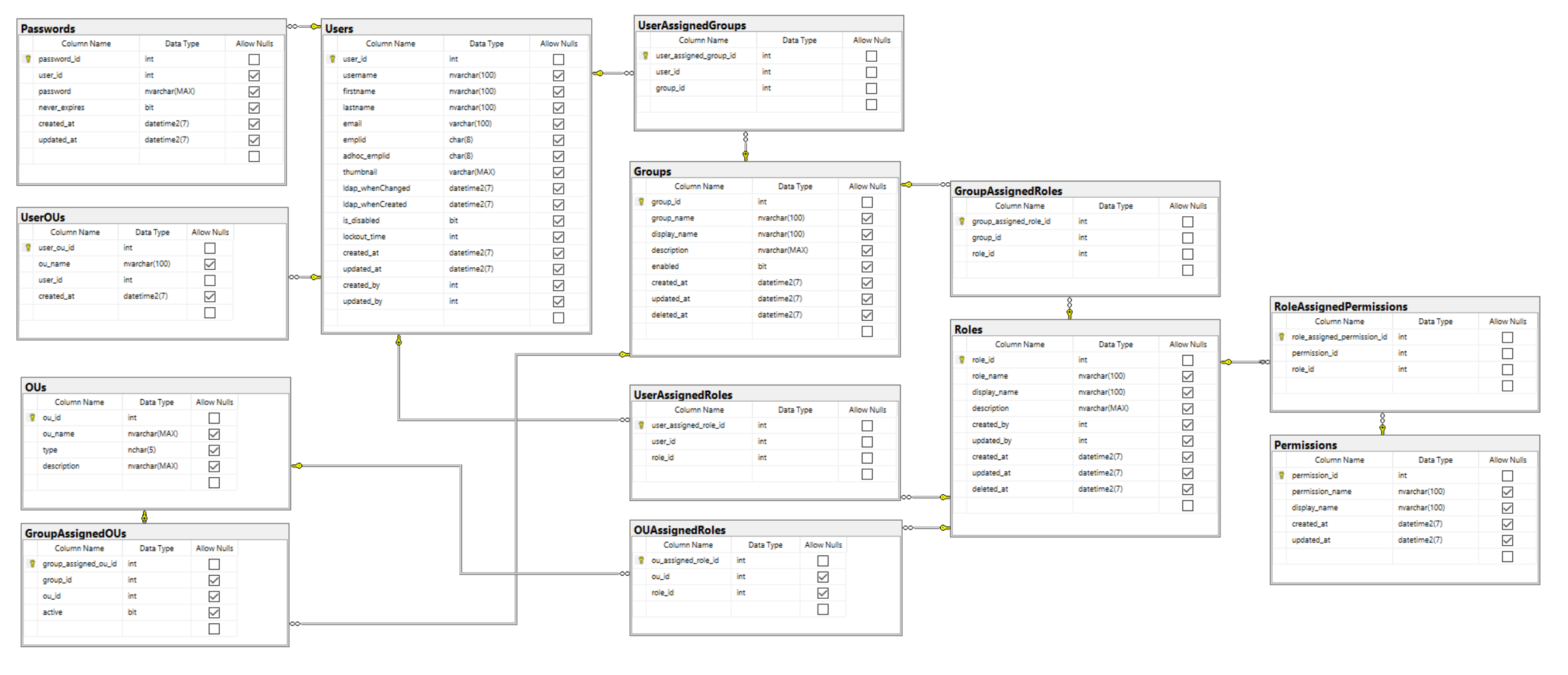
# **DotNetCore Start Up Project Procedures**

