



API DOCUMENTATION FOR QUORA

CMPE 273 | GROUP PROJECT 1

Submitted by:

Akshay Bajaj: 012705186

Amit Kamboj: 013827489

Dhruvil Bhaveshbhai Parikh: 013772720

Pardis Tashakori: 011548199

Pooja Kataria: 013818415

Shivang Mistry: 013823108

Routes

Link to API Document:

<https://documenter.getpostman.com/view/6891469/S17wP6R3#9d8c0479-490f-4add-b7ce-afa5e0d79374>

1. Creating a new account to Quora

POST Creating a new account to Quora

/signup

HEADERS

Content-Type application/json

BODY

```
{
  "firstname": "FIRSTNAME",
  "lastname": "LASTNAME",
  "email": "EMAIL",
  "password": "PASSWORD"
}
```

Example Request

Creating a new account to Quora

Example Response

200 — OK

```
{
  "status": 1,
  "msg": "Account Created Successfully"
}
```

2. Login to Quora

POST Login to Quora

/login

BODY

```
{
  "email": "EMAIL",
  "password": "PASSWORD"
}
```

Example Request

Login to Quora

Example Response

200 — OK

```
{
  "status": 1,
  "msg": "Logged in Successfully",
  "data": { All Information about user }
}
```

3. Editing User Information

PUT Editing User Information

/user/:userid

HEADERS

Content-Type application/json

PATH VARIABLES

userid 1
UNIQUE ID associated with the user

BODY

```
{
  "firstname": "FIRSTNAME",
  "lastname": "LASTNAME",
  "email": "EMAIL",
  "password": "PASSWORD",
  "city": "CITY",
  "state": "STATE",
  "country": "COUNTRY",
  "education": {
    {
```

Example Request

Editing User Information

Example Response

200 – OK

```
{
  "status": 1,
  "msg": "Updated Successfully",
  "data": {All information about user}
}
```

4. Delete an existing account:

DEL Delete an existing account

/user/:userid

PATH VARIABLES

userid 1
"UNIQUE ID associated with the user"

Example Request

/user/:userid

Example Response

200 – OK

```
{
  "status": 1,
  "msg": "Account Deleted Successfully"
}
```

5. Getting Information of the particular user

GET Getting Information of the particular user

/user/:userid

HEADERS

Content-Type	application/json
---------------------	------------------

PATH VARIABLES

userid	1 id of the user who is logged in
---------------	--------------------------------------

Example Request

Getting Information of the particular user

Example Response

200 – OK

```
{
  "profile": "profile information like name, education, city, state, etc..",
  "follower": "List of people followed this user ",
  "following": "List of people followed by this user",
  "questions": "List of questions posted by user",
  "answers": "List of answers posted by user",
  "activity": "List of activities done by user",
  "topics": "List of topics user know"
}
```

6. Follow another user

PUT Follow another user

/user/:userid/follow

HEADERS

Content-Type	application/json
---------------------	------------------

PATH VARIABLES

userid	1 id of the user who is logged in
---------------	--------------------------------------

BODY

```
{
  "anotheruserid": "2"
}
```

Example Request

Follow another user

Example Response

200 – OK

```
{
  "status": 1,
  "msg": "user followed"
}
```

7. Unfollow user

PUT Unfollow User

/user/:userid/unfollow

HEADERS

Content-Type application/json

PATH VARIABLES

userid 1
id of the user who is logged in

BODY

```
{
  "anotheruserid": 2
}
```

Example Request

Unfollow User

Example Response

200 – OK

```
{
  "status": 1,
  "msg": "user unfollowed"
}
```

8. Add Question

POST Add Question

/user/:userid/question

HEADERS

Content-Type application/json

PATH VARIABLES

userid 1
id of the user who is logged in

BODY

```
{
  "question": "Question limited to 250 characters",
  "visibilty": "public/ anonymous/ limited"
}
```

Example Request

Add Question

Example Response

200 – OK

```
{
  "status": 1,
  "msg": "Question Posted successfully"
}
```

9. View specific question

GET View specific question

./question

PATH VARIABLES

question	"What is API Document and How can it be written?" Question Description which is limited to 250 characters
-----------------	--

Example Request

View specific question

Example Response

200 – OK

```
{
  "status":1,
  "msg": "success",
  "data": {
    "question":"What is API Document and How can it be written?",
    "answers":[
      {
        "answerid":"id of the answer",
        "user": "user information who gave this answer",
        "answerinfo":"info such as upvotes, downvotes, views",

```

```
{
  "status":1,
  "msg": "success",
  "data": {
    "question":"What is API Document and How can it be written?",
    "answers":[
      {
        "answerid":"id of the answer",
        "user": "user information who gave this answer",
        "answerinfo":"info such as upvotes, downvotes, views",
        "comments": "List of comments with user information"
      }
    ]
  }
}
```

