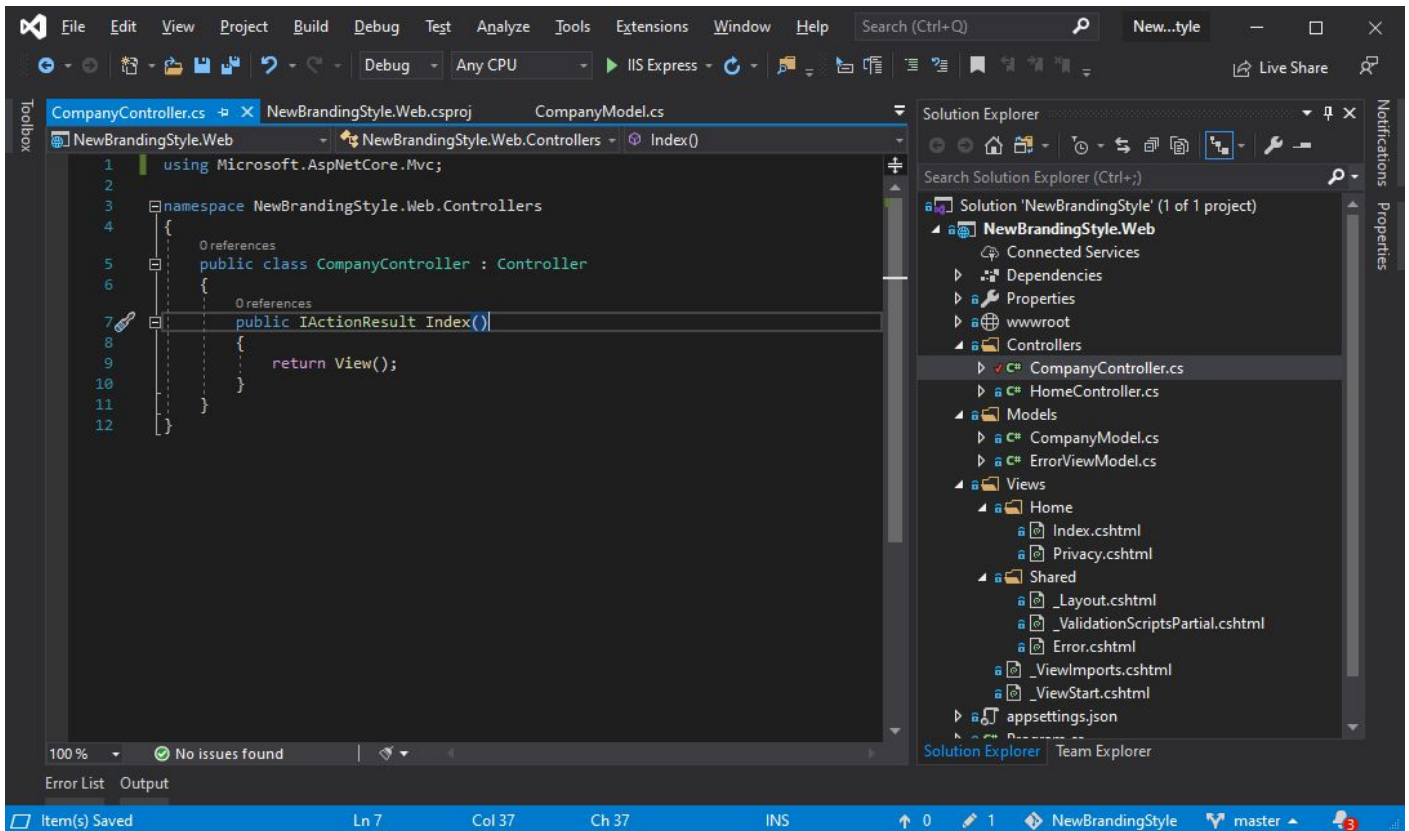
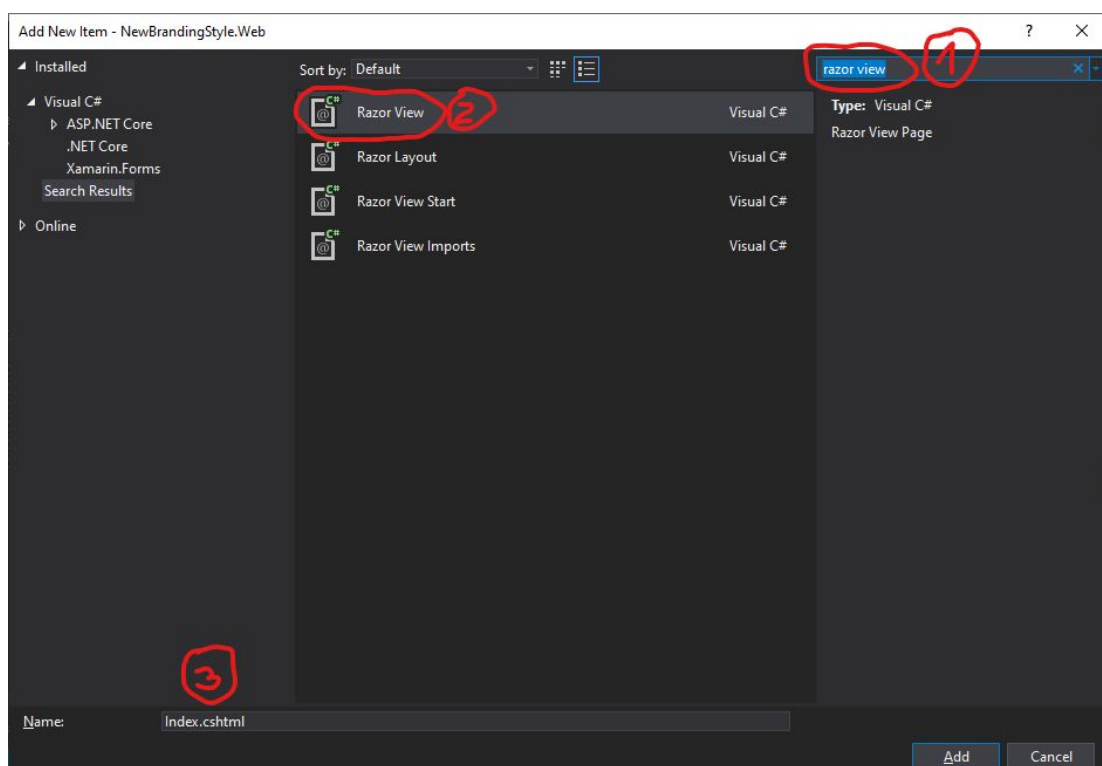


