**Breakpoints**

|  |  |  |  |
| --- | --- | --- | --- |
| **Error Type:** | **Error ID:** | **Explanation:** | **Screenshots of technique:**  **(**PLEASE ZOOM IN AND AVOID EYE STRAIN**)** |
| 1  CLOSED | Logic | I have an ASP.Net Core 2.2 web application with the backend made in Entity Framework Core from this tutorial. So am creating a new database Data isn’t being seeded to it even though I have explicitly made all the tables and data in my SchoolContext class. I am using this tutorial to complete it. <https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/complex-data-model?view=aspnetcore-2.2#update-the-database>  My version of the app is up here. <https://github.com/LayersOfAbstraction/Contoso-University-web-app-template/tree/master/ContosoUniversity>  **Steps I took to reproduce the problem.**   1. Add migration `dotnet ef migrations add ComplexDataModel` 2. Changed the connection string to add the new database, save then build the project. 3. Enter new command `dotnet ef database update`. 4. We run the app and DbInitializer.Initialize should be called. But it’s not being called at all. I can see it being skipped in the debugger.   https://github.com/LayersOfAbstraction/Contoso-University-web-app-template/tree/master/ContosoUniversity  **Expected behavior**  Data is seeded to the SQL server database and is bound to the ASP.NET Core controls.  **Actual behavior**  SQL server database is not showing the data. I can not see any databases being filled.  **Environment data**  `.NET Core SDK (reflecting any global.json):  Version: 2.2  Runtime Environment:  OS Name: Windows  OS Version: 6.1.7601  OS Platform: Windows  RID: win7-x64 `  **Fix**  Boom. I fixed the problem. Decided to look at the completed version of the project. I declared a new web host builder that didn’t have an IWebHost with the name of “host”.  And because the using statement in the Main() method of my program file was expecting one it skipped the using statement that allowed the database to be seeded. I just commented out line 20. Solved. |  |
| 2  CLOSED | Logic | I am trying to assign a course from a check box group to an Instructor. Line 226 in this picture is not executed. So the database is not assigning the CourseID value to the selected Instructor.  The problem is in the View actually. So the checkbox array was outside the class “row” so it couldn’t set the checkbox value to checked or unchecked. That being said there is a formatting problem now. So the form checkboxes aren’t perfectly aligned and appear to be trapped in the dimensions of the “row” class. |  |
| 3  OPEN | Logic | I am attempting to use a search string parameter to search through two navigational property collections but I am not getting any output when I search through the navigational properties of the User and the Job.  From the breakpoints we will get the search string but it does not result in output. |  |
| 4  OPEN | Logic | This localhost page can’t be found No web page was found for the web address: **http://localhost:5001/Account/AccessDenied?ReturnUrl=%2FUsers**  HTTP ERROR 404  We are getting this error when we are trying to authorize the user with the name [jnash486@gmail.com](mailto:jnash486@gmail.com) after we have logged in using our google account and the tenant namespace is in this format which was recommended by Auth0 in the documentation. <https://schemas.dev-dgdfgfdgf324.au.auth0.com>  I have read the quickstart guide here <https://auth0.com/docs/quickstart/webapp/aspnet-core/03-authorization> and are struggling with the last step. I believe I am not using the correct tenant address format in both my rule and my app because this was happening when I used complete sample app with a rule.  Else the Authorization attribute would fire up but isn’t even firing up and stepping through our Index method.  So I am not sure what is working and not working and must be able to effectively test isolated components. So we have to first ensure my rule is functional on the server-end in terms of connecting to our user name and tenant.  We can do that by actually debugging our rule on the Auth0 dashboard to check if the logic is correct.  We can then implement that username and tenant address in the client application. |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |