



C# Web – Razor Pages - Oefening

PRO - C# Web 1

DE HOGESCHOOL MET HET NETWERK

Hogeschool PXL – Elfde-Liniestraat 24 – B-3500 Hasselt
www.pxl.be - www.pxl.be/facebook



Doel

- ASP.Net Core toepassing
 - Razor Web Application
 - Oefening 01 – Locations
 - SelectList
 - Add Location
 - Oefening 02 – Add Klant
 - Oefening 03 - SelectList

ASP.NET CORE

Razor Web Application – Oefening01

- Project instellingen
 - Add class
 - Location.cs
 - LocationID
 - Postcode
 - City
 - Databank.cs
 - Voeg 2 records toe
 - 1-3500-Hasselt
 - 2-3600-Genk
 - Add razor Page
 - Locaties.cshtml
 - Toon een Tabel met de gegevens van de locaties uit de databank class.
 - Voeg een hyperlink toe in de Index razor page naar de nieuwe pagina

```
public class Location
{
    public Location()
    {
        LocationId = -1;
        PostCode = string.Empty;
        City = string.Empty;
    }
    public Location(int locationId, string postCode, string city)
    {
        LocationId = locationId;
        PostCode = postCode;
        City = city;
    }
    public int LocationId { get; set; }
    public string PostCode { get; set; }
    public string City { get; set; }
}
```

Browser window titled "RazorWebAppClient" showing the URL <https://localhost:44360/locaties>. The page displays a table with location data.

header

LocationId	Postcode	City
1	3500	Hasselt
2	3600	Genk

footer

```
public class Databank
{
    public static List<Klant> Klanten = new List<Klant>();
    public static List<Location> Locations = new List<Location>();

    public static void StartDatabank()
    {
        Klanten.Add(new Klant(1, "Klant A"));
        Klanten.Add(new Klant(2, "Klant B"));
        //Locations
        Locations.Add(new Location(1, "3500", "Hasselt"));
        Locations.Add(new Location(2, "3600", "Genk"));
    }
}
```

Hyperlink naar de razor pagina's in de layout page

- Add Razor page SelecteerLocatie.cshtml

_Layout.cshtml

- TagHelper
 - asp-page

<header>

[Startpagina](#)

[Klanten](#)

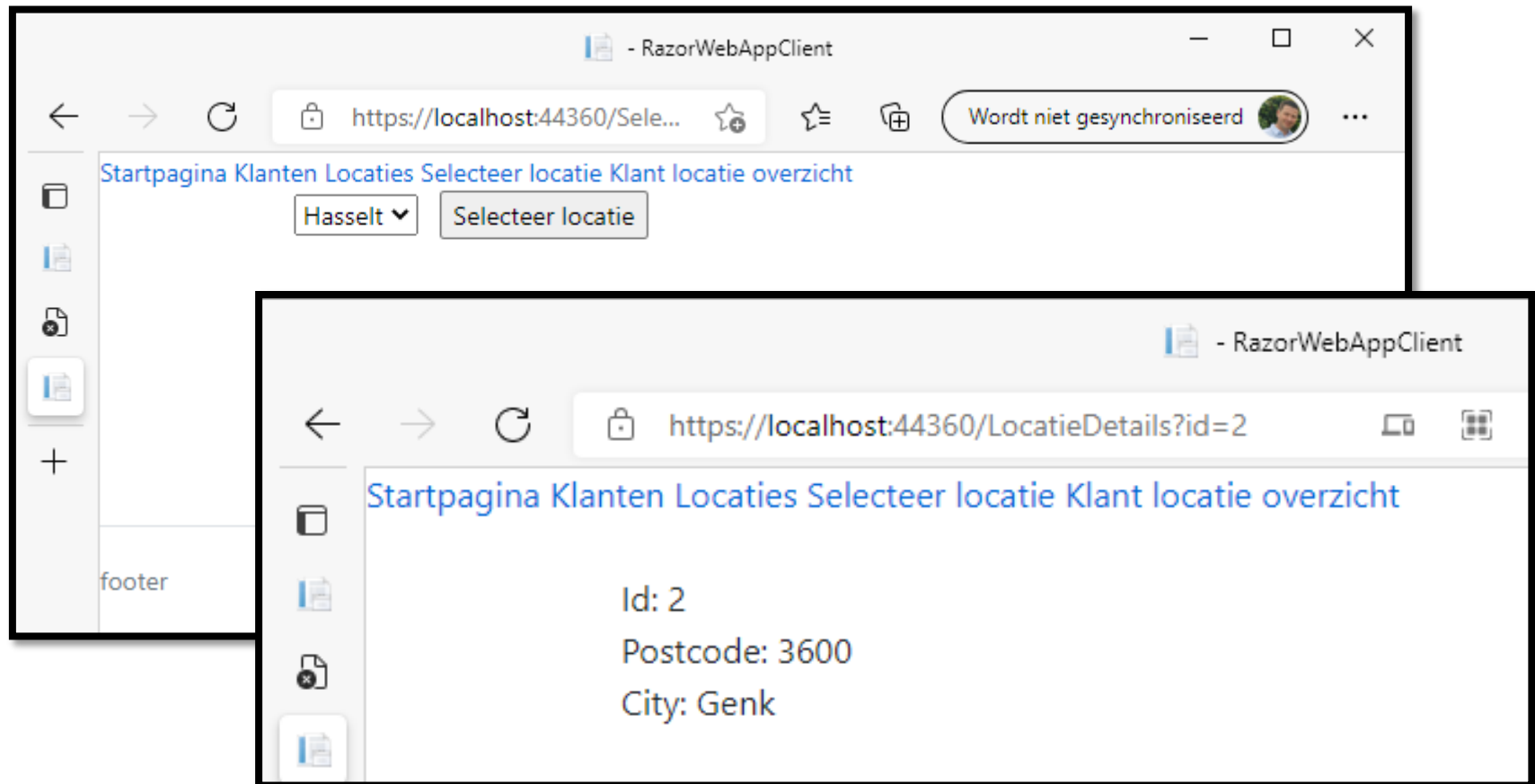
[Locaties](#)

