Contents

[1 Introduction: 3](#_Toc82984502)

[2 Register User 3](#_Toc82984503)

[1. Functional Specification: 3](#_Toc82984504)

[2. Service Operations: 3](#_Toc82984505)

[3 Login User 4](#_Toc82984506)

[1. Functional Specification: 4](#_Toc82984507)

[2. Service Operations: 4](#_Toc82984508)

[4 Add and Update New Recipe 5](#_Toc82984509)

[1. Functional Specification: 5](#_Toc82984510)

[2. Service Operations: 5](#_Toc82984511)

[5 Delete Recipe 6](#_Toc82984512)

[1. Functional Specification: 6](#_Toc82984513)

[2. Service Operations: 6](#_Toc82984514)

[6 List All Recipes 7](#_Toc82984515)

[1. Functional Specification: 7](#_Toc82984516)

[2. Service Operations: 7](#_Toc82984517)

[7 Search Recipes 8](#_Toc82984518)

[1. Functional Specification: 8](#_Toc82984519)

[2. Service Operations: 8](#_Toc82984520)

[8 Authentication & Authorization 9](#_Toc82984521)

# Introduction:

This service aims to provide recipe management in the system. In order to save recipes in the system the user must have register himself in the system. This documents illustrates how to add, update, delete and retrieve recipes. Please carry on below to get more information.

# Register User

## Functional Specification:

|  |  |
| --- | --- |
| **Service Name** | Register User |
| **Service Description** | This service allow the user to registers to the system |
| **HTTP METHOD** | Post |
| **Endpoint** | http://localhost:8082/auth/register |

## Service Operations:

1. **INPUT PARAMETERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Length** | **Optional** | **Description** |
| **name** | String |  | NO | Name Of user |
| **email** | String |  | NO | Email address |
| **password** | String |  | NO | Password |

1. **Sample input**

**{**

**"**name":"sikandar",

"email":"sikandar@gmail.com",

"password":"test123"

**}**

1. **OUTPUT JSON**

Result will be JSON objects.

**Operation Response**

**{**

"status": 200,

"message": "user created successfully",

"code": "successful",

"data": "sikandar"

**}**

**Note: - Basic authentication is required to use this service (Refer to Authentication and Authorization)**

# Login User

## Functional Specification:

|  |  |
| --- | --- |
| **Service Name** | Login User |
| **Service Description** | This service allow the user to login to the system. The response will be token which will be used to call another API throughout the service. |
| **HTTP METHOD** | POST |
| **Endpoint** | http://localhost:8081/oauth/token |

## Service Operations:

1. **INPUT PARAMETERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Length** | **Optional** | **Description** |
| **grant\_type \_** | String |  | NO | - |
| **username** | String |  | NO | Email of user |
| **password** | String |  | NO | Password of user |

1. **Sample input**

**Form Data**

grant\_type password

username [sikandar@gmail.com](mailto:sikandar@gmail.com)

password sikandar

1. **OUTPUT JSON**

Result will be JSON objects.

**Operation Response**

**{**

**"access\_token": "ed740626-edbc-43fb-b405-dfd34db72a86",**

**"token\_type": "bearer",**

**"refresh\_token": "55967444-e79c-4129-981d-1d962d482427",**

**"expires\_in": 147643,**

**"scope": "read write"**

**}**

**Note: - 1.Basic authentication is required to use this service (Refer to Authentication and Authorization)**

**2. Input parameters must be a form data**

# Add and Update New Recipe

## Functional Specification:

|  |  |
| --- | --- |
| **Service Name** | Add New Recipe |
| **Service Description** | This service allows to add a new recipe or update the existing based on **recipeId** if provided |
| **HTTP METHOD** | Post |
| **Endpoint** | http://localhost:8082/api/recipe/add |

## Service Operations:

1. **INPUT PARAMETERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Length** | **Optional** | **Description** |
| **recipeId** | Long |  | NO | If Recipe ID is provided, the existing recipe will be updated, otherwise it will create a new recipe |
| **recipeTitle** | String |  | NO | Title of Recipe |
| **isVegeterian** | String |  | NO | ‘yes’ or ‘no’ if the recipe is either vegetarian or not |
| **suitableForPeople** | Long |  | NO | Give the number of people for whom the recipe is suitable for |
| **Ingredients** | String |  | NO | Provide comma separated ingredients (view is implemented in fronted application) |
| **cookingInstructions** | String |  | NO | Provide a short description how to cook the recipe |

1. **Sample input**

**Form Data**

{

        "recipeTitle": "nuggets",

        "isVegeterian": "no",

        "suitableForPeople": 2,

        "ingredients": "chicken pieces, oil, salt, pepper",

        "cookingInstructions": "baki nuggests in hot oil"

}

1. **OUTPUT JSON**

Result will be JSON objects.

