

## Hinweis

Für die automatische DB-Migrationen:

[https://msdn.microsoft.com/en-us/library/gg679221\(v=vs.113\).aspx](https://msdn.microsoft.com/en-us/library/gg679221(v=vs.113).aspx)

Wird aber für die folgenden drei Aufgaben nicht benötigt.

## Aufgabe 1

Erstelle ein neues leeres MVC Projekt. Definiere ein Model Fahrzeug, zum Beispiel wie in Abbildung 1. Ob es ein Fahrzeug ist oder ein anderes Objekt, spielt keine Rolle.

Die Aufgabe ist, dass du das Model so definierst, dass automatisch zwei DropDown-Listen erstellt werden. Zum Beispiel wie in der Abbildung 1 die Farbe und Hersteller. Weiterhin sollen die Auswahl, bzw. Elemente der DropDown-Listen aus der Datenbank geladen werden.

Die Elemente der DropDown-Listen, wie zum Beispiel in Abbildung 2 und Abbildung 3, sollen Anfangs per Seed-Methode in die Datenbank eingefügt werden. Siehe dazu die Folie Nr. 4-13 und 4-14.

Erstelle per „Scaffolding“ lediglich den Fahrzeug-Controller und die Views dazu.

Create - My ASP.NET Applicati... X +

localhost:14530/Fahrzeugs/Create

Application name

### Create

Fahrzeug

Hubraum

PS

Farbeld

Herstellerld

Create

[Back to List](#)

Abbildung 1

## Create

Fahrzeug

---

**Hubraum**

**PS**

**Farbeld**

**HerstellerId**   
Blau  
Grün  
**Pink**  
PianoBlack  
Silver

[Back to List](#)

Abbildung 2

## Create

Fahrzeug

---

**Hubraum**

**PS**

**Farbeld**

**HerstellerId**   
Audi  
Open  
BMW  
**Renault**  
Fiat

[Back to List](#)

Abbildung 3

## Aufgabe 2

Ändere/Erweitere die Lastwagen-Stop Actionmethode aus Serie 1 wie folgt. Es darf keine Annotation Route verwendet werden.

```
[Route("Lastwagen/Stop/{sekunden}")]  
public ActionResult Stop(int sekunden)  
{ ... }
```

Passe, bzw. ändere die Actionmethode Stop() so um, dass trotzdem der übergebene Parameter sekunden gelesen werden kann.

## Aufgabe 3

Erweitere die Aufgabe 2 so, dass der Parameter sekunden per GET und POST gesendet und empfangen werden kann.