|  |  |  |
| --- | --- | --- |
| **STEPS** | **MVC 5/6** | **DDD 5/6** |
| Project Creation | * Create a new Web API solution & project with same name | * Create a new Web API solution & project with different names * Reference Clean Architecture Diagram Below |
| Structure Set Up | Solution Name   * SRC folder   + Project Name: Core / Web API / OData   + Project Name: Angular / Flutter / Redux & React / Vue JS     - Manage all session for impersonation here * TEST folder   + Unit   + Integration | Solution Name   * SRC folder   + Project Name: Core   + Project Name: Infrastructure   + Project Name: Web API / OData   + Project Name: Angular / Flutter / Redux & React / Vue JS     - Manage all session for impersonation here * TEST folder   + Core.Unit   + Core.Integration   + Infrastructure.Unit   + Infrastructure.Integration   + WebAPI.Unit   + WebAPI.Integration |
| Library References Extra Instructions | * Aspnetcore5 only use 5.x * Aspnetcore6 only use 6.x * Aspnetcore7 only use 7.x * Aspnetcore8 only use 8.x | |
| Library References | * System.Linq * System.Linq.Expressions   System.Linq.Queryable | |
| NUGET Package Extra Instructions |  | * Install all packages in Infrastructure project |
| NUGET Package Installs EF Core Support | * Microsoft.EntityFrameworkCore * Microsoft.EntityFrameworkCore.Tools * Microsoft.EntityFrameworkCore.SqlServer * Microsoft.EntityFrameworkCore.Design   Microsoft.VisualStudio.Web.CodeGeneration.Design | |
| NUGET Package Installs ODATA support Extra Instructions | * <https://docs.microsoft.com/en-us/odata/> * Startup.cs - public void ConfigureServices(IServiceCollection services)   services.AddControllers().AddOData(opt =>  {  opt.EnableQueryFeatures();  });   * Use this on top of controller 🡺 [EnableQuery()] | |
| NUGET Package Installs ODATA support | * Microsoft.AspNetCore.OData v8.0 or greater * Microsoft.AspNetCore.OData.Extensions (maybe optional) * Microsoft.OData.Core (maybe optional) * Microsoft.OData.Edm (maybe optional) * Microsoft.AspNet.OData (maybe optional) * Microsoft.OData.ModelBuilder (maybe optional) | |
| NUGET Package Installs CRON support  www.quartz-scheduler.net | * Quartz.AspNetCore (required) * Quartz.Plugins (required) * Quartz.Plugins.TimeZoneConverter (required) * Quartz.Serialization.Json (required) | |
| Quartz Scheduler Installation | * Instructions: create a class inheriting IJob * Copy and paste the following code into STARTUP to initialize   services.AddQuartz(q =>  {  // handy when part of cluster or you want to otherwise identify multiple schedulers  q.SchedulerId = "Scheduler-Core";  // you can control whether job interruption happens for running jobs when scheduler is shutting down  q.InterruptJobsOnShutdown = true;  // when QuartzHostedServiceOptions.WaitForJobsToComplete = true or scheduler.Shutdown(waitForJobsToComplete: true)  q.InterruptJobsOnShutdownWithWait = true;  // we can change from the default of 1  q.MaxBatchSize = 5;  // we take this from appsettings.json, just show it's possible  // q.SchedulerName = "Quartz ASP.NET Core Sample Scheduler";  // this is default configuration if you don't alter it  q.UseMicrosoftDependencyInjectionJobFactory();  // these are the defaults  q.UseSimpleTypeLoader();  q.UseInMemoryStore();  q.UseDefaultThreadPool(maxConcurrency: 10);  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* 8-15-2022 my first job -confirmed to work 8-15-2022 @ 347pm  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* https://www.quartz-scheduler.net/documentation/quartz-2.x/tutorial/crontriggers.html#example-cron-expressions  \* WithCronSchedule function params: Seconds,Minutes,Hours,Day-of-Month,Month,Day-of-Week,Year (optional field)  \* Individual sub-expressions can contain ranges and/or lists. For example, the day of week field in the previous (which reads "WED") example could be replaces with "MON-FRI", "MON, WED, FRI", or even "MON-WED,SAT".  \* All of the fields have a set of valid values that can be specified. These values should be fairly obvious - such as the numbers 0 to 59 for seconds and minutes, and the values 0 to 23 for hours. Day-of-Month can be any value 0-31, but you need to be careful about how many days are in a given month! Months can be specified as values between 0 and 11, or by using the strings JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV and DEC. Days-of-Week can be specified as vaules between 1 and 7 (1 = Sunday) or by using the strings SUN, MON, TUE, WED, THU, FRI and SAT  \* The '/' character can be used to specify increments to values. For example, if you put '0/15' in the Minutes field, it means 'every 15 minutes, starting at minute zero'. If you used '3/20' in the Minutes field, it would mean 'every 20 minutes during the hour, starting at minute three' - or in other words it is the same as specifying '3,23,43' in the Minutes field.  \* The '?' character is allowed for the day-of-month and day-of-week fields. It is used to specify "no specific value". This is useful when you need to specify something in one of the two fields, but not the other. See the examples below (and CronTrigger API documentation) for clarification.  \* The 'L' character is allowed for the day-of-month and day-of-week fields. This character is short-hand for "last", but it has different meaning in each of the two fields. For example, the value "L" in the day-of-month field means "the last day of the month" - day 31 for January, day 28 for February on non-leap years. If used in the day-of-week field by itself, it simply means "7" or "SAT". But if used in the day-of-week field after another value, it means "the last xxx day of the month" - for example "6L" or "FRIL" both mean "the last friday of the month". When using the 'L' option, it is important not to specify lists, or ranges of values, as you'll get confusing results.  \* Wild-cards (the '' character) can be used to say "every" possible value of this field. Therefore the '' character in the "Month" field of the previous example simply means "every month".  \* A '\*' in the Day-Of-Week field would obviously mean "every day of the week".  \*/  JobKey jobKeyEmailTest = new JobKey("Quartz Job - Email Test");  q.ScheduleJob<QuartzJob>(trigger => trigger  .ForJob(jobKeyEmailTest)  .WithCronSchedule("0 0 0 ? \* TUE,THU")  );  });  // ASP.NET Core hosting  services.AddQuartzServer(options =>  {  // when shutting down we want jobs to complete gracefully  options.WaitForJobsToComplete = true;  }); | |
| ADHOC Authentication & Authorization Schema | * Refer to **Users & Groups Permission Authorization Database Schema Design –** OU related tables are optional and required to have a process to prepopulate from LDAP | |
| Email Communications | * Use whatever is currently used to process the emails whether it be an email service or something pertaining to the current project only | |
| Package Manage Console  Extra Instructions |  | * Install EF scaffolding in Infrastructure but move all Models to Core |
| Package Manage Console (initialize) | * Scaffold-DbContext "Server=[SQL SERVER];Database=[DB NAME];Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -f * Scaffold-DbContext "Server=BWT-2F-BCHIU;Database=SeedSolution;Trusted\_Connection=True;" Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models -f | |
| Generate EF Controllers | * Within the Controllers folder | * Within the Controllers folder in the infrastructure project |
| Wire Up Services Location | * Aspnetcore5 and below use Startup.cs | * Aspnetcore6 and above use Program.cs |
| Wire Up Services Example | * Right under **services.AddControllers();** add **services.AddDbContext<[DatabaseContext]>();** add **services.AddHttpContextAccessor();** * To Initialize * private readonly IHttpContextAccessor \_httpContextAccessor;   public EmployeeRepository( IHttpContextAccessor httpContextAccessor)  {  \_httpContextAccessor = httpContextAccessor;  }   * To Use * \_httpContextAccessor.HttpContext.User.Identity.Name; | |
