RECIPE REST API USER GUIDE

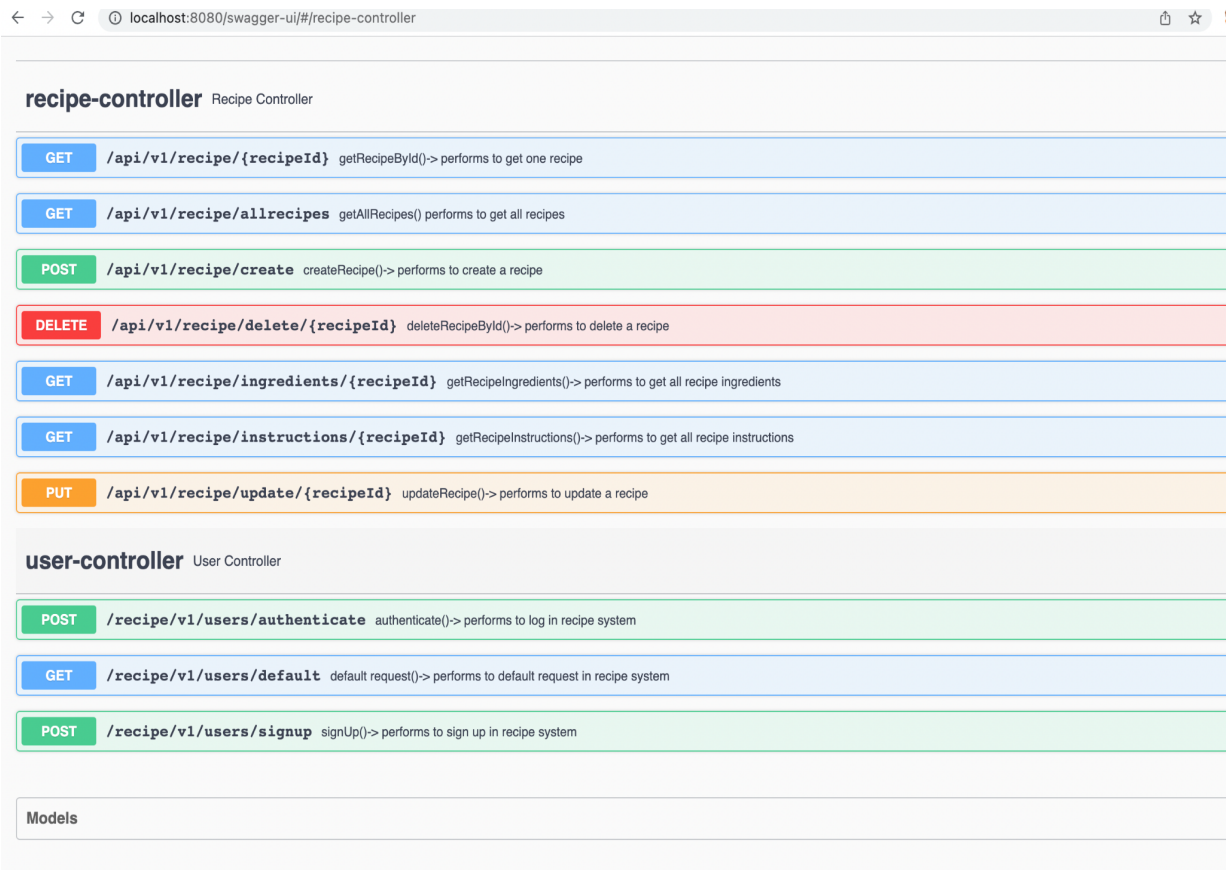
Recipe Rest API was created by using spring boot rest API frameworks.

This Rest API enables performing recipe operations like create, update, delete and list.

If you want to see these operations on swagger UI, you can visit the below swagger UI link.

swagger rest uri :
http://localhost:8080/swagger-ui/

If you entered once, you would probably see a web page like shown below.



This web page is the API documentation for the rest controller.

We can start to express the recipe rest operations.

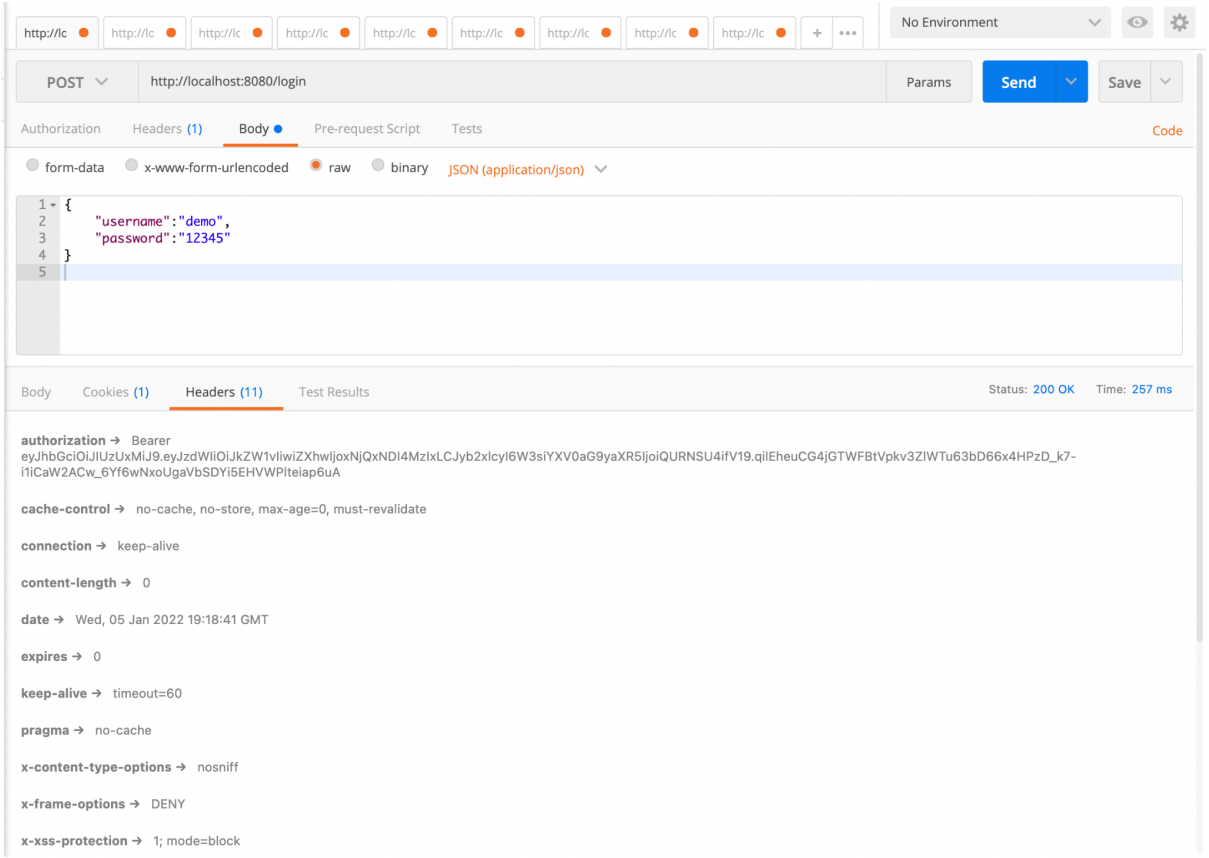
Recipe Controller :

