# **REST-API-Restaurant**

Name: Tejas Panchal ([tejaspanchal49@gmail.com](mailto:tejaspanchal49@gmail.com))

| **Technology** | **Version** |
| --- | --- |
| npm | 5.5.5 |
| Node.js | 6.11.5 |
| MongoDB | 3.4.9 |
| Redis | 4.0.8 |

## **Steps to Run the Project**

1. Install NodeJs, MongoDB, Redis.
2. Clone or download the github repository - <https://github.com/panchaltej/REST-API-Restaurant>.
3. Start mongodb and redis servers.
4. Install node modules

npm install

1. Start node server

npm start

1. To run unit tests

npm test

## **Database Schema**

Collection: restaurants

{

"\_id" : <restaurant\_id>,

"name" : <restaurant\_name>,

"location" : <restaurant\_location>,

"cuisines": [<cuisine1>, <cuisine2>],

"ratings": <restaurant\_ratings>,

"phone\_number": <restaurant\_contact\_number>,

"avg\_cost": <restaurant\_cost>,

"menus":[

{

"menu\_type": <menu type>,

"menu\_items": [

{

"item\_name": <item name>,

"item\_price": <item price>,

"desc": <item description

}

]

}

]

}

## **REST API Endpoints**

BaseURL : localhost:3001/

| **Endpoint** | **HTTP Verb** | **Functionality** | **Example** | **Success Response Code** |
| --- | --- | --- | --- | --- |
| /restaurants | POST | Create new restaurant in the database. | localhost:3001/restaurants | 201 |
| /restaurants/:restaurant\_id | GET | Get Restaurant details with the generated restaurant ID | localhost:3001/restaurants/:restaurant\_id | 200 |
| /restaurants/:restaurantID | DELETE | Delete Restaurant with the given restaurant ID | localhost:3001/restaurants/:restaurant\_id | 200 |
| /restaurants/:restaurant\_id/menu | POST | Add menu to the restaurant with name. | localhost:3001/restaurants/:restaurant\_id/menu | 201 |
| /restaurants/:restaurantID/menu | GET | Get all menus of a restaurant | localhost:3001/restaurants/:restaurant\_id/menu | 200 |
| /restaurants/:restaurant\_id/menu/:menutype | GET | Get a menu detail of a restaurant with menutype. | localhost:3001/restaurants/:restaurant\_id/menu/drinks | 200 |
| /restaurants/:restaurant\_id/menu/:menutype | DELETE | Delete menu of a restaurant with menutype. | localhost:3001/restaurants/:restaurant\_id/menu/drinks | 200 |
| /restaurants/:restaurant\_id/menu/:menutype/menuitem | POST | Add new menu item to the menu. | localhost:3001/restaurants/:restaurant\_id/menu/drinks/menuitem | 201 |
| /restaurants/:restaurant\_id/menu/:menutype/menuitem | GET | Get all menu items of the menu | localhost:3001/restaurants/:restaurant\_id/menu/drinks/menuitem | 200 |
| /restaurants/:restaurant\_id/menu/:menutype/menuitem/:menuitem | DELETE | Delete menuitem with menuid | localhost:3001/restaurants/:restaurant\_id/menu/drinks/menuitem/5a797f026d887752a071a6ae | 200 |

## **REST API Requests**

1. Create restaurant

POST http://localhost:3001/restaurants

201 Created

{

"name" : "McDonald's",

"location": "San Jose",

"cuisines": ["Snacks", "Burgers"],

"ratings": "4",

"phone\_number": "+1-408-429-0627",

"avg\_cost": "20",

"menus":[

{

"menu\_type": "sandwiches",

"menu\_items": [

{

"item\_name": "Veggie Burger",

"item\_price": "9",

"desc": "Veg Sandwich"

},

{

"item\_name": "Cheeseburger",

"item\_price": "12",

"desc": "Sandwich"

}

]

},

{

"menu\_type": "drinks",

"menu\_items": [

{

"item\_name": "Coke",

"item\_price": "3",

"desc": "Soft Drink"

},

{

"item\_name": "Sprite",

"item\_price": "3",

"desc": "Soft Drink"

}

]

}

]

