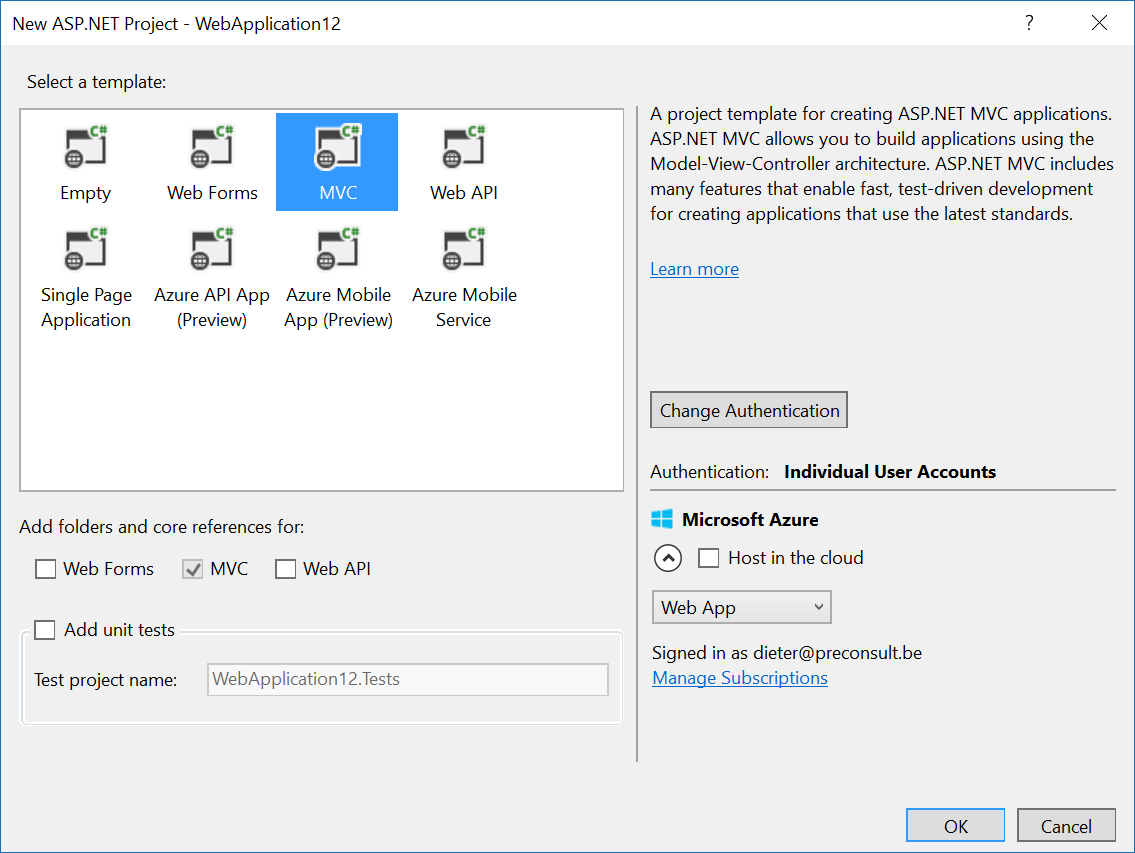
# ASP.NET MVC Labo Week 2

## Doelstelling:

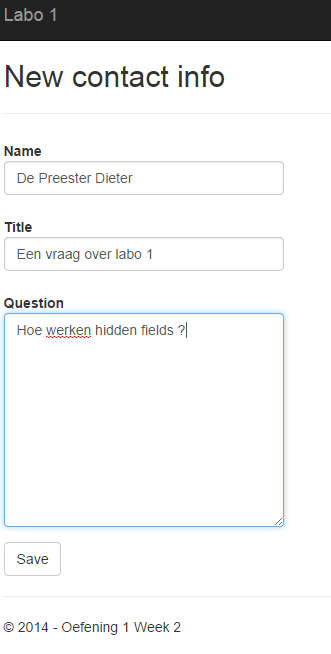
* Kennis maken met ASP.NET MVC
* Modelbinding
* Razor

## Oefening 1: Zelfstandig

De eerste oefening is een vrij eenvoudige oefening op modelbinding die u zelfstandig kan maken. Zorg zowel voor client side als server side validation. Maak een nieuw project aan in Visual Studio en kies voor een ASP.NET MVC Project

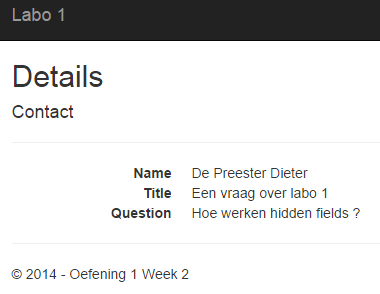


Het doel van het project is een zeer eenvoudige contact formulier met 3 verplichte velden. Maak zelf hiervoor het bijhorende model met de juiste properties. Het invulscherm ziet er als volgt uit:



Pas ook de header en footers aan van de layoutpage. Het Question veld moet ook ongeveer deze grote hebben. Waar kan je dit wijzigen?

