

# Method in HTTP

The set of common methods for HTTP / 1.1 is defined below and this set of settings can be expanded based on the requirements. These method names are case-sensitive and they must be used in uppercase.

The set of common methods for HTTP / 1.1 is defined below and this set of settings can be expanded based on the requirements. These method names are case-sensitive and they must be used in uppercase.

The **GET** Description method

GET is used to retrieve information from the server provided by using a provided URI. Requests use GET so only the data is received and should not affect the data.

## **HEAD**

Similar to GET, but it transmits the status line and the Header area.

## **POST**

A POST request is used to send data to the Server, for example, customer information, file uploads, etc., using HTML templates.

## **PUT**

Change all current representatives of the target source with the uploaded content.

## **DELETE**

Remove all the current representation of the target source by URI.

## **CONNECT**

Setting a tunnel to the Server defined by the URI provided.

## **OPTIONS**

Describe the communication functions for the target source.

## **TRACE**

Present a loop that checks the message in parallel with path to the target source.

## **GET method**

A **GET** request retrieves data from a Server by specifying the parameters in the request URL. This is the main method used to recover documents. The following example shows how to use the GET method to instruct **hello.htm** :

```
GET /hello.htm HTTP / 1.1
User-Agent: Mozilla / 4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

The server will respond to the above request as follows:

```
HTTP/1.1 200 OK Date: Mon, 27 Jul 2009 12:28:53 GMT Server: Apache/2.2.14 (Win32
```

```
Hello, World!
```

## HEAD method

The **HEAD** method is similar to GET, except that the Server responds with one line and the Headers respond, but no object body. The following example shows the usage of the HEAD method to direct Header information to **hello.htm** :

```
HEAD /hello.htm HTTP / 1.1
User-Agent: Mozilla / 4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
```

The server responding to the HEAD above request is as follows:

```
HTTP / 1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache / 2.2.14 (Win32)
Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT
ETag: "34aa387-d-1568eb00"
Vary: Authorization, Accept
Accept-Ranges: bytes
Content-Length: 88
Content-Type: text / html
Connection: Closed
```

Note that here the Server does not send any data after the Header.

## POST method

The POST method is used when you want to send some data to the Server, for example, updating files, sample data, etc. The following example shows the usage of the POST method to send a sample data to the Server, which will be processed by a process.cgi and finally a response will be returned:

```
POST /cgi-bin/process.cgi HTTP/1.1 User-Agent: Mozilla/4.0 (compatible; MSIE5.01
    xml version = "1.0" encoding = "utf-8" ?>
    span=""> xmlns = "http://clearforest.com/" > string
```

On the Server side, script process.cgi processes the transmitted data and sends the following response:

```
HTTP/1.1 200 OK Date: Mon, 27 Jul 2009 12:28:53 GMT Server: Apache/2.2.14 (Win32
```

```
Request Processed Successfully
```

## PUT method

**The PUT method** is used to request the Server to keep the object body included in a location specified by the URL provided. The following example requires Server to save the object body provided in **hello.htm** at the root of the Server:

```
PUT /hello.htm HTTP/1.1 User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT
```

```
Hello, World!
```

The server will save the object body in hello.jsp file and will send the following response back to the Client:

```
HTTP/1.1 201 Created Date: Mon, 27 Jul 2009 12:28:53 GMT Server: Apache/2.2.14 (
```

```
The file was created.
```

## DELETE method

**The DELETE method** is used to request the server to delete a file at the location specified by the URL provided. The following example requires the Server to delete the file for hello.htm at the root of the Server:

```
DELETE /hello.htm HTTP / 1.1
User-Agent: Mozilla / 4.0 (compatible; MSIE5.01; Windows NT)
Host: www.tutorialspoint.com
```

```
Accept-Language: en-us
Connection: Keep-Alive
```

The server will delete a mentioned file and will send the response back to the Client:

```
HTTP/1.1 200 OK Date: Mon, 27 Jul 2009 12:28:53 GMT Server: Apache/2.2.14 (Win32
```

```
URL deleted.
```

## CONNECT method

**The CONNECT method** is used by the Client to establish a network connection to the Server via HTTP. The following example requires a connection to a Server running on the host **tutorialspoint.com** :

```
CONNECT www.tutorialspoint.com HTTP / 1.1
User-Agent: Mozilla / 4.0 (compatible; MSIE5.01; Windows NT)
```

The connection is established with the Server and the following response is sent back to the Client:

```
HTTP / 1.1 200 Connection established
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache / 2.2.14 (Win32)
```

## OPTIONS method

**The OPTIONS method** is used by the Client to find HTTP methods and functions supported by a Server. The client can define a URL with the OPTIONS method or a \* to target the entire Server. The following example requires a list of methods supported by a Server running on tutorialspoint.com:

```
OPTIONS * HTTP / 1.1
User-Agent: Mozilla / 4.0 (compatible; MSIE5.01; Windows NT)
```

The server sends an information based on the server's current configuration, for example:

```
HTTP / 1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache / 2.2.14 (Win32)
Allow: GET, HEAD, POST, OPTIONS, TRACE
Content-Type: httpd / unix-directory
```

## TRACE method

**The TRACE method** is used to map the contents of an HTTP request to the requester that can be used for the purpose of debugging at the time of development. The following example shows how to use the TRACE method:

```
TRACE / HTTP / 1.1
Host: www.tutorialspoint.com
User-Agent: Mozilla / 4.0 (compatible; MSIE5.01; Windows NT)
```

The server will send the following message in response to the above request:

```
HTTP / 1.1 200 OK
Date: Mon, 27 Jul 2009 12:28:53 GMT
Server: Apache / 2.2.14 (Win32)
Connection: close
Content-Type: message / http
Content-Length: 39
```

```
TRACE / HTTP / 1.1
Host: www.tutorialspoint.com
User-Agent: Mozilla / 4.0 (compatible; MSIE5.01; Windows NT)
```

### **According to Tutorialspoint**

Previous post: [Feedback \(Response\) in HTTP](#)

Next article: [Encrypt status in HTTP](#)

You finished reading the article "**Method in HTTP**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.