Parent endpoint for this controller :
http://localhost:8080/api/v1/recipe

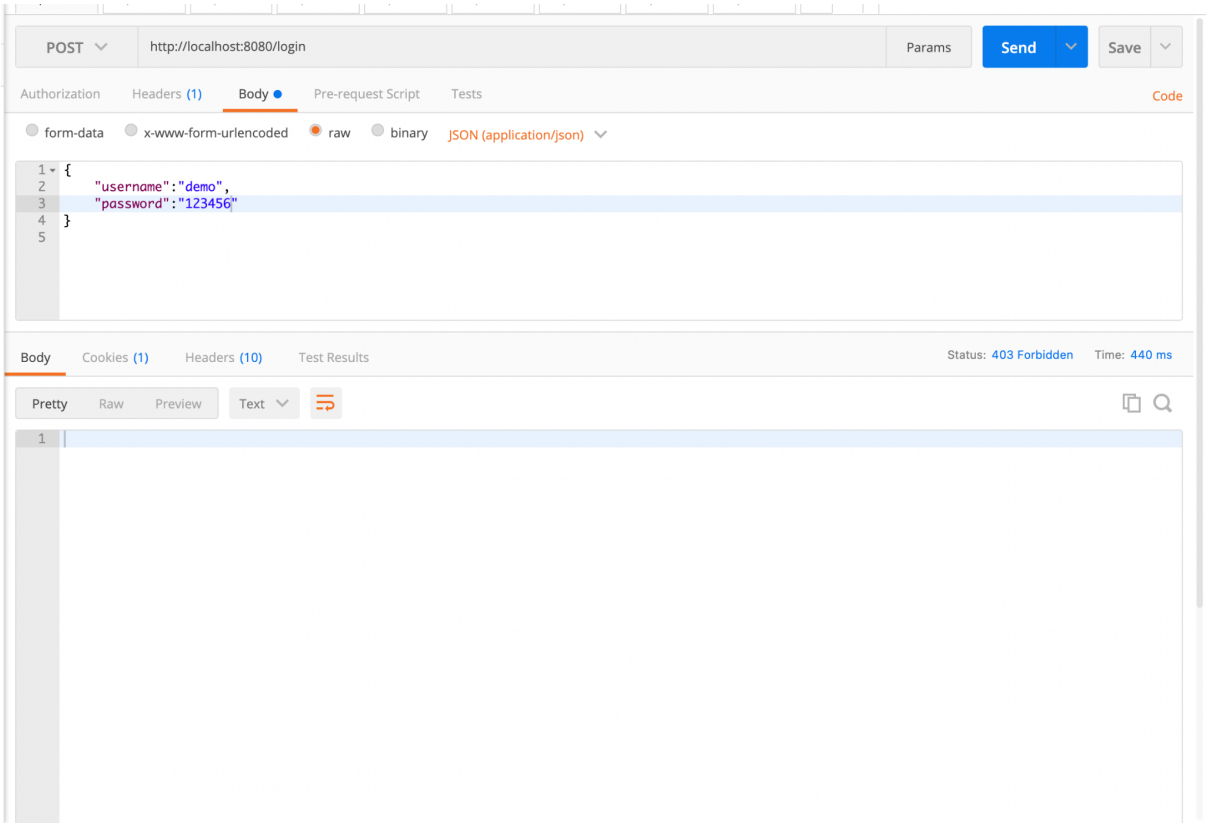
All API endpoints are secure. That's why you have to get a token. You can get the token by calling the User Controller endpoint.

We can use the login endpoint to get a bearer token.
This example was showed like below by using postman.

login post: http://localhost:8080/login



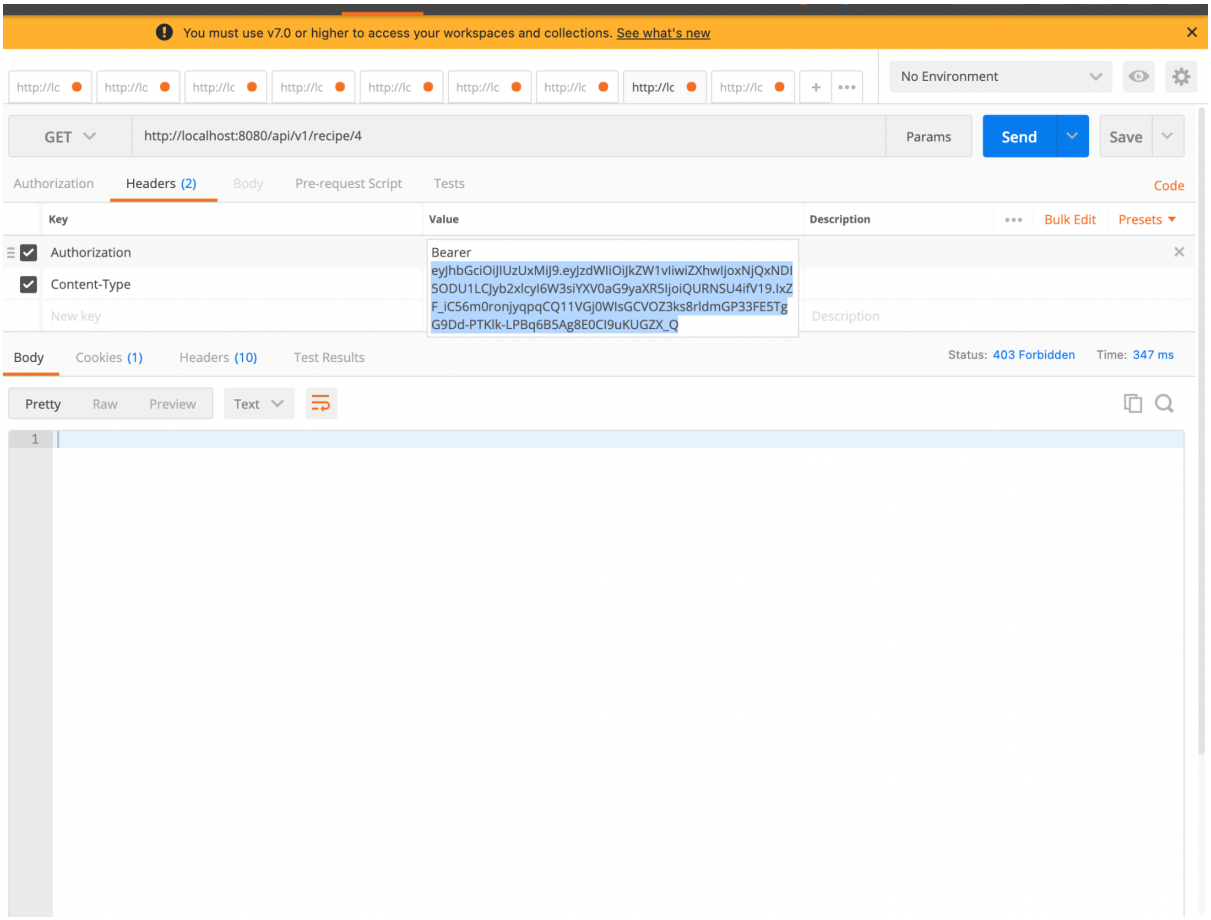
if we entered an incorrect username and password, the system would show us a 403 forbidden error like below.