Scherm Details: Dit scherm bevat gewoon de ingevulde waarden vanuit het vorige scherm.



## Oefening 2

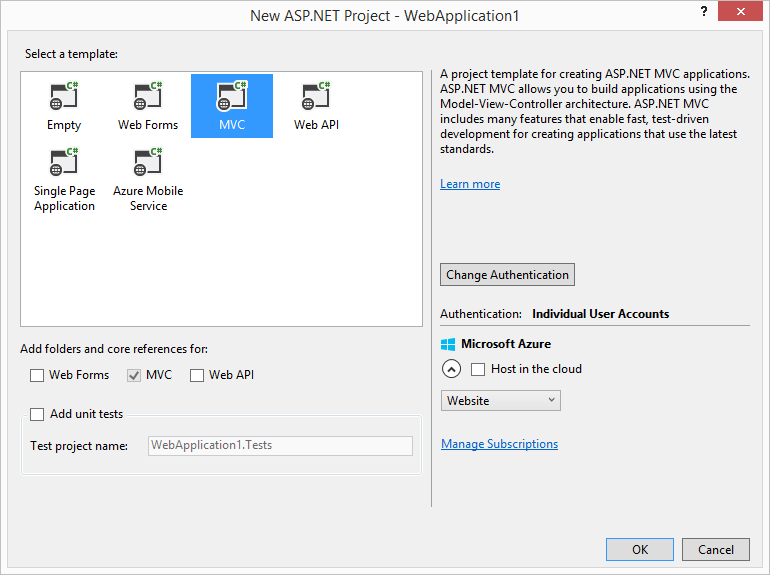
##### Opmerking

Deze oefening is een stuk uitgebreider dan de vorige oefening. Het is dan ook sterk aan te raden de grote stappen te volgen in het document. We gaan niet alles tot in detail uitleggen. Op sommige plaatsen zal u zelf moeten zoeken en uw nota’s uit de theorie gebruiken.

### Doel

We gaan een eenvoudige registratie applicatie schrijven voor mensen die zich willen registeren voor een IT Pro event. Er zal een agenda zijn op het opening scherm en een mogelijkheid tot registratie.

### Stap 1: Nieuw project

Maak een nieuw ASP.NET MVC project aan en kies voor MVC (niet empty). 

Voeg de file Data.cs toe aan het project. Deze file kan u terugvinden op LEHO en zal als “database” fungeren. Voeg ook de models “Session.cs”, “Organization.cs” en “Device.cs” toe in de map models van ons project.

