HTTP METHODS

By Naveen PS

MAKERDEMY
LEARN EXPLORE MAKE

HTTP Methods

Ever wondered what the difference is between GET and POST requests, or when to use PUT? You're not alone. Having a basic understanding of the different HTTP methods, or verbs, an API supports is a helpful knowledge when exploring and testing APIs.

HTTP Methods

1. GET
2. POST
3. PUT
4. HEAD
5. DELETE
6. PATCH
7. OPTIONS

GET requests are the most common and widely used methods in APIs and websites. Simply put, the GET method is used to retrieve data from a server at the specified resource. For example, say you have an API with user's endpoint. Making a GET request to that endpoint should return a list of all available users.

In web services, POST requests are used to send data to the API server to create or update a resource. The data sent to the server is stored in the request body of the HTTP request. The simplest example is a contact form on a website. When you fill out the inputs in a form and hit Send, that data is put in the response body of the request and sent to the server. This may be JSON, XML, or query parameters.

Similar to POST, PUT requests are used to send data to the API to update or create a resource. The difference is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly make have side effects of creating the same resource multiple times.

A PATCH request is one of the lesser-known HTTP methods, but I'm including it this high in the list since it is similar to POST and PUT. The difference with PATCH is that you only apply partial modifications to the resource.

The DELETE method is exactly as it sounds: delete the resource at the specified URL. This method is one of the more common in RESTful APIs so it's good to know how it works.

The HEAD method is almost identical to GET, except without the response body. In other words, if GET /users returns a list of users, then HEAD /users will make the same request but won't get back the list of users. HEAD requests are useful for checking what a GET request will return before actually making a GET request like before downloading a large file or response body.

Last but not least we have OPTIONS requests. OPTIONS requests are one of my favourites, though not as widely used as the other HTTP methods. In a nutshell, an OPTIONS request should return data describing what other methods and operations the server supports at the given URL.

THE END