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
  - Identity Framework
    - Identity User
    - Identity Role

Identity Framework - Overview

Solution explorer

- Identity Package
- Controllers
  - AccountController
    - Register
    - Login
    - Logout
- Models/ViewModels
  - LoginViewModel
  - RegisterViewModel
- Views
  - Views/Account
    - Login.cshtml
    - Register.cshtml
  - Views/Shared
    - _Layout
    - _LoginPartial

ASP.NET CORE

MVC Web Application
MVCGroentenEnFruit
Teams – Attachments - Identity

Identity Framework - MVCGroentenEnFruit

Solution explorer

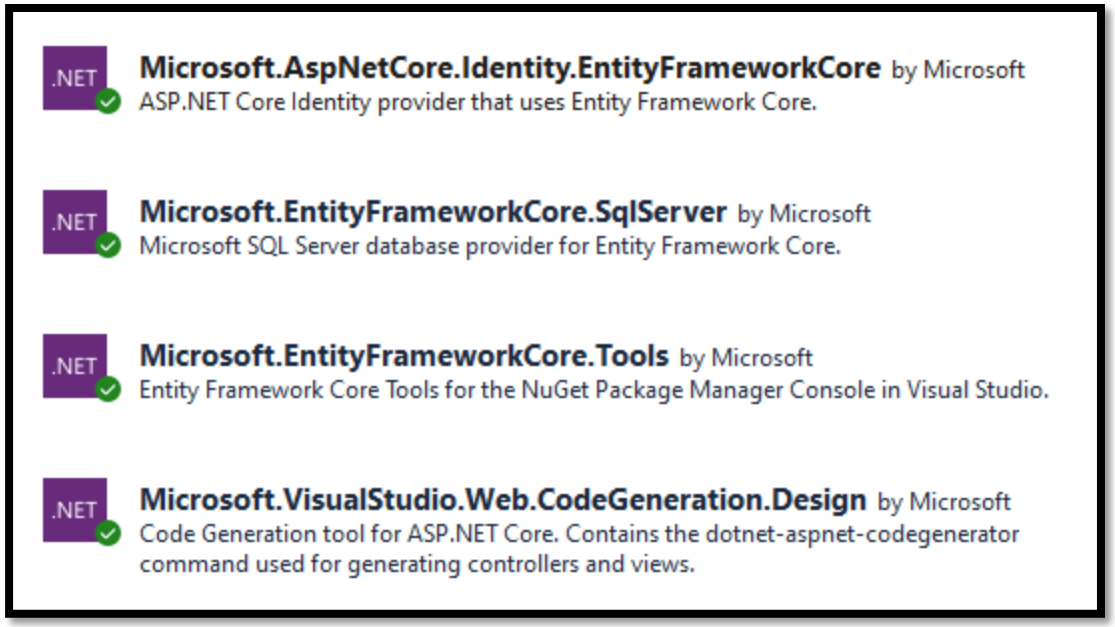
- Packages
  - EF - SQL Server
  - EF - Tools
- Controllers
  - AccountController
  - ArtikelsController
- Models/Data
  - Artikel
  - AankoopOrder
  - VerkoopOrder
- Data
  - AppDbContext
  - DefaultData
    - SeedData
- Program.cs
  - AddDbContext -> service voor sqlserver

Identity Framework - MVCGroentenEnFruit

Solution explorer

- Identity Package
- Controllers
  - AccountController
    - Register
    - Login
    - Logout
- Models/ViewModels
  - LoginViewModel
  - RegisterViewModel
- Views
  - Views/Account
    - Login.cshtml
    - Register.cshtml
  - Views/Shared
    - _Layout
    - _LoginPartial

Identity Framework – MVCGroentenEnFruit Packages



Identity Framework – Program.cs Services

Program.cs

```
builder.Services.AddControllersWithViews();

var connectionString =
        builder.Configuration.GetConnectionString("AppDbConn");
//Entity framework - SQL
builder.Services.AddDbContext<AppDbContext>(
        options => options.UseSqlServer(connectionString));
//Entity framework - Identity
builder.Services.AddIdentity<IdentityUser, IdentityRole>()
        .AddEntityFrameworkStores<AppDbContext>();
```