Now we can use this token on every request by adding it request HTTP header in the Authorization section.

Firstly, we should add a bearer token to every request before we call the rest endpoints.

We are able to add like below postman request.



ENDPOINTS:

1. */api/v1/recipe/{recipeId}*

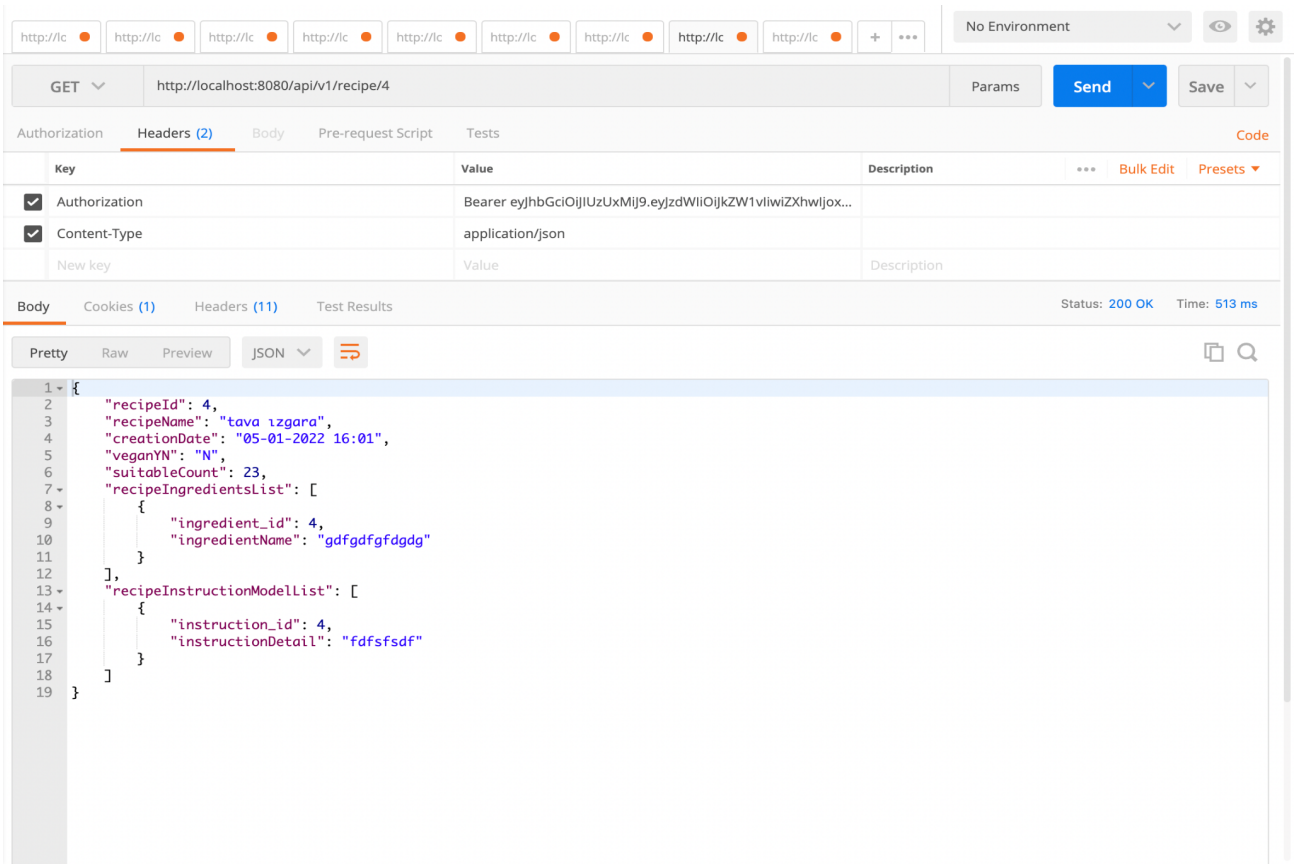
   Future: performs to get a recipe that given recipe id.

   Endpoint : http://localhost:8080/api/v1/recipe/{recipeId}
   Type : GET
   Parameter :

   ```
   {
     name : recipeId
     type  : integer
   }
   ```

   Example get request: http://localhost:8080/api/v1/recipe/4

   If you fetch this request, the result will be shown below.

