

Name :Sushma

Roll No: CS20BETCH11051

Procedure for running question-1:

Initially , no key-value present in server dictionary . we had to populate it using PUT request first
All request will be entered on terminal of client (h1)

Syntax for request (which we have to entered on terminal):

GET /assignment1?key=key1 HTTP/1.1

PUT /assignment1/key1/val1 HTTP/1.1

DELETE /assignment1/key1 HTTP/1.1

First go inside the basic folder , then:

1.make clean

2.make run

After which mininet open then runs following

3 xterm h1 <h1 terminal open>

4 xterm h2 <h2 terminal open>

Terminal of h2 :

5 bash h2-arp.sh

6 python server.py

Terminal of h1 :

7 bash h1-arp.sh

8 python client.py

Now on client terminal ,you can see

Enter the request:

/* enter the request as per syntax , since there is key-pair value in dictionary, so first run a put request*/

eg:

PUT /assignment1/key1/val1 HTTP/1.1

Response : HTTP/1.1 201 OK

GET /assignment1/key=key1 HTTP/1.1

Response : HTTP/1.1 200 OK

val1

DELETE /assignment1/key1 HTTP/1.1

Response : HTTP/1.1 200 OK

record delete successfully

Procedure for running question-2:

Here server already contain the six key value pairs , while cache has no key-value pair

I implemented two requests :GET and PUT

First go inside the star folder , then:

```
1.make clean
2.make run
After which mininet open then runs following
3 xterm h1 <h1 terminal open>
4 xterm h2 <h2 terminal open>
5 xterm h3
Terminal of h3 :
    6 bash h3-arp.sh
    7 python server.py
Terminal of h2 :
    8 bash h2-arp.sh
    9 python cache.py
Terminal of h1 :
   10 bash h1-arp.sh
   11 python client.py
```

Now on client terminal ,you can see

Enter the request:

/* enter the request as per syntax */

Eg:

```
GET /assignment1?key=key1 HTTP/1.1
```

```
Response : HTTP/1.1 200 OK
           val1
```

Responses return by my code :

```
Request: GET /assignment1?key=key1 HTTP/1.1
```

```
Response :
```

```
<If key1-val1 pair present on server dictionary >
HTTP/1.1 200 OK
val1
```

```
<if kay-pair does not present on server >
HTTP/1.1 404 NOT FOUND
```

```
Request: PUT /assignment1/key1/val1 HTTP/1.1
```

```
Response :
```

```
<If key1-val1 pair does not present on server dictionary already >
HTTP/1.1 201 Created
```

```
<if kay-pair present on server, then it will modify with new value and
returned response will be >
HTTP/1.1 200 OK
```

```
Request: DELETE /assignment1/key1 HTTP/1.1
```

```
Response :
```

<If key1-val1 pair present on server dictionary >

HTTP/1.1 200 OK

Record deleted successfully (just for understanding on client)