



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
 - Models
 - Views
 - Controllers
 - Model Validation
 - *(Custom Model Validation)*

ASP.Net Core MVC

MODEL VALIDATION

Model validation

built-in validation attributes (System.ComponentModel.DataAnnotations namespace)

- [ValidateNever]: The ValidateNeverAttribute indicates that a property or parameter should be excluded from validation.
- [CreditCard]: Validates that the property has a credit card format. Requires jQuery Validation Additional Methods.
- [Compare]: Validates that two properties in a model match.
- [EmailAddress]: Validates that the property has an email format.
- [Phone]: Validates that the property has a telephone number format.
- [Range]: Validates that the property value falls within a specified range.
- [RegularExpression]: Validates that the property value matches a specified regular expression.
- [Required]: Validates that the field is not null. See [Required] attribute for details about this attribute's behavior.
- [StringLength]: Validates that a string property value doesn't exceed a specified length limit.
- [Url]: Validates that the property has a URL format.

MVCPartyInvites

```
public class Gast
{
    [Required]
    public string? Naam { get; set; }
    [Required(ErrorMessage = "Vul je email in!")]
    [EmailAddress]
    public string? Email { get; set; }
    [Required(ErrorMessage = "Vul je telefoonnummer in!")]
    public string? Telefoon { get; set; }
    [Required(ErrorMessage = "Laat weten of je aanwezig zal zijn!")]
    public bool? Aanwezig { get; set; }
}
```

GastController

```
[HttpPost]
public IActionResult Reservatie(Gast g)
{
    if (ModelState.IsValid)
    {
        LocalData.GastList.Add(g);
        return RedirectToAction("Index", "Home");
    }
    else
    {
        return View("Index", g);
    }
}
```

Views/Gast/Index.cshtml

@model Gast

<div>

 @Html.ValidationSummary()

</div>

<form asp-action="Reservatie" method="post">

Views/Gast/Index.cshtml

@model Gast

<form asp-action="Reservatie" method="post">

 <div asp-validation-summary="All" class="text-danger"></div>

PXL Party - header

- The Naam field is required.
- The value '0' is not valid for Aanwezig.

Naam

Email

Telefoon

Aanwezig

Views/Gast/Index.cshtml

@model Gast

```
<form asp-action="Reservatie" method="post">
  <div asp-validation-summary="All" class="text-danger"></div>
  <div>
    <label asp-for="Email"></label>
    <input asp-for="Email" />
    <span asp-validation-for="Email" class="text-danger"></span>
  </div>
```

[EmailAddress]

[Required]

```
public string? Email { get; set; }
```

PXL Party - header

- The Naam field is required.
- The Email field is required.
- The value '0' is not valid for Aanwezig.

Naam

Email The Email field is required.

Telefoon

Aanwezig

Naam

Email

! Geef een '@' op in het e-mailadres. In 'aa' ontbreekt een '@'.

Validatie in de controller

Als de naam kleiner is dan 2 karakters

```
[HttpPost]
public IActionResult Reservatie(Gast g)
{
    if (ModelState.IsValid)
    {
        if (g.Naam == null || g.Naam.Length < 2)
        {
            ModelState.AddModelError("", "Naam is te kort");
            return View("Index", g);
        }
        LocalData.GastList.Add(g);
        return RedirectToAction("Index", "Home");
    }
    return View("Index", g);
}
```

Samenvatting Model validation

- **Model**

- [Required]
- [Required(ErrorMessage = "Vul je email in!")]
- [EmailAddress]

- **View**

`<div asp-validation-summary="All" class="text-danger"></div>`

- **Controller**

- `If(ModelState.IsValid())`

ASP.NET CORE

MVC Web Application

MVCModelValidation



Web Application (Model-View-Controller)

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

MVCModelValidation

- Add model
 - TestValidation
 - TestValidationId (primary key - int)
 - FirstName (string)
 - DateOfBirth (DateTime)

MVCModelValidation

- MVCModelValidation/Models

```
public class TestValidation
{
    public int? TestValidationId { get; set; }
    public string? FirstName { get; set; }
    public DateTime? DateOfBirth { get; set; }
}
```

MVCModelValidation

```
public class HomeController : Controller
{
    public IActionResult Index()
    {
        return View();
    }
    [HttpGet]
    public IActionResult TestValidation()
    {
        TestValidation tv=new TestValidation();
        return View(tv);
    }
}
```


Views/Home -> Add Empty Razor View -> TestValidation.cshtml
Views/Home/TestValidation.cshtml

```
@model TestValidation
<form asp-action="TestValidation">
    <div asp-validation-summary="All" class="text-danger"></div>
    <div class="form-group">
        <label asp-for="TestValidationId" class="control-label"></label>
        <input asp-for="TestValidationId" class="form-control" />
        <span asp-validation-for="TestValidationId" class="text-
danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="FirstName" class="control-label"></label>
        <input asp-for="FirstName" class="form-control" />
        <span asp-validation-for="FirstName" class="text-danger"></span>
    </div>
    <div class="form-group">
        <label asp-for="DateOfBirth" class="control-label"></label>
        <input asp-for="DateOfBirth" class="form-control" />
        <span asp-validation-for="DateOfBirth" class="text-danger"></span>
    </div>
    <div class="form-group">
        <input type="submit" value="Create" class="btn btn-primary" />
    </div>
</form>
```

MVCModelValidation

```
public class HomeController : Controller
{
    [HttpGet]
    public IActionResult TestValidation()
    {
        TestValidation tv=new TestValidation();
        return View(tv);
    }

    [HttpPost]
    public IActionResult TestValidation(TestValidation tv)
    {
        if(ModelState.IsValid)
        {
            return View("Index");
        }
        return RedirectToAction("TestValidation");
        return View(tv);
    }
}
```

MVCModelValidation

- Voeg een hyperlink toe in de Views/Shared/_layout naar de nieuwe actie “Home – TestValidation”
- Maak de model properties “Required” en test de model validatie

MVCModelValidation

- Add model
 - TestData
 - TestDataId (primary key)
 - Tekst (string – 50)
 - Datum (DateTime)
 - CustomDate
 - » Not > DateTime.Now
 - » Not before 1/1/1900

MVCModelValidation

```
public class TestData
{
    public int? TestDataId { get; set; }
    [StringLength(50)]
    public string? Tekst { get; set; }
    //[CustomDate]
    public DateTime? Datum { get; set; }
}
```

ASP.NET CORE

MVC Web Application – Model Validation
Oefening 06