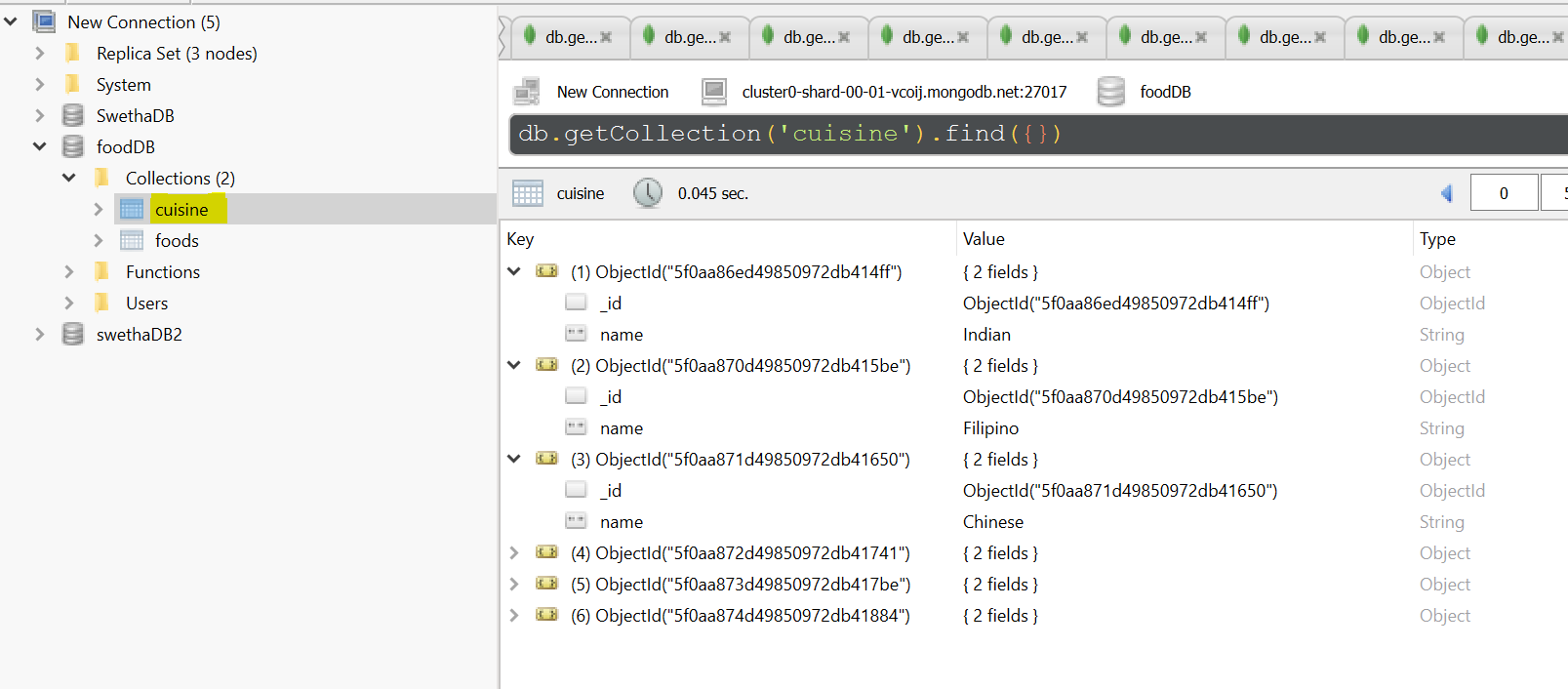
|  |  |
| --- | --- |
| THE SECRET RECIPE  GROUP PROJECT | Abstract  This is the documentation for the API created and consumed for the secret recipe application. The document also describes the mongo DB structure of collections in it.  Swetha Mary Philip, Gagandeep Kaur, Deepak Sharma |

