# CRUD – Notes

## Introduction

This API is meant to be used to store and manage notes, a note is formed by the following fields:

|  |  |  |
| --- | --- | --- |
| **Field** | **Value** | **Importance** |
| Title | String 1-40 characters | Obligatory |
| Subtitle | String >= 1 characters | Optional |
| Content | String >= 1 characters | Optional |
| Tags | Array of Strings >= 1 element | Optional |
| Id\* | Mongo Id | Obligatory |

\* Must only be provided when asked for

Note that the Fields need to be sent in lowercase in a json property, and if you are not going to send an optional field, do not declare it.

There is something else that has to be provided on every route and that is a token, an access token, the only way to get one is by following the instructions here: [JWT Auth API Docs](https://github.com/AxTulti/JWT-Auth-API#readme).

## **Routes**

### /create – POST

The request must contain a json with two properties:

|  |  |
| --- | --- |
| Property | Value |
| Token | A JWT action token gotten from the Auth API |
| Note | A basic note object |

The route is going to save the provided note in the database, the owner of the note being the owner of the token, a successful request and response should look like this:

Req.body

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiaWQiLCJlbWFpbCI6InN1cGVybWFuQGdtYWlsLmNvbSIsIm5hbWUiOiJDbGFyayIsImxhc3ROYW1lIjoiS2VudCIsImlhdCI6MTYyOTIzNzMxMn0.5gXAFKcQ8epzfBpq92fV7DtVpHlEoIjYtBrFkjXBTFY",

    "note": {

        "title": "My first note",

        "subtitle": "This is an awesome note",

        "content": "This is definitely an awesome note",

        "tags": ["first note", "my awesome note", "incredible note"]

    }

}

Res.body

(Status 201)

{

    "tags": [

        "first note",

        "my awesome note",

        "incredible note"

    ],

    "\_id": "611c2fc419f4436ea0aabfc2",

    "owner": "superman@gmail.com",

    "title": "My first note",

    "subtitle": "This is an awesome note",

    "content": "This is definitely an awesome note",

    "createdAt": "2021-08-17T21:53:08.247Z",

    "updatedAt": "2021-08-17T21:53:08.247Z",

    "\_\_v": 0

}

### /read – POST

The request must contain a json with one property:

|  |  |
| --- | --- |
| Property | Value |
| Token | A JWT action token gotten from the Auth API |

The route is going to respond with all the notes created by the user, a successful request and response should look like this:

Res.body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiaWQiLCJlbWFpbCI6InN1cGVybWFuQGdtYWlsLmNvbSIsIm5hbWUiOiJDbGFyayIsImxhc3ROYW1lIjoiS2VudCIsImlhdCI6MTYyOTIzNzMxMn0.5gXAFKcQ8epzfBpq92fV7DtVpHlEoIjYtBrFkjXBTFY"

}

(Status 200)

Res.body:

[

    {

        "tags": [

            "first note",

            "my awesome note",

            "incredible note"

        ],

        "\_id": "611c2fc419f4436ea0aabfc2",

        "owner": "superman@gmail.com",

        "title": "My first note",

        "subtitle": "This is an awesome note",

        "content": "This is definitely an awesome note",

        "createdAt": "2021-08-17T21:53:08.247Z",

        "updatedAt": "2021-08-17T21:53:08.247Z",

        "\_\_v": 0

    },

    {

        "tags": [

            "second note",

            "my awesome note",

            "incredible note"

        ],

        "\_id": "611c31fc21ec1d278cbe6485",

        "owner": "superman@gmail.com",

        "title": "My second note",

        "subtitle": "This is an awesome note",

        "content": "This is definitely an awesome note",

        "createdAt": "2021-08-17T22:02:36.339Z",

        "updatedAt": "2021-08-17T22:02:36.339Z",

        "\_\_v": 0

    },

    {

        "tags": [

            "third note",

            "my awesome note",

            "incredible note"

        ],

        "\_id": "611c330021ec1d278cbe6487",

        "owner": "superman@gmail.com",

        "title": "My third note",

        "subtitle": "This is an awesome note",

        "content": "This is definitely an awesome note",

        "createdAt": "2021-08-17T22:06:56.001Z",

        "updatedAt": "2021-08-17T22:06:56.001Z",

        "\_\_v": 0

    }

]

### /numberOfNotes – POST

The request must contain a json with one property:

|  |  |
| --- | --- |
| Property | Value |
| Token | A JWT action token gotten from the Auth API |

The route is going to respond with the number of notes that have been created by the JWT’s contained email user, a successful request and response should look like this:

