

ASP.NET MVC Labo Week 5

Doelstelling:

- Herhalings oefening modelbinding & presentation model
- Verder werken labo week 4 ASP.NET MVC
- Posten van data naar web api
- Opzetten van Post binnen een web api controller

Oefening 1: Afwerken

Werk eerst de Web API oefening van het vorige labo af.

Oefening 2: HttpClient

In het vorige labo hebben we via Web API de zoekertjes ingeladen via een http GET. De kans is groot dat u de code voor het ophalen direct in de click eventhandler van de button heeft geschreven of in het onload event van het window. In deze oefening aan we de code een beetje herstructureren in onze WPF applicatie. De aanpak die we voorstellen kan ook gebruikt worden voor Windows Phone en Windows Store applicaties.

Voeg een klasse WebserviceAccess.cs toe aan het WPF project en maak deze public.

```
0 references  
public class WebserviceAccess  
{  
  
}
```

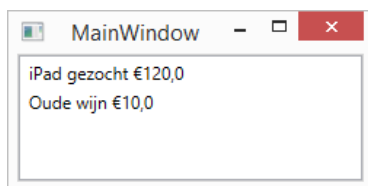
In deze klasse zullen we alle aanroepen naar de Web API afhandelen. Het eerst wat we gaan doen is de code voor het ophalen van zoekertje verplaatsen naar deze klasse. Declareer voor het gemak en duidelijkheid een private constante URL waarin we de URL van onze Web API vastleggen. Opgepast: we declareren enkel de url naar de API. Bij het gebruik van de URL zullen we nog de specifieke controller moeten opgeven. Op deze manier kunnen we de url naar de API gebruiken over meerdere methodes. Wijzig uw code zodat deze gebruik maakt van deze URL en de zoekertjes kan inladen.

```
private const string URL = "http://localhost:64041/api";

1 reference
public async void Load()
{
    using (HttpClient client = new HttpClient())
    {
        string url = string.Format("{0}{1}", URL, "/zoekertjemobile");
        HttpResponseMessage response = await client.GetAsync(url);
        if (response.IsSuccessStatusCode)
        {
            string json = await response.Content.ReadAsStringAsync();
            List<Zoekertje> result = JsonConvert.DeserializeObject<List<Zoekertje>>(json);
        }
    }
}
```

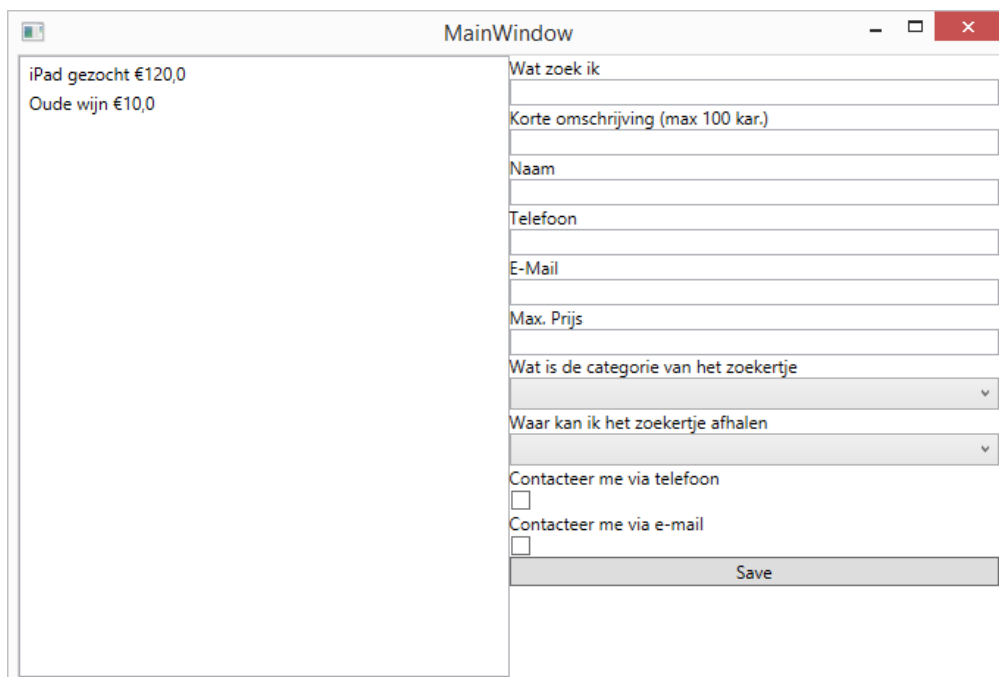
Bovenstaande code zal echter niet terugkeren uit de methode. De return waarde is void. Wijzig de code zodat deze een lijst van zoekertje zal terugkeren (List<Zoekertje>). Gebruik hiervoor de slides uit de theorie lest.