| Update launchSettings.json then build and launch each profile to test | {  "$schema": "http://json.schemastore.org/launchsettings.json",  "iisSettings": {  "windowsAuthentication": false,  "anonymousAuthentication": true,  "iisExpress": {  "applicationUrl": "http://localhost:23086",  "sslPort": 44317  }  },  "profiles": {  "IIS Express": {  "commandName": "IISExpress",  "launchBrowser": true,  "launchUrl": "swagger",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  }  },  "MVC5 Swagger": {  "commandName": "Project",  "dotnetRunMessages": "true",  "launchBrowser": true,  "launchUrl": "swagger",  "applicationUrl": "https://localhost:5001;http://localhost:5000",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  }  },  "MVC5 API": {  "commandName": "Project",  "dotnetRunMessages": "true",  "launchBrowser": true,  "launchUrl": "api/UsersAPI",  "applicationUrl": "https://localhost:5001;http://localhost:5000",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Stagging"  }  }  }  } |  |
| TSQL commands to initialize the permissions tables | use [SeedSolution];  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* DROP AND CREATE PERMISSIONS TABLE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -- REMOVES BACK THE PK/FK RELATIONSHIP  ALTER TABLE [dbo].[RoleAssignedPermissions] DROP CONSTRAINT [FK\_RoleAssignedPermissions\_Permissions]  ALTER TABLE [dbo].[Permissions] DROP CONSTRAINT [PK\_Permissions]  -- DROPS TABLE TO RESET INDEX, UNLESS I CAN FIND A WAY TO RESET INDEX WITHOUT DROPPING TABLE  DROP TABLE IF EXISTS [Permissions]  -- CREATES TABLE  CREATE TABLE [dbo].[Permissions](  [permission\_id] [int] IDENTITY(1,1) NOT NULL,  [table\_name] [nvarchar](100) NULL,  [permission\_name] [nvarchar](100) NULL,  [display\_name] [nvarchar](100) NULL,  [created\_at] [datetime2](7) NULL,  [updated\_at] [datetime2](7) NULL,  CONSTRAINT [PK\_Permissions] PRIMARY KEY CLUSTERED  (  [permission\_id] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]  ) ON [PRIMARY]  GO  -- ADDS BACK THE PK/FK RELATIONSHIP  ALTER TABLE [dbo].[RoleAssignedPermissions] ADD CONSTRAINT [FK\_RoleAssignedPermissions\_Permissions] FOREIGN KEY(permission\_id)  REFERENCES [dbo].[Permissions] (permission\_id)  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INITIALIZE PERMISSIONS TABLE \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -- WILL INITIALIZE PERMISSIONS FOR ALL TABLES WITH READ, WRITE, UPDATE, DELETE  if object\_id('#tmp1') is not null  drop table #tmp1 else drop table #tmp1  SELECT \* INTO #tmp1  FROM  (select distinct  tbl.TABLE\_NAME as 'table\_name',  tbl.TABLE\_NAME + '\_Read' as 'code' ,  tbl.TABLE\_NAME + ' Read' as 'display'  FROM INFORMATION\_SCHEMA.COLUMNS as tbl  WHERE tbl.TABLE\_NAME != 'sysdiagrams') as a;  INSERT INTO [dbo].[Permissions]([table\_name],[permission\_name],[display\_name],[created\_at])  SELECT t.table\_name,t.code, t.display, CURRENT\_TIMESTAMP  FROM #tmp1 as t  if object\_id('#tmp2') is not null  drop table #tmp2 else drop table #tmp2  SELECT \* INTO #tmp2  FROM  (select distinct  tbl.TABLE\_NAME as 'table\_name',  tbl.TABLE\_NAME + '\_Write' as 'code',  tbl.TABLE\_NAME + ' Write' as 'display'  FROM INFORMATION\_SCHEMA.COLUMNS as tbl  WHERE tbl.TABLE\_NAME != 'sysdiagrams') as b  INSERT INTO [dbo].[Permissions]([table\_name],[permission\_name],[display\_name],[created\_at])  SELECT t.table\_name,t.code, t.display, CURRENT\_TIMESTAMP  FROM #tmp2 as t  if object\_id('#tmp3') is not null  drop table #tmp3 else drop table #tmp3  SELECT \* INTO #tmp3  FROM  (select distinct  tbl.TABLE\_NAME as 'table\_name',  tbl.TABLE\_NAME +'\_Update' as 'code',  tbl.TABLE\_NAME +' Update' as 'display'  FROM INFORMATION\_SCHEMA.COLUMNS as tbl  WHERE tbl.TABLE\_NAME != 'sysdiagrams') as c  INSERT INTO [dbo].[Permissions]([table\_name],[permission\_name],[display\_name],[created\_at])  SELECT t.table\_name,t.code, t.display, CURRENT\_TIMESTAMP  FROM #tmp3 as t  if object\_id('#tmp4') is not null  drop table #tmp4 else drop table #tmp4  SELECT \* INTO #tmp4  FROM  (select distinct  tbl.TABLE\_NAME as 'table\_name',  tbl.TABLE\_NAME +'\_Delete' as 'code',  tbl.TABLE\_NAME +' Delete' as 'display'  FROM INFORMATION\_SCHEMA.COLUMNS as tbl  WHERE tbl.TABLE\_NAME != 'sysdiagrams') as d  INSERT INTO [dbo].