



C# Web – MVC

PRO/PRW - C# Web 1

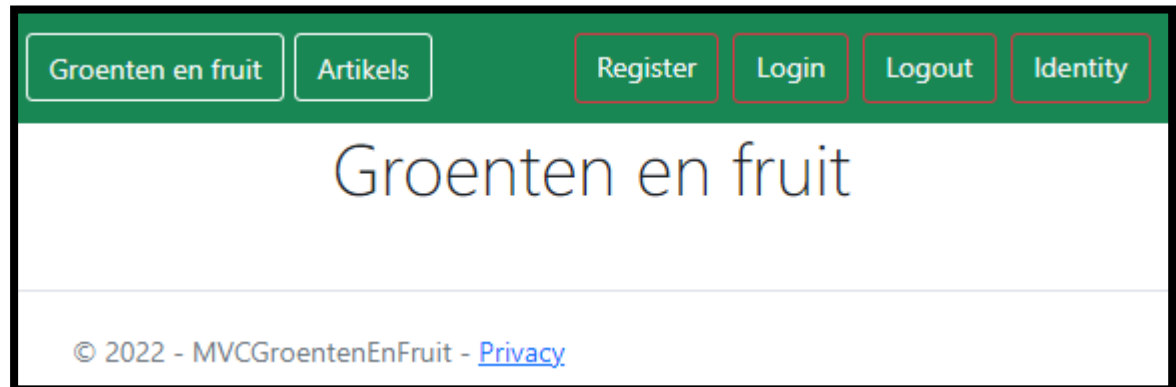
DE HOGESCHOOL MET HET NETWERK

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



Doel

- ASP.Net Core toepassing
 - Identity Framework
 - Identity User – User Claim
 - Identity Role
 - Identity Options



Oefening

- Voeg een StockController toe
- Gebruik een ViewModel
 - StockViewModel
 - ArtikelId
 - ArtikelNaam
 - Hoeveelheid
- StockController/Dependency Injection
 - ApplicationDbContext
- StockController – Index
 - Overzicht van IEnumerable<StockViewModel>
- VerkoopOrders
 - Voeg een modelstate error toe wanneer er meer artikels verkocht worden dan er in stock zijn.

Oefening

[Groenten en fruit](#)[Artikels](#)[Aankoop orders](#)[Verkoop orders](#)[Stock](#)[Register](#)[Login](#)[Logout](#)[Identity](#)

Stock

ArtikelNaam	Hoeveelheid
Ijsbergsla	4
Roma tomaat	3

AANKOPEN

Hoeveelheid	Artikel
5	Ijsbergsla
4	Roma tomaat

VERKOPEN

Hoeveelheid	Naam
1	Ijsbergsla
1	Roma tomaat

AccountController – UserClaim

Views/Account/Identity.cshtml

[Groenten en fruit](#) [Artikels](#) [Aankoop orders](#) [Verkoop orders](#) [Stock](#) [Register](#) [Login](#) [Logout](#) [Identity](#)

[Create Role](#) [Claim](#)

RoleName

Stockbeheerder

Verkoper

Aankoper

AccountController – UserClaim

Views/Account/Identity.cshtml

[Groenten en fruit](#) [Artikels](#) [Aankoop orders](#) [Verkoop orders](#) [Stock](#) [Register](#) [Login](#) [Logout](#) [Identity](#)

[Create Role](#) [Claim](#)

RoleName

[Groenten en fruit](#) [Artikels](#) [Aankoop orders](#) [Verkoop orders](#) [Stock](#)

AccountController – UserClaim

Views/Account/Identity.cshtml

[Groenten en fruit](#) [Artikels](#) [Aankoop orders](#) [Verkoop orders](#) [Stock](#) [Register](#) [Login](#) [Logout](#) [Identity](#)

[Create Role](#) [Claim](#)

RoleName

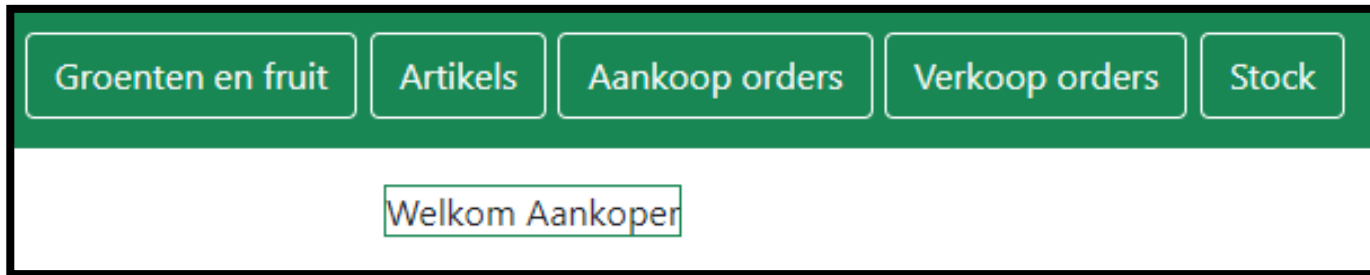
```
@model IdentityViewModel
<br />
<a class="btn btn-block link-dark border-success" asp-controller="Account"
    asp-action="CreateRole">Create Role</a>
<a class="btn btn-block link-dark border-success" asp-controller="Account"
    asp-action="UserClaim">Claim</a>
<br />
```