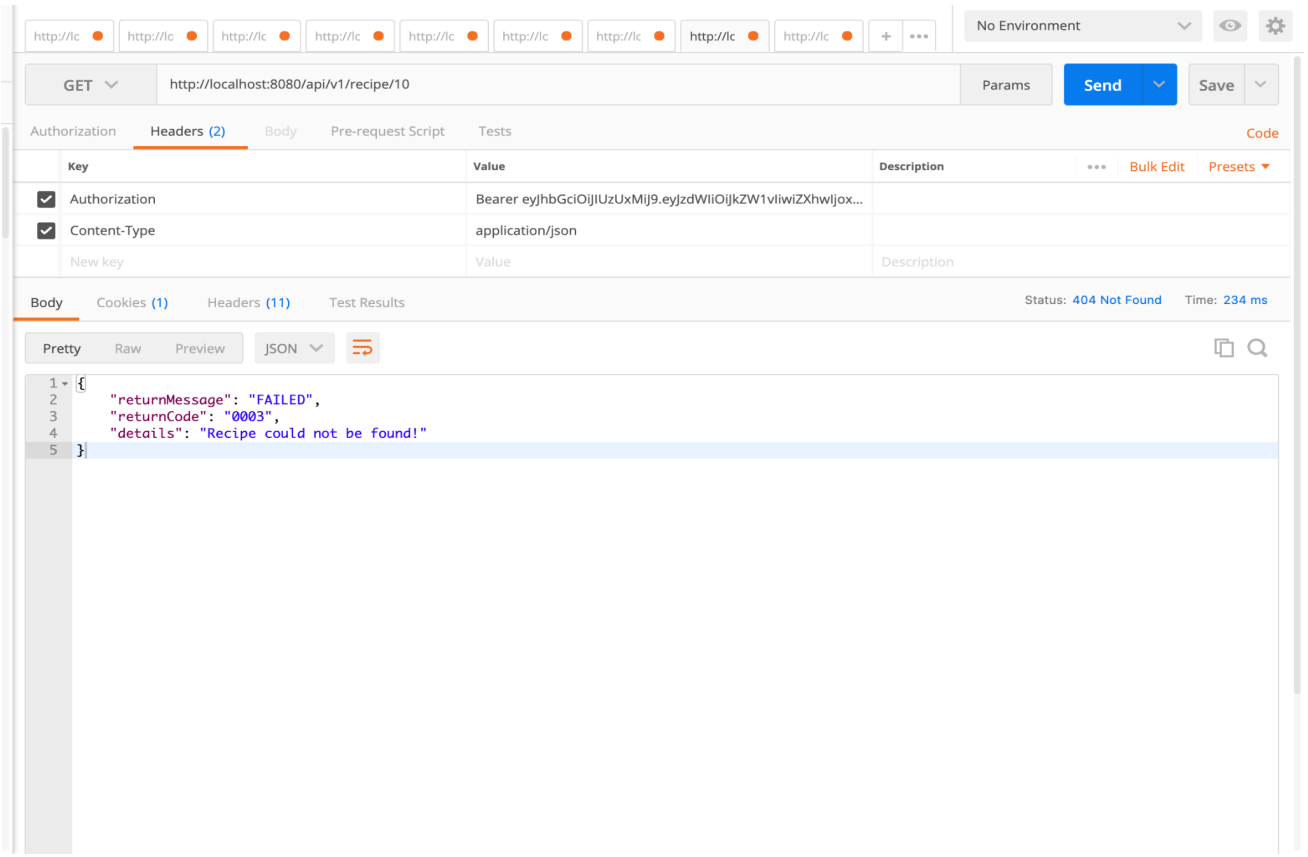


Results recipe :


```
{
    "recipeId": 4,
    "recipeName": "tava ızgara",
    "creationDate": "05-01-2022 16:01",
    "veganYN": "N",
    "suitableCount": 23,
    "recipeIngredientsList": [
        {
            "ingredient_id": 4,
            "ingredientName": "gdfgdfgfdgdg"
        }
    ],
    "recipeInstructionModelList": [
        {
            "instruction_id": 4,
            "instructionDetail": "fdfsfsdf"
        }
    ]
}
```


If we give a recipe parameter that is not registered on the system before, the system result will throw a recipe that could not be found exception like below.

Example get request: http://localhost:8080/api/v1/recipe/10



2. */api/v1/recipe/create*

Future: creates a recipe for our application by using this endpoint.

Endpoint : http://localhost:8080/api/v1/recipe/create
Type : POST
Parameter :

```
{
  "creationDate": "string",
  "recipeId": 0,
  "recipeIngredientsList": [
    {
      "ingredientName": "string",
      "ingredient_id": 0
    }
  ],
  "recipeInstructionModelList": [
    {
      "instructionDetail": "string",
      "instruction_id": 0
    }
  ],
```

```
  "recipeName": "string",
  "suitableCount": 0,
  "veganYN": "string"
}
```


Example post request: http://localhost:8080/api/v1/recipe/create

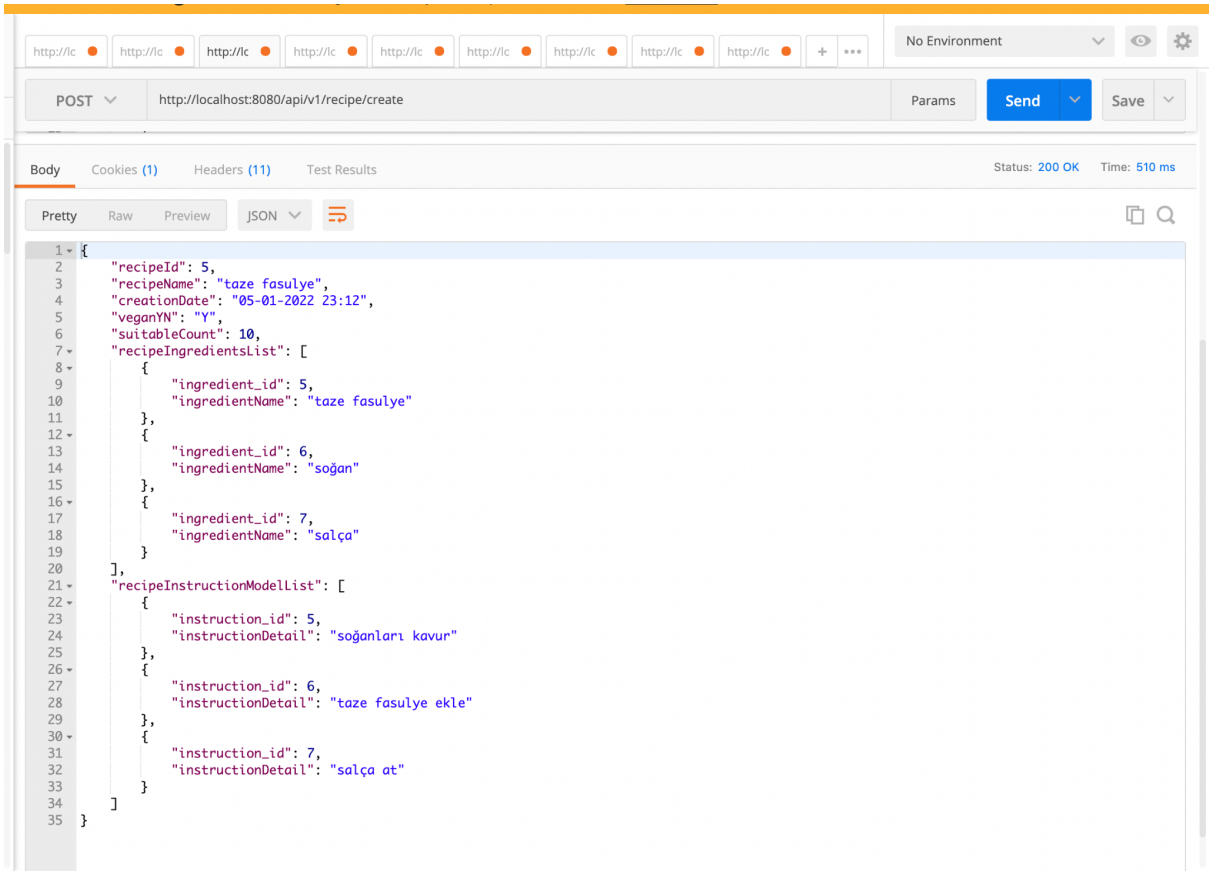
request body :


```
{
	"recipeName":"fresh bean",
	"veganYN":"Y",
	"recipeIngredientsList":
	[
		{
			"ingredientName":"fresh bean"
		},
		{
			"ingredientName":"pepper"
		},
		{
			"ingredientName":"onion"
		}
	],
	"recipeInstructionModelList":[
		{
			"instructionDetail":"hot up onion"
		},
		{
			"instructionDetail":"add bean"
		},
		{
			"instructionDetail":"put on pepper"
		}

	]

}
```


As result like below:

