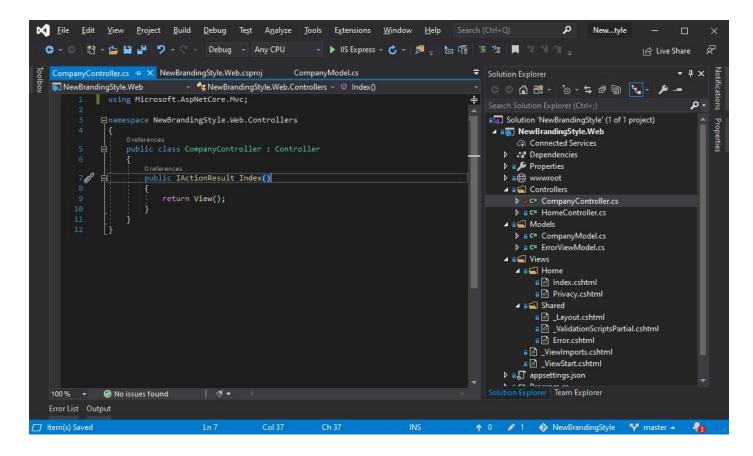
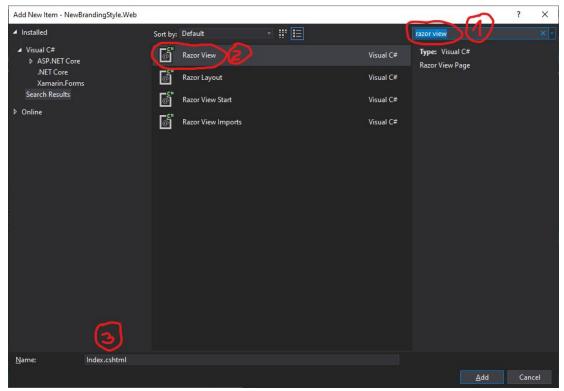
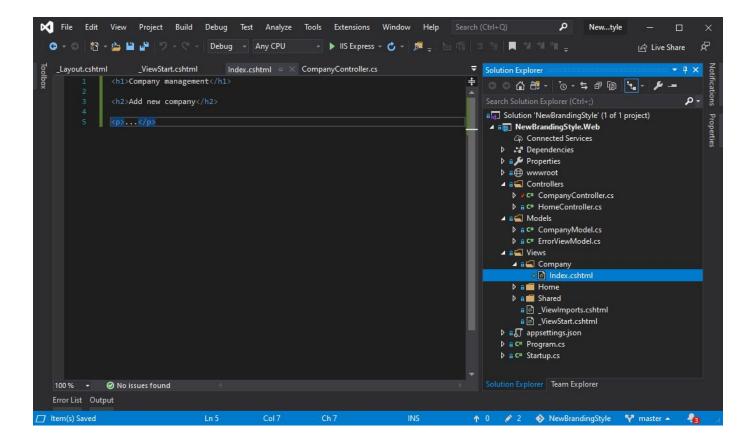
Dodanie kontrolera i widoku

Do katalogu Controllers dodajemy nową klasę kontrolera. W kontrolerze implementujemy domyślną akcję Index

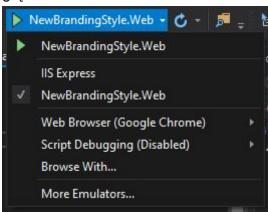


Dla nowo utworzonego kontrolera nie mamy jeszcze żadnych widoków (w katalogu Views nie ma żadnego podkatalogu Company). Tworzymy taki podkatalog. W nim dodajemy widok dla akcji Index: prawy przycisk myszy na katalog Company > "Add" > "New Item". Wyszukujemy "Razor View", plik nazywamy "Index.cshtml" i dodajemy

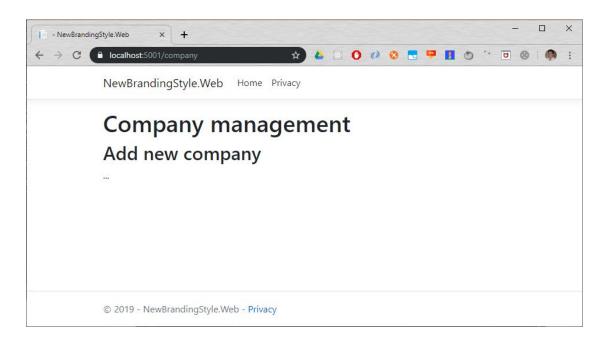




Uruchamiamy aplikację (IIS Express lub <nazwa aplikacji>). Visual Studio powinien jednocześnie uruchomić nam aplikację w nowym oknie przeglądarki



W przeglądarce przechodzimy pod "http://localhost:5000/company" (dostosuj port i nazwę kontrolera w URL do Twojego przypadku).



Dodanie modelu danych

Do katalogu Models dodajemy nową klasę naszego modelu

```
File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)
 🕒 - 🖒 📸 💾 🧬 🤣 - 🗀 Debug - Any CPU
                                                          ▼ ▶ NewBrandingStyle.Web - 💍 - 🎜 📜 🛅 🃜 📜 🗏 📗
                                                                                                                               Live Share
   launchSettings.json
                     _ViewStart.cshtml
                                          Index.cshtml CompanyModel.cs → X 

Solution Explorer
                              ᠇ 🔩 NewBrandingStyle.Web.Models.Co 🕶 🔑 Name
   NewBrandingStyle.Web
                                                                                               ○ ○ 🖸 🗂 - To - S 司 📵 🖫 - 🔑 🗕
             □namespace NewBrandingStyle.Web.Models
                                                                                              Solution 'NewBrandingStyle' (1 of 1 project)
                  public class CompanyModel
                                                                                               ▲ 🚛 NewBrandingStyle.Web
                                                                                                    (a) Connected Services
                      Oreferences public string Name { get; set; }
                                                                                                   ■ Dependencies
                                                                                                  ▲ 🕳 🔑 Properties
                                                                                                     a launchSettings.json
                                                                                                    public bool IsVisible { get; set; }
                                                                                                    ⊿ a ⊆ Models
                                                                                                    ▶ a C<sup>®</sup> CompanyModel.cs
                                                                                                    Da C# ErrorViewModel.cs

✓ B

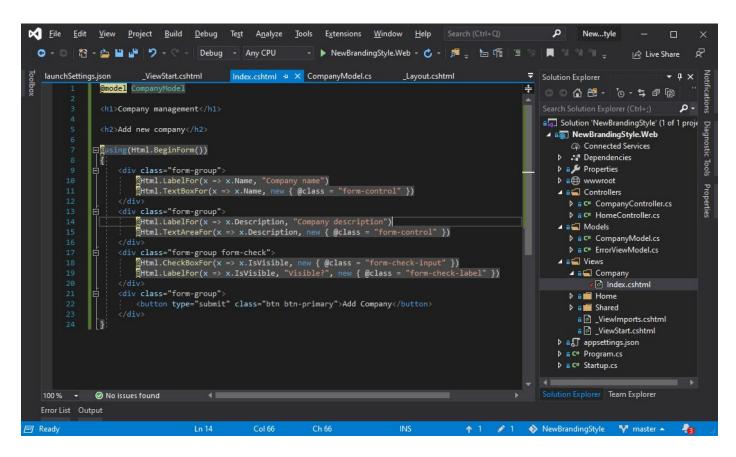
✓ Views

▲ a ⊆ Company

                                                                                                       + 🗗 Index.cshtml
                                                                                                    ▶ a Home
                                                                                                    ▶ a Shared
                                                                                                     © _ViewImports.cshtml
                                                                                                     a ☑ _ViewStart.cshtml
                                                                                                 ▶ a ☐ appsettings.json
                                                                                                  D a C# Program.cs
                                                                                                  D C# Startup.cs
                                                                                               Solution Explorer Team Explorer
   100 % ▼ Ø No issues found
                                                                                            ↑ 0 💉 2 🔷 NewBrandingStyle
```

Implementacja widoku formularza

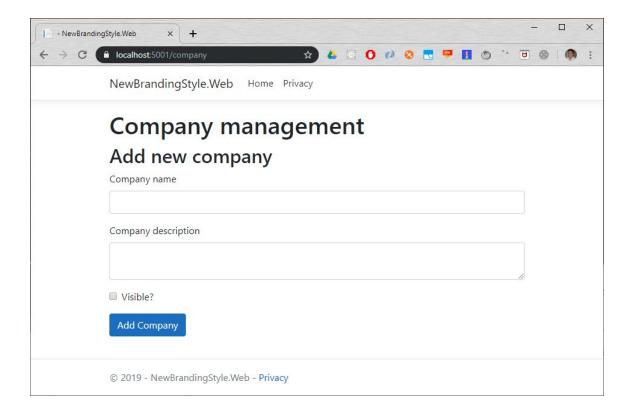
Implementujemy kod formularza. Używamy do tego Razor HTML Helpers i operujemy na klasie modelu CompanyModel, którą utworzyliśmy wcześniej



Zwracamy uwagę na

- dyrektywę @model CompanyModel na początku pliku dzięki niej Razor wie, jakiej klasy używamy jako modelu i Visual Studio może nam pomóc z podpowiadaniem kodu (Intellisense)
- wykorzystanie Html.BeginForm()
- wykorzystanie HTML Helperów i lambda expressions, które odwołują się do właściwości naszego modelu CompanyModel
- podawanie dodatkowych informacji w HTML Helpers, tj nazwy klasy CSS dla docelowego elementu