**MONGO DATABASE DESIGN**

**Collection**: Cuisine

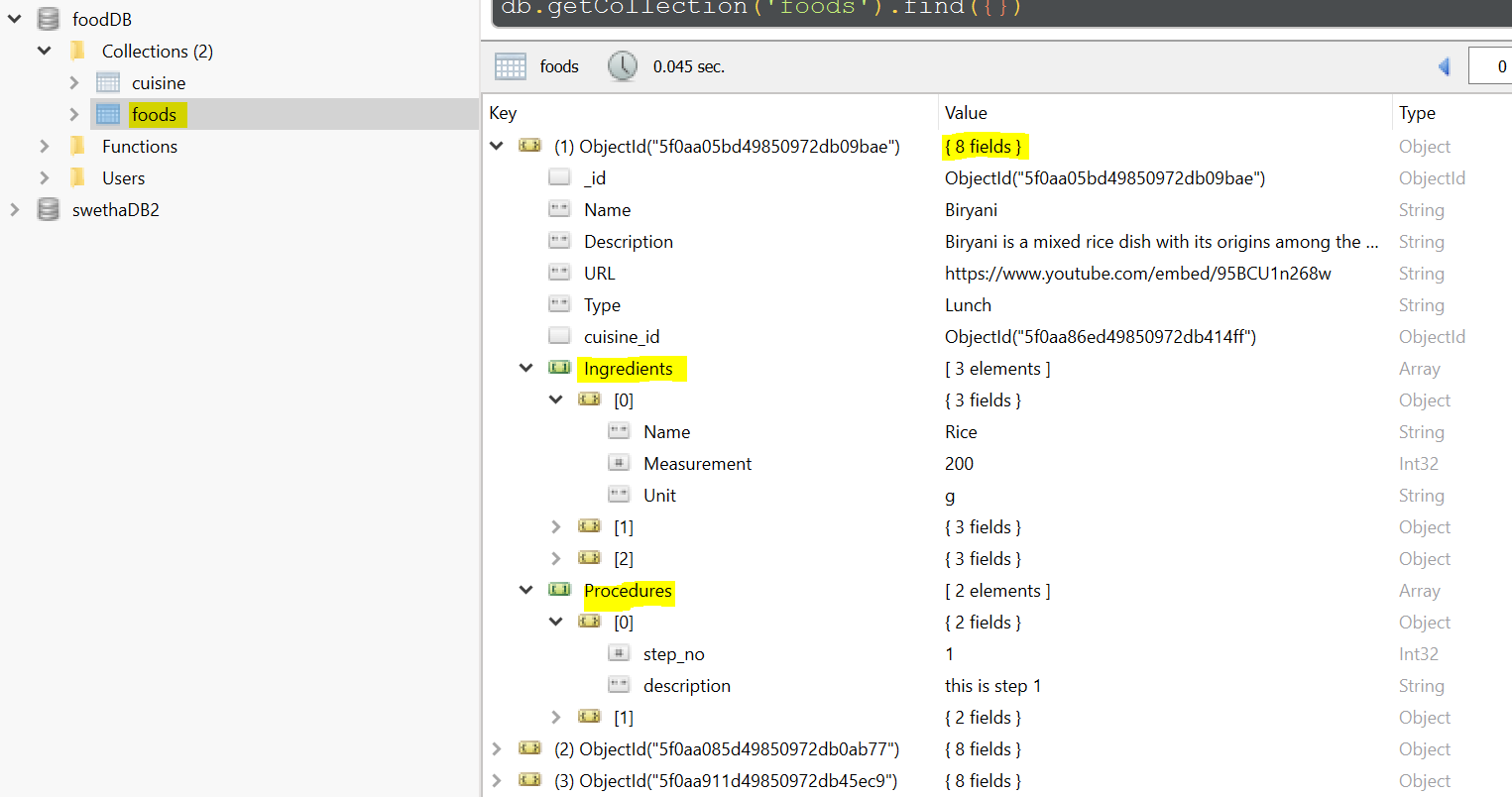
**Description**: This is a collection which maintains all the cuisines like Indian, Filipino, Japanese etc.



**Collection**: foods

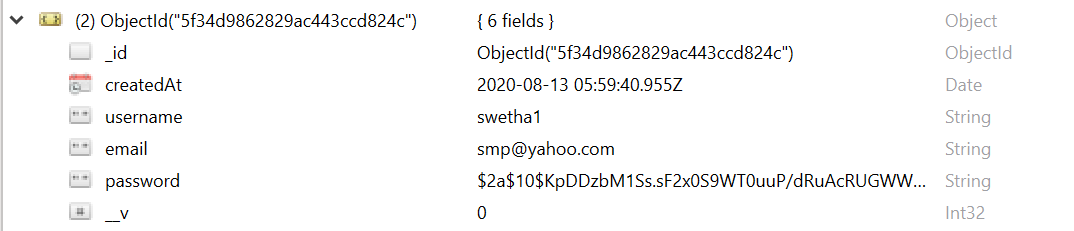
**Description**: This is the main recipe collection which maintains all the dishes with its details. It has the cuisine\_id as the foreign object\_ID from cuisine collection.