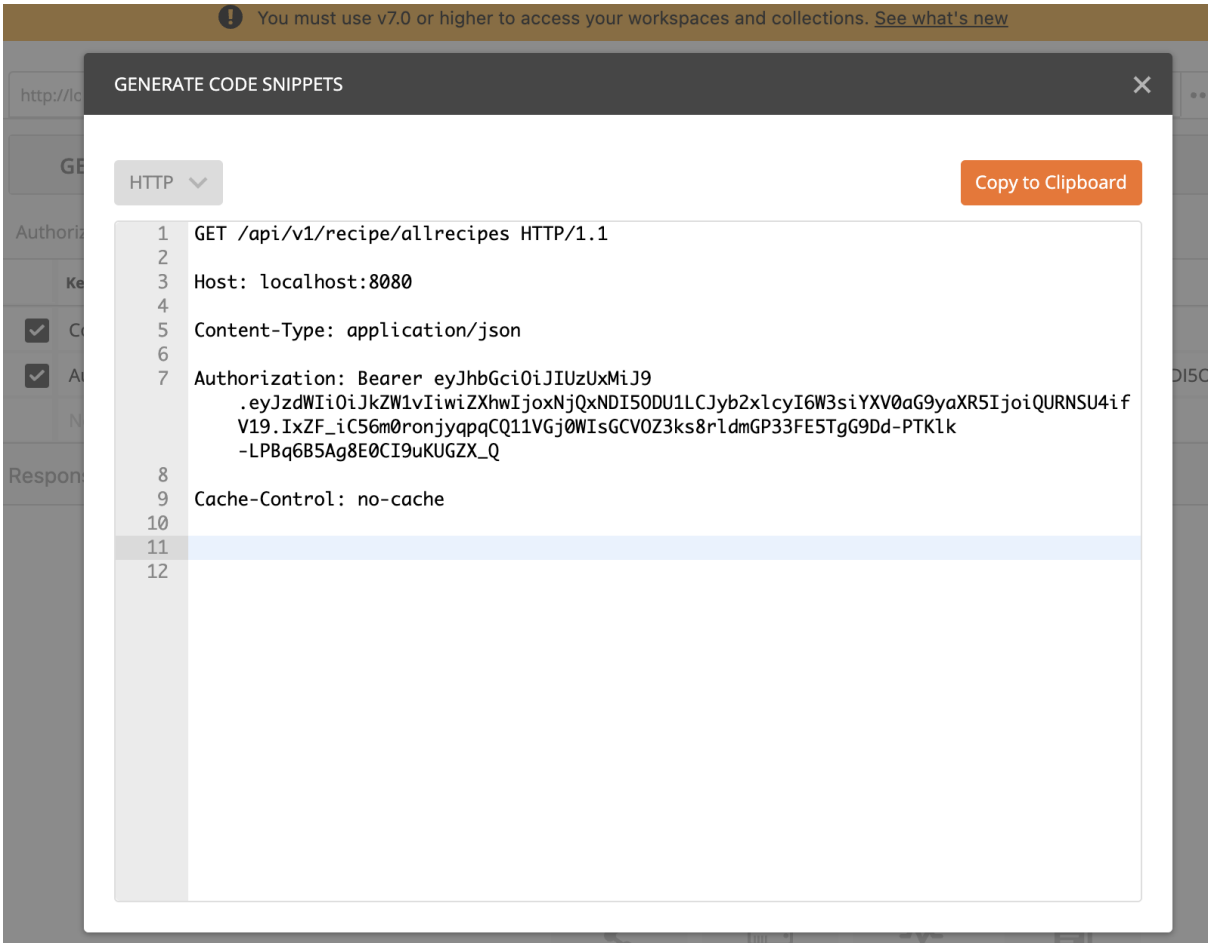


3. */api/v1/recipe/allrecipes*

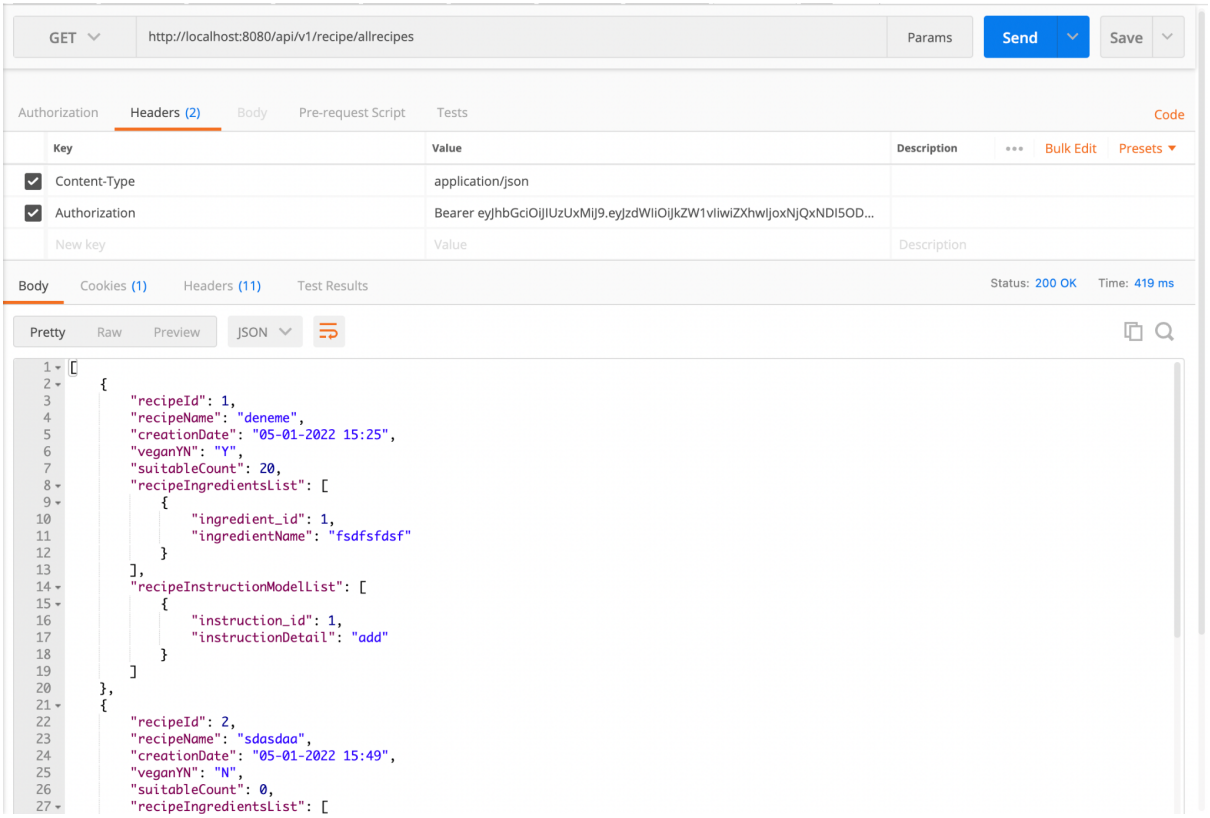
Future: performs to get all recipes

Endpoint : http://localhost:8080/api/v1/recipe/*allrecipes*
Type : GET
Parameter : No parameter

Example get request: http://localhost:8080/api/v1/recipe/*allrecipes*



Push request and we can see all of the recipes on the system.