[Permissions]([table\_name],[permission\_name],[display\_name],[created\_at])  SELECT t.table\_name,t.code, t.display, CURRENT\_TIMESTAMP  FROM #tmp4 as t  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADD SPECIAL PERMISSIONS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -- ADD WHATEVER EXTRA PERMISSIONS THAT ARE NOT REFERENCE ANY TABLE  INSERT INTO [dbo].[Permissions]([permission\_name], [display\_name], [created\_at])  VALUES  ('AdministrativeControlPanel','Administrative Control Panel',CURRENT\_TIMESTAMP)  ,('PrimaryNavigation','Primary Navigation',CURRENT\_TIMESTAMP)  ,('SecondaryNavigation','Secondary Navigation',CURRENT\_TIMESTAMP)    /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADD SPECIAL PERMISSIONS, EACH PLANT GROUP WILL HAVE THEIR CORRESPONDING PLANT PERMISSION, ADMIN GROUP WILL HAVE ALL PLANT PERMISSIONS ATTACHED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  ,('WardsIsland','Wards Island',CURRENT\_TIMESTAMP)  ,('NorthRiver','North River',CURRENT\_TIMESTAMP)  ,('HuntsPoints','Hunts Points',CURRENT\_TIMESTAMP)  ,('NewtownCreek','Newtown Creek',CURRENT\_TIMESTAMP)  ,('26thWard','26th Ward',CURRENT\_TIMESTAMP)  ,('ConeyIsland','Coney Island',CURRENT\_TIMESTAMP)  ,('RedHook','Red Hook',CURRENT\_TIMESTAMP)  ,('OwlsHead','Owls Head',CURRENT\_TIMESTAMP)  ,('TallmanIsland','Tallman Island',CURRENT\_TIMESTAMP)  ,('Jamaica','Jamaica',CURRENT\_TIMESTAMP)  ,('BoweryBay','Bowery Bay',CURRENT\_TIMESTAMP)  ,('Rockaway','Rockaway',CURRENT\_TIMESTAMP)  ,('PortRichmond','Port Richmond',CURRENT\_TIMESTAMP)  ,('OakwoodBeach','Oakwood Beach',CURRENT\_TIMESTAMP)  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADD SPECIAL PERMISSIONS, DASHBOARD WIGITS - NOT INITIALIZED YET \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\*  // COLORED BOX WIGITS  ActiveInspections: Active Containers  ContainersRemovalReady: Containers Ready For Removal  ContainersRecentlyRemoved: Containers Recently Removed  DEPContainerRemovalRequests: DEP Internal Removal Requests  ContractorContainerRemovalRequests: Contractor Removal Requests  UnassignedContractorContainerRemovals: Contractor Containers in Transit  UnassignedDepContainerRemovals: DEP Containers in Transit  ConflictedContainers: Conflicted Containers  \*/  ,('Dashboard\_ActiveInspections','Active Containers',CURRENT\_TIMESTAMP)  ,('Dashboard\_ContainersRemovalReady','Containers Ready For Removal',CURRENT\_TIMESTAMP)  ,('Dashboard\_ContainersRecentlyRemoved','Containers Recently Removed',CURRENT\_TIMESTAMP)  ,('Dashboard\_DEPContainerRemovalRequests','DEP Internal Removal Requests',CURRENT\_TIMESTAMP)  ,('Dashboard\_ContractorContainerRemovalRequests','Contractor Removal Requests',CURRENT\_TIMESTAMP)  ,('Dashboard\_UnassignedContractorContainerRemovals','Contractor Containers in Transit',CURRENT\_TIMESTAMP)  ,('Dashboard\_UnassignedDepContainerRemovals','DEP Containers in Transit',CURRENT\_TIMESTAMP)  ,('Dashboard\_ConflictedContainers','Conflicted Containers',CURRENT\_TIMESTAMP)    /\*  //GRAY BOX GRAPH WIGITS  ContainersRemovedPerDriver: Containers Removed Per Driver  \*/  ,('Dashboard\_ContainersRemovedPerDriver','Containers Removed Per Driver',CURRENT\_TIMESTAMP)  /\*  //GRAY BOX WIGITS  ContainerRemovalRequestsPerFacility: Container Removal Requests Per Facility  SystemNotificationsPanel: System Notifications Panel  \*/  ,('Dashboard\_ContainerRemovalRequestsPerFacility','Container Removal Requests Per Facility',CURRENT\_TIMESTAMP)  ,('Dashboard\_SystemNotificationsPanel','System Notifications Panel',CURRENT\_TIMESTAMP)  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INITIALIZE ROLES TABLE - NOT INITIALIZED YET \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  INSERT INTO [dbo].