Introduction to the Web Architecture Assignment

Members of group 15:

Dong Li (dl1e14)

Anis A Sharip (aas3g14)

Jack Webster (jw30g11)

Antigoni Kritioti (ak5e14)

**Overview:**

In our application for the Web Architecture assignment, we developed web pages that allow a client to create, update, get, delete a new question. These HTTP verbs also apply to the answering of a question, as well as the ability to comment on both a question and answer.

Users can run the application in a browser to achieve these functionality through our GUI. Users can also test the functions by running CURL commands.

**Functionality**

1. By using 4 HTTP method verbs including GET, POST, PUT, DELETE, users can upload, create, update, and delete the questions, answers and comments.
2. Users can obtain the head of the http request by using curl command.
3. Using HTTP status codes to present the status of response to a request.
4. The starting URL is “localhost:3000” which can list all questions and its create time.
5. Based on the accept HTTP header, content negotiation can be supported, contents can be provided in JSON and HTML format.

**Runing the application:**

The testing server system is Ubuntu Server LTS 14.04.1, which have Node and SQLite installed.

1. npm install/npm update

Open the folder where the application is in the terminal, and install required modules by command "npm install".our application have install required modules,users can

update these modules by command "npm update".

2. initialize the database

If you want work with current database, you can skip this step, otherwise, you can initialize the database by using command “node create”in terminal.

3. npm start

Start the server of the application. use command "npm start" to start the server

4.run the application in browser

Open the browser and type "http://localhost:3000" in the address bar in the browser and there will show the index page of the application, and users can use functions in the web.

**Testing the application:**

1. Using browser

After running server of the application by "npm start" command in terminal, users can test all functionality by GUI in browser.

1. Using curl commands:

We created a shell script file the name of which is “shell test.sh” containing curl calls to test the function of the RESTfull service of our project.

Besides, users can also test these services by curl commands in terminal.After running server of the application by "npm start" command in terminal, open another terminal and test the application by follow curl commands:

1. Testing head:

curl --HEAD localhost:3000/questions

Users can also test the head of the http head when implementing other methods like follow command.

curl -i http://localhost:3000/questions

1. Testing HTTP verbs:

our application supported 4 kinds of HTTP verbs(get, post, put, delete),users can test these HTTP verbs by following curl commands.(the question for which id is “2” is already exists in the database)

1. Testing GET method:

curl http://localhost:3000/questions

1. Testing post method:

curl -d “title=new question&contents=new question contents?” http://localhost:3000/questions

1. Testing put method:

curl -X PUT -d "title=modify question&contents=changed question" http://localhost:3000/questions/2

1. Testing delete method:

curl -X DELETE http://localhost:3000/questions/2

users can test 4 HTTP verbs of answers and comments by routers in API

**API**

Every part of web application can support RESTful service

1. question part:

GET method

http://localhost:3000/questions

http://localhost:3000/questions/:question\_id

POST method

http://localhost:3000/questions

PUT method

http://localhost:3000/questions/:question\_id

DELETE method

http://localhost:3000/questions/:question\_id

1. answer part:

GET method

http://localhost:3000/questions/:questin\_id/answers

http://localhost:3000/questions/:question\_id/answers/:answer\_id

POST method

http://localhost:3000/questions/:question\_id/answers

PUT method

http://localhost:3000/answers/:answer\_id

DELETE method

http://localhost:3000/answers/:answer\_id

1. question comment part:

GET method

http://localhost:3000/questions/:questin\_id/qcomments

http://localhost:3000/questions/:question\_id/qcomments/:qcomment\_id

POST method

http://localhost:3000/questions/:question\_id/qcomments

PUT method

http://localhost:3000/qcomments/:qcomment\_id

DELETE method

http://localhost:3000/qcomments/:qcomment\_id

1. answer comment part:

GET method

http://localhost:3000/answers/:answer\_id/acomments

http://localhost:3000/answers/:answer\_id/acomments/:acomment\_id

POST method

http://localhost:3000/answers/:answer\_id/acomments

PUT method

http://localhost:3000/acomments/:acomment\_id

DELETE method

http://localhost:3000/acomments/:acomment\_id

In addition, users can use the following routers to get a specific page in the browser after running the server:

1.http://localhost:3000/ show the index page

2.http://localhost:3000/about show the about page

3.http://localhost:3000/askQuestion show the page of asking question

4.http://localhost:3000/questions/:question\_id/questionEdit

show page of editing question

5.http://localhost:3000/questions/:question\_id/questionComment

show page of adding question comment

6.http://localhost:3000/questions/:question\_id/qcommentEdit

show page of editing question comment

7.http://localhost:3000/answers/:answer\_id/answerEdit

show page of editing answer

8.http://localhost:3000/answers/:answer\_id/answerCommen

show page of adding answer comment

9.http://localhost:3000/acomments/:acomment\_id/acommentEdit

show page of edting answer comment

10.http://localhost:3000/questions/:question\_id

show page of specific question and user can add answer to this question