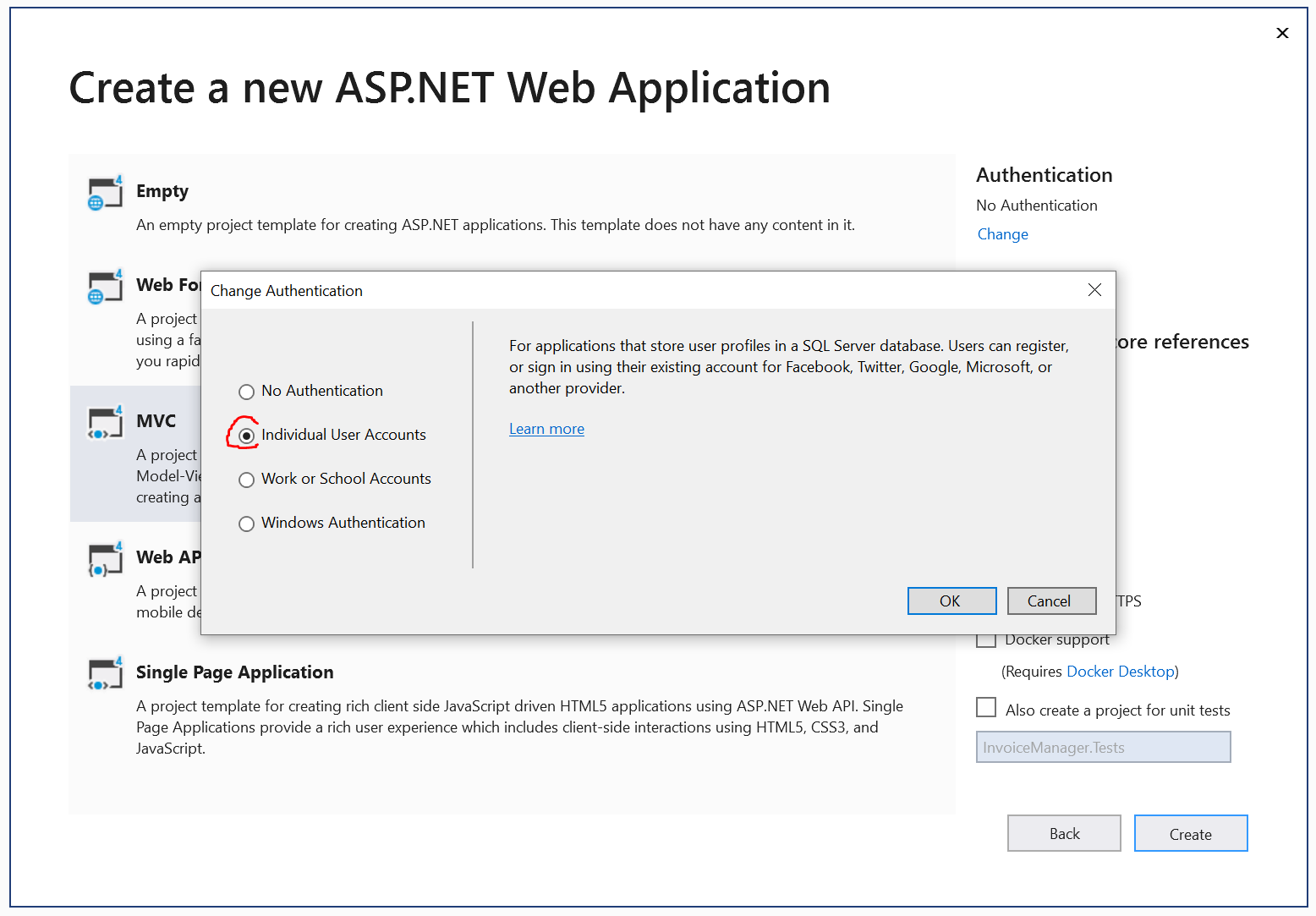
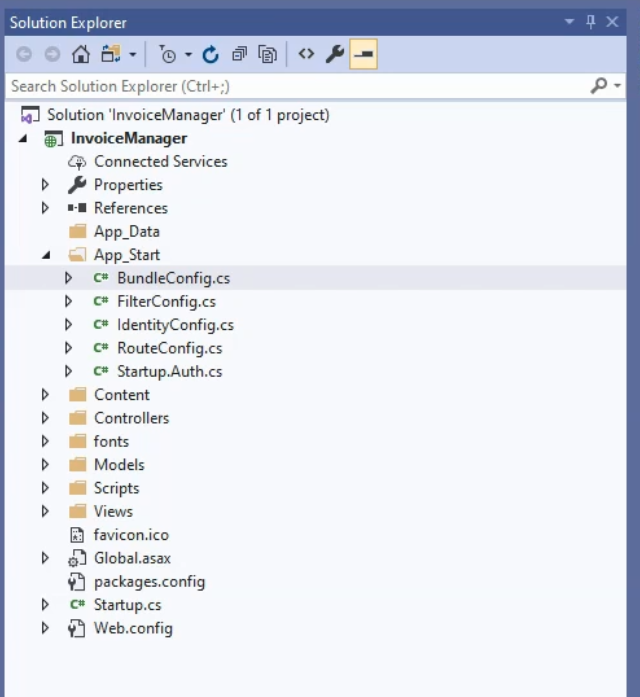