Foods collection has sub-document for details on “ingredients” and “procedure”.



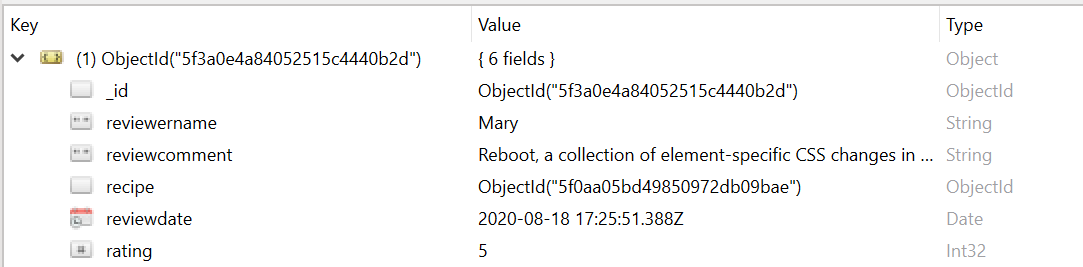
**Collection**: users

**Description**: This collection maintains the user related data like username, email and encrypted password.



**Collection**: review

**Description**: This collection maintains the reviews for every recipe posted. It has a foreign object\_id with recipe ID to link it to the recipe.



**API DOCUMENTATION**

**GET FOODS API**

This API get the list of all the recipe in the mongo database with all the related details.

|  |  |  |
| --- | --- | --- |
| **GET OPERATION** | **Name** | **DataType** |
| **Request Parameter** |  |  |
|  |  |  |
| **Response** | \_id | ObjectID |
|  | name | String |
|  | description | String |
|  | url | String |
|  | type | String |
|  | cuisine | ObjectID |
|  | **ingredients** | Array |
|  | **ingredients**.\_id | ObjectID |
|  | **ingredients**.name | String |
|  | **ingredients**.measurement | Int32 |
|  | **ingredients**.unit | String |
|  | **procedures** | Array |
|  | **procedures**.\_id | ObjectID |
|  | **procedures**.step\_no | Int32 |
|  | **procedures**.description | String |

**EXAMPLE:**

*Request Type: GET*

<http://localhost:3000/api/foods/>

*Response type: JSON:*



**GET SINGLE FOOD API**

This API get the details of all the selected recipe from the mongo database with all the related details.

|  |  |  |
| --- | --- | --- |
| **GET OPERATION** | **Name** | **DataType** |
| **Request Parameter** | \_id | ObjectID |
|  |  |  |
| **Response** | \_id | ObjectID |
|  | name | String |
|  | description | String |
|  | url | String |
|  | type | String |
|  | cuisine | ObjectID |
|  | **ingredients** | Array |
|  | **ingredients**.\_id | ObjectID |
|  | **ingredients**.name | String |
|  | **ingredients**.measurement | Int32 |
|  | **ingredients**.unit | String |
|  | **procedures** | Array |
|  | **procedures**.\_id | ObjectID |
|  | **procedures**.step\_no | Int32 |
|  | **procedures**.description | String |

**EXAMPLE:**

*Request Type: GET* 🡪 http://localhost:3000/api/foods**/{\_id}**

<http://localhost:3000/api/foods/5f20e346fe7ada460888fd8d>

*Response type: JSON:*



**DELETE FOOD API**

This API get the details of all the selected recipe from the mongo database with all the related details.

|  |  |  |
| --- | --- | --- |
| **DELETE OPERATION** | **Name** | **DataType** |
| **Request Parameter** | \_id | ObjectID |
|  |  |  |
| **Response** | Empty JSON |  |

