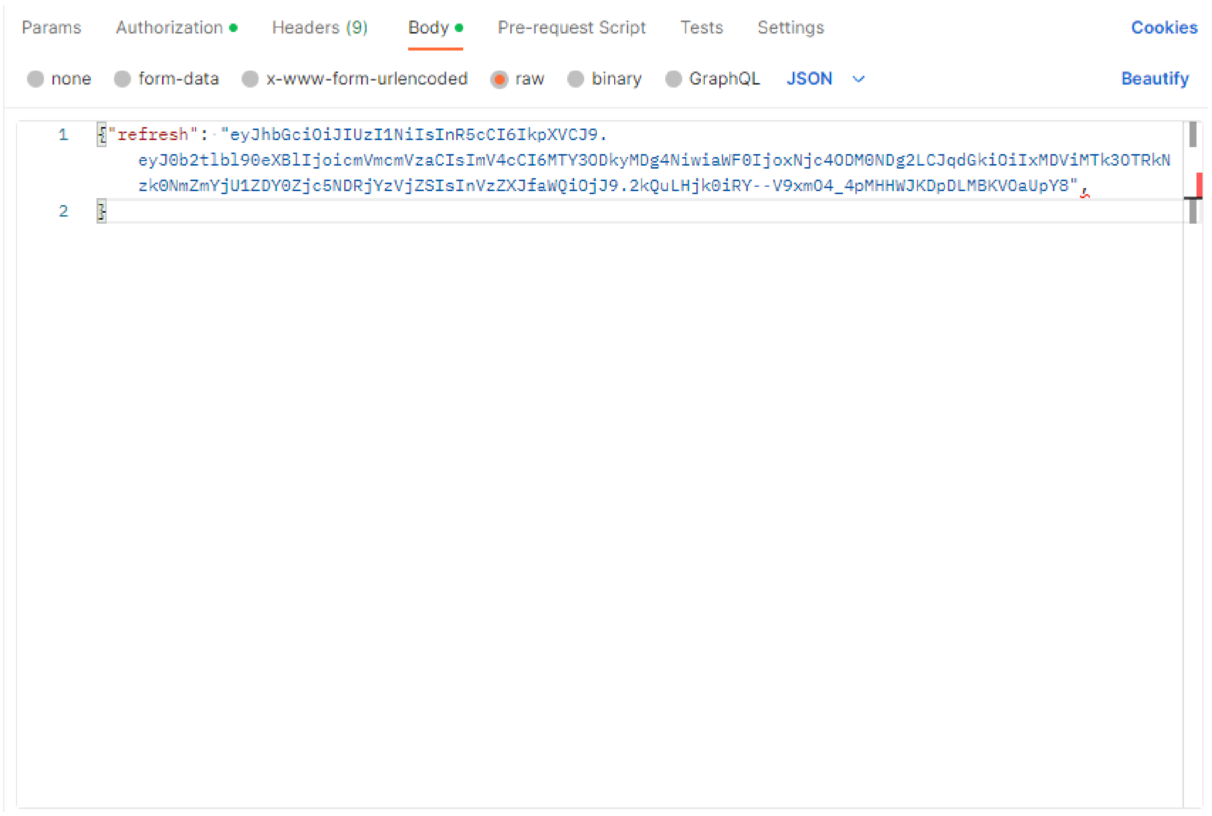
Authentication Table for accounts/:

Endpoint	Requires Authentication:
signup/	no
login/	no
logout/	yes
profile/view/	yes
profile/edit/	yes

Authentication Table for recipes/:

Endpoint	Requires Authentication
add/	yes
<recipe_id>/delete/	yes
all/	no
myrecipes/	yes
<recipe_id>/edit/	yes
<recipe_id>/	yes
<recipe_id>/like/	yes
<recipe_id>/unlike/	yes
<recipe_id>/servings/	yes
<recipe_id>/addreview/	yes
<recipe_id>/reviews/	no
<recipe_id>/editreview/	yes
<recipe_id>/deletereview/	yes
search/	no
shoppinglist/add/	yes
shoppinglist/view/	yes
autocomplete/	yes

Everytime a user logs in, they will be given both an access token and a refresh token. The user must input the access token as a bearer token in postman. The refresh token must be submitted as a raw json argument as shown below:



```json
{"refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmcmVzaCIsImV4cCI6MTY3ODkyMDg4NiwiaWF0IjoxNjc4ODM0NDg2LCJqdGkiOiIxMDViMTk3OTRkNzk0NmZmYjU1ZDY0Zjc5NDRjYzVjZSIsInVzZXJfaWQiOjJ9.2kQuLHjk0iRY--V9xmO4_4pMHHWJKDpDLMBKVOaUpY8",
}
```

