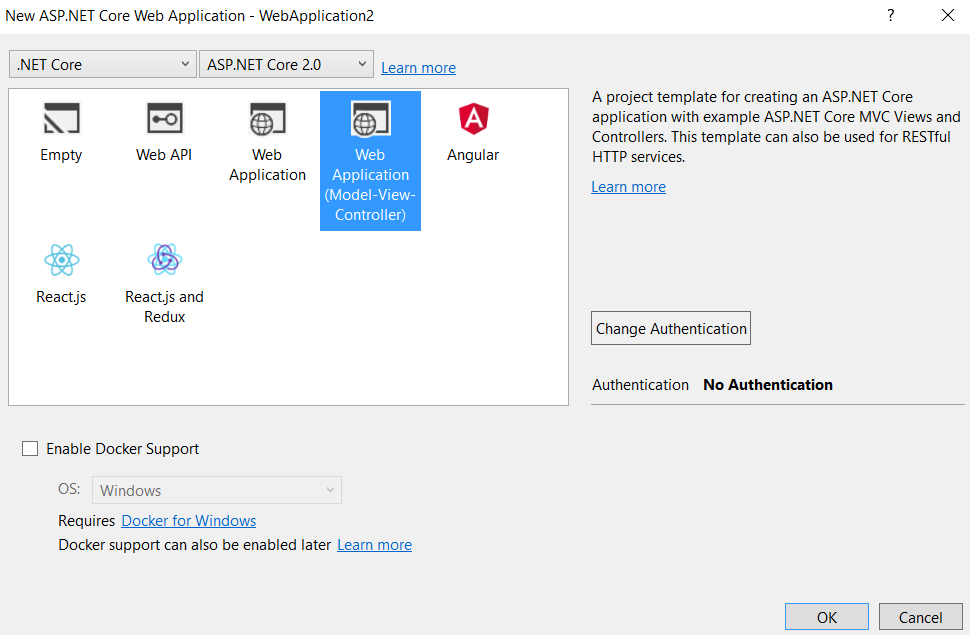
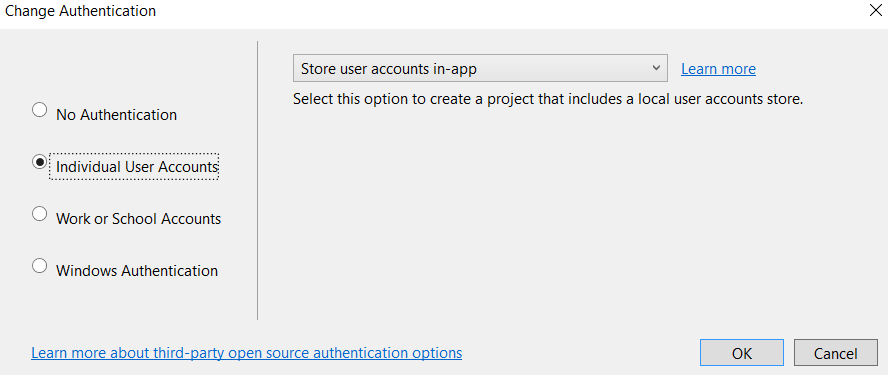
# Implementace MVC v praxi

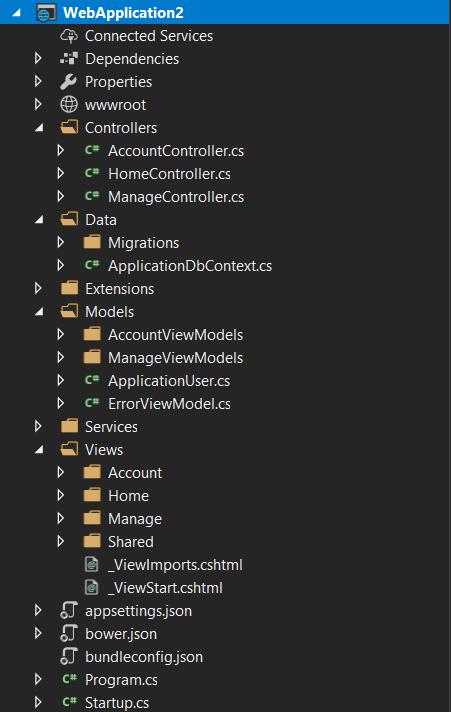
Teď, když už známe základní principy frameworku a strukturu MVC architektury, je na čase si je prakticky vyzkoušet. Jak již bylo zmíněno, MVC architektura je doporučenou a také nejčastěji využívanou při vytváření webových aplikací v ASP .NET Core frameworku, což pro nás znamená, že Microsoft zajistil, aby byla implementace této architektury do projektu velmi jednoduchá.