AccountController – UserClaim

```
[HttpGet]  
public IActionResult UserClaim()  
{  
    return View();  
}
```


AccountController – UserClaim

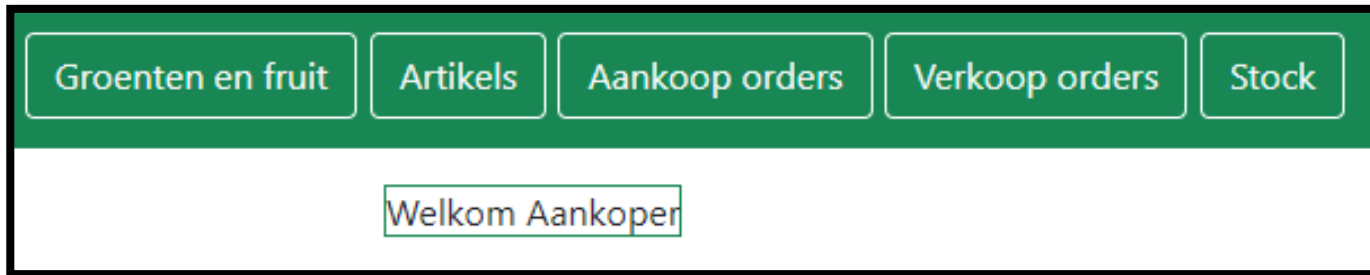
Views/_ViewImports



```
@using MVCGroentenEnFruit
@using MVCGroentenEnFruit.Models
@using MVCGroentenEnFruit.Models.ViewModels
@using Microsoft.AspNetCore.Identity;
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
@addTagHelper *, MVCGroentenEnFruit
```

AccountController – UserClaim

Views/Account/UserClaim.cshtml



```
@model IdentityUser
<div class="m-3">
    <span class="border border-success m-2">
        Welkom @Model.UserName
    </span>
</div>
```

AccountController – UserClaim

- We gebruiken het object User (System.Security.Claims.ClaimsPrincipal)
- We gaan de IdentityUser opzoeken met de Identity functie GetUser()
- Als we de IdentityUser hebben gevonden -> UserClaim View
- Anders gaan we doorverwijzen naar Login pagina

```
[HttpGet]
public async Task<IActionResult> UserClaimAsync()
{
    var user = User;
    var identityUser = await _userManager.GetUserAsync(user);
    if (identityUser != null)
        return View("UserClaim", identityUser);
    return View("Login");
}
```

AccountController – UserClaim

- We gebruiken het object User (System.Security.Claims.ClaimsPrincipal)
- We gaan de IdentityUser opzoeken met de Identity functie GetUser()
- Als we de IdentityUser hebben gevonden -> UserClaim View
- Anders gaan we doorverwijzen naar Login pagina

```
[HttpGet]
public async Task<IActionResult> UserClaim2Async()
{
    if (_signInManager.IsSignedIn(User))
    {
        var identityUser = await _userManager.GetUserAsync(User);
        if (identityUser != null)
            return View("UserClaim", identityUser);
    }
    return View("Login");
}
```

TagHelpers

TagHelper

- We gebruiken het object `User` (`System.Security.Claims.ClaimsPrincipal`)

```
public class UserTagHelper : TagHelper
{
    [ViewContext]
    [HtmlAttributeNotBound]
    public ViewContext ViewContext { get; set; }

    public override void Process (...)
    {
        var user = ViewContext.HttpContext.User;
        //user = ClaimsPrincipal dus niet de identityUser
    }
}
```

Oefening

AccountController – UserClaim

TagHelper - UserCard

The screenshot shows a web application interface. At the top is a green navigation bar with buttons for 'Groenten en fruit', 'Artikels', 'Aankoop orders', 'Verkoop orders', 'Stock', 'Register', 'Login', 'Logout', and 'Identity'. Below the navigation bar, the text 'Welkom Aankoper' is displayed. Below this, there is a white box titled 'Identity Card'. Inside this box, the label 'EMAIL' is followed by the text 'testEmail', and the label 'USERNAME' is followed by the text 'testUserName'.

```
class="card"  
Class="card-title"  
Class="card-subtitle mb-2 text-muted p-2"  
Class="card-text m-1 p-3"
```