Req.body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiaWQiLCJlbWFpbCI6InN1cGVybWFuQGdtYWlsLmNvbSIsIm5hbWUiOiJDbGFyayIsImxhc3ROYW1lIjoiS2VudCIsImlhdCI6MTYyOTIzNzMxMn0.5gXAFKcQ8epzfBpq92fV7DtVpHlEoIjYtBrFkjXBTFY"

}

(Status 200)

Res.body:

{

    "numberOfNotes": 3

}

### / readOne – POST

The request must contain a json with two properties:

|  |  |
| --- | --- |
| Property | Value |
| Token | A JWT action token gotten from the Auth API |
| Id | The Mongo id of the note we want to get. |

The route is going to respond with the specific note which the id belongs to, a successful request and response should look like this:

Req.body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiaWQiLCJlbWFpbCI6InN1cGVybWFuQGdtYWlsLmNvbSIsIm5hbWUiOiJDbGFyayIsImxhc3ROYW1lIjoiS2VudCIsImlhdCI6MTYyOTIzNzMxMn0.5gXAFKcQ8epzfBpq92fV7DtVpHlEoIjYtBrFkjXBTFY",

    "note": {

        "id": "611c2fc419f4436ea0aabfc2"

    }

}

Res.body:

(Status 200)

{

        "tags": [

            "first note",

            "my awesome note",

            "incredible note"

        ],

        "\_id": "611c2fc419f4436ea0aabfc2",

        "owner": "superman@gmail.com",

        "title": "My first note",

        "subtitle": "This is an awesome note",

        "content": "This is definitely an awesome note",

        "createdAt": "2021-08-17T21:53:08.247Z",

        "updatedAt": "2021-08-17T21:53:08.247Z",

        "\_\_v": 0

  }

### /update – PUT

The request must contain a json with three properties:

|  |  |
| --- | --- |
| Property | Value |
| Token | A JWT action token gotten from the Auth API |
| Id | The Mongo id of the note we want to update. |
| Note | A note object |

The route is going to replace the old note which the id belongs to, with the new note object sent in the request (keeping the id), a successful request and response should look like this:

Req.body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiaWQiLCJlbWFpbCI6InN1cGVybWFuQGdtYWlsLmNvbSIsIm5hbWUiOiJDbGFyayIsImxhc3ROYW1lIjoiS2VudCIsImlhdCI6MTYyOTIzNzMxMn0.5gXAFKcQ8epzfBpq92fV7DtVpHlEoIjYtBrFkjXBTFY",

    "note": {

        "title": "A different title",

        "subtitle": "A totally different subtitle",

        "content": "yeah, it is definitely a different note",

        "id": "611c2fc419f4436ea0aabfc2",

        "tags": ["other tag", "still a great note"]

    }

}

Res.body:

(Status 200)

{

    "tags": [

        "other tag",

        "still a great note"

    ],

    "\_id": "611c2fc419f4436ea0aabfc2",

    "owner": "superman@gmail.com",

    "title": "A different title",

    "subtitle": "A totally diferent subtitle",

    "content": "yeah, it is definitely a diferent note",

    "createdAt": "2021-08-17T21:53:08.247Z",

    "updatedAt": "2021-08-18T01:33:19.712Z",

    "\_\_v": 1

}

### /delete – DELETE

The request must contain a json with two properties:

|  |  |
| --- | --- |
| Property | Value |
| Token | A JWT action token gotten from the Auth API |
| Id | The Mongo id of the note we want to delete. |

The route is going to delete the note of which the id belongs to, a successful request and response should look like this:

Req.body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiaWQiLCJlbWFpbCI6InN1cGVybWFuQGdtYWlsLmNvbSIsIm5hbWUiOiJDbGFyayIsImxhc3ROYW1lIjoiS2VudCIsImlhdCI6MTYyOTIzNzMxMn0.5gXAFKcQ8epzfBpq92fV7DtVpHlEoIjYtBrFkjXBTFY",

    "note": {

        "id": "611c2fc419f4436ea0aabfc2"

    }

}

Res.body:

(Status 200)

The note has been deleted successfully

### /deleteAll – DELETE

The request must contain a json with one property:

|  |  |
| --- | --- |
| Property | Value |
| Token | A JWT action token gotten from the Auth API |

The route is going to delete all the notes that belong to that user, a successful request and response should look like this:

Req.body:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjoiaWQiLCJlbWFpbCI6InN1cGVybWFuQGdtYWlsLmNvbSIsIm5hbWUiOiJDbGFyayIsImxhc3ROYW1lIjoiS2VudCIsImlhdCI6MTYyOTIzNzMxMn0.5gXAFKcQ8epzfBpq92fV7DtVpHlEoIjYtBrFkjXBTFY"

}

Res.body:

(Status 200)

All the notes have been deleted successfully