Network Working Group
Request for Comments: 0
Category: Experimental

M. Criel
Epitech
March 2019

Protocol For Project Zia

Status of this Memo

Copyright Notice

Abstract

HTTP functions as a request–response protocol in the client–server computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

Table of Contents

1. Introduction

The protocol created by the team is used only for the Zia project. It defines the structure of requests and responses, as well as an exhaustive set of HTTP request methods and response codes. This protocol should be able suit the needs of a HTTP server, enabling typical HTTP documents and page requests, as well as CGI execution and more.

2. Request-Response

2.1 Request Structure

1. PROTOCOL
   The name of the type of protocol being used

2. PROTOCOL_VERSION
   The current version of the used protocol

3. HEADERS
   Contains additional information about the request, including the client itself

4. METHOD_STRING
   Indicates the type of request method to be performed on the ressource identified by the given PATH

5. HOST
   The domaine name of the server

6. PORT
   The TCP port number the server listen's on

7. PATH
   The request URI that identifies the resource upon which to apply the request

2.2 Response Structure

1. PROTOCOL
   The name of the type of protocol being used, same property used in the request structure

2. PROTOCOL_VERSION
   The current version of the used protocol, same property used in the request structure

3. HEADERS
   Contains additional information about the request, including the client itself, same property used in request structure

4. STATUS
   The 3-digit code that defines the nature of the response
   Example:
   2xx: Success or 5xx: Server error

3. HTTP Methods

1. GET
   The GET method is used to retrieve information from the given host using a given URI

2. POST
   The POST method is used to send data to the server

3. HEAD
   Same behaviour as the GET method, but only transfers the status line and the header

4. OPTIONS
   Describe the communication options for the target resource

5. PATCH
   The PATCH method applies partial modifications to a ressource

6. CONNECT
   Establishes a tunnel to the server identified by given URI

7. PUT
   Replaces all current representations of the target resource with the uploaded content

8. TRACE
   Performs a message loop back test along with the path to the target resource

9. DELETE
   Deletes all the current representations of the targeted resource given by URI

4. HTTP Response Codes

1. 200 OK
   The request was successful, the actual response depends on the request method called

2. 201 CREATED
   The request has been fulfilled, thus creating a new resource

3. 203 ACCEPTED
   The request has been accepted for processing, but the processing has not been completed. This response code is generally used when another process or server handles the request

4. 204 NO_CONTENT
   The server processed the request with success, but doesn't return any content. Although no content was sent, headers can still be useful

5. 300 MULTIPLE_CHOICES
   Indicates multiple options for the resource from which the client may choose

6. 301 MOVE_PERMANENTLY
   All future requests, including the current request should be directed to the given URI

7. 302 MOVE_TEMPORARILY
   The resource is temporarily directed to the given URI

8. 304 NOT_MODIFIED
   Indicates that the resource has not been modified since the specified version by the request headers

9. 400 BAD_REQUEST
   The server received a request that cannot be processed due to an invalid syntax error

10. 401 UNAUTHORIZED
    The request was not processed due to missing valid authentication information for the targeted resource

11. 403 FORBIDDEN
    Indicates that the request was valid and understood but refuses to be accepted by the server

12. 404 NOT_FOUND
    The requested resource could not be found

13. 500 INTERNAL_SERVER_ERROR
    An internal error was encountered by the server, thus preventing a response and throwing a generic error message

14. 501 NOT_IMPLEMENTED
    The request was not recognised or cannot be processed due to the server's lack of ability to fulfil the request

15. 502 BAD_GATEWAY
    The server was acting as a gateway and received an invalid response from the upstream server

16. 503 SERVICE_UNAVAILABLE
    The service is currently unavailable, generally due to an overload of the server or a maintenance