**Official Website**

<https://dapper-tutorial.net/dapper>

**Examples:**

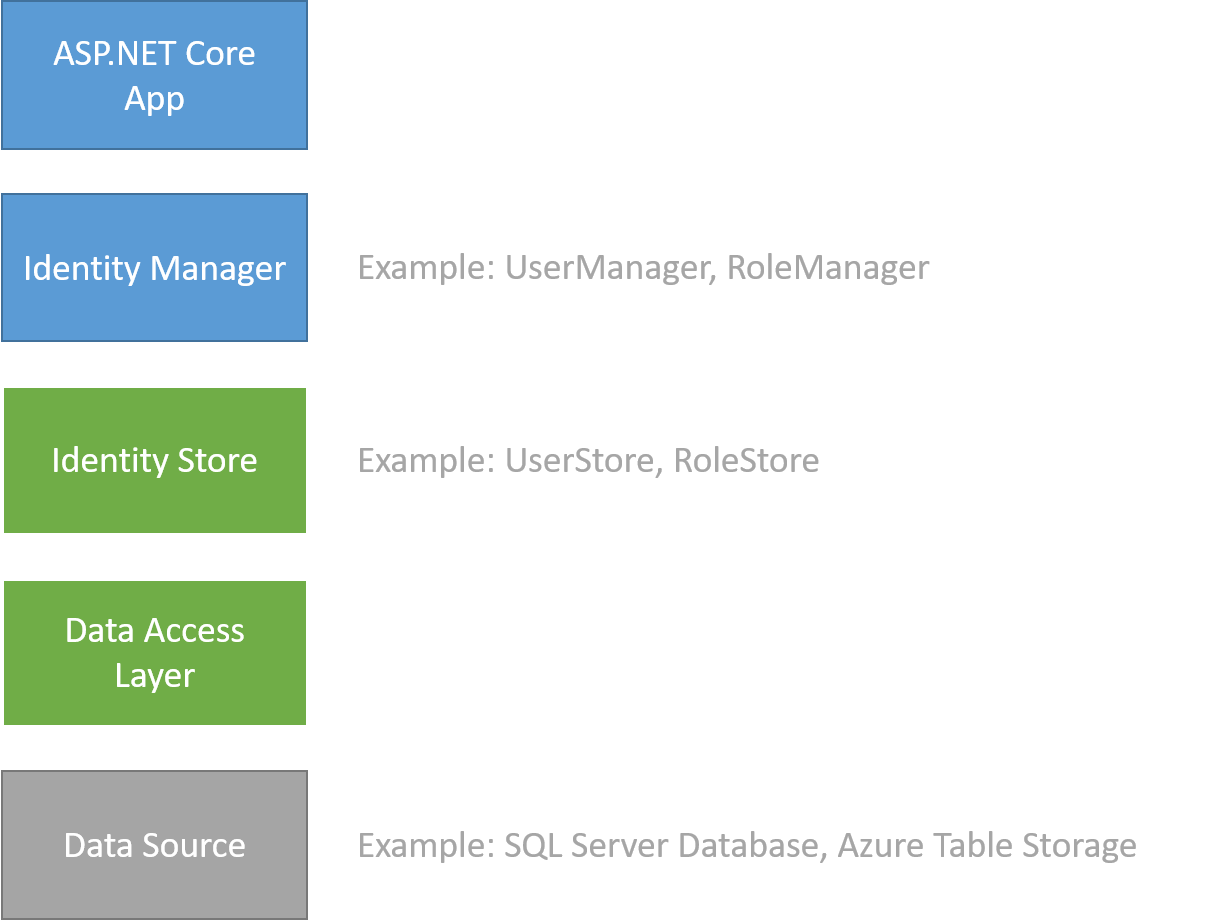
1st approach: AspNetCore.Identity replaces EntityFramework to Dapper. 🡺This one I discard when I find out RCL needs to override.

ASP.NET Core With Dapper And VS 2017 Using JWT Authentication WEB API and Consume it in Angular2 Client Application

* <https://www.codeproject.com/Articles/1221556/ASP-NET-Core-With-Dapper-And-VS-2017-Using-JWT>
* This post is focus on how to setup the JWT authentication

Custom storage providers for ASP.NET Core Identity

* https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity-custom-storage-providers?view=aspnetcore-2.2
* Sample code at <https://github.com/aspnet/Docs/tree/master/aspnetcore/security/authentication/identity-custom-storage-providers/sample/CustomIdentityProviderSample>
* This post is target on .Net Core 1.x and hard to modify due to there have a new feature of Razor Class Library after .Net Core 2.x



Replace Entity Framework to Dapper

Add, download, and delete custom user data to Identity in an ASP.NET Core project

* <https://docs.microsoft.com/en-us/aspnet/core/security/authentication/add-user-data?view=aspnetcore-2.1&tabs=visual-studio>
* This post is target on .Net Core 2.x
* If I want to replace the user class(IdentityUser) to custom user class, I have to override the Razor Class Library (RCL) of Identity. It takes a lot of time to study all the scaffold items of Identity.UI.

2nd approach: Angular client and .Net Core Web API with Dapper.

* Angular tutorial <https://angular.io/generated/zips/toh-pt6/toh-pt6.zip>
  + Comment snippet in app.module.ts
* HttpClientInMemoryWebApiModule.forRoot(
* InMemoryDataService, { dataEncapsulation: false }
* )
  + Revise URL of HeroesAPI in hero.service.ts
* private heroesUrl = 'http://localhost:5000/api/heroes'; // URL to web api
* ASP.Net Core Web API Application with Dapper – Part 1 <https://dotnetcorecentral.com/blog/asp-net-core-web-api-application-with-dapper-part-1/>
* ASP.Net Core Web API Application with Dapper – Part 2 <https://dotnetcorecentral.com/blog/asp-net-core-web-api-application-with-dapper-part-2/>
* Dapper and Repository Pattern in Web API
  + <http://www.mukeshkumar.net/articles/web-api/dapper-and-repository-pattern-in-web-api>
  + <https://code.msdn.microsoft.com/Dapper-and-Repository-68710cd>
* Dapper Contrib
  + <https://dapper-tutorial.net/dapper-contrib>
  + Dapper Contrib extend the IDbConnection interface with additional CRUD methods.