## Založení projektu implementující MVC

Založíme si tedy nový projekt pomocí stejného postupu jako minule. Tentokrát však využijeme šablonu „Web Application (Model-View-Controller)“ a do projektu přidáme autentifikaci uživatelů.





Námi nově vytvořená aplikace již obsahuje plně implementovanou architekturu MVC.

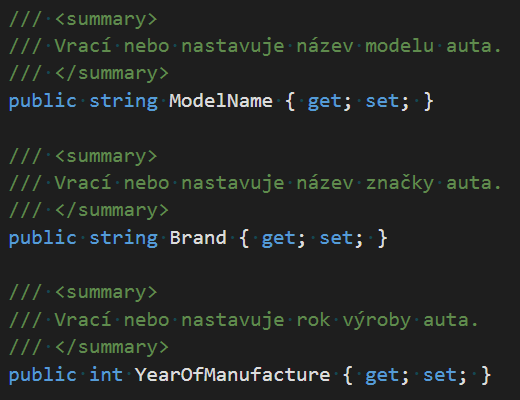
* Složka Controllers
  + Obsahuje řadiče
* Složka Models
  + Obsahuje modely
* Složka Views
  + Obsahuje uživatelská rozhraní
  + Podsložky v této složce jsou pojmenovány podle řadičů
  + Šablony v těchto podložkách jsou mají stejný název jako akce řadiče, ke které patří

## Využití MVC v praxi

### Vytvoření modelu

Jako první si vytvoříme jednoduchý model. Do složky Models přidáme třídu CarViewModel, která bude uchovávat informace o autě.

