Coding Challenge

Dawson Geist

4/18/2022

Tools Used for testing: A Google Chrome Application “Advanced Rest Client”

Notes:

4/13

Opened the project in VS19 and saw a bunch of red squiggles. -> Built the project which installed a bunch of things and the squiggles went away.

Opened the readme and identify the tasks required. read how to start the program and opened cmd line

navigated to the repository:

C:\Users\Dawso\Desktop\Coding Challenges\code-challenge>

used cmd:

dotnet run

-> The framework 'Microsoft.NETCore.App', version '2.1.0' was not found.

installed 2.1.0, reran the above command

->

warn: Microsoft.AspNetCore.Server.Kestrel[0]

Unable to bind to http://localhost:8080 on the IPv4 loopback interface: 'Error -4092 EACCES permission denied'.

warn: Microsoft.AspNetCore.Server.Kestrel[0]

Unable to bind to http://localhost:8080 on the IPv6 loopback interface: 'Error -4092 EACCES permission denied'.

crit: Microsoft.AspNetCore.Server.Kestrel[0]

Unable to start Kestrel.

Googled the error code

After using Stackoverflow I ran the following command

> netsh interface ipv4 show excludedportrange protocol=tcp

->

Protocol tcp Port Exclusion Ranges

Start Port End Port

---------- --------

5700 5700

8080 8080

8884 8884

\* - Administered port exclusions.

I am assuming we need to attach to 8080 but it is excluded....

I see some things about adjusting the dynamic range using

> netsh int ipv[46] set dynamic tcp start=49152 num=16384

(https://superuser.com/questions/1437780/how-to-fix-listen-eacces-permission-denied-on-any-port) -> (https://github.com/docker/for-win/issues/3171#issuecomment-554587817)

but Ive never used netsh and im getting issues with the command. Looking to remove 8080 from reserved so that I can minimize changes to my computer....

looking at the previous commands I see now the personal was using abbreviations...

ran

>netsh interface ipv4 set dynamic tcp start=49152 num=16384

->The requested operation requires elevation (Run as administrator).

ah.

one sec.

now rechecking the ports

>

->

Protocol tcp Port Exclusion Ranges

Start Port End Port

---------- --------

5700 5700

8080 8080

8884 8884

\* - Administered port exclusions.

ok option 2: removing 8080 from the ...

restarting the pc to let changes happen.. one sec

> cd C:\Users\Dawso\Desktop\Coding Challenges\code-challenge

>>dotnet run

->

Hosting environment: Development

Content root path: C:\Users\Dawso\Desktop\Coding Challenges\code-challenge

Now listening on: http://localhost:8080

Application started. Press Ctrl+C to shut down.

Now what. Maybe try running the tests

>cd C:\Users\Dawso\Desktop\Coding Challenges\code-challenge.Tests

>>dotnet run

->

nothing on cmd lets check directory

looks the same, wish I had created a mercurial repository.

wait let me start it then check my browser... this might of just been the server

no webpage found on port 8080 which is what i can get either using VS or cmd.

im gonna make http requests from the program.cs main function and use that as my test cases.

need to be able to do get post what ever the rest of them are

#4 (https://docs.microsoft.com/en-us/dotnet/csharp/tutorials/console-webapiclient)

…………… ………….. ……….. !

Alright we have had success. I am using a chrome extension to test the API and I have successfully broke the request... but the server is working I just got to learn json

4/14

Watching Tutorial ( https://www.youtube.com/watch?v=ZXdFisA\_hOY )

Definitions:

Web API Controllers: Handles incoming HTTP requests and send response back to the caller

Web Repositories: Handle the storage of Models/Entities in the system

Model: The collection of data models for an API

http://localhost:8080/api/reportstructure/03aa1462-ffa9-4978-901b-7c001562cf6f

" works " ... I have a controller routing the http request to my controller which then uses regualr programming logic to gather employee ID from the database and add it to a list. Only issue is that when grabbing a user from the data base their direct reports list is null

(NOTE A): There is something called Lazy loading and Eager loading, what I’ve learned is that using the include keyword allows you to specify the property you want fully? Loaded, that is, it loads the current entity and all it’s referenced entities

After I used Include I was able to consistently get the employee entity with the fully detailed direct report attribute.