**EXAMPLE:**

*Request Type: DELETE* 🡪 http://localhost:3000/api/foods**/{\_id}**

<http://localhost:3000/api/foods/5f20e346fe7ada460888fd8d>

*Response: EMPTY JSON*

**CREATE NEW RECIPE API**

This API creates a new recipe in the mongo database with all the passed details.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **POST OPERATION** | **Name** | **DataType** |  |  | **Name** | **DataType** |
| **Request Parameter** (Body Parameter) | name | String |  | **Response** | \_id | ObjectID |
|  | description | String |  |  | name | String |
|  | url | String |  |  | description | String |
|  | type | String |  |  | url | String |
|  | cuisine | ObjectID |  |  | type | String |
|  | **ingredients** | Array |  |  | cuisine | ObjectID |
|  | **ingredients**.name | String |  |  | **ingredients** | Array |
|  | **ingredients**.measurement | Int32 |  |  | **ingredients**.\_id | ObjectID |
|  | **ingredients**.unit | String |  |  | **ingredients**.name | String |
|  | **procedures** | Array |  |  | **ingredients**.measurement | Int32 |
|  | **procedures**.step\_no | Int32 |  |  | **ingredients**.unit | String |
|  | **procedures**.description | String |  |  | **procedures** | Array |
|  |  |  |  |  | **procedures**.\_id | ObjectID |
|  |  |  |  |  | **procedures**.step\_no | Int32 |
|  |  |  |  |  | **procedures**.description | String |

**EXAMPLE:**

*Request Type: POST* 🡪 <http://localhost:3000/api/foods>

****

**UPDATE RECIPE API**

This API updates the details of all the selected recipe in the mongo database.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **PUT OPERATION** | **Name** | **DataType** |  |  | **Name** | **DataType** |
| **Request Parameter** (Path Parameter) | \_id | ObjectID |  | **Response** | \_id | ObjectID |
| **Request Parameter** (Body Parameter) | name | String |  |  | name | String |
|  | description | String |  |  | description | String |
|  | url | String |  |  | url | String |
|  | type | String |  |  | type | String |
|  | cuisine | ObjectID |  |  | cuisine | ObjectID |
|  | **ingredients** | Array |  |  | **ingredients** | Array |
|  | ingredients.\_id | ObjectID |  |  | ingredients.\_id | ObjectID |
|  | ingredients.name | String |  |  | ingredients.name | String |
|  | ingredients.measurement | Int32 |  |  | ingredients.measurement | Int32 |
|  | ingredients.unit | String |  |  | ingredients.unit | String |
|  | **procedures** | Array |  |  | **procedures** | Array |
|  | procedures.\_id | ObjectID |  |  | procedures.\_id | ObjectID |
|  | procedures.step\_no | Int32 |  |  | procedures.step\_no | Int32 |
|  | procedures.description | String |  |  | procedures.description | String |

**EXAMPLE:**

*Request Type: POST* 🡪 http://localhost:3000/api/foods**/{\_id}** <http://localhost:3000/api/foods/5f2a53744ecb3b19908fcbc8>

\*Request/Response Same as Create recipe

**GET CUISINE API**

This API get the details of all the cuisines from the mongo database.

|  |  |  |
| --- | --- | --- |
| **GET OPERATION** | **Name** | **DataType** |
| **Request Parameter** |  |  |
|  |  |  |
| **Response** | \_id | ObjectId |
|  | name | String |

**EXAMPLE:**

*Request Type: GET* 🡪 http://localhost:3000/api/cuisine



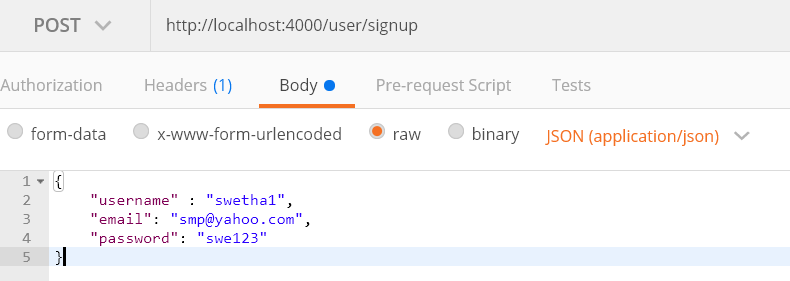
**REGISTER USER API**

This API is used to register new user into the website.

|  |  |  |
| --- | --- | --- |
| **POST OPERATION** | **Name** | **DataType** |
| **Request Parameter** | username | String |
|  | email | String |
|  | Password | String |
| **Response** | token | String |
|  |  |  |