[Roles]  --([role\_name] ,[display\_name] ,[description] ,[created\_by] ,[updated\_by] ,[created\_at] ,[updated\_at] ,[deleted\_at])  ([role\_name] ,[display\_name], [description] ,[created\_at])  --VALUES (<role\_name, nvarchar(100),> ,<display\_name, nvarchar(100),> ,<description, nvarchar(max),> ,<created\_by, int,> ,<updated\_by, int,> ,<created\_at, datetime2(7),> ,<updated\_at, datetime2(7),> ,<deleted\_at, datetime2(7),>)  VALUES  ('Administrator','Administrator',NULL,CURRENT\_TIMESTAMP)  ,('ApplicationSupport','Application Support',NULL,CURRENT\_TIMESTAMP)  ,('Executive','Executive','Commissioner, Deputy Commissioner',CURRENT\_TIMESTAMP)  ,('Manager','Manager', 'Director, Facility Manager, Plant Chief, Division Chief, Chief, Project Manager',CURRENT\_TIMESTAMP)  ,('Supervisor','Supervisor','Senior Section Stationary Engineer Electrical (SSEE), Senior Sewage Treatment Worker (SSTW)',CURRENT\_TIMESTAMP)  ,('Professional','Professional', 'Stationary Engineer Electrical (SEE), Stationary Engineer (SE), Sewage Treatment Worker (STW), Dispatchers',CURRENT\_TIMESTAMP)  ,('Administrative','Administrative', 'Drivers, Administrative Staff, Interns',CURRENT\_TIMESTAMP)  ,('Contractor','Contractor', 'Contractor',CURRENT\_TIMESTAMP)  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* INITIALIZE GROUPS TABLE - NOT INITIALIZED YET \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  INSERT INTO [dbo].[Groups] --([group\_name] ,[display\_name] ,[description] ,[enabled] ,[created\_at] ,[updated\_at] ,[deleted\_at])  ([group\_name] ,[display\_name], [description] ,[created\_at])  -- VALUES (<group\_name, nvarchar(100),> ,<display\_name, nvarchar(100),> ,<description, nvarchar(max),> ,<enabled, bit,> ,<created\_at, datetime2(7),> ,<updated\_at, datetime2(7),> ,<deleted\_at, datetime2(7),>)  VALUES  ('Administrators','Administrators',NULL,CURRENT\_TIMESTAMP)  ,('ApplicationSupports','Application Supports',NULL,CURRENT\_TIMESTAMP)  ,('Executives','Executives','Commissioner, Deputy Commissioner',CURRENT\_TIMESTAMP)  ,('Managers','Managers', 'Director, Facility Manager, Plant Chief, Division Chief, Chief, Project Manager',CURRENT\_TIMESTAMP)  ,('Supervisors','Supervisors','Senior Section Stationary Engineer Electrical (SSEE), Senior Sewage Treatment Worker (SSTW)',CURRENT\_TIMESTAMP)  ,('Professionals','Professionals', 'Stationary Engineer Electrical (SEE), Stationary Engineer (SE), Sewage Treatment Worker (STW), Dispatchers',CURRENT\_TIMESTAMP)  ,('Dispatchers','Dispatchers', 'TBD',CURRENT\_TIMESTAMP)  ,('AdministrativeStaffs','Administrative Staffs', 'This is a restricted group with limited access to adding new entries to a specific area. TBD',CURRENT\_TIMESTAMP)  ,('Drivers','Drivers', 'This is a restricted group with limited access. May just be used for the purpose of tracking their location via mobile device.',CURRENT\_TIMESTAMP)  ,('Contractors','Contractors', 'This is a restricted user with limited access. TBD',CURRENT\_TIMESTAMP)  ,('WardsIsland','Wards Island', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('NorthRiver','North River', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('HuntsPoints','Hunts Points', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('NewtownCreek','Newtown Creek', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('26thWard','26th Ward', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('ConeyIsland','Coney Island', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('RedHook','Red Hook', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('OwlsHead','Owls Head', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('TallmanIsland','Tallman Island', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('Jamaica','Jamaica', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('BoweryBay','Bowery Bay', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('Rockaway','Rockaway', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('PortRichmond','Port Richmond', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  ,('OakwoodBeach','Oakwood Beach', 'Will be able to access this plant.',CURRENT\_TIMESTAMP)  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ADD APPLICATION SPECIFIC PERMISSIONS & OMIT THE ONES ALREADY ENTERED \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  -- GET ALL TABLE THAT ALREADY HAS PERMISSIONS | |

