Official Steps for Creating Any Web App Whatsoever

# Server-Side Steps

1. Create a Web API Project in Visual Studio

Use a Web API project instead of an MVC project.

1. Domain Model

For example, if creating a Movie app then create your C# Movie class.

1. ApplicationDbContext

Add IDbSet<Movie> to the ApplicationDbContext in the IdentityModels.cs file.

1. Migrations

Enable migrations with “enable migrations”. Implement a Seed() method. Next, call “add-migration start” followed by “update-database”.

1. Generic Repository

Copy into Models folder from Coder Camps.

1. Dependency Injection

Install nInject from Nuget with “install-package Ninject.Web.WebApi” followed by “install-package Ninject.Web.WebApi.WebHost”. Configure NinjectWebCommon.cs by adding kernel.Bind<*Interface*>().To<*Interface Implementation*>().

1. Services

For example, create a MovieService. Using dependency injection, inject GenericRepository in constructor.

1. Controllers

Create an API folder and add a Web API controller to it. For example, create a MoviesController controller. Using dependency injection, injection a service in the constructor.

# Client-Side Steps

1. AngularJS

Install AngularJS and TypeScript header file using NuGet by installing “AngularJS.core” followed by “angularjs.TypeScript.DefinitelyTyped”. If you need to use AngularJS resources or routing then install “AngularJS.Route” and “AngularJS.Resource”.

1. Services

For example, create service using $http or $resource service.

1. Controllers

For example, create a moviesController controller that interacts with the movieService.

1. Application Module

Create your application module. Include all of the dependencies that you need such as ngResource and ngRoute.

1. Layout

Create your index.html page. Include all of the scripts that you need such as the AngularJS, Route, and Resource scripts. Also include your app scripts like app.ts, controllers.ts, and services.ts.