

# Moving Identity Management Data Access into Data Access Layer

08 Dec 2019

## Goals:

In "ASP.NET CORE 2.0 Web Application",

- To display how access to Identity Management data created in ASP.NET CORE Web Application can be moved to Data Access Layer as a separate project (dll).

## Keys:

Identity Management

Data Access Layer (DAL)

Data access via Data Access Layer

## Assumptions:

Visual Studio is open.

This document contains only how the identity management can be moved to Data Access Layer in a web application solution.

Default Connection for Sql Server will be used as database.

## Scenario:

- Create **new web application** using individual user accounts.
- Create **new dynamic library** as data access layer in web application solution.
- Move/Update folders and files to move identity management data access to Data Access Layer of web application solution.
- Create database (Code First)

## Result:

- Run and register to the application which uses Identity in Data Access Layer.

## Summary of Steps

<b>Part 1: Create Projects.....</b>	<b>3</b>
1. Create web application project → IdexApp.....	3
2. Create project to be used as Data Access Layer(DAL). → IdexApp.DAL.....	3
<b>Part 2: Move and update files in between Projects.....</b>	<b>3</b>
3. Move Data folder from IdexApp to IdexApp.DAL and update file as described below.....	3
4. Create Models folder within Data folder of IdexApp.DAL .....	3
5. Delete default empty Class1.cs file from IdexApp.DAL.....	3
<b>Part 3: Install necessary packages for Identity in IdexApp.DAL project.....</b>	<b>3</b>
6. Open Nuget Package Management for IdexApp.DAL.....	3
7. Build IdexApp.DAL project. It must be successful.....	4
<b>Part 4: Using IdexApp.DAL in IdexApp. ....</b>	<b>4</b>
8. Add IdexApp.DAL into references of IdexApp. ....	4
9. Update files in IdexApp solution. ....	4
10. Optional: Update password options to make it easy to register .....	4
11. Update Views in IdexApp solution. ....	5
12. Create database ready for Identity Management via IdexApp.DAL .....	5
13. Rebuild Solution and Run Project. ....	5

## Steps

### Part 1: Create Projects

We will create a Identity Example web application (IdexApp) which uses Individual User Accounts for authentication and authorization.

1. Create web application project → IdexApp
  - A. Select File/New/Project + Click OK
  - B. Select .Net Core under Visual C#
  - C. Select ASP.NET Core Web Application.
  - D. Enter the name of the application in this sample IdexApp. + Click OK
  - E. Select ASP.NET Core 2.0.
  - F. Click Change Authentication, select to Individual User Accounts. + Click OK. (Because of this step, the ASP NET Core Identity will be added into project.)
  - G. The IdexApp project will be created.
2. Create project to be used as Data Access Layer(DAL). → IdexApp.DAL
  - A. Right-click on Solution and select add new project.
  - B. Select .Net Core under Visual C#
  - C. Select class library(.net core) and enter the name of the project, in this sample it is IdexApp.DAL
  - D. Click OK.
  - E. The IdexApp.DAL will be created.
  - F. Right Click on IdexApp.DAL, select properties to check Core Version is the same with IdexApp.
  - G. **IMPORTANT:** Visual Studio creates projects with latest SDK in your machine by default. Be sure ASP.NET Core 2.0 selected. If not, select 2.0 and save it.

### Part 2: Move and update files in between Projects

3. Move Data folder from IdexApp to IdexApp.DAL and update file as described below.
  - A. On IdexApp project, select and cut Data folder via solution explorer.
  - B. Paste Data folder into IdexApp.DAL project.
  - C. Delete Migrations within Data folder.
  - D. Open ApplicationDbContext.cs within Data folder.
  - E. **Update namespace** of ApplicationDbContext class as **IdexApp.DAL.Data**
  - F. **Update using statements** of ApplicationDbContext class as **using IdexApp.DAL.Models**
  - G. Create Models folder within IdexApp.DAL
4. Create Models folder within Data folder of IdexApp.DAL
  - A. From IdexApp project, cut the **IdexApp/Models/ApplicationUser.cs**
  - B. Paste, ApplicationUser.cs file into **IdexApp.DAL/Models** folder.
  - C. Open **IdexApp.DAL/Models/ApplicationUser.cs** file. VS marks that there are many errors in file. This is quite normal. We will handle it in further steps below.
  - D. **Update namespace** of ApplicationUser class as IdexApp.DAL.Models
5. Delete default empty Class1.cs file from IdexApp.DAL

