Development Notes:

I have developed this “RestaurantReviews.sln” solution in MS Visual Studio 2015. It uses Web API 2, and attribute routing. Note that I added using NuGet several packages, one of them being the EntityFramework.6.1.3. So I expect that as you load this solution and try to build it, MS Visual Studio will try to restore these NuGet packages for you in a top “packages” folder.

As I finished this coding exercise I thought that certainly more routes like these could be developed:

1. restaurant/{restaurantApiId:int}/restaurantreview (GET say)
2. reviewer/{reviewerApiId:int}/restaurantreview (GET say)

I hope what I have is sufficient enough.

Testing Notes:

To test this “RestaurantReviews.sln” solution you will need to first go to the top “Database” project in the solution and run the Dev.publish.xml file. There specify your MS SQL Server database server, and ideally a database name that doesn’t already exist, like the default RestaurantReviewsDev. In this “Database” project’s properties I specified as the “Target platform” (lowest version) the “SQL Server 2005”.

Additionally for testing you will need to use a Rest API client. I setup the “Postman” one from: <https://www.getpostman.com/postman>.

Also note that I setup the “RestaurantReviews.Api” project to use the local port 62770, and I expect this port to be used when you run this solution. So in my below URLs I always use this 62770 local port. Finally you will need to right-click on the “RestaurantReviews.Api” project and select the “Set as StartUp Project” popup menu item.

These are some of my tests using Postman:

* **“Post a restaurant that is not in the database”:**

1. URL: http://localhost:62770/api/v1/restaurant
2. Verb: Select “POST”.
3. Headers: Make sure you add this header:

Key: Content-Type

Value: application/json

1. Body: set each one individually and then click on the “Send” button.

{

"Name": "Olive Garden #138",

"AddressLine1": "123 Main St.",

"AddressLine2": "Store 5",

"City": "Butler",

"StateProvince": "PA",

"PostalCode": "37646-3452",

"Country": "USA"

}

{

"Name": "Taco Bell #724",

"AddressLine1": "476 Rochester Rd.",

"City": "Butler",

"StateProvince": "OH",

"PostalCode": "48383-4937",

"Country": "USA"

}

{

"Name": "Grecian Delight",

"AddressLine1": "487 Freedom Rd.",

"AddressLine2": "Suite 2",

"City": "Middletown",

"StateProvince": "OH",

"PostalCode": "97381-9476",

"Country": "USA"

}

Each time you should get back the “RestaurantApiId” integer value.

If you want you can test error handling by trying to click the “Send” button again on one of them. In that case you should get back this JSON object, the “ResponseApiModel” C# class:

{

"Result": 5,

"ParameterName": null,

"ErrorMessage": "This address is already used by this restaurant: RestaurantApiId: 1 Name:'Olive Garden #1'"

}

* **“Get a list of restaurants by city”:**

With this test I pass only a city name:

1. URL: http://localhost:62770/api/v1/restaurant/?city=butler
2. Verb: Select “GET”.
3. Click on the “Send” button.

You should get back a JSON array of objects, the defined “IList<RestaurantApiModel>” C# class.

You can repeat the above test passing additionally the state, too:

http://localhost:62770/api/v1/restaurant/?city=butler&stateProvince=oh

If you want you can also test error handling by not passing the “city” parameter above.

* **“Post a review for a restaurant”:**

1. URL: http://localhost:62770/api/v1/restaurantreview
2. Verb: Select “POST”.
3. Headers: Make sure you add this header:

Key: Content-Type

Value: application/json

1. Body: set each one individually and then click on the “Send” button.

{

"RestaurantApiId": 2,

"ReviewerEmail": "sgpapakirk@yahoo.com",

"NumberOfStars": 4,

"Text": "A very good place. They are fast and friendly."

}

{

"RestaurantApiId": 2,

"ReviewerEmail": "sgpapakirk@yahoo.com",

"NumberOfStars": 3,

"Text": "So and so today. They missed some of my items."

}

{

"RestaurantApiId": 3,

"ReviewerEmail": "sgpapakirk@gmail.com",

"NumberOfStars": 5,

"Text": "This restaurant makes the best Greek food!"

}

Each time you should get back the “RestaurantReviewApiId” integer value.

If you want you can test error handling also by passing invalid “RestaurantApiId” values above.

* **“Get of a list of reviews by user”:**

1. URL: http://localhost:62770/api/v1/restaurantreview/?reviewerEmail=sgpapakirk@yahoo.com
2. Verb: Select “GET”.
3. Click on the “Send” button.

You should get back a JSON array of objects, the defined “IList<RestaurantReviewApiModel>” C# class.

If you want you can test also error handling by not passing the “reviewerEmail” parameter above.

* **“Delete a review”:**

1. URL: http://localhost:62770/api/v1/restaurantreview/2
2. Verb: Select “DELETE”.
3. Click on the “Send” button.

You should get back a Boolean (true or false) to indicate if it succeeded.

If you want you can test also error handling by passing an invalid number at the end like say 0, or 1000 (the

restaurantReviewApiId).