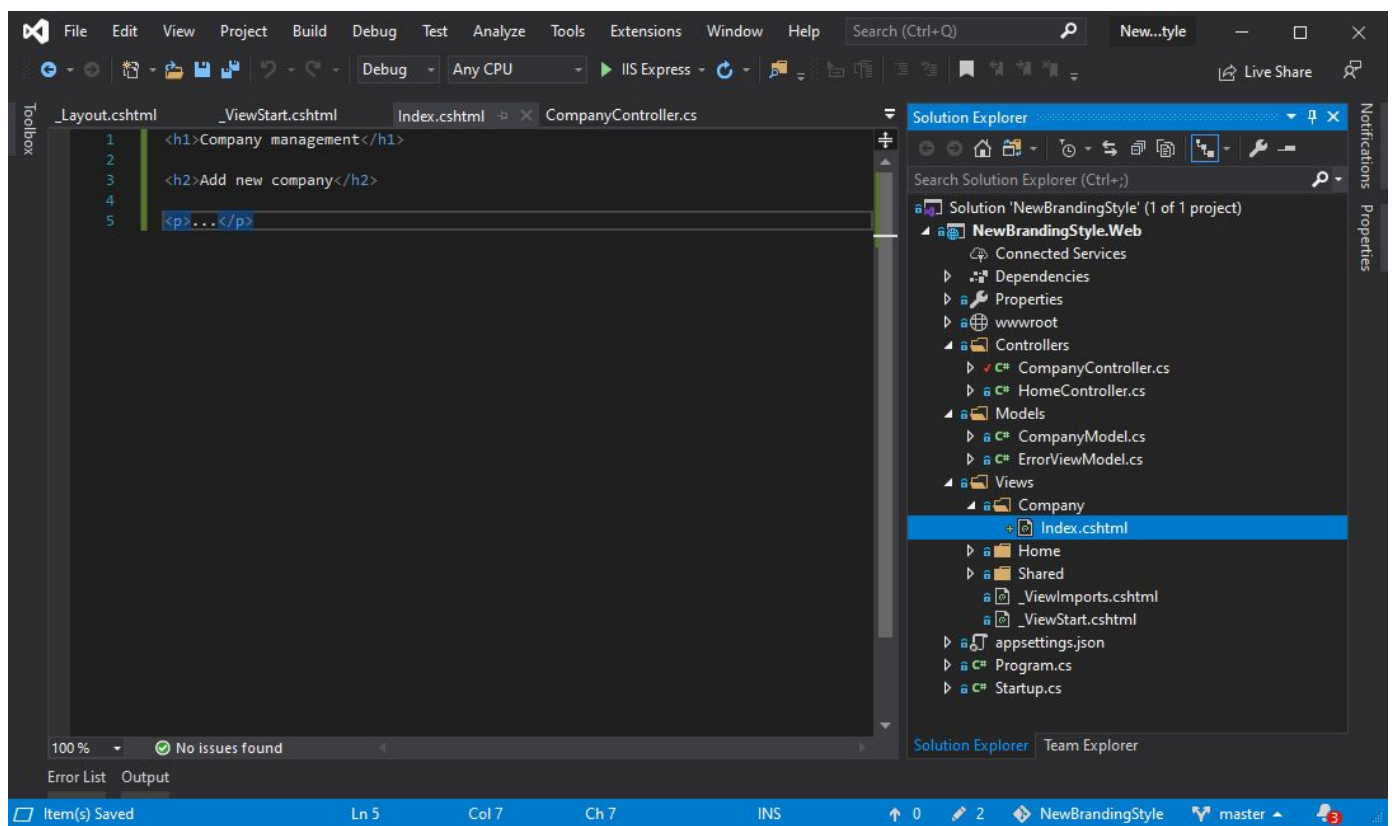
Dodanie kontrolera i widoku

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

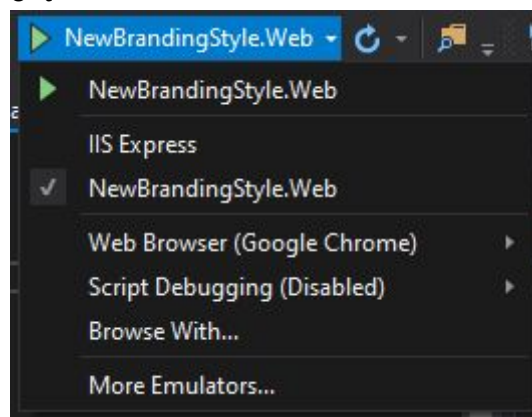


Dla nowo utworzonego kontrolera nie mamy jeszcze żadnych widoków (w katalogu Views nie ma żadnego podkatalogu Company). Tworzymy taki podkatalog. W nim dodajemy widok dla akcji Index: prawy przycisk myszy na katalog Company > "Add" > "New Item". Wyszukujemy "Razor View", plik nazywamy "Index.cshtml"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