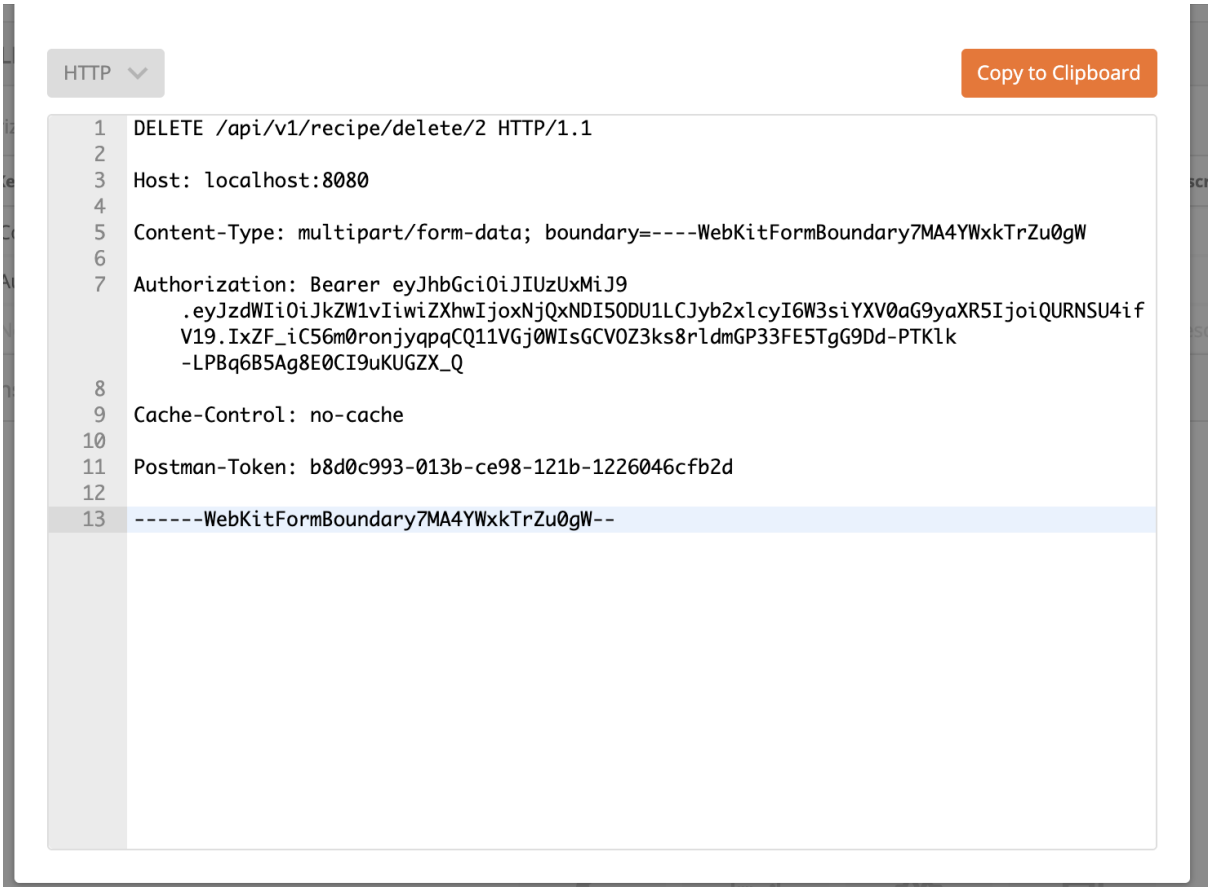
4. */api/v1/recipe/delete/{recipeId}*

Future: performs to remove that given recipe id.

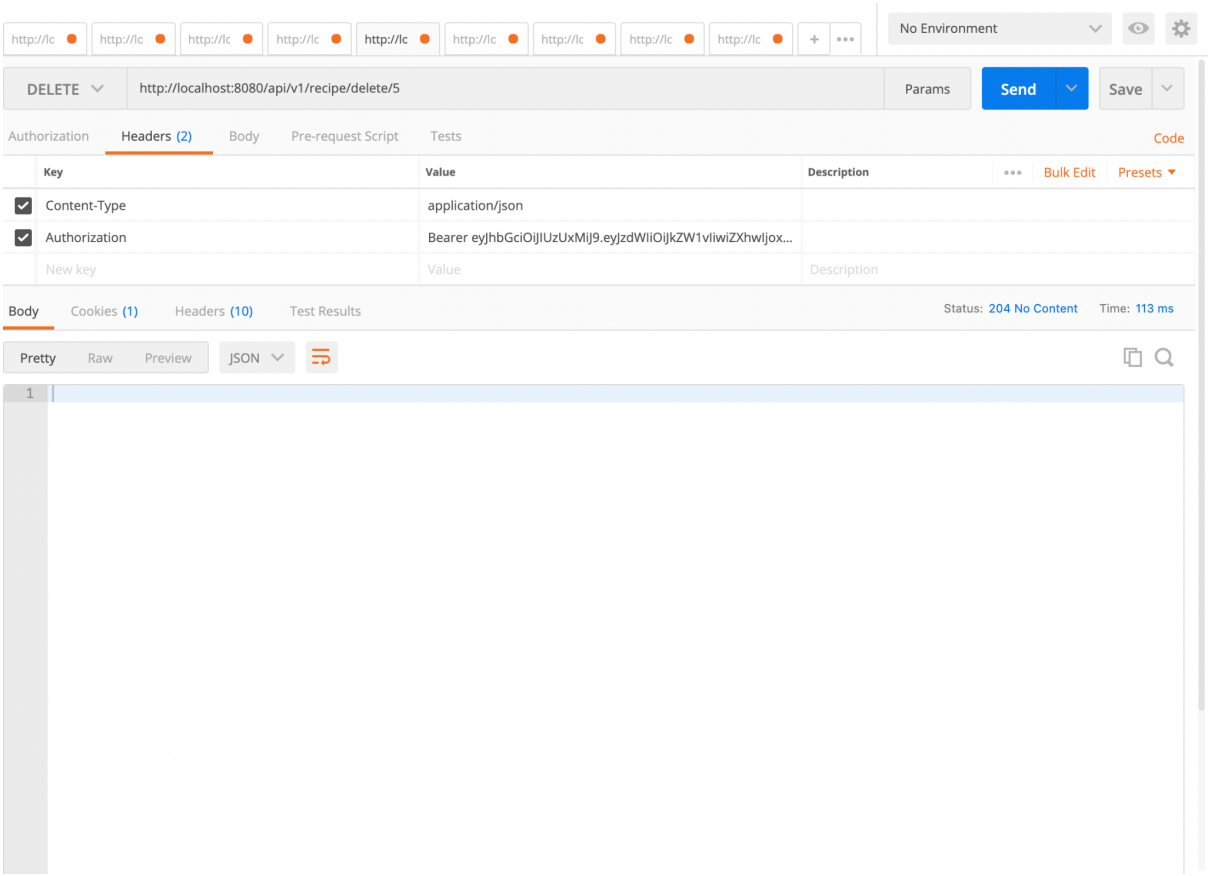
Endpoint : *http://localhost:8080/api/v1/recipe/delete/{recipeId}*
Type : DELETE
Parameter :