[Selecteer locatie](#)

</header>

We voegen een nieuwe Razor Page toe

- SelecteerLocatie.cshtml
- LocatieDetails.cshtml



We voegen een nieuwe Razor Page toe

- SelecteerLocatie.cshtml

```
@page
@model RazorWebAppClient.Pages.SelecteerLocatieModel
@{
    var locaties = Data.Databank.Locations;
    var list = locaties.Select(x => new SelectListItem
    {
        Text = x.City,
        Value = x.LocationId.ToString()
    });
}
<div>
    <form method="post">
        <select name="SelectLocatie" asp-items="@list"></select>
        <label>&nbsp;</label>
        <input type="submit" value="Selecteer locatie" class="submit" />
    </form>
</div>
```

SelecteerLocatie.cshtml

- OnPost() in codebehind page

```
namespace RazorWebAppClient.Pages
{
    public class SelecteerLocatieModel : PageModel
    {
        public void OnGet()
        {
        }
        public IActionResult OnPost()
        {
            var locId= Request.Form["SelectLocatie"].FirstOrDefault();
            return RedirectToPage("LocatieDetails",
                new { id = int.Parse(locId) });
        }
    }
}
```

We voegen een nieuwe Razor Page toe

- LocatieDetails.cshtml

```
@page
@model RazorWebAppClient.Pages.LocatieDetailsModel
@{
    var id = Model.LocationId;
    var locatieRecord = Data.Databank.Locations.Where(x =>
x.LocationId == id).FirstOrDefault();
    var postCode = locatieRecord.PostCode;
    var city = locatieRecord.City;
}
<div>
    <br />
    Id: @id
    <br />
    Postcode: @postCode
    <br />
    City: @city
</div>
```

LocatieDetails.cshtml

- OnGet() in codebehind page

```
namespace RazorWebAppClient.Pages
{
    public class LocatieDetailsModel : PageModel
    {
        public int LocationId { get; set; }
        public void OnGet(int id)
        {
            LocationId = id;
        }
    }
}
```

We voegen een nieuwe Razor Page toe

- NieuweLocatie.cshtml
- Hyperlink naar de nieuwe pagina in Locaties.cshtml

```
<a asp-page="NieuweLocatie">Nieuwe locatie</a>
```

```
<form method="post">
  <fieldset>
    <legend>Locatie toevoegen</legend>
    <div>
      <label for="PostCode">Postcode:</label>
      <input type="text" name="PostCode" />
      <br />
      <label for="PostCode">City:</label>
      <input type="text" name="City" />
      <br />
      <div>
        <label>&nbsp;</label>
        <input type="submit" value="Submit" class="submit" />
      </div>
    </div>
  </fieldset>
</form>
```

Databank klasse uitbreiden

- Functie AddLocation(string postCode, string city) toevoegen
- Gebruik van Linq voor berekening van nieuwe Id

```
public static void AddLocation(string postCode, string city)
{
    int id = Locations.Max(x => x.LocationId) + 1;
    Locations.Add(new Location(id, postCode, city));
}
```

Post methode in de code-behind pagina

- NieuweLocatie.cshtml.cs

```
public class NieuweLocatieModel : PageModel
{
    public IActionResult OnPost()
    {
        string postCode = Request.Form["PostCode"];
        string city = Request.Form["City"];
        Data.Databank.AddLocation(postCode, city);
        return RedirectToPage("Locaties");
    }
}
```


ASP.NET CORE

Razor Web Application – Oefening02

Razor Web Application – Oefening03