}

1. Get restaurant details

GET http://localhost:3001/restaurants/<restaurant-id>

GET http://localhost:3001/restaurants/5a7d201056489f10b4f5c174

200 OK

[

{

"\_id": "5a7d201056489f10b4f5c174",

"name": "McDonald's",

"location": "San Jose",

"cuisines": [

"Snacks",

"Burgers"

],

"ratings": "4",

"phone\_number": "+1-408-429-0627",

"avg\_cost": "20",

"menus": [

{

"menu\_type": "sandwiches",

"menu\_items": [

{

"item\_name": "Veggie Burger",

"item\_price": "9",

"desc": "Veg Sandwich"

},

{

"item\_name": "Cheeseburger",

"item\_price": "12",

"desc": "Sandwich"

}

]

},

{

"menu\_type": "drinks",

"menu\_items": [

{

"item\_name": "Coke",

"item\_price": "3",

"desc": "Soft Drink"

},

{

"item\_name": "Sprite",

"item\_price": "3",

"desc": "Soft Drink"

}

]

}

]

}

]

1. Delete restaurant

DELETE http://localhost:3001/restaurants/<restaurant-id>

DELETE http://localhost:3001/restaurants/5a7d201056489f10b4f5c174

200 No Content

1. Add Menu

POST http://localhost:3001/<restaurant-id>/menu

POST http://localhost:3001/restaurants/5a7d201056489f10b4f5c174/menu

201 Created

{

"menus":{

"menu\_type": "lunch",

"menu\_items": [

{

"item\_name": "demo1",

"item\_price": "20",

"desc": "demo"

},

{

"item\_name": "demo2",

"item\_price": "30",

"desc": "demo2"

}

]

}

}

1. Get Menu

GET http://localhost:3001/restaurants/<restaurant-id>/menu/<menu\_type>

GET http://localhost:3001/restaurants/5a7d201056489f10b4f5c174/menu/drinks

200 OK

{

"\_id": "5a7d201056489f10b4f5c174",

"menus": [

{

"menu\_type": "drinks",

"menu\_items": [

{

"item\_name": "Coke",

"item\_price": "3",

"desc": "Soft Drink"

},

{

"item\_name": "Sprite",

"item\_price": "3",

"desc": "Soft Drink"

}

]

}

]

}

1. Delete Menu

DELETE http://localhost:3001/restaurants/<restaurant-id>/menu/<menu\_type>

DELETE http://localhost:3001/restaurants/5a7d201056489f10b4f5c174/menu/drinks

200 No Content

1. Add MenuItem

POST http://localhost:3001/restaurants/<restaurant-id>/menu/<menu\_type>/menuitem

POST http://localhost:3001/restaurants/5a7d201056489f10b4f5c174/menu/drinks/menuitem

201 Created

{

"item\_name": "demo4",

"item\_price": "80",

"desc": "Test"

}

1. Get MenuItem

GET http://localhost:3001/restaurants/<restaurant-id>/menu/<menu\_type>/menuitem

GET http://localhost:3001/restaurants/5a7d201056489f10b4f5c174/menu/drinks/menuitem

200 OK

{

"menu\_items": [

{

"item\_name": "Coke",

"item\_price": "3",

"desc": "Soft Drink"

},

{

"item\_name": "Sprite",

"item\_price": "3",

"desc": "Soft Drink"

}

]