{
name : recipeId
type : integer
}

Example delete request:*http://localhost:8080/api/v1/recipe/delete/4*

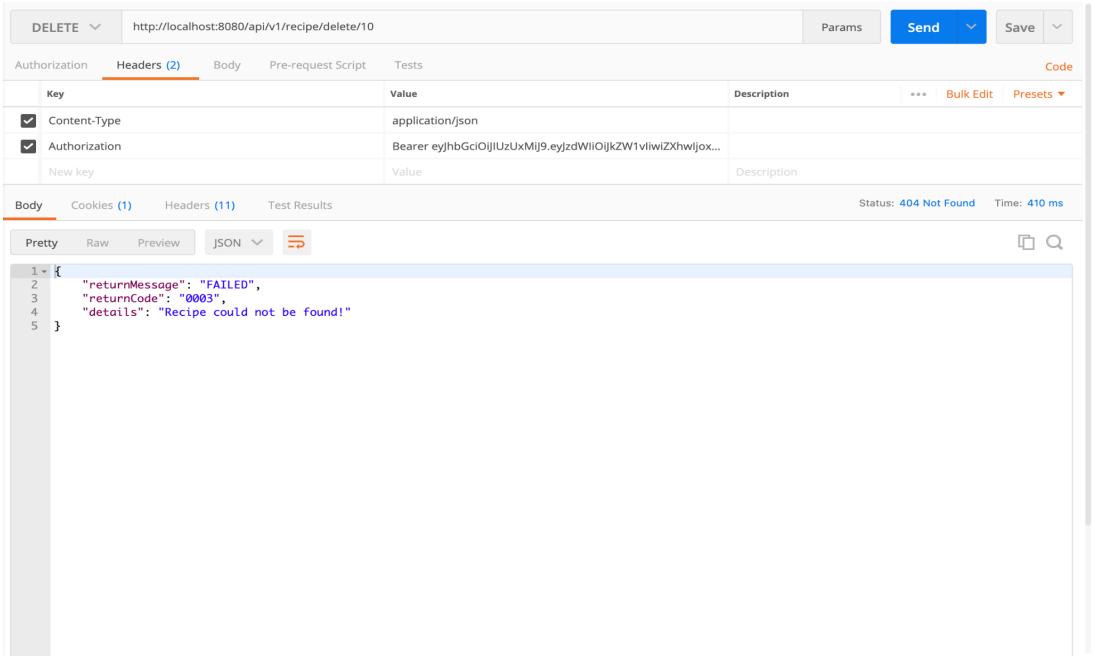


Push the delete request and we can see the response ok message.



If we give a recipe parameter that is not registered on the system before, the system result will throw a recipe that could not be found exception like below.

Example delete request: *http://localhost:8080/api/v1/recipe/delete/*10



5. */api/v1/recipe/update/{recipeId}*

Future: performs to update that given recipe object.

Endpoint : *http://localhost:8080/api/v1/recipe/update/{recipeId}*
Type : PUT
Parameter :

Request Path Parameter:

```
{
  name : recipeId
  type   : integer
}
```

Request Body Parameter:

recipeDto:

```
{
"recipeIngredientsList": [
   {
     "ingredientName": "string",
     "ingredient_id": integer
   }
 ],
 "recipeInstructionModelList": [
   {
     "instructionDetail": "string",
     "instruction_id":integer
   }
 ],
 "recipeName": "string",
 "suitableCount": integer,
 "veganYN": "string"
}
```

Example put request:*http://localhost:8080/api/v1/recipe/update/1*

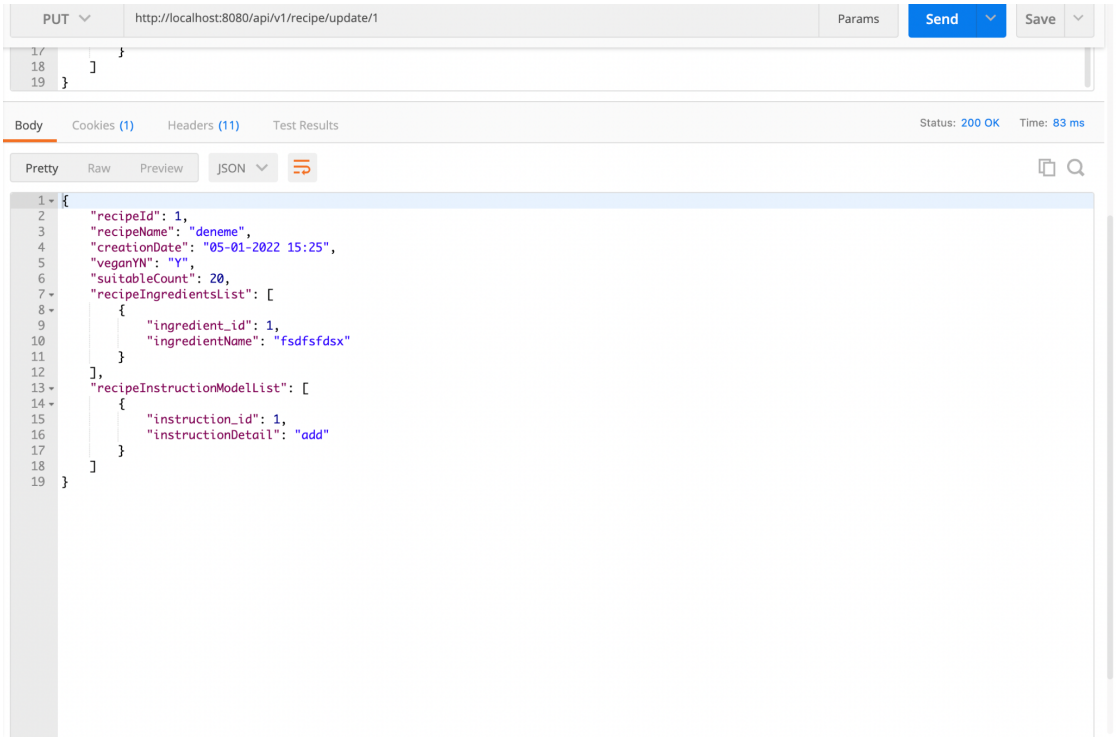
HTTP

Copy to Clipboard

```
Content-Type: application/json
Authorization: Bearer eyJhbGciOiJIUzUxMiJ9
    .eyJzdWIiOiJkZW1vIiwiZXhwIjoxNjQxNDI5ODU1LCJyb2xlcyI6W3siYXV0aG9yaXR5IjoiQURNSU4if
    V19.IxZF_iC56m0ronjyqpqCQ11VGj0WIsGCVOZ3ks8rldmGP33FE5TgG9Dd-PTKlk
    -LPBq6B5Ag8E0CI9uKUGZX_Q
Cache-Control: no-cache
Postman-Token: 71fb3fb4-76e9-abe9-7bc4-7f3249476a0f

{
    "recipeId": 1,
    "recipeName": "deneme",
    "creationDate": "05-01-2022 15:25",
    "veganYN": "Y",
    "suitableCount": 20,
    "recipeIngredientsList": [
        {
            "ingredient_id": 1,
            "ingredientName": "fsdfsfdsx"
        }
    ],
    "recipeInstructionModelList": [
        {
            "instruction_id": 1,
            "instructionDetail": "add"
        }
    ]
}
```

Push the put request and we can see the response ok message and updated recipe object.