Endpoint: recipes/add/

```
Returns a newly created recipe data
    Method : POST
    Args : "name", "description", "set_of_diets", "cuisine",
"ingredients", "servings", "steps", "time" (set of diets and
time is optional)
    Note : set of diets are a choice field and below are the
choices, user must input either "ND", "DF", "GF", "LC" or "V"
        no_diet = "ND", ("No diet")
        dairy_free = "DF", ("Dairy Free")
        gluten_free = "GF", ("Gluten Free")
        low_carb = "LC", ("Low Carb")
        vegan = "V", ("Vegan")

           cuisine is also a choice field and below are the
choices, user must input either "C", "MC", "J", "I", "INDO",
"G", "AM", or "AF"
        chinese = "C", ("Chinese")
        middle_eastern = "MC", ("Middle Eastern")
        japanese = "J", ("Japanese")
        indian = "I", ("Indian")
        indonesian = "INDO", ("Indonesian")
        german = "G", ("German")
        american = "AM", ("American")
        african = "AF", ("African")

           ingredients is a JSON field and it will be a list
of dictionaries
        format : [{"ingredient_name" : quantity},
{"ingredient_name" : quantity}]
        example : [{"flour" : 1}, {"green onion" : 2}, {"red
onion" : 3}]

           steps is also a JSON field and it will be a list of
strings
        format : ["step 1", "step 2"]
        example : ["put green onion in a plate", "cut them
into small pieces"]

           name, description is just a charfield and servings,
time is an integerfield. Time will always be in minutes so if
a recipe cooking time is 1 hour,
           the user should enter 60 in the time field.
```

Endpoint: recipes/<int:id>/delete/

```
"""
    Delete a recipe based on its id
    Method : DELETE
    Args : Just the recipe id on the URL

    """
```

Endpoint: recipes/all/

```
"""
    Returns All recipes that has been made as a list
    Method : GET
    Args : None
    """
```

Endpoint: recipes/myrecipes/

```
"""
    Returns all recipes that has the user as it's owner
    Method : GET
    Args : None
    """
```

Endpoint: recipes/<int:id>/edit/

```
"""
    Returns an edited recipe with the edited fields
    Method : POST
    Args : "name", "description", "set_of_diets", "cuisine",
"ingredients", "servings", "steps", "time" (ALL OPTIONAL)
    Note : set of diets are a choice field and below are the
choices, user must input either "ND", "DF", "GF", "LC" or "V"
    no_diet = "ND", ("No diet")
    dairy_free = "DF", ("Dairy Free")
    gluten_free = "GF", ("Gluten Free")
    low_carb = "LC", ("Low Carb")
    vegan = "V", ("Vegan")

    cuisine is also a choice field and below are the
choices, user must input either "C", "MC", "J", "I", "INDO",
"G", "AM", or "AF"
```

```
chinese = "C", ("Chinese")
middle_eastern = "MC", ("Middle Eastern")
japanese = "J", ("Japanese")
indian = "I", ("Indian")
indonesian = "INDO", ("Indonesian")
german = "G", ("German")
american = "AM", ("American")
african = "AF", ("African")

ingredients is a JSON field and it will be a list
of dictionaries
format : [{"ingredient_name" : quantity},
{"ingredient_name" : quantity}]
example : [{"flour" : 1}, {"green onion" : 2}, {"red
onion" : 3}]

steps is also a JSON field and it will be a list of
strings
format : ["step 1", "step 2"]
example : ["put green onion in a plate", "cut them
into small pieces"]

name, description is just a charfield and servings,
time is an integerfield. Time will always be in minutes so if
a recipe cooking time is 1 hour,
the user should enter 60 in the time field.
"""
```

Endpoint: recipe/<int:id>/

```
"""
    Returns details of recipe with id <id>
    Method : GET
    Args : None
    """
```

Endpoint: recipes/<int:id>/like/

```
Like the recipe with id <id>
    Method: PUT
    Args: The recipe id in the url
```

Endpoint: recipes/<int:id>/unlike/

```
Unlike a liked recipe with id <id>
Method: PUT
Args: The recipe id in the url
```

Endpoint: recipes/search/

```
Returns a list of recipes according to the search attributes
    Method : GET
    Args : Any field from a recipe such as "name",
"ingredients_name", "owner_username"
```

Endpoint: recipes/<int:id>/addreview/

```
Create a review for the recipe with id <id>
Method: POST
Arguments:
- review: a text review of the recipe (optional)
- rating: an integer rating from 1 to 5
- the recipe id provided in the url
```

Endpoint: recipes/<int:id>/reviews/

```
View all the reviews for the recipe with id <id>
Method: GET
Arguments: The recipe id in the url
```

Endpoint: recipes/<int:id>/editreview/

```
Edit an already written review for the recipe with id <id>
Method: PUT
Arguments: review, rating, id
Note: The only mandantory argument is the recipe id in the
url
```

Endpoint: Endpoint: recipes/<int:id>/deletereview/

```
Delete the current user's review for recipe with id <id>
Method: DELETE
Arguments: The recipe id in the url
```

Endpoint : recipes/autocomplete/

```
Returns a list of ingredients that contains the user input
Method : GET
Args : "input"
Note : input can be any name of incomplete ingredients,
but for the sake of the DEMO, the choices are only
('green onion', 'Green Onion'),
('red onion', 'Red Onion'),
('butter', 'Butter'),
('garlic', 'Garlic')

Example : "input" : "on" , output will return red onion
and green onion
```

Endpoint : recipes/shoppinglist/add/

```
Returns a list of ingredients with the total amount of
quantity
Method : POST
Args : "recipe_name"
```

Endpoint : recipes/shoppinglist/view/

```
Returns a list of ingredients with the total amount of
quantity
Method : GET
Args : None, just refresh token
```

Endpoint: recipes/<int:id>/servings/

```
Returns a list of ingredients with updated quantity based on
the new servings
Method : PATCH
Args : "servings"
Note : servings is an integer and cannot be negative
```