
Technologies ▼

References & Guides ▼

Feedback ▼

Sign in 

 Search

POST

The **HTTP POST method** sends data to the server. The type of the body of the request is indicated by the [Content-Type](#) header.

The difference between PUT and POST is that PUT is idempotent: calling it once or several times successively has the same effect (that is no *side* effect), where successive identical POST may have additional effects, like passing an order several times.

A POST request is typically sent via an [HTML form](#) and results in a change on the server. In this case, the content type is selected by putting the adequate string in the `enctype` attribute of the `<form>` element or the `formenctype` attribute of the `<input>` or `<button>` elements:

- `application/x-www-form-urlencoded`: the keys and values are encoded in key-value tuples separated by '&', with a '=' between the key and the value. Non-alphanumeric characters in both keys and values are percent encoded: this is the reason why this type is not suitable to use with binary data (use `multipart/form-data` instead)
- `multipart/form-data`
- `text/plain`

When the POST request is sent via a method other than an HTML form — like via an `XMLHttpRequest` — the body can take any type. As described in the [HTTP 1.1](#)

specification, POST is designed to allow a uniform method to cover the following functions:

- Annotation of existing resources
- Posting a message to a bulletin board, newsgroup, mailing list, or similar group of articles;
- Adding a new user through a signup modal;
- Providing a block of data, such as the result of submitting a form, to a data-handling process;
- Extending a database through an append operation.

Request has body	Yes
Successful response has body	Yes
<u>Safe</u>	No
<u>Idempotent</u>	No
<u>Cacheable</u>	Only if freshness information is included
Allowed in HTML forms	Yes

Syntax

POST /index.html

Example

A simple form using the default `application/x-www-form-urlencoded` content type:

```
1 | POST / HTTP/1.1
2 | Host: foo.com
3 | Content-Type: application/x-www-form-urlencoded
4 | Content-Length: 13
5 |
6 | say=Hi&to=Mom
```

A form using the multipart/form-data content type:

```
1 | POST /test.html HTTP/1.1
2 | Host: example.org
3 | Content-Type: multipart/form-data;boundary="boundary"
4 |
5 | --boundary
6 | Content-Disposition: form-data; name="field1"
7 |
8 | value1
9 | --boundary
10 | Content-Disposition: form-data; name="field2"; filename="example.txt"
11 |
12 | value2
```

Specifications


Specification	Title
RFC 7231, section 4.3.3: POST	Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

Browser compatibility

New compatibility tables are in beta ▼

Basic support

 	Yes
 	Yes
 	Yes
 	Yes
 	Yes
 	Yes
 	Yes
 	Yes
 	Yes
 	Yes
 	Yes
 	Yes
 	Yes



Full support

See also

- Content-Type
- Content-Disposition