**Users & Groups Permission Authorization Database Schema Design – Simple ( with multiple user type support )**

**Users & Groups Permission Authorization Database Schema Design – Complex ( with LDAP support, with multiple user type support, with group support )**



OUs, UserOUs, GroupAssignedOUs, and OUAssignedRoles are all optional tables. Implementation requires that UserOUs.ou\_name = OUs.ou\_name

# **Clean Architecture with ASP.NETCORE**

*https://www.youtube.com/watch?v=joNTQy-KXiU*

|  |  |
| --- | --- |
| The Core Project (domain model) Minimal dependencies – none of Infrastructure  What Goes in **Core**:   * Interfaces * Entities, Value Objects, Aggregates * Domain Services * Exceptions * Domain Events * Event Handlers * Specifications | The Infrastructure Project All dependencies on out-of-process resources  What Goes in **Infrastructure**:   * Interfaces * Repositories, EF(Core) DbContext, Cached Repositories * Web API Client * File System Accessors, Logging Adapters * Email/SMS sending * System Clock, CRON / Scheduled Tasks * Other Services |
| The Web Project All dependencies on out-of-process resources  What Goes in **Web**:   * Controllers, Views or Razor Pages * ViewModels, ApiModels, BindingModels * Filters, Binders, Tag / HTML Helpers * Other Services, Interfaces | Sharing Between Solutions: Shared Kernel Common Types May Be Shared Between Solutions. Will be referenced by Core project(s). Ideally distributed as **NUGET Packages**.  What Goes in **Shared Kernel**:   * Interface Implementations of Core and Infrastructure * Base Entity, Base Domain Event, Base Specification * Common Exceptions, Common Interfaces * Common Authentication (User Class), Common Dependency Injection, Common Logging, Common Guard Clauses (unknown unknown exceptions) |

# Solution Structure – Clean Architecture

Infrastructure

Integration  
 Test

Functional  
 Test

Web

Shared Kernel  
Create your own NUGET

Unit  
 Test

Core

# Client-Side Logo / Design Schema



**Web Colors Schema:** use DEP logo

<https://coding.tools/hex-to-rgba>

* Primary White and Black
* Primary Green | #019e57 | rgba(1,158,87,1)
* Primary Blue | #00529c | rgba(0,82,156,1)
  + Secondary Red | #9E0248 | rgba(158, 2, 72, 1)
  + Secondary Brown | #9C4900| rgba(156, 73, 0, 1)
  + Secondary Yellow | # ECEA98 | rgba(236,234,152,1)

**Web Layout:** use scavenger / ccase

# Scaffolding with EasyData

Tutorial: <https://www.youtube.com/watch?v=rW-sdkKi9So>

Source code: <https://github.com/KorzhCom/EasyData>

Note: This library is in NUGET as well. It can be configured to detect all models and generate CRUD procedures for them. Includes a very functional and rich DATAGRID.