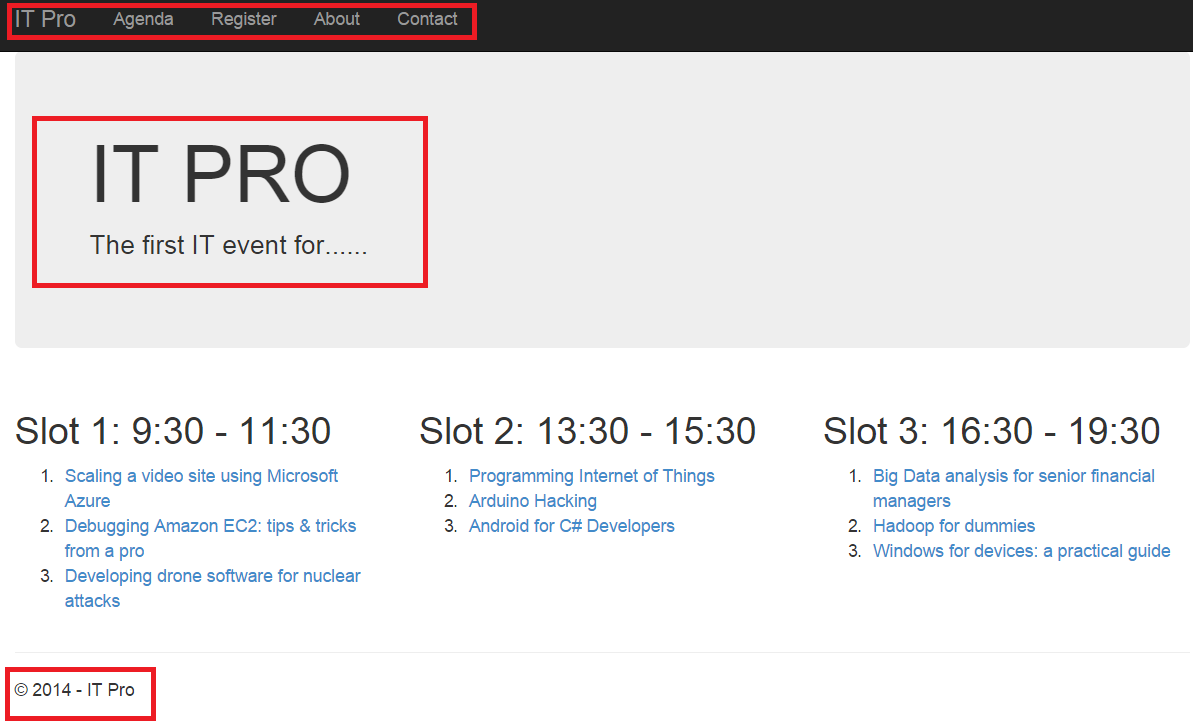
### Stap 2: Controllers

Voeg volgende controllers toe:

* RegisterController
* AgendaController
* HomeController (indien nog niet aanwezig)

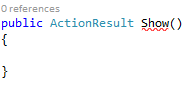
### Stap 3: Wijzig de \_Layout.cshtml

Bekijk de screenshot hieronder en pas de rode kaders aan in uw eigen applicatie zoals deze op de screenshot



### Stap 4: Agenda uitwerken

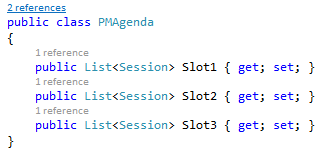
Alle data die je nodig hebt zit in “Data.cs” daar zitten ook een aantal methodes in die we nodig zullen hebben om de gegevens op te vragen. Voor dit stuk maken we gebruik van de methode GetSessions(int id). Ga naar de AgendaController.cs en voeg volgende ActionMethode toe:



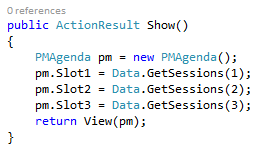
Deze methode zal on de agenda moeten weergeven die er als volgt uitziet



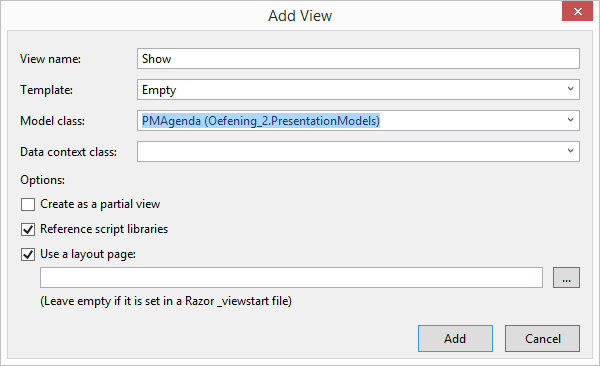
We moeten dus verschillende soorten informatie weergeven. Er zijn 3 slots met telkens keuze uit 3 sessies. Om verschillende soorten informatie weer te geven of op te vragen maken we best gebruik van een “PresentationModel”. Dit is een model die we speciaal maken voor een view. Voeg een map “PresentationModels” toe aan het project. In deze map voeg je dan de klasse “PMAgenda.cs” toe. Deze klasse beschrijft de data die we wensen weer te geven. De klasse ziet er als volgt uit:



Wijzig nu de methode Show() in de AgedaController.cs als volgt zodat we een nieuw presentation model PMAgenda aanmaken en de properties opvullen met de data afkomstig van de “Data.cs” klasse.



Voeg nu een view “Show” toe maar kies een empty template en als model class en ons presentation model PMAgenda.