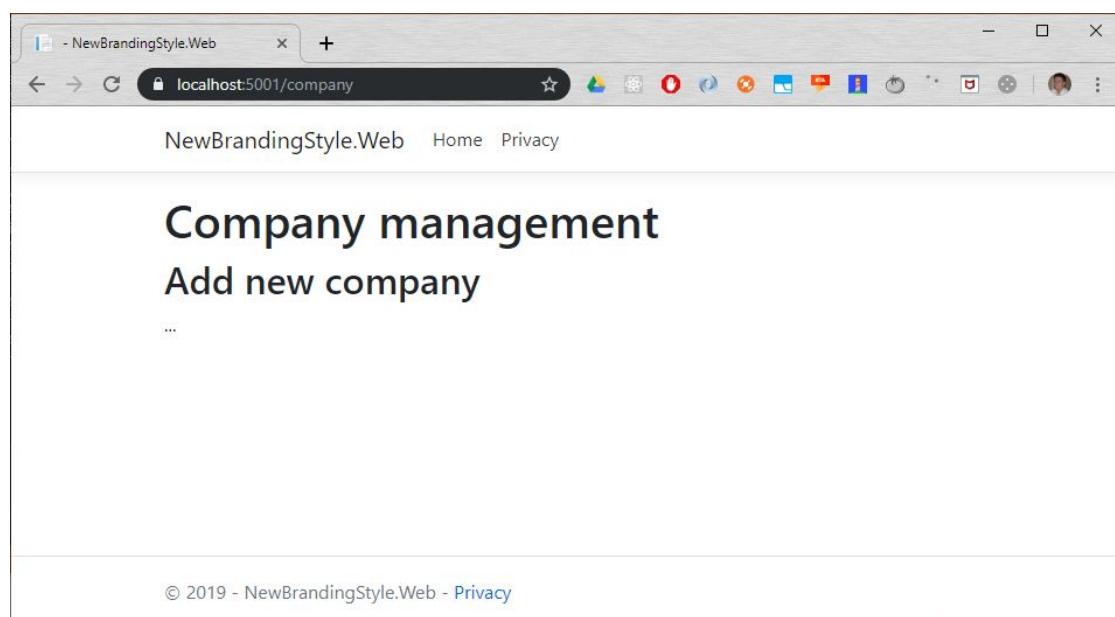




Uruchamiamy aplikację (IIS Express lub <nazwa aplikacji>). Visual Studio powinien jednocześnie uruchomić nam aplikację w nowym oknie przeglądarki