**Operation Response**

{

    "status": 200,

    "message": "Recipe saved successfully.",

    "code": "successful",

    "data": "1"

}

**Note: - Basic authentication is required to use this service (Refer to Authentication and Authorization)**

# Delete Recipe

## Functional Specification:

|  |  |
| --- | --- |
| **Service Name** | Add New Recipe |
| **Service Description** | This service allows to delete a recipe when an id is provided |
| **HTTP METHOD** | Post |
| **Endpoint** | http://localhost:8082/api/recipe/delete |

## Service Operations:

1. **INPUT PARAMETERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Length** | **Optional** | **Description** |
| **id** | Long |  | NO | The provided id if exists in the recipes, it will be deleted |

1. **Form Data**

{

        "id":"4"

}

1. **OUTPUT JSON**

Result will be JSON objects.

**Operation Response**

{

    "status": 200,

    "message": "Recipe Deleted successfully.",

    "code": "successful",

    "data": "1"

}

**Note: - Basic authentication is required to use this service (Refer to Authentication and Authorization)**

# List All Recipes

## Functional Specification:

|  |  |
| --- | --- |
| **Service Name** | Retrieve all saved recipes |
| **Service Description** | This service will list all recipes which are saved in the system. |
| **HTTP METHOD** | POST |
| **Endpoint** | http://localhost:8082/api/recipe/all |

## Service Operations:

1. **INPUT PARAMETERS**

No input parameters are required for this service

1. **OUTPUT JSON**

Result will be the list of JSON objects.

[

    {

        "id": 1,

        "recipeTitle": "maghaz",

        "isVegeterian": "No",

        "createdDate": "2021-09-18T22:16:13.000+0000",

        "updatedDate": "2021-09-19T16:04:48.000+0000",

        "suitableForPeople": 2,

        "ingredients": "dfdaf,asdf,asdf,fasdfasdf,sikandar",

        "cookingInstructions": "asdfasdfas",

        "ingredientsList": [

            "dfdaf,asdf,asdf,fasdfasdf,sikandar"

        ]

    },

    {

        "id": 3,

        "recipeTitle": "chatni roll",

        "isVegeterian": "Yes",

        "createdDate": "2021-09-18T22:19:10.000+0000",

        "updatedDate": "2021-09-19T16:05:01.000+0000",

        "suitableForPeople": 2,

        "ingredients": "asdfasdf,erte,fadfasdf,asdfasdf",

        "cookingInstructions": "asdfasdf",

        "ingredientsList": [

            "asdfasdf,erte,fadfasdf,asdfasdf"

        ]

    },

    {

        "id": 5,

        "recipeTitle": "nuggets update",

        "isVegeterian": "no",

        "createdDate": "2021-09-19T18:09:00.000+0000",

        "updatedDate": "2021-09-19T18:09:00.000+0000",

        "suitableForPeople": 2,

        "ingredients": "chicken pieces, oil, salt, pepper",

        "cookingInstructions": "baki nuggests in hot oil",

        "ingredientsList": [

            "chicken pieces, oil, salt, pepper"

        ]

    }

]

**Note: - Authorization token is required to use this service (Refer to Authentication and Authorization)**

# Search Recipes

## Functional Specification:

|  |  |
| --- | --- |
| **Service Name** | Retrieve all saved recipes |
| **Service Description** | This service will list all recipes which are saved in the system. |
| **HTTP METHOD** | POST |
| **Endpoint** | http://localhost:8082/api/recipe/search |

## Service Operations:

1. **INPUT PARAMETERS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Type** | **Length** | **Optional** | **Description** |
| **createdDate** | String |  | YES | 19-09-2021 02:16  date in this format |
| **isVegeterian** | String |  | YES | ‘yes’ or ‘no’  You want all recipes which are vegetarian or not |
| **suitableFor** | Long |  | YES | No of people for whom the recipe is suitable |
| **cookingInstructions** | String |  | YES | Search by cooking instructions, matches any word |

1. **Sample input parameters**

createdDate password

isVegeterian [‘yes’](mailto:sikandar@gmail.com)

suitableFor 2  
  
cookingInstructions baking

1. **OUTPUT JSON**

Result will be the list of JSON objects.

[

    {

        "id": 1,

        "recipeTitle": "maghaz",

        "isVegeterian": "No",

        "createdDate": "2021-09-18T22:16:13.000+0000",

        "updatedDate": "2021-09-19T16:04:48.000+0000",

        "suitableForPeople": 2,

        "ingredients": "dfdaf,asdf,asdf,fasdfasdf,sikandar",

        "cookingInstructions": "asdfasdfas",

        "ingredientsList": [

            "dfdaf,asdf,asdf,fasdfasdf,sikandar"

        ]

    }

]

**Note: - Authorization token is required to use this service (Refer to Authentication and Authorization)**

# Authentication & Authorization

**Register user** and **Login user** API must include the user id and password with a colon “:” separated base64 encoded and send this information in the request header as shown below.

For example: user id is john and password is abc123 then encode the string “john:abc123” to base64

Requests include the header name “Authorization” with value starts with “ Basic ” and concatenate with encoded string in the header as shown below.

Authorization: Basic dGVzdDpzaWthbmRhcjEyMw==

Other than login and register all API must include authorization token shown below in request header. The below token is generated each time the user logs in and remains until logout

Authorization: Bearer ed740626-edbc-43fb-b405-dfd34db72a86