

HTTP Request

HTTP Requests are messages which are sent by the client or user to initiate an action on the server.

The first line of the message includes the request message from the client to the server, the method which is applied to the resource, identifier of the resource, and the protocol version.

Syntax

1. Request = Request-Line
2. *((general-header
3. | request-header
4. | entity-header) CRLF)
5. CRLF
6. [message-body]

Request Line

The Request-Line starts with a method token, which is followed by the Request-URI, the protocol version, and ending with CRLF. Using the SP characters, the elements are separated.

Syntax

1. Request-Line = Method SP Request-URI SP HTTP-Version CRLF

1) Method

The method token is used to indicate the method which was performed on the resource identified by the Request-URI. The method is case sensitive.

Syntax

1. Method = "OPTIONS"
2. | "GET"
3. | "HEAD"
4. | "POST"
5. | "PUT"
6. | "DELETE"
7. | "TRACE"
8. | "CONNECT"
9. | extension-method
10. extension-method = token

A resource is allowed a list of methods and that methods can be specified in an Allow header field. The response's return code always notifies the client whether a method is currently allowed on a resource. Since the set of allowed methods can be changed dynamically.

Method and Description:

i) GET

This method retrieves information from the given server using a given URI. GET request can retrieve the data. It cannot apply other effects on the data.

Syntax

1. Request-Line = Method SP Request-URI SP HTTP-Version CRLF

1) Method

The method token is used to indicate the method which was performed on the resource identified by the Request-URI. The method is case sensitive.

Syntax

1. Method = "OPTIONS"
2. | "GET"
3. | "HEAD"
4. | "POST"
5. | "PUT"
6. | "DELETE"
7. | "TRACE"
8. | "CONNECT"
9. | extension-method
10. extension-method = token

A resource is allowed a list of methods and that methods can be specified in an Allow header field. The response's return code always notifies the client whether a method is currently allowed on a resource. Since the set of allowed methods can be changed dynamically.

Method and Description:

i) GET

This method retrieves information from the given server using a given URI. GET request can retrieve the data. It cannot apply other effects on the data.

The PUT method is used to replace all the current representations of the target resource with the uploaded content.

v) DELETE

The DELETE method is used to remove all the current representations of the target resource, which is given by URI.

vi) CONNECT

The CONNECT method is used to establish a tunnel to the server, which is identified by a given URI.

2) Return-URI

The Request-URI is a Uniform Resource Identifier. It is used to identify the resource upon which to apply the request.

Syntax

1. Request-URI = "*" | absoluteURI | abs_path | authority

On the nature of the request, these four options for Request-URI depend.

a) The asterisk "*" is used to show that the request does not apply to a particular resource, but it will apply to the server itself. It is allowed only when the method used does not necessarily apply to a resource.

Example

```
OPTIONS * HTTP/1.1
```

b) The **absoluteURI** form is used only when the request is being made to a proxy. The requested proxy is used to forward the request and return the response.

Example

```
GET http://www.javatpoint.com/WWW/TheProject.html HTTP/1.1
```

c) The **absolute path** can't be empty. If in the original URI, none is present, it must be given as "/".

d) The **authority** form is only used by the CONNECT method.

The Resource Identified by a Request

Using the examination of Request-URI and the Host header field, we can determine the exact resource identified by the Internet request.

An origin server must use the following rules for determining the requested resource on an HTTP/1.1 request if the origin server does differentiate based on the host requested.

1. The host will be part of the Request-URI if Request-URI is an absoluteURI.
2. The host will be determined by the Host header field value if the Request-URI is not an absoluteURI, and the request includes a header field of the host.

3. The response MUST be a 400 (Bad Request) error message if the host as determined by rule 1 or 2 is not a valid host on the server.

Request Header Fields

The request-header fields are used to allow the client to pass additional information to the server like the request and the client itself. The request header fields act as request modifiers, with semantics equivalent to the parameters on a programming language method invocation.

Syntax

1. request-header = Accept
2. | Accept-Charset
3. | Accept-Encoding
4. | Accept-Language
5. | Authorization
6. | Expect
7. | From
8. | Host
9. | If-Match
10. | If-Modified-Since
11. | If-None-Match
12. | If-Range
13. | If-Unmodified-Since
14. | Max-Forwards
15. | Proxy-Authorization
16. | Range
17. | Referer
18. | TE
19. | User-Agent

The name of the request-header field can be extended reliably only in combination with a change in the version of the protocol.