**EXAMPLE:**

*Request Type: POST* 🡪 <http://localhost:4000/user/signup>



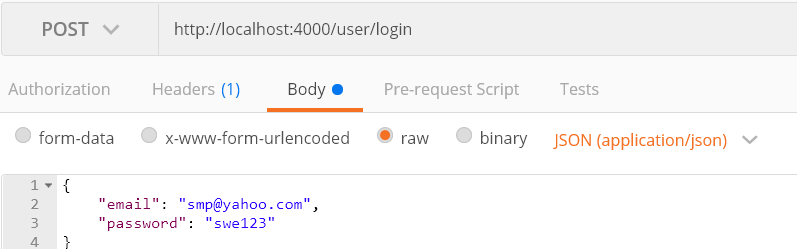
**LOGIN USER API**

This API is used to login existing user into the website.

|  |  |  |
| --- | --- | --- |
| **GET OPERATION** | **Name** | **DataType** |
| **Request Parameter** |  |  |
|  | email | String |
|  | Password | String |
| **Response** | token | String |
|  |  |  |

**EXAMPLE:**

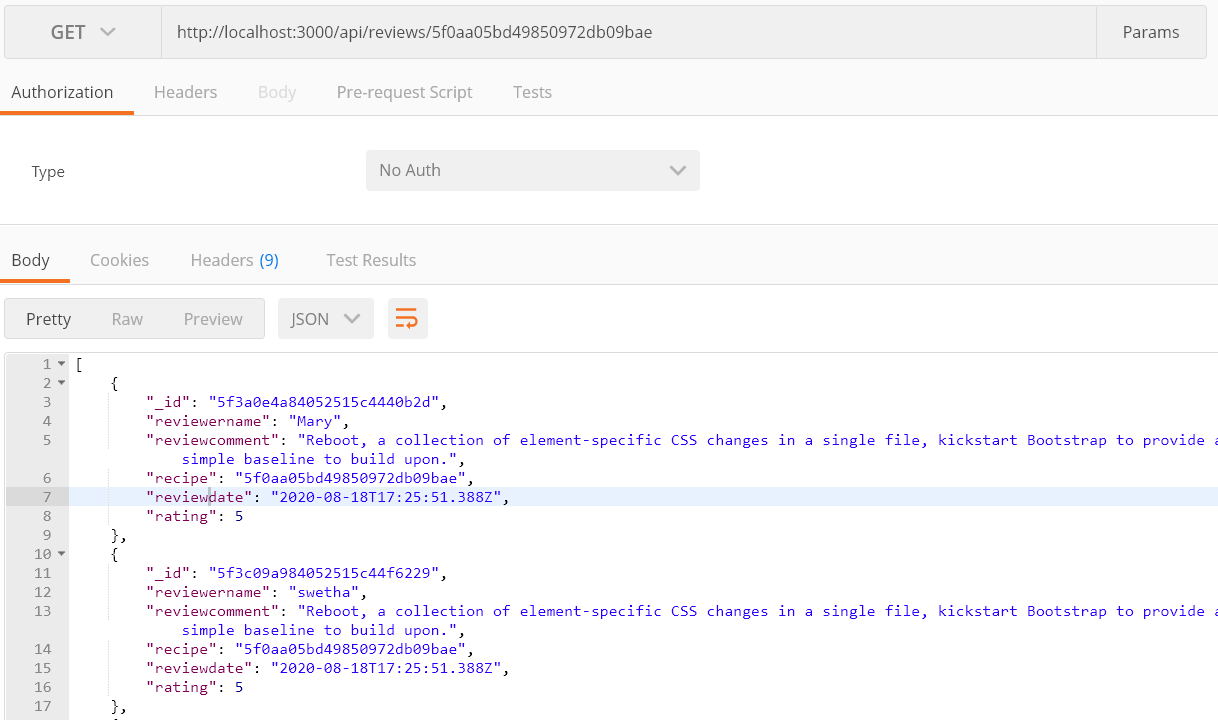
*Request Type: GET* 🡪 <http://localhost:4000/user/login>



**FETCH RECIPE REVIEWS API**

This API is used to get the reviews of the recipe based on the recipe ID which is passed in the URL

|  |  |  |
| --- | --- | --- |
| **GET OPERATION** | **Name** | **DataType** |
| **Request Parameter** | \_id | ObjectID |
|  |  |  |
|  |  |  |
| **Response** | *Please see below* | |



**CREATE RECIPE REVIEW API**

This API is used to add a new review for the recipe based on the recipe ID which is passed in the URL

|  |  |  |
| --- | --- | --- |
| **POST OPERATION** | **Name** | **DataType** |
| **Request Parameter** | \_id | ObjectID |
|  |  |  |
| **Response** | *Please see below* | |

