Note Making App Documentation

1. Login

* Endpoint: /routes/login
* Request Method: POST
* POST: Logs in the existing user
* URL Paths: POST localhost:1000/api/routes/login: Logs in the existing user

Request Body Format: For Log in of existing user:

{

    "email":"one@gmail.com",

    "password":"098765"

}

Response: *Successful Login, Status(200)*

{

    "others": {

        "\_id": "659541d794da34ac791b2890",

        "email": "one@gmail.com",

        "username": "one",

        "list": [],

        "createdAt": "2024-01-03T11:15:35.763Z",

        "updatedAt": "2024-01-03T12:38:27.264Z",

        "\_\_v": 9

    }

}

Response: *Failed Login, Status(400):*

{

    "message": "Invalid Credentials"

}

Response:  *Email not registered, Status(400):*

{"message": "Sign up first"}

1. Sign in

* Endpoint: /routes/register
* Request Method: POST
* POST: Creates a new user
* URL Paths: POST localhost:1000/api/routes/register: Creates a new user

Request Body Format: For new user:

{

    "email":"one123@gmail.com",

    "username": "one123",

    "password": "98765"

}

Response: New user created, Status(200):

{

    "user": {

        "email": "one1223@gmail.com",

        "username": "one12322",

        "password": "$2b$10$mr6wNji2EJ5r1B0tWGTwEeflMovqrKH8jJBHHlkoBe8Hg2hZ./BOW",

        "list": [],

        "\_id": "659574aed6df5b80b27d7505",

        "createdAt": "2024-01-03T14:52:30.489Z",

        "updatedAt": "2024-01-03T14:52:30.489Z",

        "\_\_v": 0

    }

}

Response: User already exist, Status(400):

{

"message": "User already exist",

}

Response: Email format, Status(400):

{

“message": "Invalid email",

}

1. Create Note

* Endpoint: routes1/create
* Request Method: POST
* POST: Creates note for user
* URL Paths: POST localhost:1000/api/routes1/create: Creates new note for user

Request Body Format: {

    "title":"One' task",

    "body":"I am writing about one's task",

    "email":"one@gmail.com"

}

Response: List successfully created, status(200)

{

    "list": {

        "title": "One' task",

        "body": "I am writing about one's task",

        "user": [

            "659541d794da34ac791b2890"

        ],

        "\_id": "659580e8aafb7c5919535d98",

        "createdAt": "2024-01-03T15:44:40.194Z",

        "updatedAt": "2024-01-03T15:44:40.194Z",

        "\_\_v": 0

    }

}

Response: User not logged in, status(400):

{

"message": "Log in First"

}

1. Update Note

* Endpoint: routes1/update/:id
* Request Method: PUT
* PUT: Updates note for user
* URL Paths: PUT localhost:1000/api/routes1/update/id: Updates note for user

Request Body Format:

{

    "title":"two' task",

    "body":"I am writing about tow's task2",

    "email":"one@gmail.com"

}

Response: Updated successfully, status(200):

{

    "message":"Task updated"

}

Response: User not logged in, status(400):

{

"message": "Log in First"

}

1. Delete Note

* Endpoint: routes1/delete/:id
* Request Method: DELETE
* DELETE: Delete note of user
* URL Paths: DELETE localhost:1000/api/routes1/delete/id: Delete note for user

Request Body format:

{

    "email": "one@gmail.com"

}

Response: Deletion successful, Status(200):

{

    "Message": "Task Deleted"

}

1. Read Note

* Endpoint: routes1/read/:id
* Request Method: GET
* GET: Gets user note from backend
* URL Paths: GET localhost:1000/api/routes1/read/id: Gets note for user from backend

Response: Data fetch successful, status(200)

{

    "list": [

        {

            "\_id": "659580e8aafb7c5919535d98",

            "title": "One' task",

            "body": "I am writing about one's task",

            "user": [

                "659541d794da34ac791b2890"

            ],

            "createdAt": "2024-01-03T15:44:40.194Z",

            "updatedAt": "2024-01-03T15:44:40.194Z",

            "\_\_v": 0

        },

        {

            "\_id": "6595554225824e974f2718b0",

            "title": "One' task",

            "body": "I am writing about one's task",

            "user": [

                "659541d794da34ac791b2890"

            ],

            "createdAt": "2024-01-03T12:38:26.944Z",

            "updatedAt": "2024-01-03T12:38:26.944Z",

            "\_\_v": 0

        },

        {

            "\_id": "6595550425824e974f2718a3",

            "title": "One' task",

            "body": "I am writing about one's task",

            "user": [

                "659541d794da34ac791b2890"

            ],

            "createdAt": "2024-01-03T12:37:24.126Z",

            "updatedAt": "2024-01-03T12:37:24.126Z",

            "\_\_v": 0

        },

Response: Data fetch unsuccessful, status(400):

{

    "message": "No Tasks"

}