Na het wijzigen van de methode zodat deze een List<Zoekertje> kan terugkeren, moeten we de code in ons WPF venster aanpassen zodat deze gebruik kan maken van de methode Load uit de klasse WebserviceAccess.cs. Het resultaat zou moeten zijn dat de zoekertjes verschijnen.



Oefening 3: Aanmaken van zoekertjes via de WPF applicatie

Zorg ervoor dat je ook zoekertjes kan aanmaken via de WPF applicatie. Maak alvast volgende UI.



Dit scherm bevat gerelateerde data. We zitten met categorieën en locaties die te selecteren zijn in onze applicatie. Deze gegevens zitten tevens ook in de Web API.

Aanpassingen Web API

Schrijf nu zelf de code om de locaties en categorieën op te halen uit de ASP.NET Web API. Maak hiervoor twee Web API controllers bij aan de server kant:

- CategorieController.cs
- LocatieController.cs

Voorzei beide controller met een Get(). Hieronder kan u de code vinden voor de CategorieController.cs

```

0 references
public class CategorieController : ApiController
{
    0 references
    public List<Categorie> Get()
    {
        return Data.GetCategories();
    }
}

```

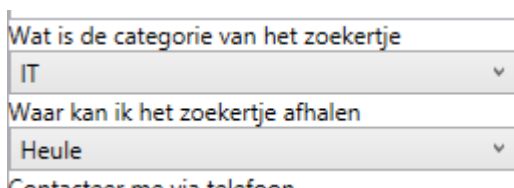
Schrijf nu zelf de LocatieController.cs en test beide uit in de browser en kijk of er data in het resultaat zit.

Ophalen locaties en categories

Als de aanpassingen aan de server kant werken kunnen we onze WebserviceAccess.cs uitbreiden. Voeg twee methodes toe:

- LoadLocaties()
- LoadCategories()

Roep deze methodes aan vanuit het WPF venster zodat de dropdownlist opgevuld worden. Het resultaat zou er ongeveer als volgt moeten uitzien:



Posten van data: nieuw zoekertje

Voeg een methode Post() toe aan de ZoekertjesController (Web API controller). Deze method zal het nieuwe zoekertje toevoegen aan de database. Voeg zelf code toe die volgende zaken doet:

- Controleren of parameter in orde is => juiste foutcode terugsturen
- Bij exception => juiste statuscode terugkeren

- Toevoegen zoekertje => juiste statuscode terugkeren

0 references

```
public HttpResponseMessage Post(Zoekertje zoekertje)
{
}
```

Aan de client kant voegen we een methode AddZoekertje toe die het nieuwe zoekertje zal Posten naar onze Web API. U moet deze code zelf schrijven.

1 reference

```
public async Task<bool> AddZoekertje(Zoekertje newZoekertje)
{
}
```

De laatste stap is deze code aanroepen vanuit het WPF venster op basis van de klik op de knop. U moet een nieuwe zoekertje object aanmaken en opvullen en dit doorsturen naar de methode AddZoekertje. U zou dit zelf moeten kunnen. Als het toevoegen lukt moet u terug alle zoekertjes ophalen zodat deze verschijnen in de lijst.

Oefening 4: JQuery overzicht

In deze oefening keren we terug naar de web kant. We gaan ons overzicht scherm van zoekertjes wijzigen zodat we deze via JQuery ophalen. Het eerste wat u moet doen is JQuery toevoegen aan de pagina bovenaan _layout.cshtml.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - My ASP.NET Application</title>
  <link href="/Content/Site.css" rel="stylesheet" type="text/css" />
  <link href="/Content/bootstrap.min.css" rel="stylesheet" type="text/css" />
  <script src="/Scripts/modernizr-2.6.2.js"></script>
  <script src="/Scripts/jquery-1.10.2.min.js"></script>
  <script src="/Scripts/jquery.validate.js"></script>
```

Voeg een nieuwe action toe aan de controller zoekertjes met als naam IndexAjax. Zoek nu zelf hoe U er kan voor zorgen dat het overzicht scherm via JQuery zal ingeladen worden. Maak gebruik van \$.get(); Maak ook gebruik van <http://api.jquery.com/>

- € 120 iPad gezocht
- € 10 Oude wijn

© 2014 - My ASP.NET Application

Oefening 5: JQuery invoer formulier

Voeg nu een invulscherm toe die we gaan posten via Ajax (\$.Post) naar de controller. Het scherm ziet er als volgt uit:

- € 120 iPad gezocht
- € 10 Oude wijn

Nieuw zoekertje

Titel

Omschrijving

Naam

Telefoon

E-Mail

Prijs

Categorie

Locaties

Contacteer via telefoon

☐

Contacteer via e-mail

☐



Na het toevoegen moet de lijst met zoekertjes terug ingeladen worden. Maak gebruik van de demo uit de theorie.