(NOTE B): Now all I had to do was use a function that would fully load an employee and all of their direct reports. So I used a loop in a function that would recursively call itself for every direct report in that entity’s list. I passed an integer variable by REFERENCE so that I could count how many entities were loaded.

Tests:

TEST 1: Ringo Starr

URL: <http://localhost:8080/api/reportstructure/03aa1462-ffa9-4978-901b-7c001562cf6f>

Expected Output:

{

"employee": {

"employeeId": "03aa1462-ffa9-4978-901b-7c001562cf6f",

"firstName": "Ringo",

"lastName": "Starr",

"position": "Developer V",

"department": "Engineering",

"directReports": [

  {

"employeeId": "62c1084e-6e34-4630-93fd-9153afb65309",

"firstName": "Pete",

"lastName": "Best",

"position": "Developer II",

"department": "Engineering",

"directReports": [],

},

  {

"employeeId": "c0c2293d-16bd-4603-8e08-638a9d18b22c",

"firstName": "George",

"lastName": "Harrison",

"position": "Developer III",

"department": "Engineering",

"directReports": [],

}

],

},

"numberOfReports": 2

}

TEST: PASS

TEST 2: John Lennon

URL: <http://localhost:8080/api/reportstructure/16a596ae-edd3-4847-99fe-c4518e82c86f>

Expected Output:

{

"employee": {

"employeeId": "16a596ae-edd3-4847-99fe-c4518e82c86f",

"firstName": "John",

"lastName": "Lennon",

"position": "Development Manager",

"department": "Engineering",

"directReports": [

  {

"employeeId": "b7839309-3348-463b-a7e3-5de1c168beb3",

"firstName": "Paul",

"lastName": "McCartney",

"position": "Developer I",

"department": "Engineering",

"directReports": [],

},

  {

"employeeId": "03aa1462-ffa9-4978-901b-7c001562cf6f",

"firstName": "Ringo",

"lastName": "Starr",

"position": "Developer V",

"department": "Engineering",

"directReports": [

  {

"employeeId": "62c1084e-6e34-4630-93fd-9153afb65309",

"firstName": "Pete",

"lastName": "Best",

"position": "Developer II",

"department": "Engineering",

"directReports": [],

},

  {

"employeeId": "c0c2293d-16bd-4603-8e08-638a9d18b22c",

"firstName": "George",

"lastName": "Harrison",

"position": "Developer III",

"department": "Engineering",

"directReports": [],

}

],

}

],

},

"numberOfReports": 4

}

TEST: PASS

End Tests:

4/16:

On Saturday I started the second part of the challenge which was half rinse and repeat of the first part and an introduction to persistence/databases. Admittedly I need to continue learning about Databases so I can understand PK/FK, The definition of Relational, and every part of implementation so I can at least describe what it is that I am trying to do…

So here were my issues and how I resolved them.

1. Was setting up the database/context. I learned that context holds the changes to the database and is what is used to update the database. (Red flags for synchronization issues)

Resolution: (I assumed since we are having our compensation entity’s persist that they would need their own separate database). Being new to the web api development world I started with the EmployeeContext class and copied over what was there into a Compensation class and hoped for the best. This lead to my second error.

1. I need to establish a primary key for the database

Thoughts: Why does the employee context not require a primary key/ already has a primary key? -> I believe it is because It has explicitly defined an ID property in the Employee Class.

1. The key that is needed and described by the challenge set is inside the Employee entity.

Thoughts: I believe this is now turning into an issue of properly defining and implementing a relationship between these two databases. I need help to understand this in a reasonable time frame.