Identity Framework – Program.cs Middleware

Program.cs

```
…
app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

…
```

Identity Framework – Program.cs default data

Program.cs

```
…
app.UseRouting();

app.UseAuthentication();

app.UseAuthorization();

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

SeedData.EnsurePopulated(app);


…
```

Identity Framework – Data AppDbContext

```
namespace MVCGroentenEnFruit.Data
{
    public class AppDbContext : IdentityDbContext
    {
        public AppDbContext(DbContextOptions<AppDbContext> options) :
                                base(options)
                                {
                                }
        public DbSet<Artikel>? Artikels { get; set; }
        public DbSet<AankoopOrder>? AankoopOrders { get; set; }
        public DbSet<VerkoopOrder>? VerkoopOrders { get; set; }
    }
}
```

Identity Framework – Data Migration

Nuget Package Manager

- Add-migration identity
- Update-database

Identity Framework - MVCGroentenEnFruit

Solution explorer

- Identity Package
- Controllers
  - AccountController
    - Register
    - Login
    - Logout
- Models/ViewModels
  - LoginViewModel
  - RegisterViewModel
- Views
  - Views/Account
    - Login.cshtml
    - Register.cshtml
  - Views/Shared
    - _Layout
    - _LoginPartial

Identity Framework - MVCGroentenEnFruit

Identity Framework - MVCGroentenEnFruit


```
[HttpGet]
Register()
{
  //Roles
}

[HttpPost]
Register(RegisterViewModel)
{
  //Create IdentityUser
}
```

```
<div class="form-group">
    <label asp-for="RoleId" class="control-label"></label>
    <select asp-for="RoleId" class="form-control" asp-items="ViewBag.RoleId"></select>
    <span asp-validation-for="RoleId" class="text-danger"></span>
</div>
```

IdentityRole - MVCGroentenEnFruit

- Data/DefaultData folder
  - Add static class Roles.cs

```
namespace MVCGroentenEnFruit.Data.DefaultData
{
    public static class Roles
    {
        public const string Aankoper = "Aankoper";
        public const string Verkoper = "Verkoper";
    }
}
```

The recommended naming and capitalization convention is to use PascalCasing for constants (Microsoft has a tool named StyleCop that documents all the preferred conventions and can check your source for compliance - though it is a little bit *too* anally retentive for many people's tastes). e.g.

private const int TheAnswer = 42;

IdentityRole – SeedData - Services

- **Data/DefaultData/SeedData.cs**

```
namespace MVCGroentenEnFruit.Data.DefaultData
{
    public static class SeedData
    {
public static void EnsurePopulated(WebApplication app)
        {
            using (var scope = app.Services.CreateScope())
            {
                var _context =
scope.ServiceProvider.GetRequiredService<AppDbContext>();
                var _roleManager =
scope.ServiceProvider.GetRequiredService<RoleManager<IdentityRole>>();
            }
        }
```

IdentityRole – SeedData - Functions

- **Data/DefaultData/SeedData.cs**

```
namespace MVCGroentenEnFruit.Data.DefaultData
{
    public static class SeedData
    {
        …
        private static async Task VoegRolToeAsync(
  RoleManager<IdentityRole> _roleManager,string roleName)
        {
                if (!await _roleManager.RoleExistsAsync(roleName))
                {
                        IdentityRole role = new IdentityRole(roleName);
                        await _roleManager.CreateAsync(role);
                }
        }
        …

     }
}
```

IdentityRole – SeedData - Functions

- **Data/DefaultData/SeedData.cs**

```
namespace MVCGroentenEnFruit.Data.DefaultData
{
    public static class SeedData
    {
        …
        private static async Task VoegRollenToeAsync(
                AppDbContext _context, RoleManager<IdentityRole> _roleManager)
        {
                if (!_context.Roles.Any())
                {
                        await VoegRolToeAsync(_roleManager, Roles.Aankoper);
                        await VoegRolToeAsync(_roleManager, Roles.Verkoper);
                }
        }
        …

    }
}
```

