

Set up Guide

On Ubuntu Server LTS 12.04 node app is started using “node app.js” on a command line. Server will start listening on a port 3000. This will also create sqlite database “ws.db” in the same folder as app.js if does not exist yet with the tables for questions, answers and comments.

API

For an API call a non-empty json object is returned whenever a request produces json representation of a question, answer or comment or a list of these objects. Empty json object is returned when a request does not produce any response such as for a DELETE method or when an error occurs. HTTP status will be visible in a header of each response from a server including error codes for failed requests.

1) Questions

a) Create a question – POST request to “<host>:<port>/questions”

POST parameters:

- title (required): title of a new question
- body (required): body text of a new question

Returns:

- status 201 Created with a Location header containing a url to a newly created question on success
- status 400 Bad Request if either title or body is missing
- status 500 Internal Server Error if a server error occurs

Example:

POST to “<host>:<port>/questions” with title=”title1” and body=”text1”

Response:

```
HTTP/1.1 201 Created
X-Powered-By: Express
Location: /questions/1
Content-Type: application/json; charset=utf-8
Content-Length: 2
Date: Tue, 17 Dec 2013 12:50:15 GMT
Connection: keep-alive
```

```
{}
```

b) Retrieve a question – GET request to “<host>:<port>/questions/:qid”

url parameters:

- qid: id of a question

Returns:

- json representation of a question on success
- status 400 Bad Request if “qid” is not a valid integer
- status 404 Not Found if a question with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

GET request to “<host>:<port>/questions/1”

Response:

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 112
ETag: "2127802202"
Date: Tue, 17 Dec 2013 12:50:15 GMT
Connection: keep-alive
```

```
{
  "title": "title1",
  "body": "text1",
  "date": 1387284614864,
  "answers_count": 0,
  "comments_count": 0
}
```

“date” is a timestamp when question was created

“answers_count” is a number of answers for this question

“comments_count” is a number of comments for this question

c) Update a question – PUT request to “<host>:<port>/questions/:qid”

url parameters:

- qid: id of a question

PUT parameters:

- title (optional): new title of a question
- body (optional): new body text of a question

Returns:

- status 204 No Content on success
- status 400 Bad Request if “qid” is not a valid integer
- status 404 Not Found if a question with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

PUT request to “<host>:<port>/questions/1” with title=”title2”
Response:

HTTP/1.1 204 No Content
X-Powered-By: Express
Date: Tue, 17 Dec 2013 12:50:15 GMT
Connection: keep-alive

d) Delete a question – DELETE request to “<host>:<port>/questions/:qid”

url parameters:
- qid: id of a question

Returns:

- status 204 No Content on success
- status 400 Bad Request if “qid” is not a valid integer
- status 404 Not Found if a question with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

DELETE request to “<host>:<port>/questions/1”

Response:

HTTP/1.1 204 No Content
X-Powered-By: Express
Date: Tue, 17 Dec 2013 12:50:16 GMT
Connection: keep-alive

e) Retrieve a list of all questions - GET request to “<host>:<port>/questions”

query parameters:
- count (optional): number of questions to return starting from the first
- start (optional): starting index of a question from a list
- end (optional) ending index of a question from a list

“start” and “end” must both be present if one of them is present. If they are absent then number of questions is given by “count”. If all parameters are absent then all questions are returned.

Returns:

- json array of questions in descending order on success
- status 400 Bad Request if “count”, “start” or “end” are not valid integers or do not represent valid ranges.
- status 500 Internal Server Error if a server error occurs

Example:

GET request to “<host>:<port>/questions”

Response:

HTTP/1.1 200 OK

X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 130
ETag: "-1980824993"
Date: Tue, 17 Dec 2013 12:50:16 GMT
Connection: keep-alive

```
[
  {
    "title": "title1",
    "body": "text1",
    "date": 1387284616226,
    "answers_count": 0,
    "comments_count": 0
  }
]
```

2) Answers

- a) Create an answer – POST request to
“<host>:<port>/questions/:qid/answers”

url parameters:
- qid: id of a question

POST parameters:
– body (required): body text of a new answer

Returns:

- status 201 Created with a Location header containing a url to a newly created answer on success
- status 400 Bad Request if a body is missing or “qid” is not a valid integer
- status 404 Not Found if a question with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

POST request to “<host>:<port>/questions/2/answers” with
body=”text2”

Response:

HTTP/1.1 201 Created
X-Powered-By: Express
Location: /questions/2/answers/1
Content-Type: application/json; charset=utf-8
Content-Length: 2
Date: Tue, 17 Dec 2013 12:50:17 GMT
Connection: keep-alive

{}

b) Retrieve an answer – GET request to
“<host>:<port>/questions/:qid/answers/:aid”

url parameters:
- qid: id of a question
- aid: id of an answer

Returns:

- json representation of an answer on success
- status 400 Bad Request if “qid” or “aid” is not a valid integer
- status 404 Not Found if a question or an answer with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

GET request to “<host>:<port>/questions/2/answers/1”

Response:

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 69
ETag: "-389286306"
Date: Tue, 17 Dec 2013 12:50:17 GMT
Connection: keep-alive
```

```
{
  "body": "text2",
  "date": 1387284616726,
  "comments_count": 0
}
```

