Meals And Recipes API

The Meals & Recipes Application, built using Java, Spring framework, MongoDB, and Postman, offers a user-friendly platform for exploring and interacting with various meal categories like Italian, French, and Fast-food. Users can browse through specific foods within each category, access detailed information on meals including ingredients, recipes, and user comments, and have the ability to star and comment on recipes. The app includes features for account creation, login, managing favorites, and displays starred meals in a dedicated favorites section. **USER ENDPOINTS**

* **Register a User**

-URL: ‘/api/users/register’

-Method: POST

* Data constraints:

{

"username": "[valid username]",

"password": "[password in plain text]"

}

* Success Response:
  + Code: ‘200 OK’
  + Content: ‘User [username] registered successfully’
* Error Response:
  + Code: ‘400 BAD REQUEST’
  + Content: ‘Username already exists’
* **Login User**

-URL: ‘/api/users/login’

-Method: POST

* Data constraints:

{

"username": "[username]",

"password": "[password]"

}

* Success Response:
  + Code: ‘200 OK’
* Error Response:
  + Code: ‘401 UNAUTHORIZED’
  + Content: ‘Invalid username or password’
* **Get User Favorites**
* URL: ‘**/**api/users/{userId}/favorites**’**
* Method: GET
* URL Params: ‘username =[String]’
* Success Response:
  + Code: ‘200 OK’
  + Content:

[

{

"userId": “1123123”,

"username": "newuser",

“favorites”: null

},  
….

]

* Error Response:
  + Code: ‘404 NOT FOUND’
* **Get All Users**
* URL: ‘/api/users’
* Method: GET
* Success Response:
  + Code: ‘200 OK’
  + Content:

[

    {

         "userId": "6576043fee0d170d08572f1a",

        "username": "newuser",

        "password": "password123",

        "favorites": **null**

    },

….

]

* Error Response:
  + Code: ‘404 NOT FOUND’
* **Add meal to User’s Favorites**

- URL: ‘/api/users/{userId}/favorites/add/{mealId}’

- Method: POST

- Success Response:

- Code: ‘200 OK’

- Content: Updated User object with favorites

- Error Response:

- Code: ‘404 NOT FOUND’

- Content: ‘User/Meal not found’

* **Remove meal from User’s Favorites**

- URL: ‘/api/users/{userId}/favorites/remove/{mealId}’

- Method: DELETE

- Success Response:

- Code: ‘200 OK’

- Content: Updated User object without the removed favorite

- Error Response:

- Code: ‘404 NOT FOUND’

- Content: ‘User/Meal not found’

**CATEGORY ENDPOINTS**

* **Get All Categories**
* URL: ‘/api/categories‘
* Method: GET
* Auth required: No
* Success Response:
  + Code: ‘200 OK’
  + Content:

[

{

"categoryId": “1”,

"categoryName": "Italian"   
“meals”: [

“mealName”: “Pizza”,  
 “recipe”: “recipe”,

“comments” : null,

“mealId”: “meal1”

],  
..

},  
…

]

* **Get Meals by Category by ID**

- URL: ‘/api/categories/{categoryId}/meals’

-Method: GET

-URL Params: ‘categoryId=[string]‘

-Success Response:

o Code: ‘200 OK’

oContent:

[

{

"mealName": “Pizza”,

"recipe": “recipe”,

"comments" : null,

"mealId": “meal1”

},

...

]

-Error Response:

oCode: ‘404 NOT FOUND ‘

* **Add Category**

- URL: ‘/api/categories’

- Method: POST

- Auth required: No

- Data constraints:

{

"categoryName": "[category name]",

"meals": [

{

"mealName": "[meal name]",

"recipe": "[recipe details]",

// other meal fields...

},

// ...other meals

]

}

- Success Response:

- Code: ‘201 Created’

- Content:

{

"categoryId": "[newly created category ID]",

"categoryName": "[category name]",

"meals": [

{

"mealName": "[meal name]",

"recipe": "[recipe details]",

// other meal fields...

},

// ...other meals

]

}

- Error Response:

- Code: ‘400 BAD REQUEST’

- Content: ‘Error message if any validation fails or category cannot be created’

* **Delete Category**

- URL: ‘/api/categories/{categoryId}’

- Method: DELETE

- Auth required: Yes (typically you would restrict delete operations to authorized users)

- URL Params: ‘categoryId=[String]’

- Success Response:

- Code: ‘200 OK’

- Content: ‘Category successfully deleted’ or no content

- Error Response:

- Code: ‘404 NOT FOUND’

- Content: ‘Category not found with id: [categoryId]’

* **Add Meal to Category**
* URL: /api/categories/{categoryId}/meals
* Method: POST
* Auth Required: Yes
* URL Params: categoryId=[string]
* Data Constraints:

{

"mealName": "[meal name]",

"recipe": "[recipe details]",

// other meal fields...

}

* Success Response:
  + - Code: 201 Created
    - Content:

{

"mealId": "[newly created meal ID]",

"mealName": "[meal name]",

"recipe": "[recipe details]",

// other meal fields...

}

* Error Response:
  + - Code: 404 NOT FOUND
    - Content: Category not found with id: [categoryId]
* **Delete Meal From Category**
* URL: /api/categories/{categoryId}/meals/{mealId}
* Method: DELETE
* Auth Required: Yes
* URL Params: categoryId=[string], mealId=[string]
* Success Response:
  + - Code: 200 OK
    - Content: Meal successfully deleted from category
* Error Response:
  + - Code: 404 NOT FOUND
    - Content: Category or Meal not found

**MEAL ENDPOINTS**

* **Get All Meals (All meals in the database)**
* URL: ‘/api/meals‘
* Method: GET
* Auth required: No
* Success Response:
  + Code: ‘200 OK’
  + Content: [

{

“mealName”: “Pizza”,  
 “recipe”: “recipe”,

“comments” : null,

“mealId”: “meal1”

},

//other meals…

]

* **Get Meal by Meal Id**
* URL: ‘/api/meals/{mealId}’
* Method: GET
* Auth required: No
* Success Response:
  + Code: ‘200 OK’
  + Content: {

“mealName”: “Pizza”,  
 “recipe”: “recipe”,

“comments” : null,

“mealId”: “meal1”

}

* **Post a Comment on a Meal**
* URL: ‘/api/meals/{mealId}/addComment’
* Method: POST
* Auth required: No
* Success Response:
  + Code: ‘201 Created
  + Content:
* Data Constraints:  
  {

  "commentText": "This is a great meal!",

  "userId": "6576043fee0d170d08572f1a"

}