# Tvorba formulářů

## Formuláře obecně

Formulář slouží ke shromáždění informací zadaných uživatelem. Tyto informace se pak po potvrzení formuláře odešlou na server, kde jsou zpracovány akcí v nějakém řadiči.

## Praktická ukázka

Základní princip formulářů není nijak složitý. Dle mého názoru bude nejlepší si vše ukázat v praxi. Vytvoříme si tedy jednoduchý formulář.

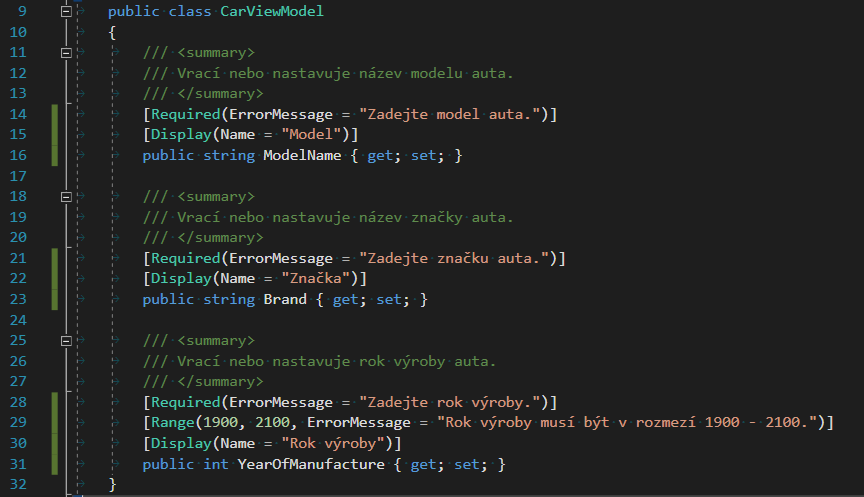
Námi vytvořený formulář bude sloužit k vyplnění základních informací o autě. Formulář bude využívat datových anotací pro popisky vstupních polí a určení podmínek, které musí data splňovat, aby prošla validací.

### Model formuláře

Pro náš formulář si nejdříve musíme vytvořit datový model. V momentě, kdy odešleme data formuláře na server, si framework sám vytvoří model tohoto typu a naplní jen přijatými daty. Model se také využívá při vytváření formuláře pro dosazení výchozích hodnot.

#### Vytvoření modelu

Náš model obsahuje 3 vlastnosti, kterými jsou značka auta, model auta a rok výroby. Jak můžete vidět na obrázku, nad každou vlastností jsou umístěny atributy, které specifikují text popisku vstupního pole ve formuláři (atribut Display) a podmínky, které musí hodnoty splňovat. Např. vlastnost *YearOfManufacture* musíme ve formuláři vyplnit hodnotu (atribut *Required*), která se může pohybovat v rozmezí 1900 až 2000 (atribut *Range*).

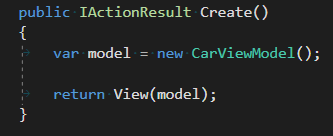


### Vytvoření potřebných akcí v řadiči

Pro vytvoření formuláře potřebujeme do řadiče přidat celkově 2 akce. První akce nám vrátí formulář, který budeme moct vyplnit a druhá bude data odeslaná formulářem zpracovávat.

#### Vytvoření akce pro vrácení formuláře

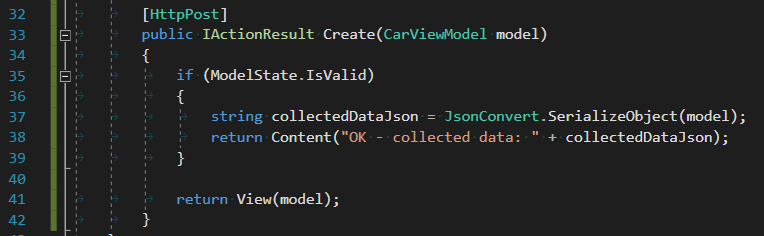
Tato akce pouze vytváří prázdnou instanci modelu, kterou předá šabloně formuláře. Je nutno dodat, že předávat šabloně formuláře prázdný model není nutností. Například, při editaci záznamu v databázi by se šabloně předával model s předvyplněnými hodnotami.



