**HTTP Methods**

The HTTP method provide the operations available on Rule Execution Server artefact

Mainly there are two type of Methods

1. **Safe Methods**

Implementers should be aware that software represent the user over the internet, and should be careful to allow the user to be aware of any action they might take which may have unexpected outcome.

e.g.

Methods like GET and HEAD are design to only for retrieval data and should not have significance of taking other action

These method ought to be considered as “safe”.

This allow implementer to represent methods, such as POST, PUT and DELETE, in such a way that the user is made aware of the fact that a possibly unsafe action is being requested.

1. **Idempotent Methods**

If an identical request can be made once or several times in a row with the same effect while leaving the server in same state.

**HTTP request methods:**

1. **GET**:-

* The GET method requests a representation of the specified resources. Requests using GET should only retrieve data.

Syntax:-

GET /index.html

1. **HEAD**:-

* The HEAD method asks for a response identical to GET request, but without the response body.
* If the response to a HEAD request shows that cached URL response is now outdated, the cached copy is invalidated even if no GET request was made.

Syntax:-

HEAD /index.html

1. **POST**:-

* The POST method is used to submit an entity to the specified resource, often causing a change in state or side effects on the server
* The type of the body of the request is indicated by the content-type header (it tells client what the content type of the returned content actually is).
* Successive identical POST request may have additional effects, which may cause additional delay.

Syntax:-

POST /test HTTP/1.1

1. **PUT**

* The PUT method replaces all current representations of the target resource with the request payload.
* PUT is idempotent whereas POST is not
* Calling it once or several times successively has the same effect.
* If the target resource does not have current representation then the origin server must inform the user agent by sending a 201(request has succeeded and has led to creation of a resource).
* If the target resource does have a current representation and the representation is successfully modified, then the origin must send either 200(resource has been updated) or a 204(No need to update as no update required).

Syntax:-

PUT /new.html HTTP/1.1

1. **DELETE**:-

* The DELETE method deletes the specified resource.
* A 202 (Accepted) status code if the action will likely succeed but has not yet been enacted.
* A 204 (No Content) status code if the action has been enacted and no further information is to be supplied.
* A 200 (OK) status code if the action has been enacted and the response message includes a representation describing the status.

Syntax:-

DELETE /file.html HTTP/1.1

1. **CONNECT**

* HTTP CONNECT method start two way communications with requested resource
* It is a hop-by-hop method
* It is used to access websites that uses SSL i.e. encrypted connection
* Connection oriented protocol i.e. TCP is used to establish connection to the requested destination
* HTTP proxy server is used to tunnel the connection between client and server.

Syntax:-

CONNECT [www.example.com:443](http://www.example.com:443) HTTP/1.1

1. **OPTIONS**

* It requests permitted communication options for give URL or server.
* For referring entire server asterisk (\*) is used.
* Client can request
  + Access-Control-Allow-Origin
  + Access-Control-Allow-Methods
  + Access-Control-Allow-Headers
  + Access-Control-Max-Age

Syntax

OPTIONS /index.html HTTP/1.1

OPTIONS \* HTTP/1.1

1. **TRACE**

* It is useful for debugging mechanism
* It perform a message loop-back test along the path the target resource.
* Final recipient is declared when Max-Forwards value is 0 in the request it may be origin server or the first server.
* Final recipient of the request should exclude following fields:
  + Message body of a 200 response
  + Content-Type of message/http

Syntax

TRACE /index.html

|  |  |  |
| --- | --- | --- |
| Method | Safe | Idempotent |
| GET | Yes | Yes |
| HEAD | Yes | Yes |
| POST | No | No |
| PUT | No | Yes |
| DELETE | No | Yes |
| CONNECT | No | No |
| OPTIONS | Yes | Yes |
| TRACE | Yes | Yes |