**Demonstrate how to work with POST requests:**

HTTP POST Request Method

[Articles](https://reqbin.com/articles) / [Top 9 HTTP Methods](https://reqbin.com/Article/HttpMethods) / [What is HTTP POST Request Method?](https://reqbin.com/Article/HttpPost)

POST is an HTTP method designed to send data to the server from an HTTP client. The HTTP POST method requests the web server accept the data enclosed in the body of the POST message. HTTP POST method is often used when submitting login or contact forms or uploading files and images to the server.

What is HTTP?

The [Hypertext Transfer Protocol](https://reqbin.com/Article/HttpProtocol) (HTTP) is designed to support communication between clients (browser or application) and web servers. HTTP is designed to send information in a format that both the client and the server can understand. HTTP works as a stateless request-response protocol between the client and the web server. HTTP is based on several request methods, or "verbs", including the HTTP POST and [HTTP GET](https://reqbin.com/Article/HttpGet) request methods, and several others. Web browsers usually only use HTTP GET and HTTP POST, but RESTful desktop and mobile applications use many others. Sending data to the server over HTTP can be done using several HTTP request methods. The HTTP POST request method is one of them. The HTTP POST method asks the web server to accept the data contained in the body of the message. The data type in the HTTP POST body is indicated by the [Content-Type](https://reqbin.com/Article/ContentType) header.

What is the HTTP POST request method used for?

The HTTP POST method is used to create or add a resource on the server. Typically, the POST request adds a new resource to the server, while the PUT request replaces an existing resource on the server. For example, the HTTP POST request method is used by browsers when submitting HTML form data to the server or when submitting data using jQuery/AJAX requests. Unlike GET and [HEAD](https://reqbin.com/Article/HttpHead) requests, the HTTP POST requests may change the server state.

Sending HTTP headers with HTTP POST request

If you are sending data with your POST request, you must provide the Content-Type and Content-Length HTTP headers that indicate the type and size of the data in your POST message. These HTTP headers will help the server interpret and process the sent data correctly. Alternatively, you can omit the [Content-Length](https://reqbin.com/Article/ContentLength) header for your POST request and use the Transfer-Encoding header instead. If the Content-Length and Transfer-Encoding headers are missing, the connection MUST be closed at the end of the response.

POST /post HTTP/1.1

Content-Type: application/json

Content-Length: 1024

Host: reqbin.com

Sending data with HTTP POST method

To send data using the HTTP POST method, you must include the data in the body of the HTTP POST message and specify the MIME type of the data with a Content-Type header. Below is an example of an HTTP POST request to send JSON data to the server.  
  
The size and data type for HTTP POST requests is not limited. But you must specify the data type in the Content-Type header and the data size in the Content-Length header fields. The HTTP POST requests can also send data to the server using the URL parameters. In this case, you are limited to the maximum size of the URL, which is about 2000 characters (it depends on the browser).  
  
The HTTP POST method is not idempotent, which means that sending an identical POST request multiple times may additionally affect the state or cause further side effects (eg. on financial transactions).

HTTP POST Request Example

The following HTTP POST request example demonstrates sending a POST request to the server. In this example, the 'Content-Type: application/json' request header indicates the media type of the resource, and the 'Content-Length: 85' request header indicates the size of the data in the request body.

HTTP POST Request Example[Run Example](https://reqbin.com/req/zvtstmpb)

POST /echo/post/json HTTP/1.1

Authorization: Bearer mt0dgHmLJMVQhvjpNXDyA83vA\_Pxh33Y

Accept: application/json

Content-Type: application/json

Content-Length: 85

Host: reqbin.com

{

"Id": 12345,

"Customer": "John Smith",

"Quantity": 1,

"Price": 10.00

}

And the server response to our request.

Server Response to HTTP POST Request

HTTP/1.1 200 OK

Content-Length: 19

Content-Type: application/json

{"success":"true"}

Submitting HTML forms using the HTTP POST method

[HTML forms](https://reqbin.com/req/yjok4snr/post-html-form-example) are used to collect user input and submit it to a web server. There are two options for submitting web forms to the server: using the HTTP POST or HTTP GET methods. In the example below, the "method=POST" form attribute tells the browser to submit the webform using the HTTP POST method, and the "action=/login" attribute specifies the destination URL.

