

GO CRUD API : endpoints document

Post endpoints :

Path	Http method	Description
/post	GET	Get all the posts.
/post/:id	GET	Get one post, by post id.
/post	POST	Create a post, with sending request body datas.
/post/:id	PUT	Update a post of a given post id, with sending request body datas.
/post/:id	DELETE	Delete a post of a given post id.

User endpoints :

Path	Http method	Description
/user/sign-up	POST	Sign up a user, with sending request body datas.
/user/log-in	POST	Log in a user, with sending request body datas.
/user/me	GET	Get a user datas, of sending Auth bearer JWT token.

Examples :

GET : /post

Get all posts inside the database.

Example : localhost:3000/post

Auth bearer token : None

Request : None

Response : 200

```
{
  "posts": [
    {
      "id": "1d6dc138-464e-498b-948c-44c370958adb",
      "title": "This is a Title",
      "body": "This is a Body of a post",
      "created_at": "2022-11-29T15:38:14.321175+13:00",
      "updated_at": "2022-11-29T15:38:14.321175+13:00",
      "user": {
        "email": "test@test.com",
        "name": "tester"
      }
    },
    ...
  ]
}
```

Or

Error Response : 500

```
{
  "error": "Some error message"
}
```

GET : /post/:id

Get one post, a post of the path id.

Example : localhost:3000/post/eaf7e459-10da-4884-856f-520cb608c9e1

Auth bearer token : None

Request : None

Response : 200

```
{
  "post": {
    "id": "1d6dc138-464e-498b-948c-44c370958adb",
    "title": "This is a Title",
    "body": "This is a Body of a post",
    "created_at": "2022-11-29T15:38:14.321175+13:00",
    "updated_at": "2022-11-29T15:38:14.321175+13:00",
    "user": {
      "email": "test@test.com",
      "name": "tester"
    }
  }
}
```

Or

Error Response : 404 or 500

```
{
  "error": "Some error message"
}
```

POST : /post

Create a post, from a sending request body datas of post title and body.
The server app would not allow a new post to have same title, with posts stored in the database.

Example : localhost:3000/post

Auth bearer token : Required

Request :

```
{
  "title" : "This is a Title",
  "body" : "This is a Body of a post"
}
```

Response : 201

```
{
  "post" : {
    "id" : "1d6dc138-464e-498b-948c-44c370958adb",
    "title" : "This is a Title",
    "body" : "This is a Body of a post",
    "created_at" : "2022-11-29T15:38:14.321175+13:00",
    "updated_at" : "2022-11-29T15:38:14.321175+13:00",
    "user" : {
      "email" : "test@test.com",
      "name" : "tester"
    }
  }
}
```

Or

Error Response : 400, 401, or 500

```
{
  "error" : "Some error message"
}
```

PUT : /post/:id

Update a post of a path id, with a sending request body datas of post title and body.
Only a post' author is allowed to update the post.

The server app would not allow an update post to have same title, with posts stored in the database.

Example : localhost:3000/post/eaf7e459-10da-4884-856f-520cb608c9e1

Auth bearer token : Required

Request :

```
{
  "title" : "This is an UPDATED Title",
  "body" : "This is an UPDATED Body of a post"
}
```

Response : 200

```
{
  "post" : {
    "id" : "1d6dc138-464e-498b-948c-44c370958adb",
    "title" : "This is an UPDATED Title",
    "body" : "This is an UPDATED Body of a post",
    "created_at" : "2022-11-29T15:38:14.321175+13:00",
    "updated_at" : "2022-11-29T15:38:14.321175+13:00",
    "user" : {
      "email" : "test@test.com",
      "name" : "tester"
    }
  }
}
```

Or

Error Response : 400, 401, 404, or 500

```
{
  "error" : "Some error message"
}
```

DELETE : /post/:id

Delete a post of a path id. Only a post' author is allowed to delete the post.

Example : localhost:3000/post/eaf7e459-10da-4884-856f-520cb608c9e1

Auth bearer token : Required

Request : None

Response : 200

```
{  
  "message" : "Deleted success"  
}
```

Or

Error Response : 401, 404, or 500

```
{  
  "error" : "Some error message"  
}
```

POST : /user/sign-up

Sign up a user account, from a sending request body datas of user : email, name, and password.

The server app would not allow a duplicate email and name, inside the database.

Example : localhost:3000/user/sign-up

Auth bearer token : None

Request :

```
{
  "email" : "test@test.com",
  "name" : "tester",
  "password" : "password"
}
```

Response : 201

```
{
  "user" : {
    "email" : "test@test.com",
    "name" : "tester",
    "jwt" :
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2NzIyOTk4MDUsInVzZXJfaWQiOiJjNzZmZjE3My1jMzgwLTQ3M2ltYmRINS02ZDVhZTE0MzYyZGUifQ.7UZc2c1WzyV3RWveLZQ6azCzBbZMr2aE0J7RWwz8X5U"
  }
}
```

Or

Error Response : 400 or 500

```
{
  "error" : "Some error message"
}
```

POST : /user/log-in

Log in a user account, with a sending request body datas of user : email and password.
To get user datas of email, name and jwt token.

Example : localhost:3000/user/log-in

Auth bearer token : None

Request :

```
{
  "email" : "test@test.com",
  "password" : "password"
}
```

Response : 200

```
{
  "user" : {
    "email" : "test@test.com",
    "name" : "tester",
    "jwt" :
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2NzIyOTk4MDUsInVzZXJfaWQiOiJjNzZmZjE3My1jMzgwLTQ3M2ltYmRINS02ZDVhZTE0MzYyZGUifQ.7UZc2c1WzyV3RWveLZQ6azCzBbZMr2aE0J7RWwz8X5U"
  }
}
```

Or

Error Response : 400, 401 or 500

```
{
  "error" : "Some error message"
}
```


GET : /user/me

Get a user data of a sending bearer token.
To get user datas of email and name.

Example : localhost:3000/user/me

Auth bearer token : Required

Request : None

Response : 200

```
{
  "user" : {
    "email" : "test@test.com",
    "name" : "tester"
  }
}
```

Or

Error Response : 400 or 500

```
{
  "error" : "Some error message"
}
```