|  |  |  |  |
| --- | --- | --- | --- |
| **Error Type:** | **Error ID:** | **Explanation:** | **Screenshots of technique:**  **(**PLEASE ZOOM IN AND AVOID EYE STRAIN**)** |
| Logic  (CLOSED) | 1 | I cannot cycle to the next 3 PaginatedList values in a HTML table in ASP.NET Core. So I am using this tutorial for it. <https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/sort-filter-page?view=aspnetcore-2.2>  The debugger keeps saying in PaginatedList that the count is gone up to 9 when I try to render the values by clicking the “Next” button. Except I can’t see those values rendering. It stays on the same page.  Also I’m noticing the “CurrentSort” and CurrentFilter values are null which the view needs to:   1. Provide the current sort order to the view so especially while paging and… 2. Maintain the filter settings during paging.   Here is my code for the files I had to alter/create in the tutorial in order for paging to work.  The solution was to revert to a previous backup and copy and paste the code from the tutorial instead of hand writing it. |  |
| Logic  (CLOSED) | 2 | Failure to transfer data into new SQL Server DB with ASP.NET Core and Entity Framework Core.  **Steps to reproduce**  I have an ASP.Net Core 2.2 web application with the backend made in Entity Framework Core from this tutorial. <https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/migrations?view=aspnetcore-2.2>  To summarise the steps.   1. Download the project in the link. 2. Go to `appsettings.json`. 3. Change database name in connection string to ` ContosoUniversity2` 4. Save your changes and build the project. 5. Ensure no instance of the program is running and use CLI (Command Line Interface) or PMC (Package Management Console) and navigate to the project folder and enter this command. `dotnet ef migrations add InitialCreate`   Note, tables and columns are generated here.  **Expected behaviour**  Entity Framework migrates data from previous database named ContosoUniversity1 to ContosoUniversity2 including all the columns and tables. It has worked for the author in Visual Studio 2019.  **Actual behaviour**  I am only seeing the creation of all tables and columns. There is no data other than in \_\_EFMigrationsHistory which has a single row for the migration history. I get this output from .NET CLI.  `Build started...  Build succeeded.  info: Microsoft.EntityFrameworkCore.Infrastructure[10403]  Entity Framework Core 2.2.6-servicing-10079 initialized 'SchoolContext' us  ing provider 'Microsoft.EntityFrameworkCore.SqlServer' with options: None  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (666ms) [Parameters=[], CommandType='Text', CommandTime  out='60']  CREATE DATABASE [ContosoUniversity2];  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (161ms) [Parameters=[], CommandType='Text', CommandTime  out='60']  IF SERVERPROPERTY('EngineEdition') <> 5  BEGIN  ALTER DATABASE [ContosoUniversity2] SET READ\_COMMITTED\_SNAPSHOT ON;  END;  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (18ms) [Parameters=[], CommandType='Text', CommandTimeo  ut='30']  CREATE TABLE [\_\_EFMigrationsHistory] (  [MigrationId] nvarchar(150) NOT NULL,  [ProductVersion] nvarchar(32) NOT NULL,  CONSTRAINT [PK\_\_\_EFMigrationsHistory] PRIMARY KEY ([MigrationId])  );  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (3ms) [Parameters=[], CommandType='Text', CommandTimeou  t='30']  SELECT OBJECT\_ID(N'[\_\_EFMigrationsHistory]');  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (1ms) [Parameters=[], CommandType='Text', CommandTimeou  t='30']  SELECT [MigrationId], [ProductVersion]  FROM [\_\_EFMigrationsHistory]  ORDER BY [MigrationId];  info: Microsoft.EntityFrameworkCore.Migrations[20402]  Applying migration '20191227004521\_InitialCreate'.  Applying migration '20191227004521\_InitialCreate'.  