

Why You Should Use WordPress HTTP Functions to Make API Requests?

[WordPress](#) HTTP comprises numerous functions and methods. These resources make HTTP calls an easier and expendable ride. You don't need to goof around with various cURL contents anymore. A unified interface like WordPress HTTP makes the job easier. It's an excellent alternative for communicating with third-party APIs like Mailchimp, Twitter, and many others.

In this article, we will look at the WordPress HTTP API functions and why you should utilize them.



Basic Ingredients of HTTP

In one way or the other, you have come across how HTTP works. Even if you are a first-timer in terms of web service, you must have probably heard about how HTTP makes things work. HTTP is an internet protocol that is used to transfer files and contents across the webs.

There are two ingredients to consider when dealing with HTTP. They include the “[response](#)” and “request” sections. The response and request transactions have a similar pattern. But they comprise four sections. Let's dive into them.

- A blank line.
- Header lines.
- Initial line.
- Body Content.

Header Lines

The header lines comprise numerous pieces of information relating to the response or the request transactions. For instance, HTTP 1.1 allows faster delivery of internet pages than regular HTTP. It also minimizes web traffic. It makes the response and request transactions easier. In any case, the HTTP 1.1 itemizes forty-six types of headers but only one out of them does the request transaction. It's known as the "[host header](#)."

The host header indicates the server's port number and the requesting host.

Initial Line

The request transactions leverage the initial line to relay three bits of information. They include the [URL path](#), resource name, and the HTTP model. For instance, if you are viewing the infotohow blog page, you would encounter this in the initial line of the request.

GET/blog/HTTP/1.1

The responses typically offer three bits of information. They include the HTTP model, the response key, and an illustration of the response. When requesting infotohow blog page, it will relay an HTTP response with the initial line below:

HTTP/1.0 200 OK

Body Content

The body typically contains a piece of information. If you convey a "GET" request to the main infotohow blog page, you will receive an HTML that will provide the page or resource in the body.

What is HTTP API?

It's an accepted fact that HTTP represents Hypertext Transfer Protocol. It is the most significant communication protocol for the entire web. For the most part, HTTP operates in this way:

- Hello server DYX, may I please have file xyz.html"
- Hello dear client, yes, you may. You are here, here it is"

Some numerous methods and functions allow you to send HTTP demands in PHP. The goal of the WordPress HTTP API is to leverage as many functions and methods as possible. However, you should utilize the one that is best suited for the specific request.

You can also leverage the WordPress HTTP API to fraternize with other APIs like Google Maps.

WordPress HTTP Functions and Methods

The WordPress HTTP has numerous functions and methods that allow you to make [API](#) requests. They comprise verbs that help to illustrate or elucidate specific types of actions. Although there are more verbs to explore, WordPress has pre-configured functions for the three most prominent ones. Once the HTTP requests the API, the transaction passes a resource that helps the server to figure out the kind of action that the client wants to perform.

Post #

Research from the [college paper writer](#) has it that POST is used to send a piece of information to the server so that it can work around it. For instance, address or contact forms. Once you enter the piece of information in the form and tap the submit button, the browser receives the data and sends a POST message to the server. It sends the message alongside the data you inserted in the form. This way, the server will optimize the contact transaction or request.

GET #

You use "GET" to restore or salvage data. It is the most prominent verb. In other words, it is the most used request verb. As soon as you visit a website or takes data from an API, you will see the outcome of a GET request. Your service provider sends a GET request to the internet as you view this posy. It requests the resources that are used to create this article.

HEAD #

HEAD is not as common or popular as the other two. It is similar to the GET request but it does not salvage data. All it does is restore the piece of information about the data. The information describes things like what type of resource is, when the resource was last updated, and many others.

Your service providers will relay HEAD requests to previously visited pages to figure out any updates. If there are no updates, you will encounter a previously installed copy of the page.

Most API clients leverage the HEAD transaction before engaging in a GET request to salvage the bandwidth.

Custom Functions

There are other numerous HTTP functions or methods or resources on WordPress. They range from “CONNECT”, “TRACE” to “DELETE.” We cannot touch these functions in this post as WordPress does not provide pre-configured methods to leverage them. It’s not common for APIs either.

You can also adopt more personalized functions depending on how you’ve optimized your server. However, it’s always risky to go outside the conventional methods. It limits other developers carving out clients to leverage your website or API. But it is possible to leverage any function you desire. Let’s look at how to use the conventional methods to make API requests via WordPress HTTP.

Response Keys

HTTP leverages both the integer and string response keys. Rather than delve deeper into them, we will provide the conventional response keys. You can personalize your response keys when carving out or making APIs. However, it’s always an excellent move to stick to the conventional codes. You may opt for the unconventional ones only when there’s an overwhelming need for them.

Let’s look at the commonest keys or codes that you will probably encounter.

301	The resource was moved permanently
302	The resource was moved temporarily
500	Internal server error
200	Request was successful

Why You Should Utilize WordPress HTTP Functions for API Requests

The WordPress HTTP will help you to make any sort of request. It follows that there are numerous reasons you should leverage it to make API requests. Let’s look at them.

Customizable

WordPress HTTP allows you to personalize your API requests. It means that it gives room for numerous options that will help you to modify your requests.

```
$args = array( Copy
    // Increase the timeout from the default
    'timeout' => 10,

    // Overwrite the default: "WordPress/5.8
    'user-agent' => 'My special WordPress in

    // Add a couple of custom HTTP headers
    'headers' => array(
        'X-Custom-Id' => 'ABC123',
        'X-Secret-Thing' => 'secret',
    ),

    // Skip validating the HTTP servers SSL
    'sslverify' => false,
);

$response = wp_remote_get( 'https://www.ex
```

Capable

It has capable and substantial features. You can add the header lines and switch the agent string. As a user, you can also indicate your location for SSL certifications.

Large HTTP Libraries

Numerous helper libraries make it easier to create HTTP and API requests. You do not have to add other libraries since you already have access to an authentic library.

Closing Thoughts

WordPress HTTP comes with substantial features which are well-suited for almost any API request. There are numerous benefits of using it over other options. Not only developers enjoy these benefits but also the entire WordPress sector.

Author Bio

Charles Normandin writes unique articles for the [best essay writing services](#). He is a professional in writing unique texts and also speaks 3 languages at a high level. He is fond of volunteering, drawing, and

rock climbing. In his view, days spent helping animals at the shelter make their life better. Meet him on Twitter @Charles60104524.

Sources

<https://deliciousbrains.com/wordpress-http-api-requests/>

Pixabay