

Recipe Api

Github Repo Link.....	1
Description.....	1
Documenting Endpoints	1
Endpoints	1
Request Body and Response Formats.....	2
Errors.....	3

Recipe API

Github Repo Link

https://github.com/vaangis93/project_recipe_api

Description

I want to create an API to manage and share the recipes I've developed and collected over the years. The primary goal is to have an admin interface where I can read, create, and modify recipes. Additionally, I plan to implement roles based access control, allowing more restricted access for other users. For example, the public can view the recipes but won't be able to edit or add new ones. Over time, I aim to expand the API's functionality by incorporating more advanced features, such as enhanced visual elements and detailed descriptions. However, for now, the focus is on building a simple, functional API that will allow me to easily store my recipes and share them with coworkers and others.

Documenting Endpoints

Endpoints

Method	URL	Request Body (JSON)	Response (JSON)	Error (e)
GET	/api/recipes		[recipe, recipe, ...](1)	(e1)

GET	/api/recipes/{id}		Recipe (1)	(e1)
POST	/api/admin/recipes	recipe(1) without id		(e2) if wrong user-role
UPDATE	/api/admin/recipes	recipe(1) without id	recipe (1)	(e2) if wrong user-role
UPDATE	/api/admin/recipes/{id}	recipe(1) without id	Recipe (1)	(e2) if wrong user-role
DELETE	/api/admin/recipes/{id}	recipe(1) without id		(e2) if wrong user-role
GET	/api/users		[user, user, ...](2)	(e1)
GET	/api/users/{id}		user (2)	(e1)
POST	/api/users	user(2) without id		(e3)
UPDATE	/api/users/{id}	user(2) without id	user (2)	
DELETE	/api/admin/users/{id}	user(2) without id		(e2) if wrong user-role

Request Body and Response Formats

(1) Recipe format (Json) (No ID provided for POST):

```
{
  "id": Number,
  "title": String,
  "IngredientsInGrams": String,
  "description": String,
  "diffuculty": String
}
```

(2) User format (Json) (No ID provided for POST):

```
{
  "id": Number,
  "name": String,
  "password": String,
  "roles": ["ADMIN", "REGULAR"]
}
```

Errors

(e) All errors are reported using this format (with the HTTP-status code matching the number)

```
{ status : statusCode, "msg": "Explains the problem" }
```

- (e1) : { status : 404, "msg": "No content found for this request" }
- (e2): {status : 401 "msg" : Unauthorized: Authentication is required, and it has either failed or has not been provided.}
- (e3) : { status : 400, "msg": "Field 'xxx' is required" } (for example, no name provided)
-