and response, provides information about request status, and so on.

DATA

REST API requests might include data for the POST, PUT, and other HTTP methods to work successfully.

Parameters

RESTful API requests can include parameters that give the server more details about what needs to be done. The following are some different types of parameters:

Path parameters that specify URL details.

Query parameters that request more information about the resource.

Cookie parameters that authenticate clients quickly.

RESTful API authentication methods?

A RESTful web service must authenticate requests before it can send a response. Authentication is the process of verifying identity. For example, you can prove your identity by showing an ID card or driver's license. Similarly, RESTful service clients must prove their identity to the server to establish trust.

RESTful API has four common authentication methods:

HTTP authentication

HTTP defines some authentication schemes that you can use directly when you are implementing REST API. The following are two of these schemes:

Basic authentication

In basic authentication, the client sends the user name and password in the request header. It encodes them with base64, which is an encoding technique that converts the pair into a set of 64 characters for safe transmission.

Bearer authentication

The term bearer authentication refers to the process of giving access control to the token bearer. The bearer token is typically an encrypted string of characters that the server generates in response to a login request. The client sends the token in the request headers to access resources.

API keys

API keys are another option for REST API authentication. In this approach, the server assigns a unique generated value to a first-time client. Whenever the client tries to access resources, it uses the unique API key to verify itself. API keys

are less secure because the client has to transmit the key, which makes it vulnerable to network theft.

OAuth

OAuth combines passwords and tokens for highly secure login access to any system. The server first requests a password and then asks for an additional token to complete the authorization process. It can check the token at any time and also over time with a specific scope and longevity.

What does the RESTful API server response contain?

REST principles require the server response to contain the following main components:

Status line

The status line contains a three-digit status code that communicates request success or failure. For instance, 2XX codes indicate success, but 4XX and 5XX codes indicate errors. 3XX codes indicate URL redirection.

The following are some common status codes:

200: Generic success response

201: POST method success response

400: Incorrect request that the server cannot process

404: Resource not found

Message body

Message body

The response body contains the resource representation. The server selects an appropriate representation format based on what the request headers contain. Clients can request information in XML or JSON formats, which define how the data is written in plain text. For example, if the client requests the name and age of a person named John, the server returns a JSON representation as follows:

```
'{"name": "Mohamad", "age": 27}'
```

Headers

The response also contains headers or metadata about the response. They give more context about the response and include information such as the server, encoding, date, and content type.

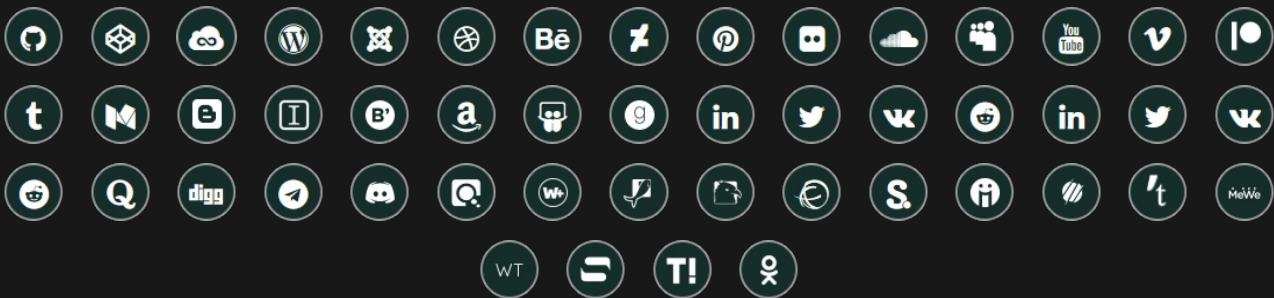
Conclusion

In this article, you learned about the way the RESTful APIs work, client requests, and server responses in these APIs. By knowing the way the RESTful API works, you can use any kind of framework or language to write a microservice or an API using the REST architecture. Flask, Django, PHP, or any other kind of language or framework or micro-framework can help you create this kind of API. One of the most popular and simple applications created using the RESTful API is CRUD (Create, Read, Update, Delete) application and it can be used for creating a user database or a website or a web application.

Join Arashtad Community

Follow Arashtad on Social Media

We provide variety of content, products, services, tools, tutorials, etc. Each social profile according to its features and purpose can cover only one or few parts of our updates. We can not upload our videos on SoundCloud or provide our eBooks on Youtube. So, for not missing any high quality original content that we provide on various social networks, make sure you follow us on as many social networks as you're active in. You can find out Arashtad's profiles on different social media services.



Get Even Closer!

Did you know that only one universal Arashtad account makes you able to log into all Arashtad network at once? Creating an Arashtad account is free. Why not to try it? Also, we have regular updates on our newsletter and feed entries. Use all these beneficial free features to get more involved with the community and enjoy the many products, services, tools, tutorials, etc. that we provide frequently.

[SIGN UP](#)[NEWSLETTER](#)[RSS FEED](#)