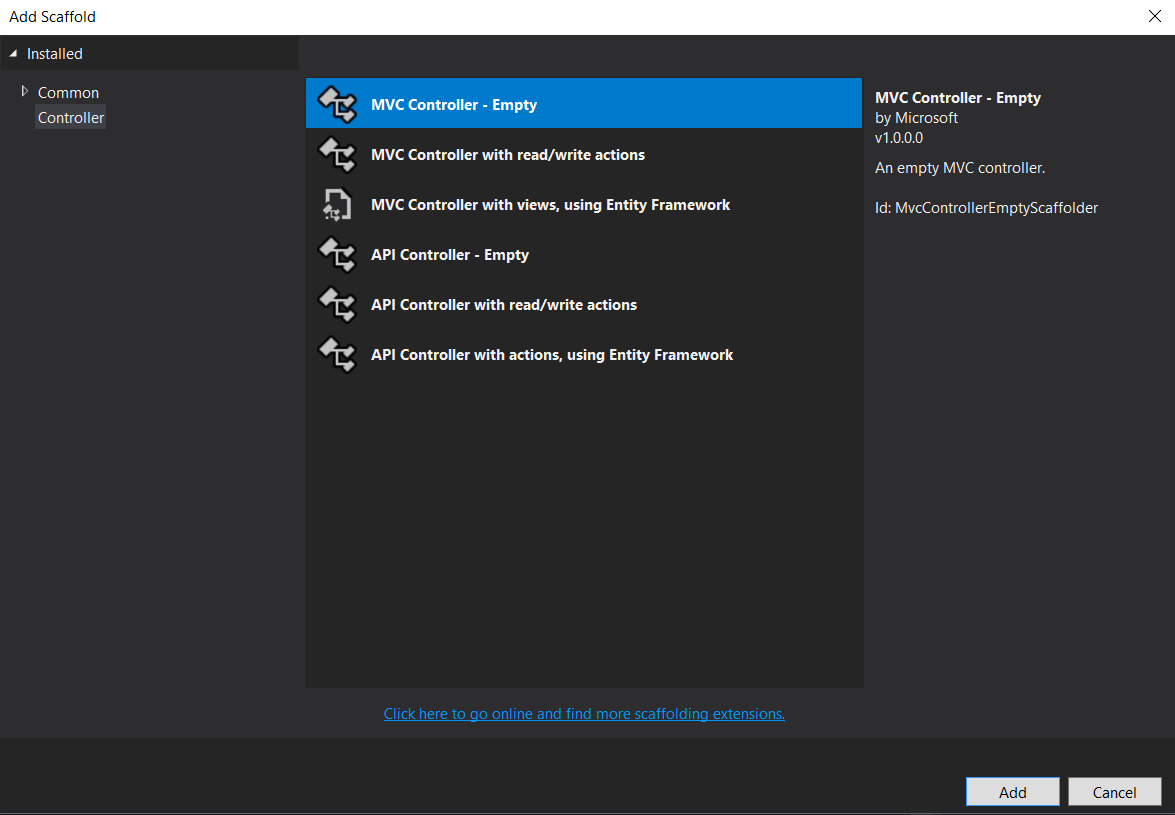
**Obsah třídy CarViewModel**

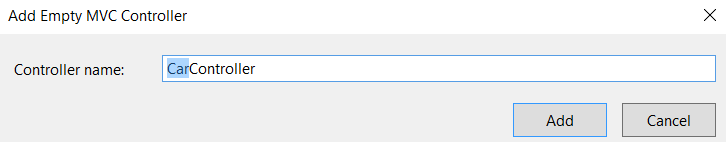


### Vytvoření řadiče

Postup

* Klikneme pravým tlačítkem myši na složku Controllers
* Klikneme na Add
* Klikneme na Controller
* Zvolíme šablonu MVC Controller – Empty
* Pojmenujeme kontroler a potvrdíme

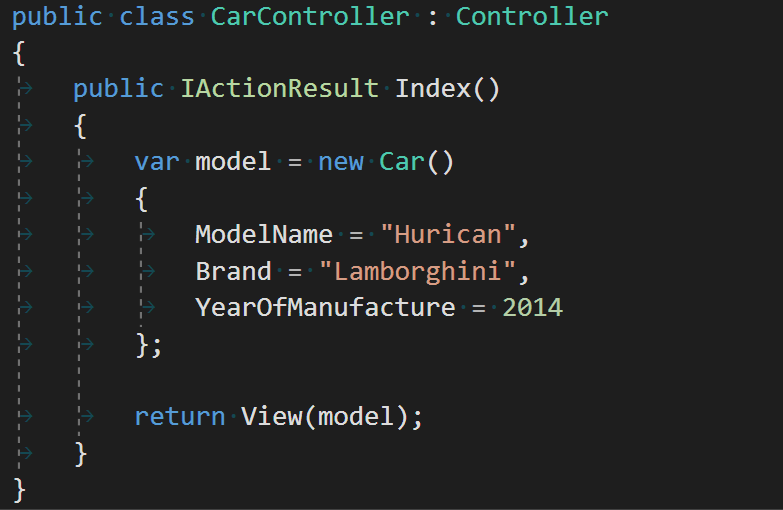




**Struktura kontroleru**

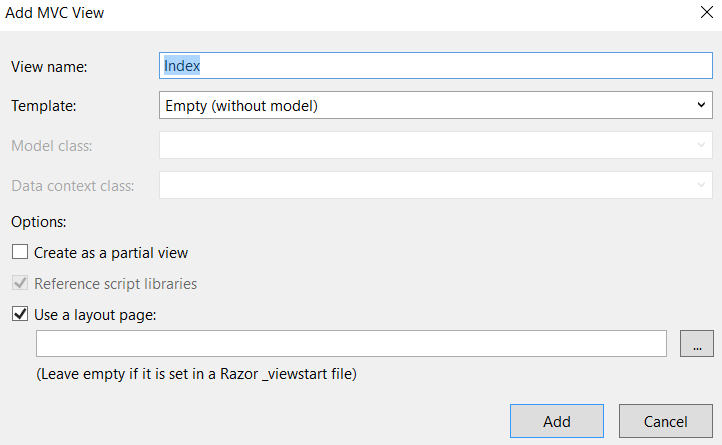
Každý řadič obsahuje tzv. akce. Tyto akce jsou v podstatě jen metody nacházející se v řadiči. Při zpracovávání dotazu se router rozhoduje, které akci, v jakém řadiči, data předá. Routerem zvolená akce pak data zpracuje a vrátí odpověď, kterou může být například html stránka.