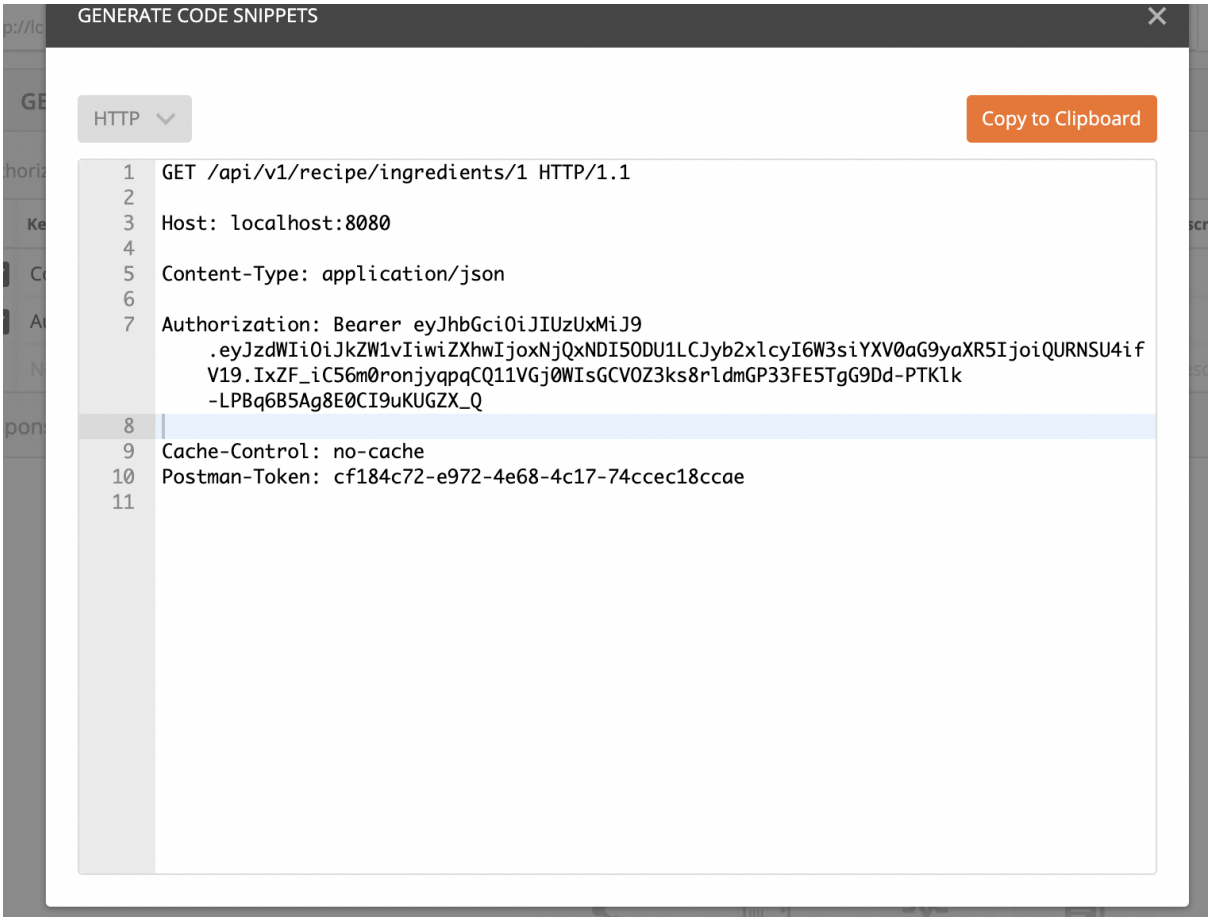
6. *[/api/v1/recipe/ingredients/{recipeId}](#)*

_____Future: performs to get all recipe ingredients that given recipe id.

Endpoint : http://localhost:8080/api/v1/recipe/ingredients/{recipeId}
Type : GET
Parameter :

```
{
  name : recipeId
  type  : integer
}
```

Example get request: http://localhost:8080/api/v1/recipe/ingredients/1



If you fetch this request, the result will be shown below.

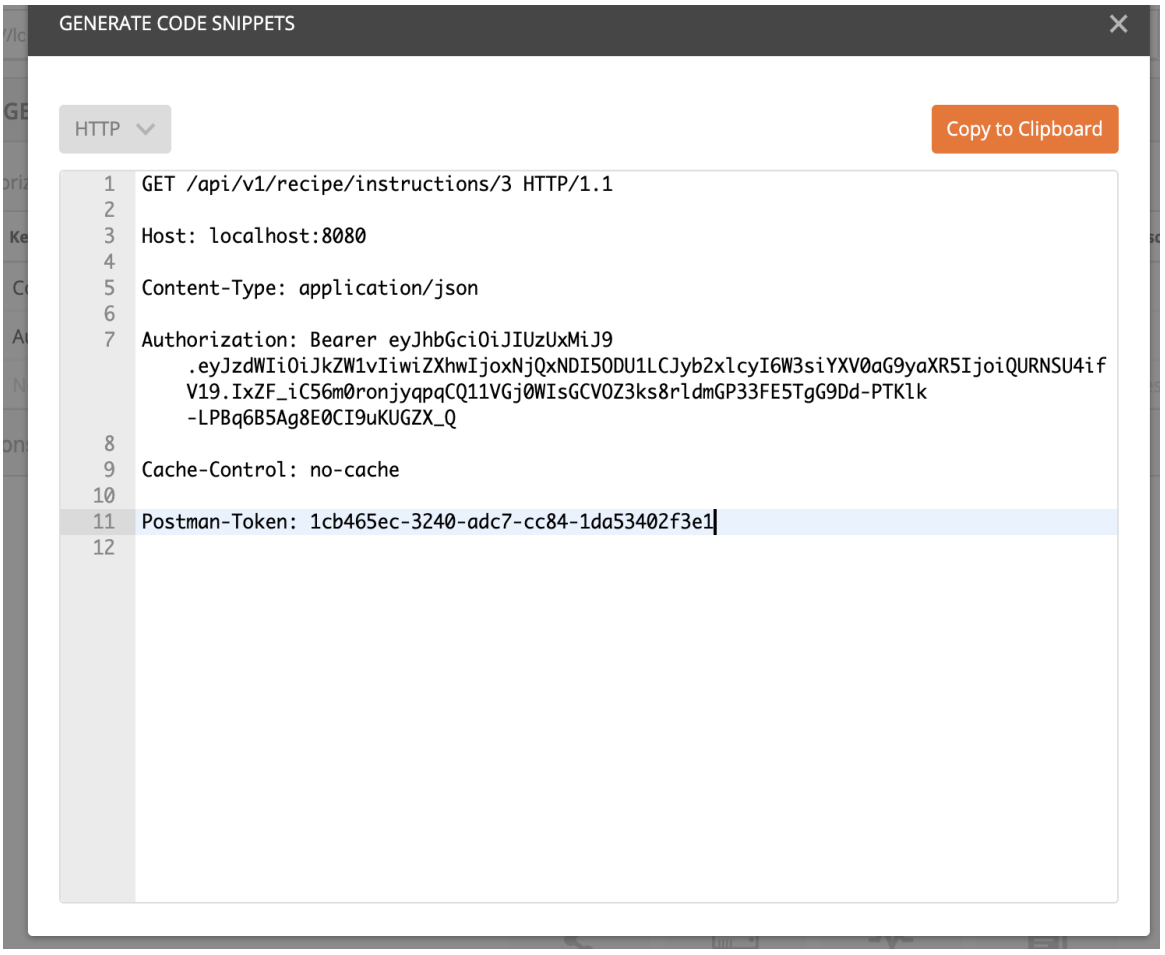
7. *[/api/v1/recipe/instructions/{recipeId}](#)*

_____Future: performs to get all recipe ingredients that given recipe id.

Endpoint : http://localhost:8080/api/v1/recipe/instructions/{recipeId}
Type : GET
Parameter :

```
{
  name : recipeId
  type  : integer
}
```

Example get request: http://localhost:8080/api/v1/recipe/instructions//1



If you fetch this request, the result will be shown below.