What I did to overcome this: After trying to establish a Foreign key in one of the databases, which came after using has one with one keywords to describe the relationship between the two entities/ table columns? I cheated and added a primary key Attribute to the compensation class and (NOTE C): declared that as my primary key and added Code so that the Compensation ID matched the Employee ID whenever a new compensation was created. (There was also some additions made that would create an employee if the employee ID was not found in the employee repository… as a way to maintain the 1-1 database relationship I struggled with earlier)

Finally I replaced some existing code with conditional expressions.

4/17

Tests:

TEST: Create Compensation with Direct Reports

URL:

http://localhost:8080/api/compensation

JSON BODY:

{

"employee":

{

"employeeId": "Dawson-Palmer-Geist",

"firstName": null,

"lastName": null,

"position": null,

"department": null,

"directReports": [

{

"employeeId": "AAAA",

"firstName": "Paul",

"lastName": "RUDD",

"position": "Developer I",

"department": "Engineering",

"directReports": null

},

{

"employeeId": "BBBB",

"firstName": "Ringo",

"lastName": "WINGO",

"position": "Developer V",

"department": "Engineering",

"directReports": null

}

]

},

"salary": 90000,

"effectiveDate": "2022-18-4Z"

}

TEST: PART A PASS

TEST: Get Compensation (ID values change so grab them from the response body)

URL: e94977e6-d355-4d8d-93b4-fea547faa35b

Expected Output:

{

"compensationId": "e94977e6-d355-4d8d-93b4-fea547faa35b",

"employee": {

"employeeId": "e94977e6-d355-4d8d-93b4-fea547faa35b",

"firstName": null,

"lastName": null,

"position": null,

"department": null,

"directReports": null

},

"salary": 90000,

"effectiveDate": "2022-18-4Z"

}

TEST: PASS

TEST: Get Reporting structure for Employee with the Compensation

URL: <http://localhost:8080/api/reportstructure/e94977e6-d355-4d8d-93b4-fea547faa35b>

Expected Response:

{

"employee": {

"employeeId": "e94977e6-d355-4d8d-93b4-fea547faa35b",

"firstName": null,

"lastName": null,

"position": null,

"department": null,

"directReports": [

  {

"employeeId": "0629df45-f2bd-4d0c-8ac5-3347d4118e54",

"firstName": "Paul",

"lastName": "RUDD",

"position": "Developer I",

"department": "Engineering",

"directReports": [],

},

  {

"employeeId": "3557367f-7d48-43ac-a41d-c39d114ee038",

"firstName": "Ringo",

"lastName": "WINGO",

"position": "Developer V",

"department": "Engineering",

"directReports": [],

}

],

},

"numberOfReports": 2

}

TEST: PASS

TEST: Get Each Employee From the Compensation’s Employee’s Direct Reports

URL(s):

http://localhost:8080/api/employee/0629df45-f2bd-4d0c-8ac5-3347d4118e54,

http://localhost:8080/api/employee/3557367f-7d48-43ac-a41d-c39d114ee038

Expected Output A:

{

"employeeId": "0629df45-f2bd-4d0c-8ac5-3347d4118e54",

"firstName": "Paul",

"lastName": "RUDD",

"position": "Developer I",

"department": "Engineering",

"directReports": [],

}

Expected Output B:

{

"employeeId": "3557367f-7d48-43ac-a41d-c39d114ee038",

"firstName": "Ringo",

"lastName": "WINGO",

"position": "Developer V",

"department": "Engineering",

"directReports": [],

}

TEST A-B: PASS

End Tests

Final Notes:

This was a really fun coding challenge. It involved technology I have never worked with before and I think the skills I learned are incredibly valuable. Moving forward I am going to revisit this project and take what I learned and use it to open an API for an application I have been working on as a side project. I definitely plan on taking a LinkedIn learning course on databases and EF core if they are available.

Thank you for taking the time to review my code. I would appreciate any and all feedback and critiques on what I’ve provided.

Thank you for the opportunity to learn more about the company and the technology it uses.