“date” is a timestamp when an answer was created

“comments_count” is a number of comments for this answer

c) Update an answer – PUT request to
“<host>:<port>/questions/:qid/answers/:aid”

url parameters:
- qid: id of a question
- aid: id of an answer

PUT parameters:

- body (optional): new body text of an answer

Returns:

- status 204 No Content on success
- status 400 Bad Request if “qid” or “aid” is not a valid integer
- status 404 Not Found if a question or an answer with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

PUT request to “<host>:<port>/questions/2/answers/1” with
body=”text3”

Response:

HTTP/1.1 204 No Content
X-Powered-By: Express
Date: Tue, 17 Dec 2013 12:50:18 GMT
Connection: keep-alive

d) Delete an answer – DELETE request to
“<host>:<port>/questions/:qid/answers/:aid”

url parameters:

- qid: id of a question
- aid: id of an answer

Returns:

- status 204 No Content on success
- status 400 Bad Request if “qid” or “aid” is not a valid integer
- status 404 Not Found if a question or an answer with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

DELETE request to “<host>:<port>/questions/2/answers/1”

Response:

HTTP/1.1 204 No Content
X-Powered-By: Express
Date: Tue, 17 Dec 2013 14:28:36 GMT
Connection: keep-alive

e) Retrieve a list of all answers for a question - GET request to
“<host>:<port>/questions/:qid/answers”

url parameters:

- qid: id of a question

query parameters:

- count (optional): number of answers to return starting from the first
- start (optional): starting index of an answer from a list
- end (optional) ending index of an answer from a list

“start” and “end” must both be present if one of them is present. If they are absent then number of answers is given by “count”. If all parameters are absent then all answers are returned.

Returns:

- json array of answers in descending order on success
- status 400 Bad Request if “count”, “start” or “end” are not valid integers or do not represent valid ranges or “qid” is not a valid integer
- status 404 Not Found if a question with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

GET request to “<host>:<port>/questions/2/answers”

Response:

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 83
ETag: "1433029315"
Date: Tue, 17 Dec 2013 12:50:20 GMT
Connection: keep-alive
```

```
[
  {
    "body": "text2",
    "date": 1387284619509,
    "comments_count": 0
  }
]
```

3) Question Comments

- a) Create a comment – POST request to
“<host>:<port>/questions/:qid/comments”

url parameters:

- qid: id of a question

POST parameters:

- body (required): body text of a new comment

Returns:

- status 201 Created with a Location header containing a url to a newly created comment on success
- status 400 Bad Request if a body is missing or “qid” is not a valid integer
- status 404 Not Found if a question with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

POST request to “<host>:<port>/questions/2/comments” with
body=”text4”

Response:

```
HTTP/1.1 201 Created
X-Powered-By: Express
Location: /questions/2/comments/1
Content-Type: application/json; charset=utf-8
Content-Length: 2
Date: Tue, 17 Dec 2013 12:50:21 GMT
Connection: keep-alive
```

```
{}
```

b) Retrieve a comment – GET request to
“<host>:<port>/questions/:qid/comments/:cid”

url parameters:

- qid: id of a question
- cid: id of a comment

Returns:

- json representation of a comment on success
- status 400 Bad Request if “qid” or “cid” is not a valid integer
- status 404 Not Found if a question or a comment with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

GET request to “<host>:<port>/questions/2/comments/1”

Response:

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 46
ETag: "-1294402213"
Date: Tue, 17 Dec 2013 12:50:21 GMT
Connection: keep-alive
```



```
{  
  "body": "text4",  
  "date": 1387284620488  
}
```

c) Update a comment – PUT request to
“<host>:<port>/questions/:qid/comments/:cid”

url parameters:

- qid: id of a question
- cid: id of a comment

PUT parameters:

- body (optional): new body text of a comment

Returns:

- status 204 No Content on success
- status 400 Bad Request if “qid” or “cid” is not a valid integer
- status 404 Not Found if a question or a comment with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

PUT request to “<host>:<port>/questions/2/comments/1” with
body=”text5”

Response:

```
HTTP/1.1 204 No Content  
X-Powered-By: Express  
Date: Tue, 17 Dec 2013 14:28:39 GMT  
Connection: keep-alive
```

d) Delete a comment – DELETE request to
“<host>:<port>/questions/:qid/comments/:cid”

url parameters:

- qid: id of a question
- cid: id of a comment

Returns:

- status 204 No Content on success
- status 400 Bad Request if “qid” or “cid” is not a valid integer
- status 404 Not Found if a question or a comment with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

DELETE request to "<host>:<port>/questions/2/comments/1"

Response:

HTTP/1.1 204 No Content
X-Powered-By: Express
Date: Tue, 17 Dec 2013 14:28:40 GMT
Connection: keep-alive

e) Retrieve a list of all comments for a question - GET request to
"<host>:<port>/questions/:qid/comments"

url parameters:

- qid: id of a question

query parameters:

- count (optional): number of comments to return starting from the first comment
- start (optional): starting index of a comment from a list
- end (optional) ending index of a comment from a list

"start" and "end" must both be present if one of them is present. If they are absent then number of comments is given by "count". If all parameters are absent then all comments are returned.

