



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

ASP.NET CORE

MVC Web Application
MVCPartyInvites

Identity Framework

PXL Party - header

Register

Login

[Schrijf je hier in!](#)

Aanwezige gasten	
Naam	Email
Kristof	kristof.palmaers@pxl.be

Afwezige gasten	
Naam	Email
Koen	koen.bloemen@pxl.be

PXL Party - footer

Identity Framework

- Register action

```
var result = await _userManager.CreateAsync(  
    user, Input.Password);
```

- Signin action

```
await _signInManager.SignInAsync(user, isPersistent: false);
```

- Errors – Identity result

```
foreach (var error in result.Errors)  
{  
    ModelState.AddModelError(string.Empty, error.Description);  
}
```

MVCPartyInvites - Identity

- LoginViewModel

```
namespace MVCPartyInvites.ViewModels
{
    public class LoginViewModel
    {
        public int Id { get; set; }
        [EmailAddress]
        public string Email { get; set; }
        public string Password { get; set; }
    }
}
```

MVCPartyInvites - Identity

- LoginViewModel

```
namespace MVCPartyInvites.ViewModels
{
    public class LoginViewModel
    {
        public int Id { get; set; }
        [EmailAddress]
        public string Email { get; set; }
        [DataType(DataType.Password)]
        public string Password { get; set; }
    }
}
```

MVCPartyInvites - Identity

- RegisterViewModel

```
namespace MVCPartyInvites.ViewModels
{
    public class RegisterViewModel : LoginViewModel
    {
        //->compare
        public string ConfirmPassword { get; set; }
    }
}
```


MVCPartyInvites - Identity

- RegisterViewModel

```
namespace MVCPartyInvites.ViewModels
{
    public class RegisterViewModel : LoginViewModel
    {
        [DataType(DataType.Password)]
        [Compare(nameof(Password),
            ErrorMessage = "Passwords do not match!")]
        public string ConfirmPassword { get; set; }
    }
}
```

MVCPartyInvites - Identity

- Login

PXL Party - header

Register

Login

Login

LoginViewModel

Email

Password

Login

[Registreer een nieuwe gebruiker](#)

MVCPartyInvites - Identity

- Login

```
#region Login
[HttpGet]
public IActionResult Login()
{
    return View();
}
[HttpPost]
public IActionResult Login(LoginViewModel loginViewModel)
{
    return View();
}
#endregion
```

MVCPartyInvites - Identity

- Login

```
[HttpPost]
public IActionResult Login(LoginViewModel loginViewModel)
{
    if (ModelState.IsValid)
    {
        return RedirectToAction("Index", "Home");
    }
    return View();
}
```

MVCPartyInvites - Identity

- Login – DI - SignInManager

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

MVCPartyInvites - Identity

- Login – SignInManager - PasswordSignIn

```
public IActionResult Login(LoginViewModel loginViewModel)
{
    if (ModelState.IsValid)
    {
        var identityUser = new IdentityUser();
        identityUser.Email = loginViewModel.Email;
        identityUser.UserName = loginViewModel.Email;
        var result = _signInManager.PasswordSignInAsync(
            identityUser, loginViewModel.Password,
            false, false);

        ...
    }
}
```

MVCPartyInvites - Identity

- Login – SignInManager - Async

```
public async Task<IActionResult> LoginAsync(LoginViewModel  
                                            loginViewModel)  
{  
    if (ModelState.IsValid)  
    {  
        ...  
        var result = await _signInManager.PasswordSignInAsync(  
            identityUser, loginViewModel.Password,  
            false, false);  
    }  
}
```

MVCPartyInvites - Identity

- Login – SignInManager - Async

```
public async Task<IActionResult> LoginAsync(LoginViewModel
                                            loginViewModel)
{
    if (ModelState.IsValid)
    {
        //var identityUser = new IdentityUser();
        var identityUser =
await _userManager.FindByEmailAsync(loginViewModel.Email);
        if (identityUser != null)
        {
            var result = await _signInManager.PasswordSignInAsync(
                identityUser, loginViewModel.Password,
                false, false);
        }
    }
    return View();
}
```


MVCPartyInvites - Identity

- Login – SignInManager - Async

```
private async Task<IdentityUser> SearchIdentityUserAsync(string email)
{
    return await _userManager.FindByEmailAsync(email);
}

public async Task<IActionResult> LoginAsync(LoginViewModel loginViewModel)
{
    if (ModelState.IsValid)
    {
        var identityUser = await SearchIdentityUser(
            loginViewModel.Email);
    }
}
```

MVCPartyInvites - Identity

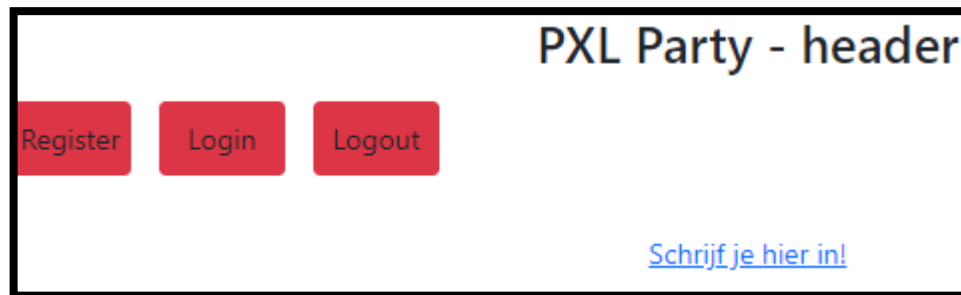
- Login – SignInResult - Errors

```
if (result.Succeeded)
{
    return RedirectToAction("Index", "Home");
}
else
{
    if (result.IsNotAllowed)
    {
        ModelState.AddModelError("", "User not allowed!");
    }
    else
    {
        ModelState.AddModelError("", "Can not login!");
    }
}
```

MVCPartyInvites - Identity

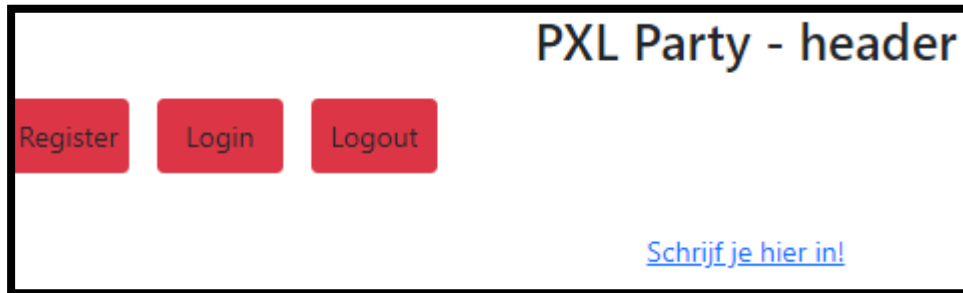
- Logout - _LoginPartial.cshtml

```
<div class="row">
  <a class="btn bg-danger col-1 m-2 p-2"
    asp-controller="Account" asp-action="Register">Register</a>
  <a class="btn bg-danger col-1 m-2 p-2"
    asp-controller="Account" asp-action="Login">Login</a>
  <a class="btn bg-danger col-1 m-2 p-2"
    asp-controller="Account" asp-action="Logout">Logout</a>
</div>
```



MVCPartyInvites - Identity

- AccountController - Logout



```
public async Task<IActionResult> LogoutAsync()
{
    await _signInManager.SignOutAsync();
    return View("Login");
}
```

ASP.NET CORE

Oefening

MVCVoertuig

- Register
- Login
- Logout