Chcemy aby nasza aplikacja miała już zaimplementowaną logikę dotyczącą rejestracji i logowania użytkowników.







BundleConfig.cs klasa w której wiążemy kilka skryptów css lub js w jeden plik, którego kod ładuje się szybciej, łatwiej możemy się do niego odwołać w kodzie.

|  |
| --- |
| public class BundleConfig  {  // For more information on bundling, visit https://go.microsoft.com/fwlink/?LinkId=301862  public static void RegisterBundles(BundleCollection bundles)  {  bundles.Add(new ScriptBundle("~/bundles/jquery").Include(  "~/Scripts/jquery-{version}.js"));  bundles.Add(new ScriptBundle("~/bundles/jqueryval").Include(  "~/Scripts/jquery.validate\*"));  // Use the development version of Modernizr to develop with and learn from. Then, when you're  // ready for production, use the build tool at https://modernizr.com to pick only the tests you need.  bundles.Add(new ScriptBundle("~/bundles/modernizr").Include(  "~/Scripts/modernizr-\*"));  bundles.Add(new ScriptBundle("~/bundles/bootstrap").Include(  "~/Scripts/bootstrap.js"));  bundles.Add(new StyleBundle("~/Content/css").Include(  "~/Content/bootstrap.css",  "~/Content/site.css"));  }  } |

Dzięki zapisowi version w nawiasach klamrowych, jak wersja się zmieni to nadal zapis ten będzie działał.  
Dalej zostaną załączone wszystkie skrypty, które rozpoczynają się od jquery.validate.

FilterConfig.cs tu można własny filtr, który będzie wywoływany przed każdym requestem.  
RouteConfig.cs – można tu zdefiniować jak ma działać ruting w naszej aplikacji, jak mają wyglądać adresy Url i jakie akcje mają być wywoływane

Folder Content – css, logo , obrazki i inne zasoby, bootstrapowe css’y   
Site.css – będziemy definiować własny css.

**Controllers** mamy zdefiniowane na wstępie 3 kontrolery. AccountControler.cs I ManageController.cs mają już w sobie mnóstwo logiki odpowiedzialnej za logowanie i zarządzanie kontami. HomeController.cs jest prawie pusty.

Controller to zwykła klasa, która dziedziczy po klasie Controller.  
Metody kontrolera nazywamy Akcjami.

**Models** – zawiera modele związane z logiką związaną z użytkownikami

**Scripts** – zawiera skrypty w JavaScript

**Views** – zawiera widoki. Posiada on kolejne foldery, które mają i muszą mieć taką samą nazwę powiązaną z kontrolerami.  
Dla HomeController.cs jest tu katalog Home, dla ManageController.cs jest katalog Manage