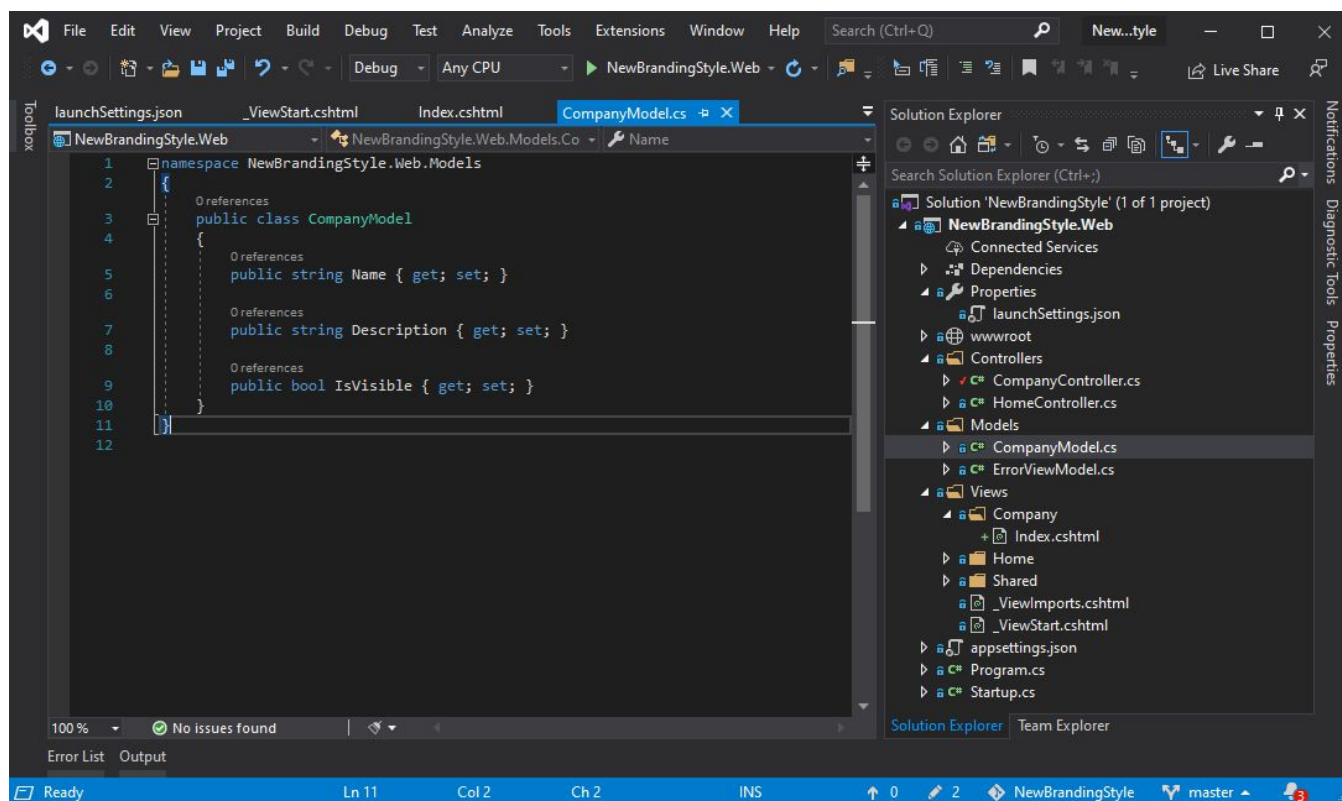


W przeglądarce przechodzimy pod <http://localhost:5000/company> (dostosuj port i nazwę kontrolera w URL do Twojego przypadku).



