**Procedure to run program and API payload**

1. Go to GitHub, click on clone or download.
2. Copy the clone URL for the repository.
3. Open Git bash
4. Change the current working directory to the location where you want to clone the directory to be made.
5. Type **git clone** and then paste the URL copied URL.
6. Press Enter ‘sample’ project will created.
7. Open that folder in any tool(visual studio code, WebStrom etc)
8. Open terminal.
9. Types in the terminal npm install to get all dependencies.
10. Types in the terminal **node app.js** to run project or **nodemon** in the terminal, if the nodemon is not installed type **npm install nodemon –save**

* First created project by using **npm init,** it will ask few project related questions such as project name, version, description, author etc.
* After that created **app.js** file in this folder imported express and body-parser module and port to listen the server request etc.
* Created **routes** folder in that creates two files such as **products.js** and **category.js** file for writing the routes.
* Created **service** folder in that creates two files such as **product\_service.js** and **category\_service.js** for writing the services.
* Created **controllers** folder in that creates three files such as **products.js, category.js** for writing logic operation **and index.js** for creating the entire controllers variable name.
* Created **utils** folder in that created **database.js** file for connecting to mysql.
* **Db.sql** file contains the database and tables.

**Category API’s**

1) **Add category**

**Method:** post

**Path:** <http://localhost:3000/category/addCategory>

**Payload:**

{

"title": "FROZEN",

"description": "Frozen category products is ready to cook"

}

**Response:**

{

"success": true,

"message": "category has been added successfully"

}

**2) Get all categories**

**Method:** get

**Path:** <http://localhost:3000/category/getAllCategory>

**Response:**

[

{

"id": 4,

"title": "FROZEN",

"description": "Frozen category products is ready to cook",

"createdDate": "2019-07-31T07:12:51.000Z"

},

{

"id": 5,

"title": "RAW",

"description": "Raw category are fresh products directly obtaining from land",

"createdDate": "2019-07-31T12:21:20.000Z"

}

]

3) **Get category by category id**

**Method:** get(params)

**Path:** <http://localhost:3000/category/getAllCategoryByCategoryId/4>

**Response:**

[

{

"id": 4,

"title": "FROZEN",

"description": "Frozen category products is ready to cook",

"createdDate": "2019-07-31T07:12:51.000Z"

}

]

4) **Update category**

**Method:** put

**Path:** <http://localhost:3000/category/updateCategoryByCategoryId>

**Payload:**

{

"title": "raw",

"description": "raw category products directly get from land fertile",

"id": 4

}

**Response:**

{

"success": true,

"message": "Category has been updated successfully"

}

5) **If delete one category related products should delete.**

**Method:** delete

**Path:** <http://localhost:3000/category/deleteCategoryByCategoryId>

**Payload:**

{

"categoryId": 1

}

**Response:**

{

"deleted products are": [

{

"id": 12,

"categoryId": 4,

"name": "SWEET CORN",

"description": "SWEET CORN ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:42:48.000Z"

},

{

"id": 13,

"categoryId": 4,

"name": "GARLIC",

"description": "GARLIC ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:43:05.000Z"

},

{

"id": 14,

"categoryId": 4,

"name": "GREEN PEAS",

"description": "GREEN PEAS ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:43:32.000Z"

}

]

}

**Products API’s**

**1) Add products**

**Method:** post

**Path:** <http://localhost:3000/product/addProducts>

**Payload:**

{

"categoryId": 5,

"name": "Tomato",

"description": "tomato ",

"uom": "KG",

"preservationMethods" : "keep in refrigerators",

"gst": "1"

}

**Response:**

{

"success": true,

"message": "products has been added successfully"

}

**2) Get all products**

**Method:** get

**Path:** <http://localhost:3000/product/getAllProducts>

**Response:**

[

{

"id": 7,

"categoryId": 5,

"name": "Tomato",

"description": "tomato ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:39:53.000Z"

},

{

"id": 8,

"categoryId": 5,

"name": "POTATO",

"description": "POTATO ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:41:35.000Z"

},

{

"id": 9,

"categoryId": 5,

"name": "GARLIC",

"description": "GARLIC ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:42:01.000Z"

},

{

"id": 10,

"categoryId": 5,

"name": "ONION",

"description": "ONION ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:42:12.000Z"

},

{

"id": 12,

"categoryId": 4,

"name": "SWEET CORN",

"description": "SWEET CORN ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:42:48.000Z"

},

{

"id": 13,

"categoryId": 4,

"name": "GARLIC",

"description": "GARLIC ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:43:05.000Z"

},

{

"id": 14,

"categoryId": 4,

"name": "GREEN PEAS",

"description": "GREEN PEAS ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:43:32.000Z"

}

]

**3) Get all products by product id**

Method: get (params)

Path: <http://localhost:3000/product/getProductsByProductId/12>

Response:

[

{

"id": 12,

"categoryId": 4,

"name": "SWEET CORN",

"description": "SWEET CORN ",

"uom": "KG",

"preservationMethods": "keep in refrigerators",

"gst": 1,

"isEnabled": 1,

"createdDate": "2019-07-31T12:42:48.000Z"

}

]

**4) Update products**

**Method:** put

**Path:** <http://localhost:3000/product/updateProductsByProductId>

**Payload:**

{

"categoryId": 1,

"name": "Potato",

"description": "Potato ",

"uom": "KG",

"preservationMethods" : "keep in refrigerators",

"gst": "1",

"isEnabled": false,

"id": 1

}

**Response:**

{

"success": true,

"message": "products has been updated successfully"

}