Returns:

- json array of comments in descending order on success
- status 400 Bad Request if "count", "start" or "end" are not valid integers or do not represent valid ranges or "qid" is not a valid integer
- status 404 Not Found if a question with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

GET request to "<host>:<port>/questions/2/comments"

Response:

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 58
ETag: "-738684138"
Date: Tue, 17 Dec 2013 14:28:40 GMT
Connection: keep-alive

```
[  
{  
  "body": "text4",
```

```
"date": 1387290520096
}
]
```

4) Answer Comments

- a) Create a comment – POST request to
“<host>:<port>/questions/:qid/answers/:aid/comments”

url parameters:

- qid: id of a question
- aid: id of an answer

POST parameters:

- body (required): body text of a new comment

Returns:

- status 201 Created with a Location header containing a url to a newly created comment on success
- status 400 Bad Request if a body is missing, “qid” or “aid” is not a valid integer
- status 404 Not Found if a question or an answer with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

POST request to “<host>:<port>/questions/2/answers/2/comments”
with body=”text6”

Response:

```
HTTP/1.1 201 Created
X-Powered-By: Express
Location: /questions/2/answers/2/comments/3
Content-Type: application/json; charset=utf-8
Content-Length: 2
Date: Tue, 17 Dec 2013 14:28:41 GMT
Connection: keep-alive
```

```
{}
```

- b) Retrieve a comment – GET request to
“<host>:<port>/questions/:qid/answers/:aid/comments/:cid”

url parameters:

- qid: id of a question
- aid: id of an answer
- cid: id of a comment

Returns:

- json representation of a comment on success
- status 400 Bad Request if “qid”, “aid” or “cid” is not a valid integer
- status 404 Not Found if a question, answer or a comment with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

GET request to “<host>:<port>/questions/2/answers/2/comments/1”

Response:

```
HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 46
ETag: "-560382944"
Date: Tue, 17 Dec 2013 14:28:41 GMT
Connection: keep-alive
```

```
{
  "body": "text6",
  "date": 1387290520929
}
```

c) Update a comment – PUT request to

“<host>:<port>/questions/:qid/answers/:aid/comments/:cid”

url parameters:

- qid: id of a question
- aid: id of an answer
- cid: id of a comment

PUT parameters:

- body (optional): new body text of a comment

Returns:

- status 204 No Content on success
- status 400 Bad Request if “qid”, “aid” or “cid” is not a valid integer
- status 404 Not Found if a question, answer or a comment with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

PUT request to “<host>:<port>/questions/2/answers/2/comments/3”
with body=”text7”

Response:

HTTP/1.1 204 No Content
X-Powered-By: Express
Date: Tue, 17 Dec 2013 14:28:42 GMT
Connection: keep-alive

d) Delete a comment – DELETE request to
“<host>:<port>/questions/:qid/answers/:aid/comments/:cid”

url parameters:
- qid: id of a question
- aid: id of an answer
- cid: id of a comment

Returns:

- status 204 No Content on success
- status 400 Bad Request if “qid”, “aid” or “cid” is not a valid integer
- status 404 Not Found if a question, answer or a comment with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

DELETE request to
“<host>:<port>/questions/2/answers/2/comments/3”

Response:

HTTP/1.1 204 No Content
X-Powered-By: Express
Date: Tue, 17 Dec 2013 14:28:43 GMT
Connection: keep-alive

e) Retrieve a list of all comments for an answer - GET request to
“<host>:<port>/questions/:qid/answers/:aid/comments”

url parameters:
- qid: id of a question
- aid: id of an answer

query parameters:
- count (optional): number of comments to return starting from the first comment
- start (optional): starting index of a comment from a list
- end (optional) ending index of a comment from a list

“start” and “end” must both be present if one of them is present. If they are absent then number of comments is given by “count”. If all parameters are absent then all comments are returned.

Returns:

- json array of comments in descending order on success
- status 400 Bad Request if “count”, “start” or “end” are not valid integers or do not represent valid ranges or “qid”, “aid” is not a valid integer
- status 404 Not Found if a question or an answer with a given id does not exist
- status 500 Internal Server Error if a server error occurs

Example:

GET request to “<host>:<port>/questions/2/answers/2/comments”

Response:

HTTP/1.1 200 OK
Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 58
ETag: "-759102184"
Date: Tue, 17 Dec 2013 14:28:43 GMT
Connection: keep-alive

```
[  
  {  
    "body": "text6",  
    "date": 1387290523107  
  }  
]
```