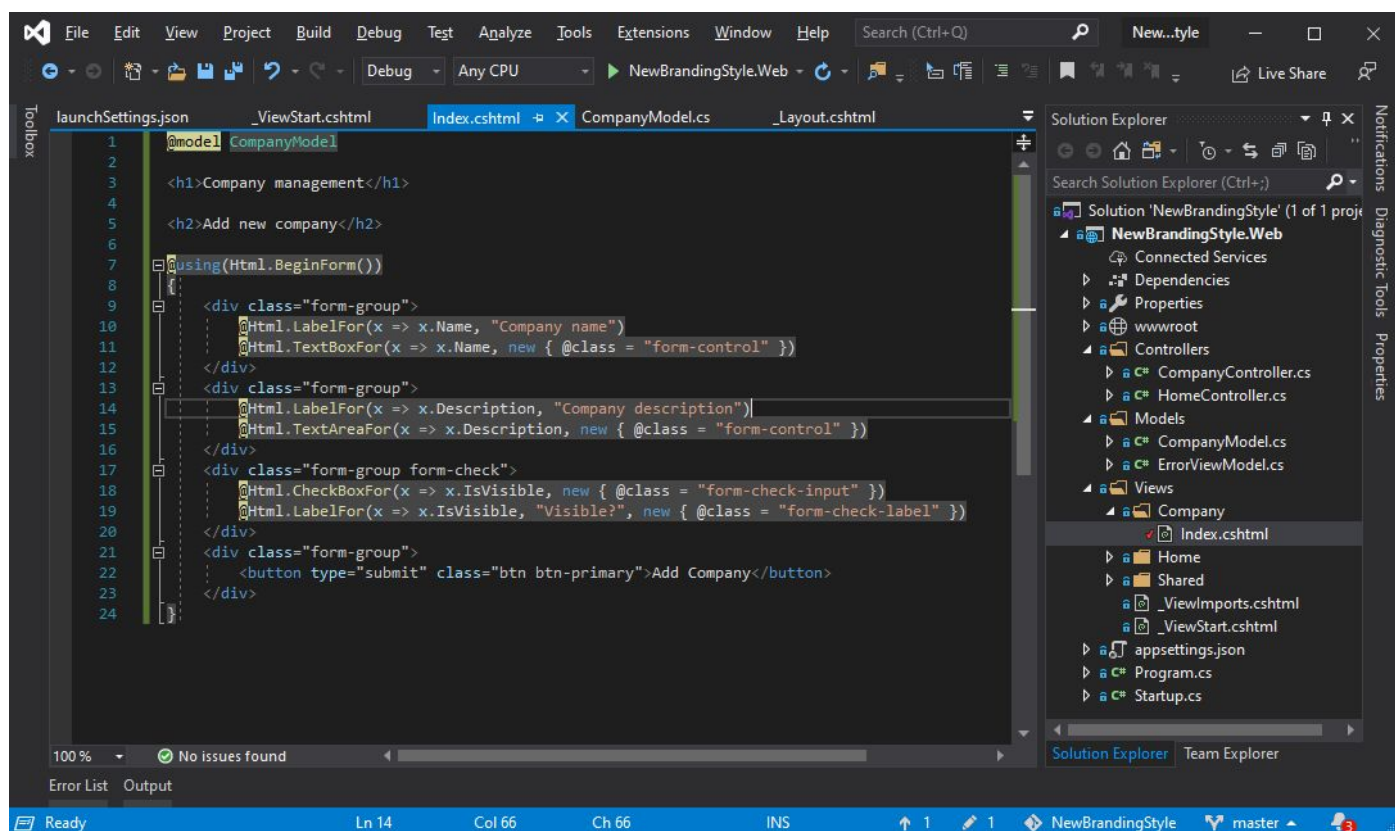
Dodanie modelu danych

Do katalogu Models dodajemy nową klasę naszego modelu



Implementacja widoku formularza

