

# C# Web - MVC

PRO - C# Web 1

## DE HOGESCHOOL MET HET NETWERK

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



# Doel

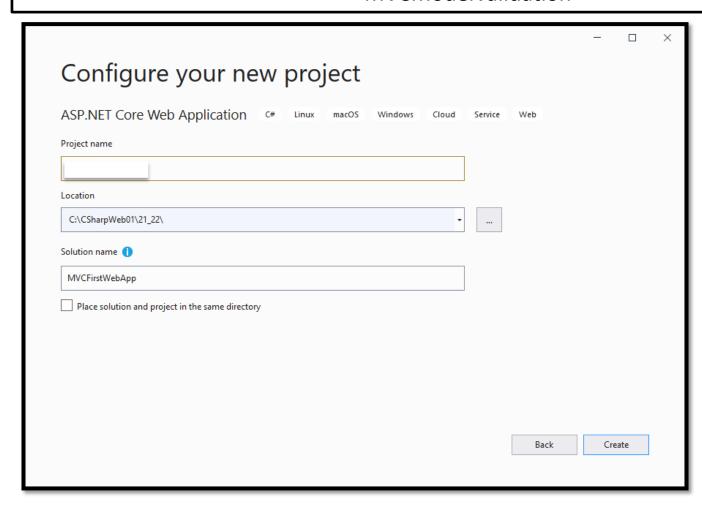
- ASP.Net Core toepassing
  - MVC Web Application
    - Models
    - Views
    - Controllers
  - Custom Model Validation



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

ASP.Net Core MVC

# **CUSTOM 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.



## Models & Data

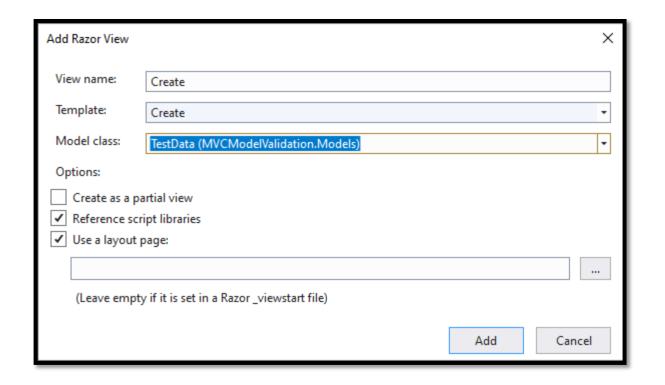
```
namespace MVCModelValidation.Models
{
    public class TestData
    {
        public int TestDataId { get; set; }
            [Required]
            [StringLength(10)]
            public string Tekst { get; set; }
            [CustomDate]
            public DateTime Datum { get; set; }
    }
}
```

```
namespace MVCModelValidation.Data
{
    public class LocalData
    {
       public static List<TestData> TestDataList = new List<TestData>();
    }
}
```

# Add Folder • CustomModelValidation • Add class • CustomDate.cs using Microsoft.AspNetCore.Mvc.ModelBinding.Validation;

```
public class CustomDate : Attribute, IModelValidator
    public IEnumerable<ModelValidationResult> Validate(ModelValidationContext context)
        var dtm = DateTime.Now;
        var lst = new List<ModelValidationResult>();
        if (DateTime.TryParse(context.Model.ToString(), out dtm))
            if (dtm > DateTime.Now)
              lst.Add(new ModelValidationResult("", "Date of Birth cannot be in the future"));
          else if (dtm < new DateTime(1900, 1, 1))</pre>
              lst.Add(new ModelValidationResult("", "Date of Birth should not be before 1900"));
        else
            lst.Add(new ModelValidationResult("", "Not a valid date!"));
        return 1st;
```

- Add Controller TestDataController
- Add Folder Views/TestData
  - Add Razor View (Model TestData)
    - Index Template(List)
    - Create Template(Create)





```
public class TestDataController : Controller
    public IActionResult Index()
        return View(LocalData.TestDataList);
    [HttpGet]
    public IActionResult Create()
        var testData = new TestData();
        return View(testData);
    [HttpPost]
    public IActionResult Create(TestData testData)
        if (ModelState.IsValid)
            LocalData.TestDataList.Add(testData);
            return RedirectToAction("Index");
        return View(testData);
```

## **ASP.NET CORE**

**MVC Web Application – Oefening06**