Submit HTML form using POST method

<form action="/login" method="POST">

<input type="text" name="login">

<input type="password" name="password">

<input type="submit" value="Submit">

</form>

Sending HTTP POST requests with JavaScript

In [JavaScript](https://reqbin.com/code/javascript), you can send HTTP requests using the [XMLHttpRequest](https://reqbin.com/code/javascript/car0zbiq/javascript-xmlhttprequest-example) object or the new Fetch web API. In the example below, we are making an HTTP POST request using the XMLHttpRequest object.

JavaScript HTTP POST Request Example

var url = "https://reqbin.com/echo/post/json";

var xhr = new XMLHttpRequest();

xhr.open("POST", url);

xhr.setRequestHeader("Accept", "application/json");

xhr.setRequestHeader("Content-Type", "application/json");

xhr.onreadystatechange = function () {

if (xhr.readyState === 4) {

console.log(xhr.status);

console.log(xhr.responseText);

}};

xhr.send('{"Id": 78912, "Customer": "Jason Sweet"}');

Sending HTTP POST requests with Python

Python has an excellent "requests" library for making HTTP requests. Using the "requests" library, you can send POST requests with just a few lines of code.

Python HTTP POST Request Example

import requests

from requests.structures import CaseInsensitiveDict

url = "https://reqbin.com/echo/post/json"

headers = CaseInsensitiveDict()

headers["Accept"] = "application/json"

headers["Content-Type"] = "application/json"

data = """

{

"Id": 78912,

"Customer": "Jason Sweet",

}

"""

resp = requests.post(url, headers=headers, data=data)

print(resp.status\_code)

Sending HTTP POST requests with PHP

For [PHP](https://reqbin.com/code/php), you can use the built-in [Curl](https://reqbin.com/curl) module to make HTTP requests. Following is an example HTTP POST request to the ReqBin echo URL.

PHP HTTP POST Request Example

<?php

$url = "https://reqbin.com/echo/post/json";

$curl = curl\_init($url);

curl\_setopt($curl, CURLOPT\_URL, $url);

curl\_setopt($curl, CURLOPT\_POST, true);

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, true);

$headers = array(

"Accept: application/json",

"Content-Type: application/json",

);

curl\_setopt($curl, CURLOPT\_HTTPHEADER, $headers);

$data = <<<DATA

{

"Id": 78912,

"Customer": "Jason Sweet",

}

DATA;

curl\_setopt($curl, CURLOPT\_POSTFIELDS, $data);

$resp = curl\_exec($curl);

curl\_close($curl);

var\_dump($resp);

?>

HTTP POST Method Specification

|  |  |
| --- | --- |
| Safe | No |
| Idempotent | No |
| Cacheable | No |
| Can have a body | Yes |

Some notes on HTTP POST requests

POST requests are never cached

POST requests do not remain in the browser history

POST requests cannot be bookmarked

HTTP POST vs GET Method

|  |  |  |
| --- | --- | --- |
|  | GET | POST |
| Browser BACK button/Reload | Harmless | Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted) |
| Bookmarked | Can be bookmarked | Cannot be bookmarked |
| Cached | Can be cached | Not cached |
| History | Parameters remain in browser history | Parameters are not saved in browser history |
| Restrictions on data length | Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters) | No restrictions |
| Restrictions on data type | Only ASCII characters allowed | No restrictions. Binary data is also allowed |
| Security | GET is less secure compared to HTTP POST because data sent is part of the URL. Never use GET when sending passwords or other sensitive information! |  |
| Visibility | Data is visible to everyone in the URL | Data is not displayed in the URL |

HTTP POST vs PUT Method

The fundamental difference between the HTTP POST and [PUT](https://reqbin.com/Article/HttpPut) requests is reflected in the different meaning of the Request-URI. The URI in a POST request identifies the resource that will handle the enclosed entity. In contrast; the URI in a PUT request identifies the entity enclosed with the request. Practically speaking, POST is used to append a resource to an existing collection, while PUT is used to update an existing resource.