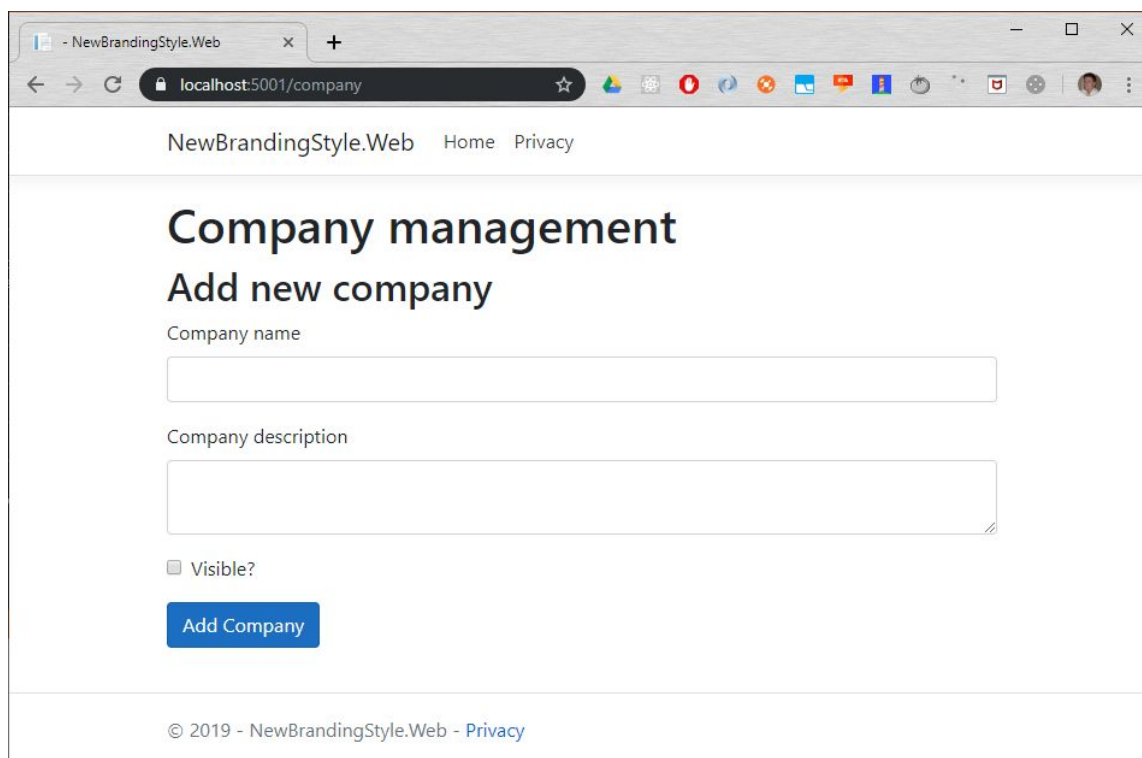
Implementujemy kod formularza. Używamy do tego Razor HTML Helpers i operujemy na klasie modelu CompanyModel, którą utworzyliśmy wcześniej