V našem případě tedy máme jednu akci jménem Index nacházející se v řadiči Car. V této akci vytváříme model, který pak předáváme metodě View. Nemáme však ještě uživatelské rozhraní (view) patřící k této akci.



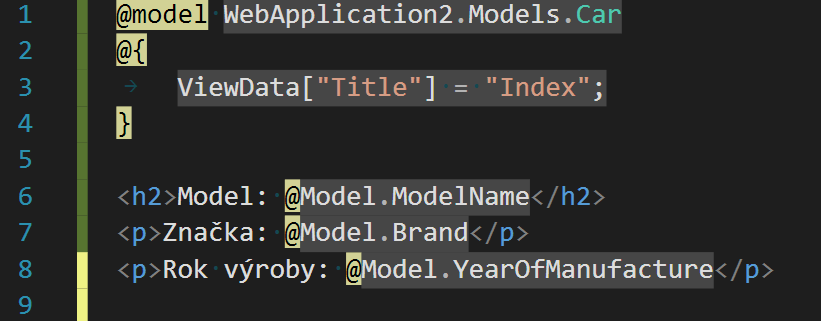
### Vytvoření šablony

Klikneme pravým tlačítkem myši uvnitř metody Index a zvolíme *„Add View“*. Otevře se nám dialogové okno, které jen potvrdíme.

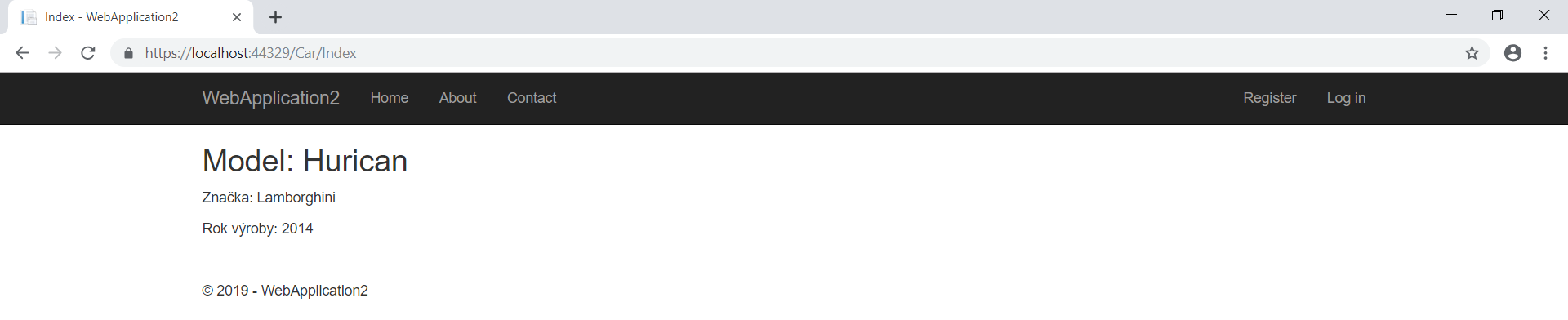


Po potvrzení se nám vygeneruje šablona šablona s příponou *„.cshtml“*. Do šablony vepíšeme kód z obrázku. Jedná se o velmi jednoduchý pohled, který pouze vypíše data z modelu. Na 1. řádku můžete vidět, že jsme specifikovali datový typ modelu. Tento model pak můžeme kdekoliv v šabloně využívat pomocí @Model.vlastnost.

Pozn.: Akce řadiče **musí** uživatelskému rozhraní předat model stejného typu, jaký vyžaduje šablona.



Spuštění webové aplikace



Na obrázku můžeme vidět, že to vše skutečně funguje. Data z modelu se nám zobrazila přesně tak, jak by se dalo očekávat.

Kromě dat z modelu však vidíme na obrázku navíc hlavičku s navigačním menu a patičku stránky. To je způsobeno tím, že naše šablona byla vložena do tzv. sdíleného Layoutu, který se nachází v Views/Shared/Layout.cshtml.

V defaultní šabloně MVC projektu nám toho Microsoft připravil do souboru \_Layout.cshtml opravdu hodně. Nás však v tuhle chvíli zajímá pouze část, kde se volá metoda RenderBody. Právě do tohoto místa se vkládají vyrenderované šablony, které využívají tento layout.