Uruchamiamy aplikację, przechodzimy pod "http://localhost:5000/company"



Przesłanie formularza: odebranie danych od użytkownika

Do kontrolera dodajemy przeciążenie metody Index z parametrem CompanyModel i atrybutem [HttpPost]. Wcześniejszą metodę Index dla porządku dekorujemy atrybutem [HttpGet]. Implementujemy podstawowe przetwarzanie danych podanych przez użytownika

```
File Edit View Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)
 🕒 - 🖒 🏥 💾 🤣 🤣 - 🗀 Debug - Any CPU
                                                           🔻 🕨 NewBrandingStyle.Web - 💍 - 🎜 📜 🔚 🏗 🖫 📜 👭 👭 👭
                                                                                                                                  🖄 Live Share
   CompanyAddedViewModel.cs _Layout.cshtml CompanyController.cs + X _ViewStart.cshtml
   NewBrandingStyle.Web
                                    - NewBrandingStyle.Web.Controllers.Comr - 🛈 Index(CompanyModel company)
                                                                                                              namespace NewBrandingStyle.Web.Controllers
                                                                                                               a Solution 'NewBrandingStyle' (1 of 1 proje
                                                                                                                ▲ 👼 NewBrandingStyle.Web
                                                                                                                    Connected Services
                                                                                                                    Dependencies
                                                                                                                  ▶ a  Properties
                       public IActionResult Index()
                          return View():

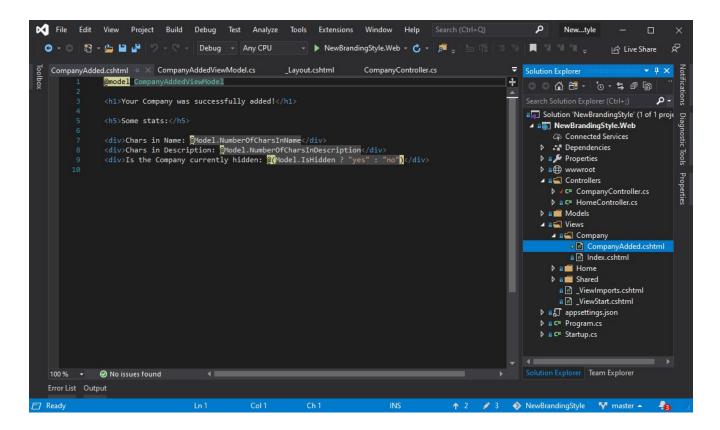
▲ a Controllers

                                                                                                                    Da C# HomeController.cs
                                                                                                                  ▶ a ■ Models
                                                                                                                   public IActionResult Index(CompanyModel company)
                           var viewModel = new CompanyAddedViewModel
                                                                                                                    ▶ a Home
                                                                                                                    ▶ a Shared
                              NumberOfCharsInName = company.Name.Length,
NumberOfCharsInDescription = company.Description.Length,
IsHidden = !company.IsVisible
                                                                                                                      © _ViewImports.cshtml
                                                                                                                      ■ ViewStart.cshtml
                                                                                                                  ▶ a∬ appsettings.json
                                                                                                                  return View("CompanyAdded", viewModel);
                                                                                                                    on Explorer Team Explorer
   100 % ▼ ⊘ No issues found
  Error List Output
```

Zwracamy uwagę na:

- obecność parametru CompanyModel company
- operacje wykonywane na danych przekazanych jako parametr
- nazwę widoku (pierwszy argument "CompanyAdded" przekazany do metody View())
- przekazanie modelu do widoku (drugi argument View())
- typ przekazanego modelu do widoku (CompanyAddedViewModel nowa klasa tworzymy ją sami w katalogu Models, analogicznie do CompanyModel)

Dodajemy widok potwierdzający przyjęcie danych od użytkownika (tj dodanie Company). Widok dodajemy również w katalogu Views/Company. Używamy nazwy pliku takiej jak wartość argumentu przekazana do metody View() w kontrolerze "CompanyAdded"



Zwracamy uwagę na:

- dyrektywę @model CompanyAddedViewModel na początku pliku
- wypisywanie danych za pomocą @Model.NumberOfCharsInName etc bezpośrednio z klasy CompanyAddedViewModel
- użycie operatora ternarnego w kodzie Razor wypisanie informacji w oparciu o Model.IsHidden.
 Składnia: @ (...kod C#...)

Testujemy. Ponownie uruchamiamy aplikację z Visual Studio, przechodzimy pod URL formularza dodawania Company. Wypełniamy formularz, klikamy przycisk "Add Company". W rezultacie otrzymujemy stronę z potwierdzeniem