Oefening

AccountController – UserClaim TagHelper - UserCard

- Dependency Injection – UserManager
- ViewContext – Property – [ViewContext] Attribute

```
namespace MVCGroentenEnFruit.TagHelpers
{
    [HtmlTargetElement("user")]
    public class UserTagHelper : TagHelper
    {
        UserManager<IdentityUser> _userManager;
        public UserTagHelper(UserManager<IdentityUser> userManager)
        {
            _userManager = userManager;
        }
        [ViewContext]
        [HtmlAttributeNotBound]
        public ViewContext ViewContext { get; set; }
```

Oefening

AccountController – UserClaim

TagHelper - UserCard

```
private TagBuilder UserCard(IdentityUser user)
{
    TagBuilder divCard = new TagBuilder("div");
    return divCard;
}
private TagBuilder UserCardTitle(string title)
{
    TagBuilder titleH5 = new TagBuilder("h5");
    return titleH5;
}
private TagBuilder UserCardItem(string header, string item)
{
    TagBuilder cardItem = new TagBuilder("h6");
    return cardItem;
}
```


Oefening

AccountController – UserClaim TagHelper - UserCard

```
public override async Task ProcessAsync(TagHelperContext context, TagHelperOutput output)
{
    var user = ViewContext.HttpContext.User;
    var identityUser = await _userManager.GetUserAsync(user);
    output.Content.SetHtmlContent(UserCard(identityUser));
}
```

Oefening

AccountController – UserClaim TagHelper - UserCard

```
private TagBuilder UserCard(IdentityUser user)
{
    TagBuilder divCard = new TagBuilder("div");
    divCard.Attributes["class"] = "card";
    divCard.Attributes["style"] = "width: 18rem;";
    //Title
    divCard.InnerHtml.AppendHtml(UserCardTitle("Identity Card"));
    if (user != null)
    {
        //email
        divCard.InnerHtml.AppendHtml(UserCardItem("EMAIL", user.Email));
        //username
        divCard.InnerHtml.AppendHtml(UserCardItem("USERNAME", user.UserName));
    }
    return divCard;
}
```

Oefening

AccountController – UserClaim

TagHelper - UserCard

```
private TagBuilder UserCardTitle(string title)
{
    TagBuilder titleH5 = new TagBuilder("h5");
    titleH5.Attributes["class"] = "card-title";
    titleH5.InnerHtml.Append(title);
    return titleH5;
}

private TagBuilder UserCardItem(string header, string item)
{
    TagBuilder cardItem = new TagBuilder("h6");
    cardItem.Attributes["class"] = "card-subtitle mb-2 text-muted p-2";
    cardItem.InnerHtml.Append(header);
    TagBuilder cardText = new TagBuilder("p");
    cardText.Attributes["class"] = "card-text m-1 p-3";
    cardText.InnerHtml.Append(item);
    cardItem.InnerHtml.AppendHtml(cardText);
    return cardItem;
}
```

AccountController – UserClaim

Views/Account/UserClaim.cshtml

UserTagHelper

```
@model IdentityUser
<div class="m-3">
    <span class="border border-success m-2">
        Welkom @Model.UserName
    </span>
</div>

<user></user>
```

Identity Framework – Program.cs

Identity options (default properties)

```
builder.Services.Configure<IdentityOptions>(options =>
{
    // Password settings.
    options.Password.RequireDigit = true;
    options.Password.RequireLowercase = true;
    options.Password.RequireNonAlphanumeric = true;
    options.Password.RequireUppercase = true;
    options.Password.RequiredLength = 6;
    options.Password.RequiredUniqueChars = 1;

    // Lockout settings.
    options.Lockout.DefaultLockoutTimeSpan = TimeSpan.FromMinutes(5);
    options.Lockout.MaxFailedAccessAttempts = 5;
    options.Lockout.AllowedForNewUsers = true;

    // User settings.
    options.User.AllowedUserNameCharacters =
        "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-._@+";
    options.User.RequireUniqueEmail = false;
});
```

Identity options

Password

- 5 chars
- No capital
- No alpha-numeric
- Number is required

```
builder.Services.Configure<IdentityOptions>(
options =>
{
    // Password settings.
    options.Password.RequireDigit = true;
    options.Password.RequireLowercase = true;
    options.Password.RequireNonAlphanumeric = true;
    options.Password.RequireUppercase = true;
    options.Password.RequiredLength = 6;
    options.Password.RequiredUniqueChars = 1;
```

Identity options

Password

- 5 chars
- No capital
- No alpha-numeric
- Number is required

```
builder.Services.Configure<IdentityOptions>(options =>
{
    // Password settings.
    options.Password.RequireDigit = true;
    options.Password.RequireNonAlphanumeric = false;
    options.Password.RequireUppercase = false;
    options.Password.RequiredLength = 5;
});
```