Web Server

This POC project has been implemented using python programming language. Here the team has tried to implement GET, HEAD, POST & PUT HTTP requests as per the RFC2616. As this implementation has started from the scratch, not all of the aspects of these HTTP requests are covered in the given short time span & there are some missing features of the requests.

As mentioned above 4 HTTP requests are implemented & tested locally using cURL tool. Here mainly the server will have one web page & a data.json file. So the main intention of the server is to maintain the (key, value) pair data & the same is save in the data.json file as json data format. All the (key, value) pair data are saved in a list which will be accessed by their indices. So the web server has simple layout containing web page & a json dat file.

Following are the HTTP request implementation details & their usage using cURL tool:

As the testing is done locally on the same system on which the web server is run, to access the server we need to address to the following address: <http://localhost:5555>. To access the data from the server URI to be used is <http://localhost:5555/post> & each (key, value) data pair can be accessed with their respect indices as <http://localhost:5555/post/1> & etc.

**GET request:**

As per the RFC, GET will fetch information from the server through a URI location mentioned in the request.

Following is the command to be used to trigger GET request to the server.

Ex 1:

$ curl -i -X GET <http://localhost:5555>

$ HTTP/1.1 200 OK

Date: Thu, 20 Sep 2018 13:34:33 GMT

Connection: close

Content-Type: text/html

<html><head><meta http-equiv="Content-Type" content="text/html; charset=utf-8"></head><body><h1>Aricent Web Server</h1><h2>Welcome to our Web Page</h2></body></html>

Ex 2:

$ curl -i -X GET http://localhost:5555/post/1

$ HTTP/1.1 200 OK

Date: Thu, 20 Sep 2018 14:35:08 GMT

Content-Type: application/json; charset=utf-8

Connection: close

{

"Hello": "Hello World"

}

**HEAD request:**

HEAD request is more or less same as the GET request except this request will not get any message body as part of the response to the request. Following is the request & response formats:

Ex 1:

$ curl -i -X HEAD <http://localhost:5555>

$ HTTP/1.1 200 OK

Date: Thu, 20 Sep 2018 13:34:33 GMT

Connection: close

Content-Type: text/html

Ex 2:

$ curl -i -X HEAD http://localhost:5555/post/1

$ HTTP/1.1 200 OK

Date: Thu, 20 Sep 2018 14:35:08 GMT

Content-Type: application/json; charset=utf-8

Connection: close

**POST request:**

POST request is used to create a (key, value) data pair in the server & hence any data transferred to the server should be sent in the (key, value) data pair. The data can be sent in normal text or json format and the same is shown in the below examples. POST request will basically create new entry in the server.

Ex1:

$ curl -i -X POST --data "key=Hi&value=Hello World!" <http://localhost:5555/post>

$ HTTP/1.1 201 Created

Date: Thu, 20 Sep 2018 16:36:10 GMT

Location: http://localhost:5555/3

Content-Type: application/json; charset=utf-8

Content-Length: 27

Connection: keep - alive

Server: Aricent Web Server

Expires: -1

{

"Hi": "Hello World"

}

Ex2:

$ curl -i -X POST -H 'Content-Type: application/json; charset=utf-8' --data '{"Hi": "Hello World"}' <http://localhost:5555/post>

$ HTTP/1.1 201 Created

Date: Thu, 20 Sep 2018 16:40:42 GMT

Location: http://localhost:5555/3

Content-Type: application/json; charset=utf-8

Content-Length: 27

Connection: keep - alive

Server: Aricent Web Server

Expires: -1

{

"Hi": "Hello World"

}

**PUT request:**

PUT request can be used to create a new data entry or to update the existing one. Here in this implementation the PUT request is used only to update the existing data entry. For PUT requests we need to provide exact path in the URI.

Ex1:

$ curl -i -X PUT --data "key=Hi&value=How are you!" <http://localhost:5555/post/3>

$ HTTP/1.1 200 OK

Date: Thu, 20 Sep 2018 16:48:27 GMT

Location: http://localhost:5555/post/3

Content-Length: 27

Content-Type: application/json; charset=utf-8

Connection: keep - alive

Server: Aricent Web Server

Expires: -1

{

"Hi": "How are you"

}

Ex2:

$ curl -i -X PUT -H 'Content-Type: application/json; charset=utf-8' --data '{"Hi": "Hello World! How are you!"}' <http://localhost:5555/post/2>

$ HTTP/1.1 200 OK

Date: Fri, 21 Sep 2018 00:38:53 GMT

Location: http://localhost:5555/post/2

Content-Length: 41

Content-Type: application/json; charset=utf-8

Connection: keep - alive

Server: Aricent Web Server

Expires: -1

{

"Hi": "Hello World! How are you!"

}