

Project 1 CPSC 476

Project Name: Blog

Team:

Libraries Used:

Hashlib : hashing the password with md5

Sqlite3 : RDBMS server

flask_basicauth : For authorization

flask: for creating a service endpoints

Structure:

- user.py
- comments.py
- tags.py
- article.py
- test_user.tavern.yaml
- test_article.tavern.yaml
- test_comment.tavern.yaml
- test_tag.tavern.yaml
- Main.db
- Main.db.sql
- Procfile

Micro-service: User - user.py

Functionality: Create new user

Url: <http://localhost:5000/user>

Method: POST

Curl command example: curl -POST http://127.0.0.1:5000/user -d '{"user_name": "Shekhar", "hashed_password": "shekhar12345", "full_name": "ashdjk", "email_id": "asjdkhjas@akdhkasj"}'

Input variables :

user_name :user name will be unique key

hashed_password :Provide password for user

full_name:Full name of user

email_id:Email of user

All the variables mentioned above are in lower case.

Functionality: Delete existing user

Url: <http://localhost:5000/user>

Method: DELETE

Login by entering the user id and password.

Curl command example: curl -X DELETE http://127.0.0.1:5000/article -d '{" user_name ":"Amey", "hashed_password ":"passbhor"}' -u

Input variables :

user_name: Provide user name

hashed_password: Password for authentication

All variables mentioned above are in lower case.

Functionality: Change existing user's password

Url: <http://localhost:5000/user>

Method: PATCH

Login by entering the user id and password.

```
curl -X PATCH http://127.0.0.1:5000/user -d '{"hashed_password":"xyz"}' -u
```

Shekhar12345:shekhar12345

Input variables :

 hashed_password: Update current password

All variables mentioned above are in lower case.

Micro-service: Articles - article.py

Functionality: Post a new article

Url: <http://localhost:5200/article>

Method: POST

Curl command example: `curl -GET "http://127.0.0.1:5200/article"`

Input variables:

title: Provide title of the article

content: Provide the content of the article.

All the variables mentioned above are in lower case.

Functionality: Retrieve an individual article

Url: <http://localhost:5200/article>

Method: GET

Curl command example: `curl -GET "http://127.0.0.1:5200/article?limit=1"`

Login by entering the user id and password.

Input variables:

article_id : Provide article id for retrieving article

All the variables mentioned above are in lower case.

Functionality: Edit an individual article. The last-modified timestamp should be updated.

Url: <http://localhost:5200/article>

Method: PUT

Curl command example: `curl http://127.0.0.1:5200/article -d '{"title": "Edited Title", "content": "Edited content"}' -u parag:bhingre`

Login by entering the user id and password.

Input variables :

article_id: Provide article id to edit

content: Will replace existing content of given article id

All the variables mentioned above are in lower case.

Functionality: Delete a specific existing article

Url: <http://localhost:5200/article>

Method: DELETE

Curl command example: `curl -X DELETE http://127.0.0.1:5200/article -d '{"article_id":8}' -u parag:bhingre`

Login by entering the user id and password.

Input variables :

article_id: provide the article id.

All the variables mentioned above are in lower case.

Functionality: Retrieve the entire contents (including article text) for the n most recent articles

Url: <http://localhost:5200/article>

Method: GET

Curl command example: curl -GET "http://127.0.0.1:5200/article?limit=2"

Input variables:

limit : Retrieve all data of article.

All the variables mentioned above are in lower case.

Functionality: Retrieve metadata for the n most recent articles, including title, author, date, and URL

Url: <http://localhost:5200/article>

Method: GET

Curl command example: curl -GET "http://127.0.0.1:5200/article?metadata=2"

Input variables:

metadata: Retrieve metadata of given number

All the variables mentioned above are in lower case.

Micro-service: Comments - comment.py

Functionality: Add a new comment.

Url: <http://127.0.0.1:5100/comment>

Method: POST

You can post a comment by your login credentials, if you want your username to appear in the comment section.

Curl command example:

Input variables:

comment: comment Text.

article_id: ID of the article

variables mentioned above are in lower case.

Functionality: Delete a comment. (Authentication required for this operation).

Url: http://127.0.0.1:5100/comment?comment_id=

Method: DELETE

Sign in by your login credentials

Curl command example:

Input variables:

comment_id: comment id to delete the comment.

variables mentioned above are in lower case.

Functionality: Get n number of comments of an article.

Url: http://127.0.0.1:5100/comment?article_id=

Method: GET

Authentication not required for this operation

Curl command example:

Input variables:

article_id: article ID.

variables mentioned above are in lower case.

Functionality: Get n recent comments of an article.

Url: “http://127.0.0.1:5100/comment?article_id=&number=”

Method: GET

Authentication not required for this operation

Input variables:

article_id: article ID.

number: provide the number of comments to be retrieved.

variables mentioned above are in lower case.

Functionality: UPDATE comment.

Url: “<http://127.0.0.1:5100/comment>”

Method: PUT

Authentication not required of updating comments posted by Anonymous Coward

Curl command example:

Input variables:

comment: New comment

comment_id: ID of the specific comment

Micro-service: Tags - tags.py

Functionality: Add tags for a new URL

Url: `http://127.0.0.1:5300/tags`

Method: POST

You may login by entering the user id and password.

Curl command example :`curl -POST http://127.0.0.1:5003/tags -d`

`'{"article_id":3,"tag_name":"#great1"}'`

Input variables:

artid: provide the article id.

tag: provide tag to be added.

All the variables mentioned above are in lower case.

Functionality: Delete the link of tag and article.

Url: `http://127.0.0.1:5300/tags`

Method: DELETE

You may login by entering the user id and password.

Curl command example:`curl -X DELETE http://127.0.0.1:5003/tags -d`

`'{"article_id":1,"tag_name":"#fullerton"}'`

Input variables:

article_id: Provide the article id.

tag_name: Provide tag name

All the variables mentioned above are in lower case.

Functionality: Retrieve the tags for an individual URL

Url: `http://127.0.0.1:5300/tags`

Method: GET

You may login by entering the user id and password.

Curl command example: `curl -GET http://127.0.0.1:5003/tags/7`

Input variables:

article_id: Provide the article id.

All the variables mentioned above are in lower case.

Functionality: Retrieve a list of URLs with a given tag.

Url: `http://127.0.0.1:5300/tags`

Method: GET

You may login by entering the user id and password.

Curl command example:`curl -GET http://127.0.0.1:5003/tags?tag=%23fullerton`

Input variables:

tag: provide tag name

All the variables mentioned above are in lower case.