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"](http://localhost:5000/company)



NewBrandingStyle.Web Home Privacy

Company management

Add new company

Company name

Company description

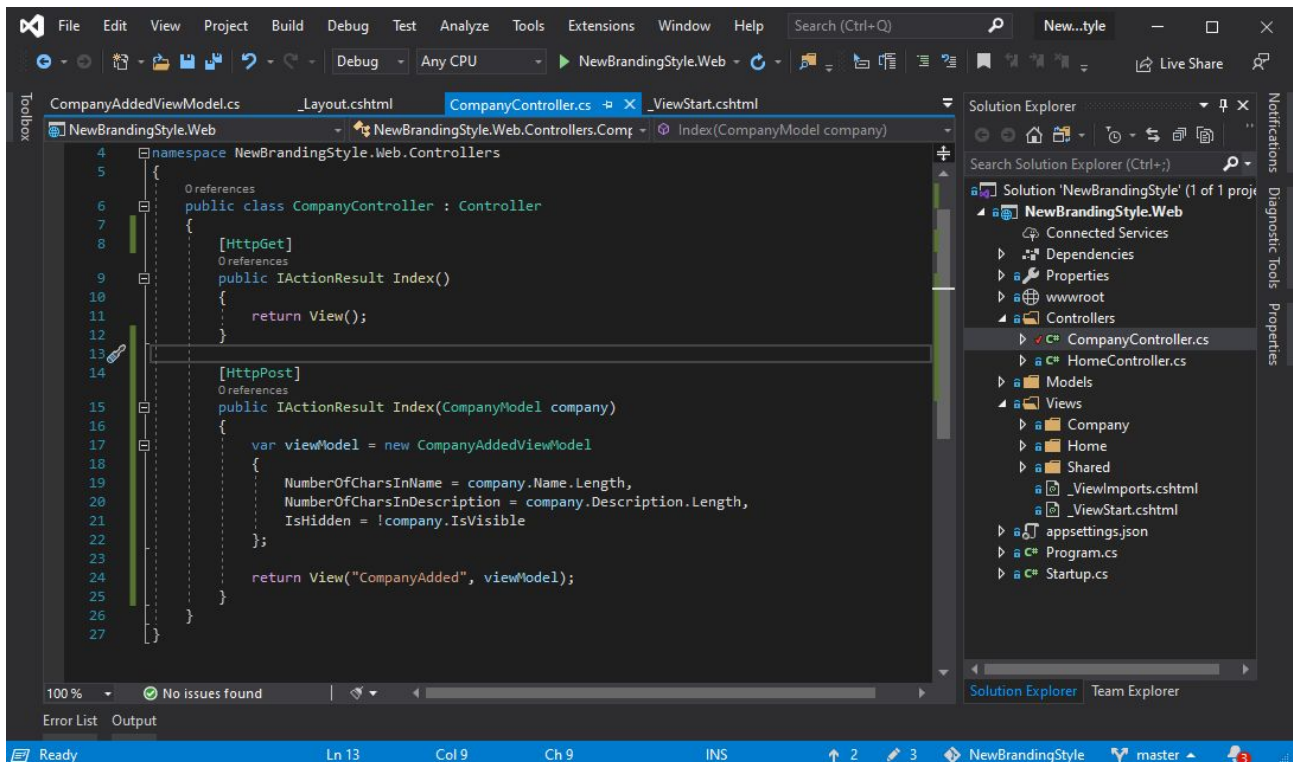
☐ Visible?

