API Documentation

Users API

1. Create new user

Endpoint: <http://127.0.0.1:5000/users>

Request Method: POST

Authentication: Not required

Request Parameters: We need to pass parameters in json fields:

1. username
2. email\_id
3. password

Sample Request:

{

"username": "dileep3",

"email\_id": "dileep3@gmail.com",

"password": "password3"

}

Response Parameters: We will send response in json:

1. status
2. message

Sample response

{

"message": "dileep3@gmail.com was added!",

"status": "success"

}

2. Delete existing user

Endpoint: [http://127.0.0.1:5000/users/<id](http://127.0.0.1:5000/users/%3cid)> #pass userid in url

E.g. URL:http://127.0.0.1:5000/users/3

Request Method: DELETE

Authentication: Pass username and password in authentication field

Request Parameters: No need to pass request filed in json

Response Parameters: We will send response in json:

1. status
2. message

Sample response

{

"message": "dileep2 is deleted!",

"status": "success"

}

Sample failure response

URL: <http://127.0.0.1:5000/users/10>

{

"message": "No user exist of given ID!!",

"status": "fail"

}

3. Change existing user’s password

Endpoint: [http://127.0.0.1:5000/users/<id](http://127.0.0.1:5000/users/%3cid)> #pass userid in url

E.g. URL:http://127.0.0.1:5000/users/3

Request Method: PUT

Authentication: Pass username and password in authentication field

Request Parameters: We need to pass parameters in json fields:

1. password

Sample Request:

{

"password": "password1"

}

Response Parameters: We will send response in json:

1. status
2. message

Sample Response:

{

"message": "dileep1 password is modified",

"status": "success"

}

Sample Fail Response:

{

"message": "No user exist for given ID!!",

"status": "fail"

}

Articles API

1. Post a new article

Endpoint: http://127.0.0.1:5001/articles

Request Method: POST

Authentication: Pass username and password in authentication field

Request Parameters: We need to pass parameters in json fields:

1. title
2. body
3. article\_url

Sample Request:

{

"title": "title11",

"body": "this is body 11",

"article\_url": "url11"

}

Response Parameters: We will send response in json:

1. status
2. message

Sample response

{

"message": "article title11 was added!",

"status": "success"

}

Sample fail response:

{

"message": "Error while adding article",

"status": "fail"

}

2. Retrieve an individual article

Endpoint:  [http://127.0.0.1:5001/article/<id](%20http://127.0.0.1:5001/article/%3cid)> #pass userid in url

E.g. URL: http://127.0.0.1:5001/article/11

Request Method: GET

Authentication: Not required

Request Parameters: no need to pass parameters in json

Response Parameters: We will send response in json:

1. status
2. title
3. body
4. author

Sample Response:

{

"author": "dileep3",

"body": "this is body 5",

"status": "success",

"title": "title5"

}

Sample fail response:

{

"message": "No article exist of given ID!!",

"status": "fail"

}

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

Endpoint: http://127.0.0.1:5001/article/<id> #pass userid in url

E.g. URL: http://127.0.0.1:5001/article/2

Request Method: PUT

Authentication: Pass username and password in authentication field

Request Parameters: We need to pass parameters in json fields:

1. title
2. body

Sample Request:

{

"title": "title2 updating",

"body": "this is body 2 updating"

}

Response Parameters: We will send response in json:

1. status
2. message

Sample Response:

{

"message": "title2 updating is updated!!",

"status": "success"

}

Sample Fail Response:

{

"message": "No article exist for given ID!!",

"status": "fail"

}

4. Delete a specific existing article

Endpoint: http://127.0.0.1:5001/article/<id> #pass userid in url

E.g. URL: http://127.0.0.1:5001/article/5

Request Method: DELETE

Authentication: Pass username and password in authentication field

Request Parameters: No need to pass request filed in json

Response Parameters: We will send response in json:

1. status
2. message

{

"message": "title2 updating is deleted!",

"status": "success"

}

Sample Fail Response:

{

"message": "No article exist for given ID!!",

"status": "fail"

}

5. Retrieve the entire contents (including article text) for the n most recent articles

Endpoint: http://127.0.0.1:5001/articles/<n> #pass number n in url

E.g. URL: http://127.0.0.1:5001/articles/3

Request Method: GET

Authentication: Not required

Request Parameters: no need to pass parameters in json

Response Parameters: We will send response in json:

1. status
2. data

Sample Response:

{

"data": {

"articles": [

{

"article\_url": "url11",

"author": "dileep3",

"body": "this is body 11",

"created\_timestamp": "Tue, 12 Mar 2019 11:22:46 GMT",

"title": "title11"

},

{

"article\_url": "url10",

"author": "dileep1",

"body": "this is body 10",

"created\_timestamp": "Tue, 12 Mar 2019 11:21:42 GMT",

"title": "title10"

},

{

"article\_url": "url9",

"author": "dileep1",

"body": "this is body 9",

"created\_timestamp": "Mon, 11 Mar 2019 21:06:03 GMT",

"title": "title9"

}

]

},

"status": "success"

}

6. Retrieve metadata for the n most recent articles, including title, author, date, and URL

Endpoint: http://127.0.0.1:5001/articlesMetaData/<n> #pass number n in url

E.g. URL: http://127.0.0.1:5001/articlesMetaData/3

Request Method: GET

Authentication: Not required

Request Parameters: no need to pass parameters in json

Response Parameters: We will send response in json:

1. status
2. data

Sample Response:

{

"data": {

"articles": [

{

"article\_url": "url11",

"author": "dileep3",

"body": "this is body 11",

"created\_timestamp": "Tue, 12 Mar 2019 11:22:46 GMT",

"id": 11,

"modified\_timestamp": "Tue, 12 Mar 2019 11:22:46 GMT",

"title": "title11",

"user\_id": 3

},

{

"article\_url": "url10",

"author": "dileep1",

"body": "this is body 10",

"created\_timestamp": "Tue, 12 Mar 2019 11:21:42 GMT",

"id": 10,

"modified\_timestamp": "Tue, 12 Mar 2019 11:21:42 GMT",

"title": "title10",

"user\_id": 1

},

{

"article\_url": "url9",

"author": "dileep1",

"body": "this is body 9",

"created\_timestamp": "Mon, 11 Mar 2019 21:06:03 GMT",

"id": 9,

"modified\_timestamp": "Mon, 11 Mar 2019 21:06:03 GMT",

"title": "title9",

"user\_id": 1

}

]

},

"status": "success"

}