#### Vytvoření akce pro zpracování dat

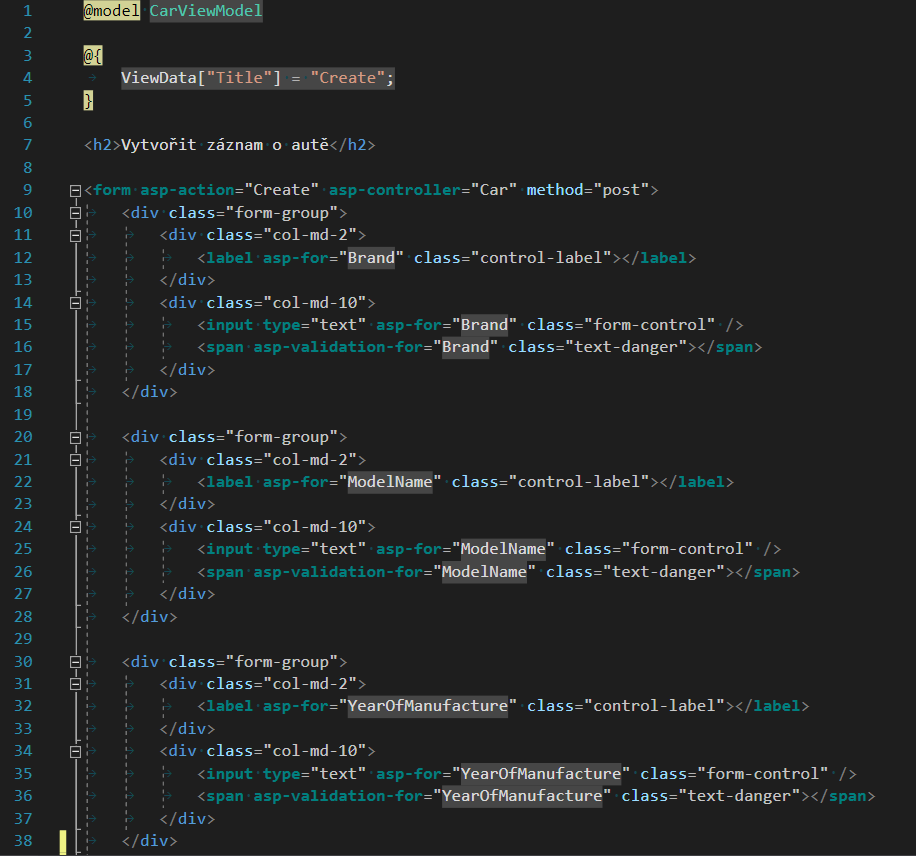
Tato akce přijímá jako parametr objekt typu *CarViewModel*. Všimněte si, že se nad metodou nachází atribut *HttpPost*, který specifikuje, že tuto akci si můžeme vyžádat pouze pomocí požadavku typu POST.

Na řádku 35 můžeme vidět podmíněné větvení na základě vlastnosti *ModelState.IsValid*. Hodnota této vlastnosti nabývá hodnot *true* nebo *false*, kde *true* znamená, že data jsou validní a *false*, že nejsou validní. Pokud jsou data validní, vrátí akce textový řetězec s informacemi o autě, pokud ne, tak vrátí stránku s formulářem s chybovými hláškami, které jsme specifikovali pomocí atributů v našem modelu.



### Vytvoření formuláře pomocí HTML Tag Helpers

Nyní nám už jen zbývá vytvořit šablonu formuláře. Kód šablony můžete vidět na následujícím obrázku.



#### Určení modelu

Každá šablona využívající data z nějakého modelu musí mít datový typ modelu specifikovaný pomocí příkazu @model. V našem případě se jedná o příkaz *@model CarViewModel*, který se v šabloně nachází na řádku číslo 1.

#### Formulář (<form> tag)

Kromě klasického atributu *method*, který určuje způsob předávání dat, se nastavují ještě atributy *asp-action* a *asp-controller*. Hodnoty těchto atributů určují, jaké akci, v kterém řadiči se data odešlou.

#### Popisek vstupního pole (<label> tag)

Tento tag nám slouží jako popisek polí. Pomocí atributu *asp-for* určuje, ke kterému poli se popisek váže. Popisek se pak zobrazí vedle vstupního pole jakožto text, jemuž bude dosazena hodnota, kterou jsme specifikovali v modelu pomocí atributu *Display.*

#### Vstupní pole (<input> tag)

Jedná se o klasické vstupní pole, které se váže k určité vlastnosti v modelu. To, ke které vlastnosti se váže určuje hodnota atributu *asp-for*.

#### Validační zpráva (<span> tag)

Tato zpráva se zobrazí, pokud data formuláře po odeslání neprojdou validací. Text validační zprávy jsme specifikovali pomocí atributu již v modelu. Stejně jako u vstupního pole a jeho popisku, musíme specifikovat, ke které vlastnosti modelu se váže. To se určuje pomocí atributu *asp-validation-for*.

## Závěrem

Pro jednoduchost jsem do ukázky zařadil pouze pár HTML Tag Helperů. K dispozici jich však máme mnohem víc. Pokud Vás zajímají další, doporučuju se kouknout na jejich využití v ukázkovém projektu „InternetForum“.