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeou  t='30']  CREATE TABLE [Course] (  [CourseID] int NOT NULL,  [Title] nvarchar(max) NULL,  [Credits] int NOT NULL,  CONSTRAINT [PK\_Course] PRIMARY KEY ([CourseID])  );  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeou  t='30']  CREATE TABLE [Student] (  [ID] int NOT NULL IDENTITY,  [LastName] nvarchar(max) NULL,  [FirstMidName] nvarchar(max) NULL,  [EnrollmentDate] datetime2 NOT NULL,  CONSTRAINT [PK\_Student] PRIMARY KEY ([ID])  );  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (3ms) [Parameters=[], CommandType='Text', CommandTimeou  t='30']  CREATE TABLE [Enrollment] (  [EnrollmentID] int NOT NULL IDENTITY,  [CourseID] int NOT NULL,  [StudentID] int NOT NULL,  [Grade] int NULL,  CONSTRAINT [PK\_Enrollment] PRIMARY KEY ([EnrollmentID]),  CONSTRAINT [FK\_Enrollment\_Course\_CourseID] FOREIGN KEY ([CourseID]) RE  FERENCES [Course] ([CourseID]) ON DELETE CASCADE,  CONSTRAINT [FK\_Enrollment\_Student\_StudentID] FOREIGN KEY ([StudentID])  REFERENCES [Student] ([ID]) ON DELETE CASCADE  );  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (1ms) [Parameters=[], CommandType='Text', CommandTimeou  t='30']  CREATE INDEX [IX\_Enrollment\_CourseID] ON [Enrollment] ([CourseID]);  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (1ms) [Parameters=[], CommandType='Text', CommandTimeou  t='30']  CREATE INDEX [IX\_Enrollment\_StudentID] ON [Enrollment] ([StudentID]);  info: Microsoft.EntityFrameworkCore.Database.Command[20101]  Executed DbCommand (2ms) [Parameters=[], CommandType='Text', CommandTimeou  t='30']  INSERT INTO [\_\_EFMigrationsHistory] ([MigrationId], [ProductVersion])  VALUES (N'20191227004521\_InitialCreate', N'2.2.6-servicing-10079');  Done.`  **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 `  SOLUTION: The tutorial I was using was confusing because <https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/migrations?view=aspnetcore-2.2> there is an image of data in the database even though we are creating a new one from scratch. So I am just going to continue to use the original DB for upcoming tutorials. |  |
| 3  CLOSED | Logic | Had a problem using a migration. When I run command “dotnet ef database update” this was the error. There is already an object named 'Student' in the database.  The way past it was to comment out the code for the “Up” migration method in “20191229082704\_MaxLengthOnNames.cs”. |  |
| 4  CLOSED | Logic | Unable to render scrollable image background homepage  I am using the "ASP.NET Core Web Application" template in (2.2 MVC).  I attempted to use a w3schools HOWTO sample to render scrollable image background to my home page which is written in asp.net core razor cshtml file.  <https://www.w3schools.com/howto/howto_css_bg_change_scroll.asp>  The text box in the centre worked so I got rid of it. But I cannot render any image from my file in wwwroot\lib\bootstrap\dist\css\bootstrap.css to the homepage view and the image path appears to be correct as intellisense found the path to all images.  The difficulty is that both these html selectors “html” and “body” are combined into one in the tutorial and in my page both selectors are separated and I don’t know which one should have the height property set to 100% as it is in the tutorial.  I tried setting both to 100% with no effect or resolve of my issue.  Here is my summary code for the bootstrap.css file containing the scrollable image background.  Turned out that the code from the layout page, he said will make the attributes invalid. To solve it he suggested that I write the needed css under “wwwroot\css” instead of overwriting the default css file. I did this and it worked except I had to copy and paste all the code from the default css file in order to keep the navbar. | html {  font-family: sans-serif;  line-height: 1.15;  -webkit-text-size-adjust: 100%;  -webkit-tap-highlight-color: rgba(0, 0, 0, 0);  }  \* {  box-sizing: border-box;  }  .bg-image {  /\* Full height \*/  height: 50%;  /\* Center and scale the image nicely \*/  background-position: center;  background-repeat: no-repeat;  background-size: cover;  }  /\* Images used \*/  .img1 {background-image: url("../../images/pitcher1.jpg");}  .img2 {background-image: url("../../images/pitcher2.jpg");}  .img3 {background-image: url("../../images/pitcher3.jpg");}  .img4 {background-image: url("../../images/pitcher4.jpg");}  article, aside, figcaption, figure, footer, header, hgroup, main, nav, section {  display: block;  }  body {  margin: 0;  font-family: "Lato", -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto, "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol";  font-size: 1rem;  font-weight: 400;  line-height: 1.5;  color: #EBEBEB;  text-align: left;  background-color: #2B3E50;  } |
