This page was translated from English by the community. Learn more and join the MDN Web Docs community.

#### **POST**

O método HTTP POST envia dados ao servidor. O tipo do corpo da solicitação é indicado pelo cabeçalho Content-Type.

A diferença entre PUT e POST é que PUT é idempotente: chamá-lo uma vez ou várias vezes sucessivamente tem o mesmo efeito (ou seja, nenhum efeito colateral), onde sucessivos POST idênticos podem ter efeitos adicionais, assim como passando uma ordem várias vezes.

Uma solicitação POST geralmente é enviada por meio de um formulário HTML e resulta em uma alteração no servidor. Nesse caso, o tipo de conteúdo é selecionado colocando a string adequada no atributo enctype do elemento <form> ou o atributo formenctype dos elementos <input> ou <but> ou <but > 0 <but >

- application/x-www-form-urlencoded: as chaves e valores são codificados em tuplas de valor-chave separadas por '&', com um '=' entre a chave e o valor. Caracteres não alfanuméricos em chaves e valores são percent encoded (en-US): este é o motivo pelo qual esse tipo não é adequado para uso com dados binários (ao invés disso, use multipart/form-data)
- multipart/form-data
- text/plain

Quando a requisição POST é enviada através de um método diferente de um formulário HTML - como por meio de um XMLHttpReguest - o corpo pode assumir qualquer tipo. Conforme descrito na especificação HTTP 1.1, o POST é projetado para permitir que um método uniforme cubra as seguintes funções:

- Anotação de recursos existentes
- Postar uma mensagem em um quadro de avisos, newsgroup, lista de emails ou grupo similar de artigos;
- Adicionando um novo usuário através de um modal de inscrição;
- Fornecendo um bloco de dados, como o resultado do envio de um formulário, para um processo de manipulação de dados;
- Estendendo um banco de dados por meio de uma operação de append.

Requisição tem corpo	Sim
Resposta bem-sucedida tem corpo	Sim
Seguro	Não
Idempotente	Não
Cacheável	Somente se as informações de atualização estiverem incluídas
Permitido em <u>formulários HTML</u>	Sim

## Sintaxe

POST /index.html

#### Exemplo

Um simples formulário utilizando o padrão content type application/x-www-form-urlencoded:

POST / HTTP/1.1 Host: foo.com

Content-Type: application/x-www-form-urlencoded

Content-Length: 13

say=Hi&to=Mom

Um formulário utilizando o content type multipart/form-data:

POST /test.html HTTP/1.1 Host: example.org

Content-Type: multipart/form-data;boundary="boundary"

--boundary

Content-Disposition: form-data; name="field1"

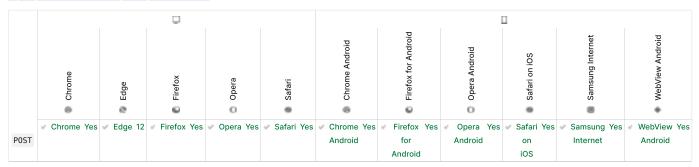
value1
--boundary
Content-Disposition: form-data; name="field2"; filename="example.txt"
value2

# Especificações

Especificação	Título
RFC 7231, sessão 4.3.3: POST ■	Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

## Compatibilidade com navegadores

Report problems with this compatibility data on GitHub



Tip: you can click/tap on a cell for more information.

Full support

# Veja também

- Content-Type
- Content-Disposition

Last modified: 4 de nov. de 2022, by MDN contributors