### Part 3: Install necessary packages for Identity in IdexApp.DAL project

6. Open Nuget Package Management for IdexApp.DAL
  - A. Right Click on IdexApp.DAL project, and choose Manage Nuget Packages.
  - B. Click Browse tab.
  - C. Search **Microsoft.AspNetCore.Identity**
  - D. Select **latest** v2.0.x (Because our project is 2.0.0)
  - E. Install Microsoft.AspNetCore.Identity 2.0.x

- F. Search **Microsoft.EntityFrameworkCore**
- G. Select **latest** v2.0.x (Because our project is 2.0.0)
- H. Install Microsoft.EntityFrameworkCore 2.0.x
- I. Search **Microsoft.AspNetCore.Identity.EntityFrameworkCore**
- J. Select **latest** v2.0.x (Because our project is 2.0.0)
- K. Install Microsoft.AspNetCore.Identity.EntityFrameworkCore
- L. Search **Microsoft.EntityFrameworkCore.SqlServer**
- M. Select **latest** v2.0.x (Because our project is 2.0.0)
- N. Install Microsoft.EntityFrameworkCore.SqlServer 2.0.x

#### 7. Build IdexApp.DAL project. It must be successful.

- A. If not succesful, check if you've installed the correct versions of packages. Or you can try to start from the beginning because steps below cannot be used if this step is not successful.

## Part 4: Using IdexApp.DAL in IdexApp.

#### 8. Add IdexApp.DAL into references of IdexApp.

- A. Right-click on **IdexApp** project
- B. Click add choose reference
- C. Click on **IdexApp.DAL**

#### 9. Update files in IdexApp solution.

- A. VS tell us that there are many errors in file. This is quite normal because we had moved Data folder and the ApplicationUser class to IdexApp.DAL.
- B. Open IdexApp/Controllers/**AccountController.cs**
- C. Update using statement "using IdexApp.Models;" as "using IdexApp.**DAL**.Models;"
- D. Open IdexApp/Controllers/**ManageController.cs**
- E. Update using statement "using IdexApp.Models;" as "using IdexApp.**DAL**.Models;"
- F. Open IdexApp/**Startup.cs**
- G. Update using statement "using IdexApp.Data;" as "using IdexApp.**DAL**.Data;"
- H. Update using statement "using IdexApp.Models;" as "using IdexApp.**DAL**.Models;"

#### 10. Optional: Update password options to make it easy to register

- A. In IdexApp/**Startup.cs** file, add code below into previous line before "services.AddMvc();" in ConfigureServices method.

```
services.Configure<IdentityOptions>(options =>
{
    // Password settings updated to ease this demo.
    //Do NOT use values below in your projects.
    options.Password.RequireDigit = false;
    options.Password.RequireLowercase = false;
    options.Password.RequireNonAlphanumeric = false;
    options.Password.RequireUppercase = false;
    options.Password.RequiredLength = 6;
    options.Password.RequiredUniqueChars = 1;

    // Lockout settings.
    options.Lockout.DefaultLockoutTimeSpan = TimeSpan.FromMinutes(5);
    options.Lockout.MaxFailedAccessAttempts = 5;
    options.Lockout.AllowedForNewUsers = true;

    // User settings.
    options.User.AllowedUserNameCharacters =
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-._@+#!";
    options.User.RequireUniqueEmail = false;
});
```

### 11. Update Views in IdexApp solution.

PS: You can find files to be updated by searching ApplicationUser string and recognize cshtml files in results.

- A. Open Login.cshtml, add "**@using IdexApp.DAL.Models**" statement on top of the file.
- B. Open \_LoginPartial.cshtml, add "**@using IdexApp.DAL.Models**" statement on top of the file.
- C. Open \_ManageNav.cshtml, add "**@using IdexApp.DAL.Models**" statement on top of the file.

### 12. Create database ready for Identity Management via IdexApp.DAL

- A. Open Package Manager Console
  - (a) Right click on **IdexApp.DAL** project.
  - (b) Click on Manage Nuget Packages.
  - (c) Click on package manager console tab.
- B. Set **default project to IdexApp.DAL**
- C. Type and execute: Add-Migration initializeDB
- D. Type and execute: update-database

### 13. Rebuild Solution and Run Project.

- A. Register to application to see it works correct.