



C# Web – MVC

PRO/PRW - 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
 - MVC Web Application
 - TagHelper
 - SportStore
 - ViewComponent
 - SportStore
 - New Models
 - Migration
 - » Table Reference keys
 - Overview

ASP.NET CORE

MVC Web Application

MVCSportStore

- TagHelper
- ViewComponent
- Models
 - Table Reference keys

Models - Reseller

```
namespace MVCSportStore.Models
{
    public class Reseller
    {
        public int ResellerId { get; set; }
        public string ResellerName { get; set; }
        public string ContentManagementGuid { get; set; }
    }
}
```

Controllers - ResellerController

Add Empty MVC Controller

- ResellerController

```
using Microsoft.AspNetCore.Mvc;

namespace MVCSportStore.Controllers
{
    public class ResellerController : Controller
    {
        public IActionResult Index()
        {
            return View();
        }
    }
}
```

Views – Views/Reseller

×

Add Razor View

View name

Index

Template

List

Model class

Reseller (MVCSportStore.Models)

DbContext class

StoreDbContext (MVCSportStore.Data)

Options

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page

...

(Leave empty if it is set in a Razor _viewstart file)

Add

Cancel

Data – StoreDbContext Resellers

```
public class StoreDbContext : DbContext
{
    public StoreDbContext(DbContextOptions<StoreDbContext> options) :
        base(options)
    {
    }

    //DbSet - -> kopellingen naar onze tabellen
    public DbSet<Product>? Products { get; set; }
    //DbSet - -> kopellingen naar onze tabellen
    public DbSet<Reseller>? Resellers { get; set; }
}
```

Controllers - ResellerController

- Dependency Injection – StoreDbContext
 - `_context.Resellers`

```
public class ResellerController : Controller
{
    StoreDbContext _context;
    public ResellerController(StoreDbContext context)
    {
        _context = context;
    }
    public IActionResult Index()
    {
        return View(_context.Resellers);
    }
}
```


Nuget Package Manager - Console

- Migration
 - Reseller

```
Package Manager Console
Package source: nuget.org
Default project: MVCSportStore

PM> add-migration reseller
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Infrastructure[10403]
    Entity Framework Core 6.0.23 initialized 'StoreDbContext'
To undo this action, use Remove-Migration.
PM> update-database
```

Views/Shared - _layout

```
<span class="navbar-brand ml-2">  
  <a class="btn border border-secondary m-1 text-white"  
    asp-controller="Home" asp-action="Index">  
    SportStore - Start page</a>  
  <a class="btn border border-secondary m-1 text-white"  
    asp-controller="Settings" asp-action="PagingSettingsView">  
    SportStore - Page settings</a>  
  
  <a class="btn border border-secondary m-1 text-white"  
    asp-controller="Reseller" asp-action="Index">  
    SportStore - Resellers</a>  
</span>
```

SportStore - Start page

SportStore - Page settings

SportStore - Resellers

Chess

Soccer

Watersports

Resellers

ResellerName

ContentManagementGuid

SportStore - Start page

SportStore - Page settings

SportStore - Resellers

Data/DefaultData - SeedData

```
public static void EnsurePopulated(WebApplication app)
{
    using (var scope = app.Services.CreateScope())
    {
        var context =
scope.ServiceProvider.GetRequiredService<StoreDbContext>();
        if (!context.Products.Any())
        {
            ...
        }
        MatchResellerData(context);
    }
}
private static void MatchResellerData(StoreDbContext context)
{ }
```

Program.cs – DataFolder key

```
...
app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");

var dataFolder =
    builder.Configuration.GetSection("DataFolder").Value;

AppStartBus.SetDataFolder(dataFolder);

SeedData.EnsurePopulated(app);

app.Run();
```

Data/DefaultData - SeedData

```
public static void EnsurePopulated(WebApplication app)
{
    using (var scope = app.Services.CreateScope())
    {
        var context =
scope.ServiceProvider.GetRequiredService<StoreDbContext>();
        if (!context.Products.Any())
        {
            ...
        }
        MatchResellerData();
    }
}
public static void MatchResellerData()
{ }
```

Data/DefaultData/SeedData - Match

```
public static void MatchResellerData(StoreDbContext context)
{
    var resellers = ResellerBus.ResellerEntities;
    bool entityAdded = false;
    foreach (var reseller in resellers)
    {
        if (!context.Resellers
            .Where(x => x.ContentManagementGuid ==
                      reseller.ContentManagementGuid)
            .Any())
        {
            //add reseller
        }
    }
}
```