}

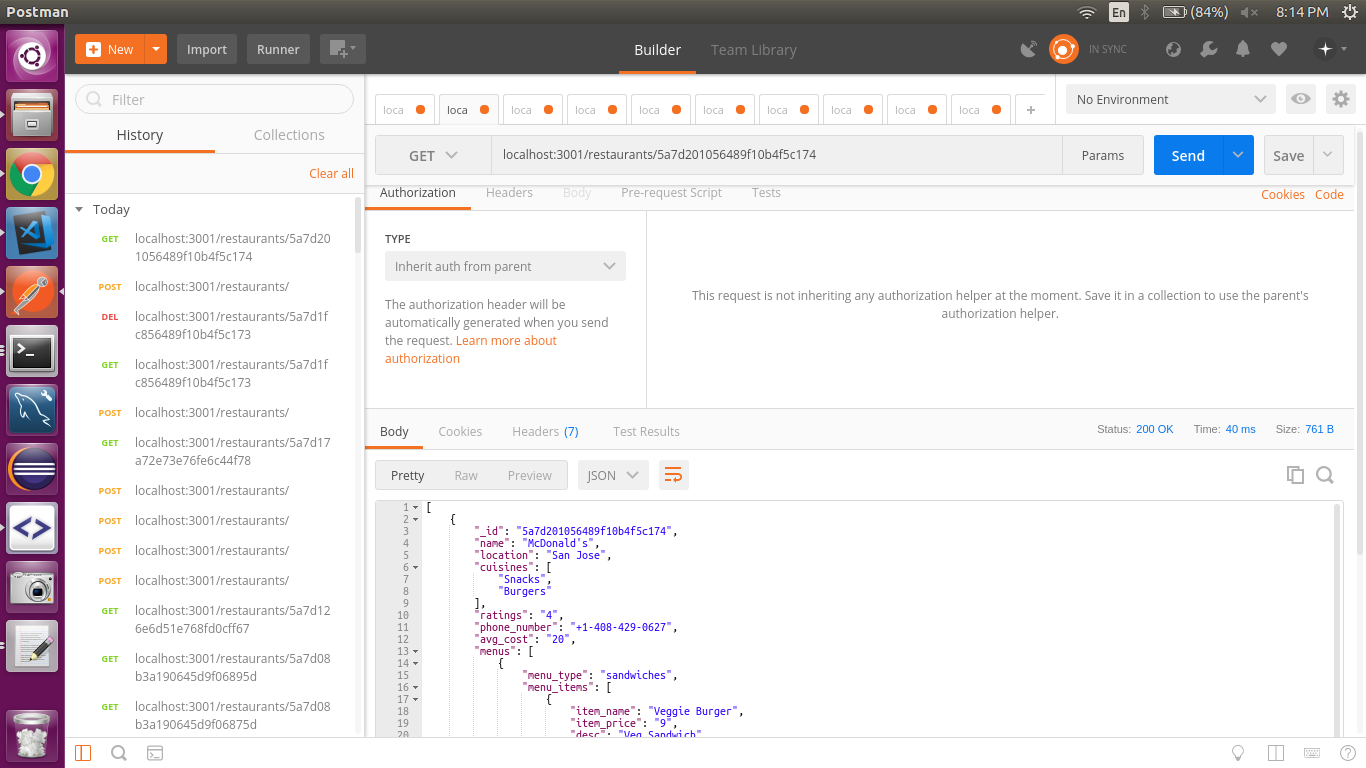
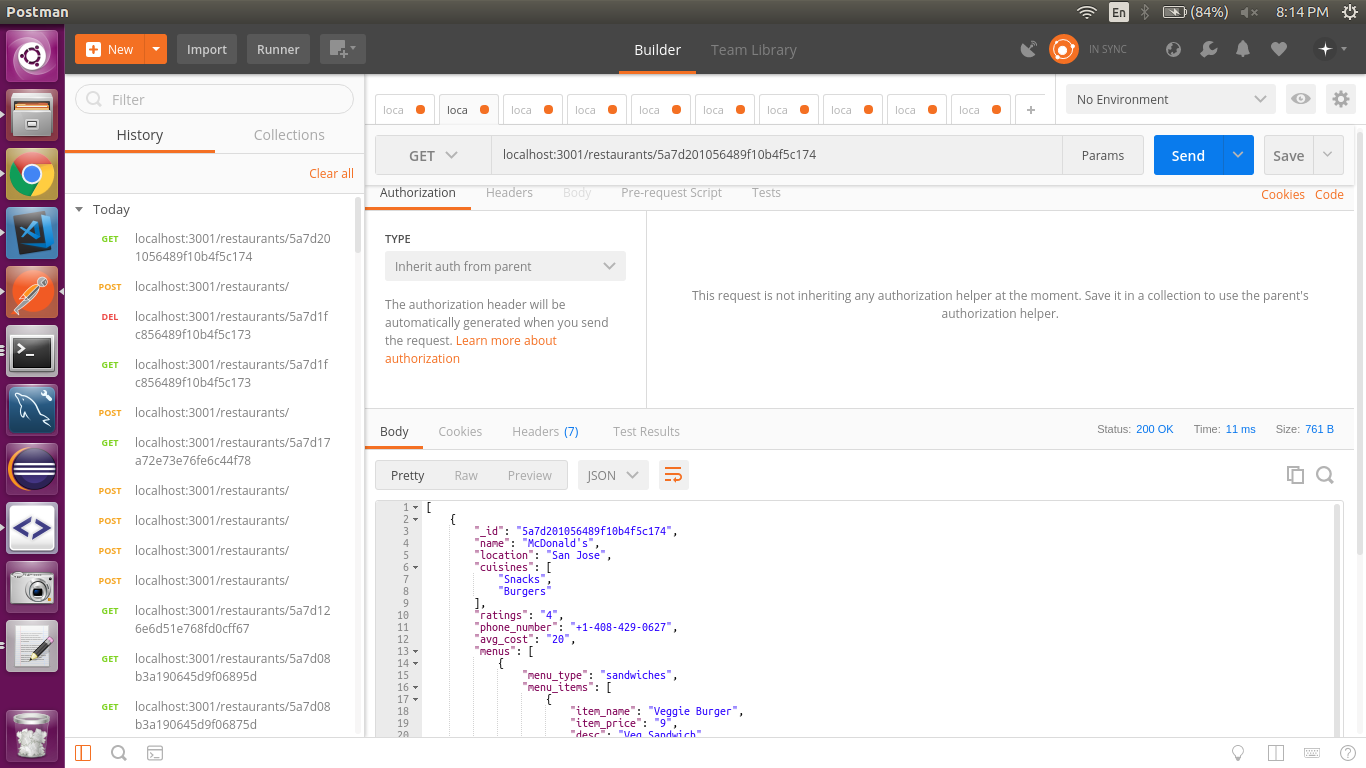
1. Delete MenuItem

DELETE http://localhost:3001/restaurants/<restaurant-id>/menu/<menu\_type>/menuitem/<item\_name>

DELETE http://localhost:3001/restaurants/5a7d201056489f10b4f5c174/menu/drinks/menuitem/coke

200 No Content

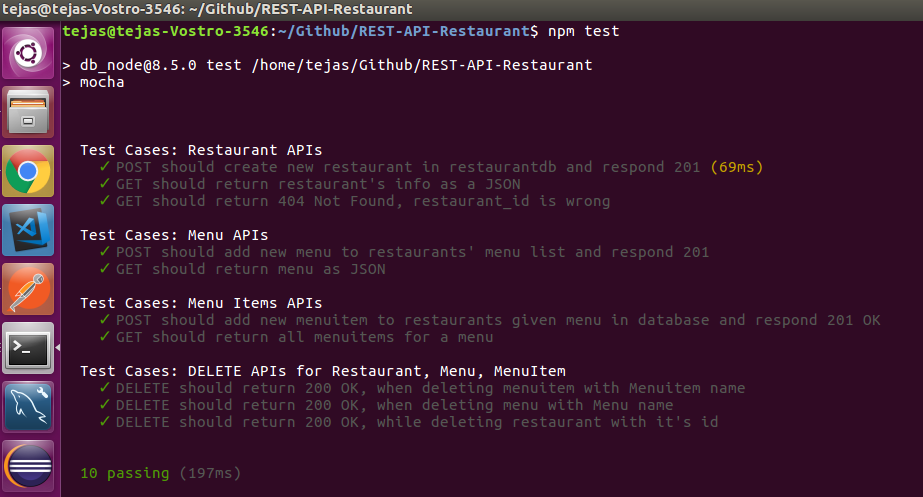
## **Redis for Caching**

* I have attached some screenshots displaying the response time for both the scenario.
  1. MongoDB Access : 40ms
  2. Redis Cache : 11ms
* Fetching from mongoDB
* Fetching from Redis 

## **Unit Tests**

Run following command to run unit tests

npm test



## **Handling large data of restaurants all over USA**

To handle millions of requests simultaneously and to scale the accordingly, we can host the server on AWS ec2 instances with load balancing. Each instances serves all the API requests and the AWS ELB balances load on all the instances.

To overcome data consistency problem, we can host mongodb database on a separate instance serving the requests being attanded to on different instances.