10. Upvote an answer

PUT Upvote an answer

/answer/:answerid/upvote

HEADERS

Content-Type application/json

PATH VARIABLES

answerid 1
unique answerid

BODY

```
{
  "userid": 1
}
```

Example Request

Upvote an answer

Example Response

200 — OK

```
{
  "status": 1,
  "msg": "Answer upvoted successfully"
}
```

11. Downvote an answer

PUT Downvote an answer

/answer/:answerid/downvote

HEADERS

Content-Type application/json

PATH VARIABLES

answerid 1
unique answerid

BODY

```
{
  "userid": 1
}
```

Example Request

Downvote an answer

Example Response

200 — OK

```
{
  "status": 1,
  "msg": "Answer downvoted successfully"
}
```

12. Getting questions from a specific user

GET Getting Questions from specific user

/user/:userid/questions

PATH VARIABLES

userid	2
	unique id of the user

Example Request

Getting Questions from specific user

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "success"
  "data": {
    "questions": ["List of Questions and their information"]
  }
}
```

13. Getting answers given by a specific user

GET Getting answers given by a specific user

/user/:userid/answers

PATH VARIABLES

userid	2
	id of the user

Example Request

Getting answers given by a specific user

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "success"
  "data": {
    "questions": ["List of Questions and their information"],
    "answers": ["List of Answers and their information"]
  }
}
```


14. Bookmark an answer

PUT Bookmark an answer

/bookmark/:answerid

HEADERS

Content-Type	application/json
---------------------	------------------

PATH VARIABLES

answerid	1 id of the answer
-----------------	-----------------------

BODY

```
{
  "userid": "id of the user who click on bookmark"
}
```

Example Request

Bookmark an answer

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "answer bookmarked successfully"
}
```

15. Getting all bookmarked answers

GET Getting all bookmarked answers

/bookmarks

HEADERS

Content-Type	application/json
---------------------	------------------

BODY

```
{
  "userid": "id of the user who is logged in"
}
```

Example Request

Getting all bookmarked answers

Example Response

200 - OK

```
{
  "status" : 1,
  "msg" : "success"
  "data": ["All Answers bookmarked"]
}
```

16. Getting inbox

GET Getting inbox

/messages

HEADERS

Content-Type application/json

BODY

```
{
  "userid": "id of the user who is logged in"
}
```

Example Request

Getting inbox

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "success",
  "data": [
    "Getting all conversion of this user"
  ]
}
```

17. Sending a message to other user

POST Sending a message to other user

/messages/new

BODY

```
{
  "userid": "id of the user logged in",
  "name" : "name of other user",
  "message": "new message to be sent"
}
```

Example Request

Sending a message to other user

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "success",
  "data": ""
}
```

18. Getting older conversation

GET Getting older conversation

/messages/thread/:thread_id

HEADERS

Content-Type application/json

PATH VARIABLES

thread_id 1
id of the conversation

Example Request

Getting older conversation

Example Response

200 - OK

```
{
  "status" : 1,
  "msg" : "success"
  "data": ["thread with all messages in thread"]
}
```

19. Getting user content

GET Getting user content

/content

HEADERS

Content-Type application/json

BODY

```
{
  "userid": "id of the user logged in"
}
```

Example Request

Getting user content

Example Response

200 - OK

```
{
  "status" : 1,
  "msg" : "success"
  "data": ["all the content in chronological order"]
}
```

20. Getting questions from the given topic

GET getting questions from the given topic

/topic/:topicname

PATH VARIABLES

topicname	What is the best app for making video fan edits? name of the topic
------------------	---

Example Request

getting questions from the given topic

Example Response

200 - OK

```
{
  "status" : 1,
  "msg" : "success"
  "data": ["all the questions from the given topic"]
}
```

21. Getting questions to answer from topics followed

GET getting questions to answer from topics followed

/answer

HEADERS

Content-Type	application/json
---------------------	------------------

BODY

```
{
  "userid": "id of the user logged in"
}
```

Example Request

getting questions to answer from topics followed

Example Response

200 - OK

```
{
  "status" : 1,
  "msg" : "success"
  "data": ["List of question to answer from topics followed"]
}
```

22. Getting unanswered question

GET Getting unanswered question

/unanswered/:question_name

HEADERS

Content-Type application/json

PATH VARIABLES

question_name

Example Request

Getting unanswered question

Example Response

200 - OK

```
{
  "status" : 1,
  "msg" : "success"
  "data": "Unanswered question"
}
```

23. Answer a question

POST Answer a question

/:questionid/answer

HEADERS

Content-Type application/json

PATH VARIABLES

questionid 1
id of the question

Example Request

Answer a question

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "successfully posted an answer"
}
```

24. Anonymously answer a question

POST anonymously answer a question

`/:questionid/answer/anonymous`

HEADERS

Content-Type `application/json`

PATH VARIABLES

questionid

BODY

```
{
  "answerlink": "URL of the anonymous answer"
}
```

Example Request

`anonymously answer a question`

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "successfully posted an answer anonymously"
}
```

25. Edit the answer

PUT Edit the answer

`/:questionid/answer/:answer`

PATH VARIABLES

questionid

answer

Example Request

`Edit the answer`

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "successfully updated an answer"
}
```

26. Search the question, answer, topic, profile

GET Search question, answer, topic, profile

/search	
HEADERS	
Content-Type	application/json
BODY	
<pre>{ "search_string": "any string to search" }</pre>	

Example Request

Search question, answer, topic, profile

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "success",
  "data": "List of all results with type of the result "
}
```

27. Getting Stats of the user to show different types of graphs

GET Getting Stats of the user to show different types of graphs

/stats	
HEADERS	
Content-Type	application/json
BODY	
<pre>{ "userid": "id of the user logged in" }</pre>	

Example Request

Getting Stats of the user to show different types of graphs

Example Response

200 - OK

```
{
  "status": 1,
  "msg": "success",
  "data": "List of top answers with its views, upvotes or downvotes"
}
```