Data/DefaultData/SeedData - Match

```
if (!context.Resellers
    .Where(x => x.ContentManagementGuid ==
              reseller.ContentManagementGuid)
    .Any())
{
    var newReseller = new Reseller();
    newReseller.ResellerName = reseller.ResellerName;
    newReseller.ContentManagementGuid =
        reseller.ContentManagementGuid;
    context.Resellers.Add(newReseller);
    entityAdded = true;
}
```


Resellers

SPORT STORE

Chess

Soccer

Watersports

Resellers

ResellerName	ContentManagementGuid
Bol.com	ff3ddf87-a2da-4c3f-92ff-fa7816d62364

SportStore - Start page

SportStore - Page settings

SportStore - Resellers

Oefening

- Voeg een model ResellerStock toe!
 - Properties
 - ResellerStockId
 - ResellerId
 - ProductId
 - Amount
- Voeg de table reference keys toe!
- Maak een migratie aan!
- Update de database!

Models - ResellerStock

```
public class ResellerStock
{
    public int ResellerStockId { get; set; }
    public int ResellerId { get; set; }
    public long ProductId { get; set; }
    public int Amount { get; set; }

    public Reseller? Reseller { get; set; }
    public Product? Product { get; set; }
}
```

Models - Reseller

```
public class Reseller
{
    public int ResellerId { get; set; }
    public string ResellerName { get; set; }
    public string ContentManagementGuid { get; set; }

    public ICollection<ResellerStock>? ResellerStocks { get; set; }
}
```

Models - Product

```
public class Product
{
    ...
    //get property
    public decimal ProductPrice =>
        (Price == null) ? 0 : (decimal)Price;

    public ICollection<ResellerStock>? ResellerStocks { get; set; }
}
```

Data – StoreDbContext ResellerStocks

```
public class StoreDbContext : DbContext
{
    public StoreDbContext(... options) : base(options)
    { }
    public DbSet<Product> Products { get; set; }
    public DbSet<Reseller>? Resellers { get; set; }
    public DbSet<ResellerStock>? ResellerStocks { get; set; }
}
```

Nuget Package Manager - Console

- Migration
 - ResellerStock

Data/DefaultData - SeedData

```
public static void EnsurePopulated(WebApplication app)
{
    using (var scope = app.Services.CreateScope())
    {
        var context =
scope.ServiceProvider.GetRequiredService<StoreDbContext>();
        if (!context.Products.Any())
        {
            ...
        }
        MatchResellerData(context);
        MatchResellerStockData(context);
    }
}

public static void MatchResellerStockData(StoreDbContext context)
{ }
```


ViewModels - ResellerViewModel

```
private static void MatchResellerStockData(StoreDbContext context)
{
    var resellerStocks = ResellerStockBus.ResellerStockEntities;
    bool entityAdded = false;
    foreach (var resellerStock in resellerStocks)
    {
        if (!context.ResellerStocks.Any())
        {
            var newResellerStock = new ResellerStock();
            var reseller = context.Resellers
                .Where(x => x.ContentManagementGuid == resellerStock.ResellerGuid)
                .FirstOrDefault();
            newResellerStock.ResellerId = reseller.ResellerId;
            newResellerStock.ProductId = resellerStock.ProductId;
            newResellerStock.Amount = resellerStock.Amount;
            context.ResellerStocks.Add(newResellerStock);
            entityAdded = true;
        }
    }
    if(entityAdded)
    { context.SaveChanges(); }
}
```

ViewModels - ResellerViewModel

```
namespace MVCSportStore.ViewModels
{
    public class ResellerViewModel
    {
        public int ResellerId { get; set; }
        public string ResellerName { get; set; }
        public IEnumerable<ResellerStock> ResellerStocks { get; set; }
    }
}
```

ViewModels - ProductModel

```
namespace MVCSportStore.ViewModels
{
    public class ProductModel
    {
        public IEnumerable<Product> Products { get; set; }
        public PagingInfo PagingInfo { get; set; }
        public IEnumerable<ResellerViewModel> Resellers { get; set; }
    }
}
```

ASP.NET CORE

MVC Web Application
Oefening 14

- **ViewComponent**