IdentityRole – SeedData - Functions

- **Data/DefaultData/SeedData.cs**

```
public static class SeedData
{
    public static async Task EnsurePopulatedAsync(WebApplication app)
    {
        using (var scope = app.Services.CreateScope())
        {

var _context = scope.ServiceProvider.GetRequiredService<AppDbContext>();
var _roleManager =
scope.ServiceProvider.GetRequiredService<RoleManager<IdentityRole>>();

        await VoegRollenToeAsync(_context, _roleManager);

        }
}
```

Identity - Models/ViewModels

```
namespace MVCGroentenEnFruit.Models.ViewModels
{
    public class LoginViewModel
    {
        public string? Email { get; set; }
        public string? Password { get; set; }
    }
}
```

SCAFFOLD VIEW Login -> Views/Account (Add Temp int Id)

Identity - Models/ViewModels

```
namespace MVCGroentenEnFruit.Models.ViewModels
{
    public class RegisterViewModel : LoginViewModel
    {
        public string? RoleId { get; set; }
    }
}
```

Identity Framework - MVCGroentenEnFruit


```
[HttpGet]
Register()
{
  //Roles
}

[HttpPost]
Register(RegisterViewModel)
{
  //Create IdentityUser
}
```

```
<div class="form-group">
    <label asp-for="RoleId" class="control-label"></label>
    <select asp-for="RoleId" class="form-control" asp-items="ViewBag.RoleId"></select>
    <span asp-validation-for="RoleId" class="text-danger"></span>
</div>
```

AccountController – Dependency Injection

```
public class AccountController : Controller
{
    AppDbContext _context;
    public AccountController(AppDbContext context)
    {
        _context = context;
    }
    …
```

IdentityRole – AccountController - Register

```
[HttpGet]
public IActionResult Register()
{
    ViewData["RoleId"] = new SelectList(_context.Roles, "Id", "Name");
    return View();
}
```

Views/Shared/_Layout

```
<ul class="navbar-nav flex-grow-1">
    <li class="nav-item">
        <a class="nav-link text-dark" asp-controller="Home"
            asp-action="Index">Home</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-controller="Account"
            asp-action="Register">Register</a>
    </li>
    <li class="nav-item">
        <a class="nav-link text-dark" asp-controller="Account"
            asp-action="Login">Login</a>
    </li>
</ul>
```

IdentityRole – AccountController – Register Dependency injection

```
public class AccountController : Controller
{
    AppDbContext _context;
    UserManager<IdentityUser> _userManager;
    public AccountController(AppDbContext context,
                             UserManager<IdentityUser> userManager)
    {
        _context = context;
        _userManager = userManager;
    }
```

IdentityRole – AccountController - Register

```
[HttpPost]
public async Task<IActionResult> RegisterAsync(RegisterViewModel registerViewModel)
{
    if (ModelState.IsValid)
    {
        if (registerViewModel.RoleId != null)
        {
            var identityUser = new IdentityUser();
            identityUser.Email = registerViewModel.Email;
            identityUser.UserName = registerViewModel.Email;
            var identityResult = await _userManager.CreateAsync(identityUser,
                                                    registerViewModel.Password);
            if (identityResult.Succeeded)
            {
                return View("Login");
            }
            foreach (var error in identityResult.Errors)
            {
                ModelState.AddModelError("", error.Description);
            }
        }
        else
        {
            ModelState.AddModelError("", "Geen rol geselecteerd!");
        }
    }
    return View();
}
```

IdentityRole – AccountController - Register

```
…
var identityResult = await _userManager.CreateAsync(
         identityUser, registerViewModel.Passwoord);

if (identityResult.Succeeded)
{
    var roleName = _context.Roles.Where(x=>x.Id == rvm.RoleId).First().Name;
    var roleResult = await _userManager.AddToRoleAsync(
         identityUser, roleName);

    if (roleResult.Succeeded)
        return View("Login");
    else
    {
        ModelState.AddModelError("", "Problemen met toekennen van rol!");
        return View();
    }
}
..
```

IdentityUser - Register

https://localhost:7230/Account/Register

Register users

- aankoper@pxl.be
  - Aankoper00!
    - Role: **Aankoper**
- verkoper@pxl.be
  - Verkoper00!
    - Role: **Verkoper**