



**Question - 1**  
**Restaurant Collection**

SCORE: 50 points

.NET   Easy   .NET Core 3.0   Back-End Development

**Restaurant Collection API**

A company is launching a new service that provides details of restaurants spread across multiple cities. A 'restaurant collection' needs to be created, and as part of this challenge, you are required to come up with a service to maintain this restaurant collection.

As step 1, create a service that supports REST APIs for creating, deleting, and updating restaurant data. Restaurant data will have details such as restaurant name, city name, the estimated cost for dining, user rating, etc. A few more APIs required would be to fetch the restaurant details, the restaurants in a particular city, sort the restaurants as per the user rating,, etc. A detailed explanation about the APIs and data is given below.

Each restaurant object is a JSON object with the following keys -

- *city* - City in which the restaurant is located. [STRING]
- *name* - Name of the restaurant. [STRING]
- *estimatedCost* - the estimated cost for 2 people at the restaurant. [INTEGER]
- *averageRating* - Average rating of the restaurant. [STRING] (A decimal number in string format).
- *votes* - Total reviews. [INTEGER].
- *id* - Unique id of the restaurant as generated by the system. [INTEGER]

▼ **Example**

```
{
  "city": "Miami",
  "name": "Byg Company",
  "estimatedCost": 1600,
  "averageRating": "4.9",
  "votes": "16203",
  "id": "1"
}
```

▼ **APIs**

The following APIs need to be implemented:

1. Adding a new restaurant - POST request should be created to add a new restaurant. The API endpoint would be /restaurant. The request body contains the details of the restaurant. HTTP response should be 201.

2. Updating restaurant rating - averageRating and votes can be updated by PUT request to endpoint /restaurant/{id}. The request body would contain averageRating and votes.
3. Getting all restaurants - GET request to endpoint /restaurant should return the entire list of restaurants. The HTTP response code should be 200.
4. Getting all restaurants in a particular city - GET request to endpoint /restaurant/query?city={city} should return the entire list of restaurants in that city. The HTTP response code should be 200.
5. Getting restaurant by id - GET request to endpoint /restaurant/query?id={id} should return the details of the restaurant with this unique id. The HTTP response code should be 200.
6. Deleting restaurant by id - DELETE request to endpoint /restaurant/{id} should delete the corresponding restaurant.
7. Sort the restaurants according to rating - GET request to endpoint /restaurant/sort should return restaurants sorted as per their rating.