| 5  CLOSED | Logic | By default when I generate the views for registration I find that because EFC doesn’t know which field the user needs to make sense out of UserID it uses the UserContactEmail field.  I wanted something like full name but I could only do First or last name so what I did was make this property in the User model.  public string UserFullname => string.Format("{0} {1}", UserFirstName, UserLastName);  It allowed me to then just change all the values in the controller and the views from UserContactEmail to UserFullName. I didn’t even need to recreate the database or migrate this property as a data column in the database. |  |
| 6  CLOSED | Syntax | Cannot implicitly convert type 'System.Linq.IOrderedQueryable<Pitcher.Models.Registration>' to 'Microsoft.EntityFrameworkCore.Query.IIncludableQueryable  The way around this first was to use IQueryable in place of the var keyword which I have highlighted in the code snippet.  That would give us another error for each switch case for example…  'IQueryable' does not contain a definition for 'OrderByDescending' and no accessible extension method 'OrderByDescending' accepting a first argument of type 'IQueryable' could be found (are you missing a using directive or an assembly reference?) [Pitcher]  …To fix that up right next to the keyword IQueryable I put the word <Registration>. Done.  IQueryable <Registration> registrations = \_context.Registrations.Include(r => r.Job).Include(r => r.User); |  |
| 7  CLOSED | Logic | Unable to calculate total users by job title.  **Steps to reproduce**  <https://docs.microsoft.com/en-us/aspnet/core/data/ef-mvc/sort-filter-page?view=aspnetcore-2.2>   1. Create the view model called “ProjectTotalsGroup.cs” 2. Modify home controller and add this.   private readonly TeamContext \_context;          public HomeController(TeamContext context)          {              \_context = context;          }          public IActionResult Index()          {              return View();          }          public IActionResult Privacy()          {              return View();          }          [ResponseCache(Duration = 0, Location = ResponseCacheLocation.None, NoStore = true)]          public IActionResult Error()          {              return View(new ErrorViewModel { RequestId = Activity.Current?.Id ?? HttpContext.TraceIdentifier });          }            //TO DO: GROUP UserCount by JobName.          public async Task<ActionResult> About()          {              IQueryable<ProjectTotalsGroup> data =                  from jobs in \_context.Jobs                  group jobs by jobs.JobTitle into projectGroup                  select new ProjectTotalsGroup()                  {                      JobName = projectGroup.Key,                      UserCount = projectGroup.Count()                  };              return View(await data.AsNoTracking().ToListAsync());          }   1. Create About view.   @model IEnumerable<Pitcher.Models.TeamViewModels.ProjectTotalsGroup>  @{      ViewData["Title"] = "Project Body Statistics";  }  <h2>Student Body Statistics</h2>  <table>      <tr>          <th>              Job Name          </th>          <th>              Staff total          </th>      </tr>        @foreach (var item in Model)      {          <tr>              <td>                  @Html.DisplayFor(modelItem => item.JobName)              </td>              <td>                  @item.UserCount              </td>          </tr>      }  </table>  **Expected output**  I see total users in each Job shown on screen.  **Actual output**  Unable to calculate total users by job title.  **Algorithm to solve my problem**  //PROBLEM: TO GROUP userCount by jobTitle.  //GET Users  //GET Registrations  //SELECT all Users where ID = tblRegistration.JobID  //GROUP Users by tblRegistration.JobID  **SUCCESS!!!**  Well that algorithm solved most of the problem but I realized I could just group the registrations by registrations.Job.JobTitle. As in access the navigational property that leads to the entity set “Job” and then the entity called JobTitle. Simple. |  |