Add Company

© 2019 - NewBrandingStyle.Web - Privacy

Przesłanie formularza: odebranie danych od użytkownika

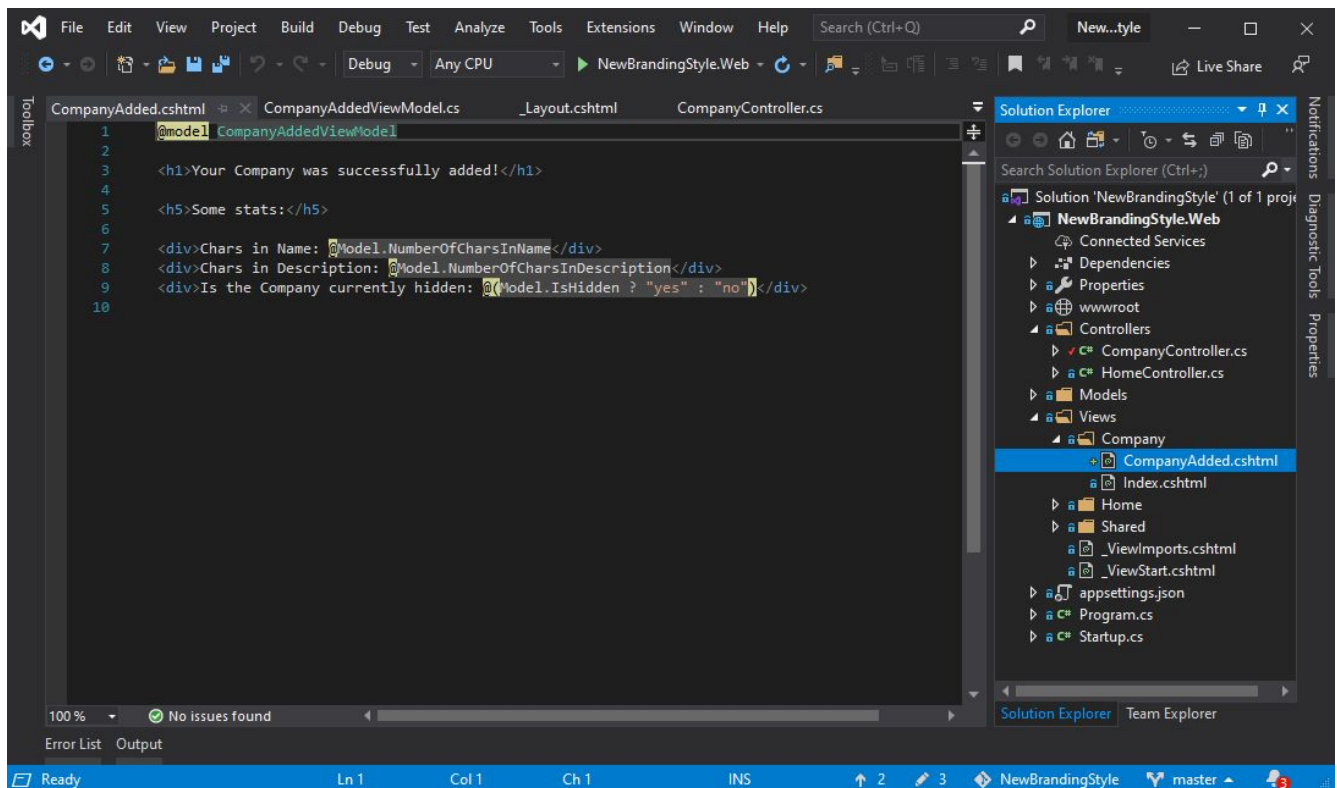
Do kontrolera dodajemy przeciążenie metody Index z parametrem CompanyModel i atrybutem [HttpPost]. Wcześniejszą metodę Index dla porządku dekorujemy atrybutem [HttpGet]. Implementujemy podstawowe przetwarzanie danych podanych przez użytkownika



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

