

Microblog Services

There are two microservices which are 1) User and 2) Timeline services.

1) Users microservice :

- **createUser(username, email, password):**

Description: This api is for new user registration. If all the parameters i.e, username, email and password are passed successfully and are not null then we get a success message with status code 200. If the user already exists or null or empty value for username is passed, then we get 400 statusCode with Bad Request error.

Note: We are generating hash value of the password using **generate_password_hash** from **werkzeug.security** and then storing the hashed value in the database.

Endpoint : /register

Method: POST

Parameters : Username , Email , Password

Sample JSON Request :

```
{
  "username": "Pavani",
  "email": "nagisetipavani@gmail.com",
  "Password": "asjd@273@"
}
```

Success Response : {

```
"statusCode": 200,
"message": "You have successfully registered!"
}
```

Error Response: {

```
"statusCode": 400,
"error": "Bad Request",
"message": "Invalid parameter(s)"
}
```

- **authenticateUser(username, password)**

Description: This api is for validating the user login. If the username and password match the values in the database we return the value "true". If the validation returns false, we return 401 Status code which is "Unauthorized error".

Endpoint : /login

Method : POST

Parameters : username , password

Sample JSON request :

```
{
  Username : "Pavani"
  Password : "asjd@273@"
}
```

Success Response : {

```
  true
}
```

Error Response: {

```
  "statusCode": 401,
  "error": "Unauthorized",
  "message": "Login failed: Invalid username or password"
}
```

- **addFollower(username, usernameToFollow)**

Description: This api takes two parameters username and usernameToFollow. If the username or usernameToFollow is None or empty we return statusCode: 400 and the error is "Bad Request" with message as "Invalid parameters(s). If all the parameters provided are valid, we return a success response with statusCode: 200 and message as "You are successfully following the { usernameToFollow}!".

Endpoint : /follow

Method: POST

Parameters : Username ,Username To follow

Sample JSON Request :

```
{
  "username" : "Pavani",
  "usernameToFollow" : "Rushabh"
}
```

Success Response : {

```
  "statusCode": 200,
  "message": "You are successfully following Rushabh!"
}
```

Error Response: {

```
  "statusCode": 400,
  "error": "Bad Request",
  "message": "Invalid parameter(s)"
}
```

- **removeFollower(username, usernameToRemove)**

Description: This api takes two parameters username and usernameToRemove. If the username or usernameToRemove is None or empty we return statusCode: 400 and the error is “Bad Request” with message as “Invalid parameters(s). If all the parameters provided are valid, we return a success response with statusCode: 200 and message as “You have successfully unfollowed { usernameToRemove}!”.

Endpoint : /unfollow

Method: POST

Parameters : Username ,Username To Remove

Sample JSON Request :

```
{
  "username" : "Pavani",
  "usernameToRemove" : "Rushabh"
}
```

Success Response : {

```
  "statusCode": 200,
  "message": "You have successfully unfollowed Rushabh!"
}
```

Error Response: {

```
  "statusCode": 400,
  "error": "Bad Request",
  "message": "Invalid parameter(s)"
}
```

2) Timelines microservice:

- **getUserTimeline(username)**

Description: This api takes username as parameter. If the username is None or empty we return statusCode: 400 and the error is “Bad Request” with message as “Invalid parameter”. If all the parameters provided are valid, we return a success response of the latest tweets. We have limited the tweets to 25 and they are also returned in the order of the latest timestamp.

Endpoint : /getUserTimeline

Method: GET

Parameters : Username

Sample JSON Request :

```
{
  "username" : "Anna"
}
```

Success Response :

```
[
  {
    "text": "Super excited!",
    "timestamp": "2020-10-06 06:03:47.989280",
    "username": "Anna"
  }
]
```

Error Response: {

```
  "statusCode": 400,
  "error": "Bad Request",
  "message": "Invalid parameter"
}
```

- **getPublicTimeline()**

Description: This api returns the recent tweets from all the users. We limited the number of tweets to 25 and also tweets are shown with the most recent tweet at the top.

Endpoint : /getPublicTimeline

Method: GET

Parameters : N/A

Success Response :

```
[
  {
    "text": "Super excited!",
    "timestamp": "2020-10-06 06:03:47.989280",
    "username": "Anna"
  },
  {
    "text": "Hey there! This is super awesome class",
    "timestamp": "2020-10-06 06:03:47.989280",
    "username": "Pavani"
  },
]
```

- **getHomeTimeline(username)**

Description: This api takes username as parameter and returns the recent tweets from all the users that this user follows. We limited the number of tweets to 25 and also tweets are shown with the most recent tweet at the top. If the username parameter is empty or None we are sending a response with statusCode as 400, error as “Bad Request” and message as “Invalid parameter”.

Endpoint : /getHomeTimeline

Method: GET

Parameters : username

Sample JSON Request :

```
{  
  "username": "Anna"  
}
```

Note: Let's assume Anna is following Rushabh and Pavani.

Success Response :

```
[  
  {  
    "text": "Super excited!",  
    "timestamp": "2020-10-06 06:03:47.989280",  
    "username": "Rushabh"  
  },  
  {  
    "text": "Hey there! This is super awesome class",  
    "timestamp": "2020-10-06 06:03:47.989280",  
    "username": "Pavani"  
  },  
]
```

Error Response: {

```
  "statusCode": 400,  
  "error": "Bad Request",  
  "message": "Invalid parameter"  
}
```

- **postTweet(username, text)**

Description: This api returns the recent tweets from all the users. We limited the number of tweets to 25 and also tweets are shown with the most recent tweet at the top.

Endpoint : /postTweet

Method: POST

Parameters : username, text

Sample JSON Request :

```
{  
  "username": "Anna"  
  "text": "Super excited!"  
}
```

Success Response :

```
{  
  "statusCode": 200  
}
```

Error Response: {

```
"statusCode": 400,  
"error": "Bad Request",  
"message": "Invalid parameter"  
}
```

If user doesn't exist in the users table we send the following error response

```
{  
  "message": "Username doesn't exist. If you are a new user please register, or  
if you are an existing user please sign in."  
}
```