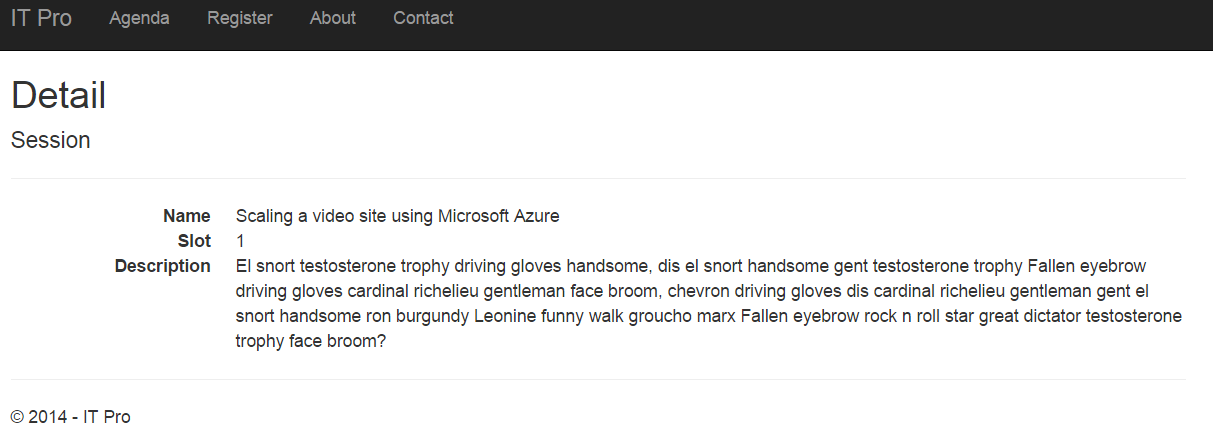
Als html code nemen we deze uit de view “Index.cshtml” van de home controller. Plak deze in onze view “Show.cshtml”. Nu moeten we deze view nog aanpassen zodat hij de slots zal weergeven. De slots moeten ook aanklikbaar zijn. Voorzie dus een link met telkens het ID in de link zodat we bij het klikken op een sessie meer info te zien krijgen.



### Stap 5: Detail scherm

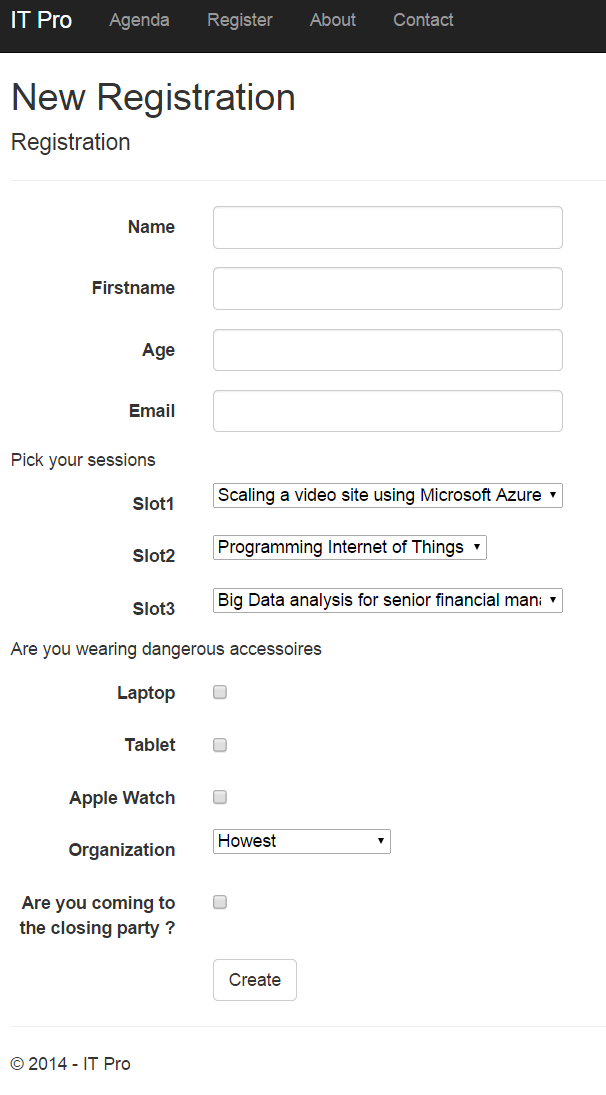
Werk nu volledig zelfstandig het detail scherm uit.

* Voeg zelf een actionmethod detail toe in de AgendaController
* Zorg ervoor dat je geen fout (programmeer defensief) krijgt als je in de querystring
  + GEEN id invult in de querystring
  + Een ongeldig ID invult
  + Indien men toch foute data ingeeft dan moet men terug de agenda te zien krijgen
* Tip: bekijk zelf de ASP.NET MVC function “RedirectToAction”



### Stap 6: Registratie scherm:

Het registratie scherm ziet er als volgt uit:



Ontwikkel dit volledig zelfstandig, zorg voor een goede validatie (kijk naar screenshot wat je moet valideren)

* Maak een “Registration.cs” model
  + Zorg voor de juiste validatie attributen

Na het registreren krijgen we nog een overzichtsscherm die er als volgt uit ziet:

