

Restaurant Suggestion and Management Application.

- Document provided had a lot of typos in url and field names, so i fixed that to have uniform fields. (like: restaurant had a lot of typos). Due to this the request payload in documnet will not work unless typos are not fixed there too. I have provided fixed payload in this document below.
- I am using a Embedded H2 database that is setup using provided schema.sql and data.sql files on application startup.
- This is a maven project.
- DataBase code is added. Used Spring Data JPA.
- Unit tests added

1) add new restaurants.(Done)

POST <http://localhost:8080/api/v1/restaurants>

Request Body:

```
{
  "restaurant_name": "RestaurantC",
  "restaurant_cusines": [
    "cusineX",
    "cusineY"
  ],
  "restaurant_chefs": [
    {
      "chef_name": "chefG",
      "chef_salary": 4000,
      "chef_cusines": [
        "cusineX",
        "cusineR"
      ]
    },
    {
      "chef_name": "chefH",
      "chef_salary": 2000,
      "chef_cusines": [
        "cusineY",
        "cusineZ"
      ]
    }
  ],
  "restaurant_rating": 6,
  "restaurant_city": "brampton",
  "restaurant_province": "ON"
}
```

Response :

Valid Request Status code : 201

```
{
  "restaurant_id": 3,
  "create_date": "2021-04-30T05:39:31.6745509"
}
```

Incase of Invalid Request body (fields are null/empty/not present)

Status code : 400

```
{
  "error": "Bad Request",
  "code": "com.restaurant.service.request.body.invalid",
  "message": "Validation error in request body",
  "details": [
    {
      "error": "restaurant_name",
      "message": "must not be blank"
    },
    {
      "error": "restaurant_rating",
      "message": "must be less than or equal to 10"
    },
    {
      "error": "restaurant_city",
      "message": "must not be blank"
    }
  ]
}
```

Incase of Internal Server Error

Status code : 500

2)update details in existing restaurants.(Done)

PUT <http://localhost:8080/api/v1/restaurants/3>

Request Body:

```
{
  "restaurant_name": "RestaurantC",
  "restaurant_cusines": [
    "cuisineY"
  ],
  "restaurant_chefs": [
    {
      "chef_name": "chefH",
      "chef_salary": 2000,
      "chef_cusines": [
        "cuisineY",
        "cuisineZ"
      ]
    }
  ]
}
```

```
    ]
  }
],
"restaurant_rating": 4,
"restaurant_city": "missi",
"restaurant_province": "ON"
}
```

Response :

Valid Request Status code : 200

```
{
  "restaurant_id": 3,
  "update_date": "2021-04-30T05:39:31.6745509"
}
```

Incase of Invalid Request body (fields are null/empty/not present)

Status code : 400

```
{
  "error": "Bad Request",
  "code": "com.restaurant.service.request.body.invalid",
  "message": "Validation error in request body",
  "details": [
    {
      "error": "restaurant_name",
      "message": "must not be blank"
    },
    {
      "error": "restaurant_rating",
      "message": "must be less than or equal to 10"
    },
    {
      "error": "restaurant_city",
      "message": "must not be blank"
    }
  ]
}
```

Incase of Internal Server Error

Status code : 500

Incase of restaurant_id not found in database

Status code : 404

```
{
  "error": "Not Found",
  "code": "com.restaurant.service.entity.notfound",
  "message": "Restaurant with id : 4 doesn't exist.",
}
```

```
"details": []  
}
```

3) Delete existing restaurants.(Done)

DELETE <http://localhost:8080/api/v1/restaurants/3>

Response :

Valid Request Status code : 200

```
{  
  "restaurant_id": 3,  
  "delete_date": "2021-04-30T05:39:31.6745509"  
}
```

Incase of Internal Server Error

Status code : 500

Incase of restaurant_id not found in database

Status code : 404

```
{  
  "error": "Not Found",  
  "code": "com.restaurantservice.entity.notfound",  
  "message": "Restaurant with id : 4 doesn't exist.",  
  "details": []  
}
```

4) Search restaurant based on restarunt_name , restarunt_id, restarunt_city, restarunt_province,restarunt_rating , restarunt_cusines available (Done)

1. GET <http://localhost:8080/api/v1/restaurants/1>
2. GET http://localhost:8080/api/v1/restaurants?restaurant_name=RestaurantB
3. GET http://localhost:8080/api/v1/restaurants?restaurant_city=brampton
4. GET http://localhost:8080/api/v1/restaurants?restaurant_province=ON
5. GET http://localhost:8080/api/v1/restaurants?restaurant_rating=8
6. GET http://localhost:8080/api/v1/restaurants?restaurant_cuisine=cuisineA

Response for above api's will be a list a restaurants based on query parameters

4Bouns) you'll have extra credit if you can provide
search based on multiple parameters like being able to

search on rating and cuisine at the same time. (Done)

GET [http://localhost:8080/api/v1/restaurants?
restaurant_cuisine=cuisineA&restaurant_rating=5](http://localhost:8080/api/v1/restaurants?restaurant_cuisine=cuisineA&restaurant_rating=5)

The service will handle any combinations of QueryParameter provided. Parameter Should have only one value and not multiple values

5) Search chefs based on restarunt_id he is working in , cuisines he can make (Done)

1. GET http://localhost:8080/api/v1/chefs?restaurant_id=1
2. GET http://localhost:8080/api/v1/chefs?restaurant_id=1&cuisine=cuisineA
3. GET http://localhost:8080/api/v1/chefs?restaurant_id=1&name=chefA
4. GET <http://localhost:8080/api/v1/chefs?name=chefB>
5. GET <http://localhost:8080/api/v1/chefs?cuisine=cuisineC>

Response for above api's will be a list a chefs based on query parameters

Status code : 200

Response:

```
[
  {
    "chef_name": "chefA",
    "chef_salary": 2000,
    "chef_cusines": [
      "cuisineA",
      "cuisineB"
    ]
  },
  {
    "chef_name": "chefB",
    "chef_salary": 2000,
    "chef_cusines": [
      "cuisineC",
      "cuisineD"
    ]
  },
  {
    "chef_name": "chefC",
    "chef_salary": 4000,
    "chef_cusines": [
      "cuisineF",
      "cuisineG"
    ]
  },
  {
    "chef_name": "chefD",
    "chef_salary": 4000,
```

```
    "chef_cusines": [  
      "cuisineE"  
    ]  
  }  
]
```