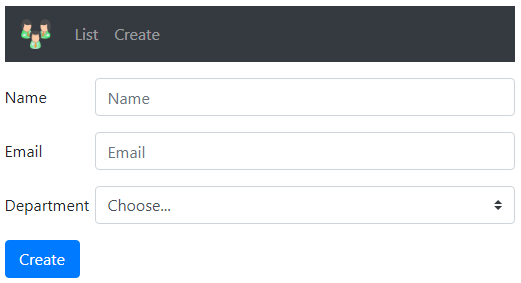
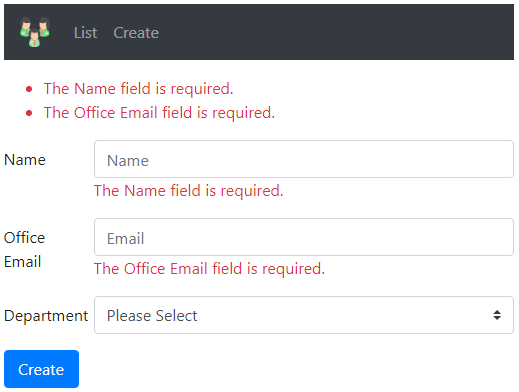
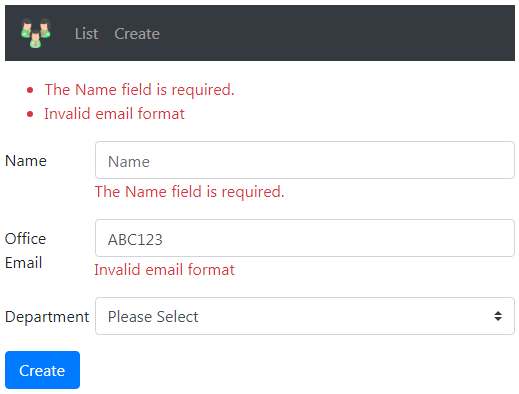
**Model Validation Example**  
  
Consider the following *Create Employee Form*  
  
  
  
  
We want to make both *Name*and *Office Email*fields required. If the required values are not provided and the form is submitted we want to display required validation errors as shown below.  
  
  
  
If invalid *email* is provided we want to display *Invalid Email Format*validation error as shown below.  
  
  
  
To make *Name*field a required field, apply *Required*attribute on the *Name* property of the Employee model class. *Required*attribute is in *System.ComponentModel.DataAnnotations* namespace.

public class Employee

{

    public int Id { get; set; }

    [Required]

    public string Name { get; set; }

    public string Email { get; set; }

    public Dept Department { get; set; }

}

**ModelState.IsValid Property**

* When the form is submitted, the following *Create()* action method is executed
* The model for *Create Employee Form* is Employee class
* When the form is submitted, model binding maps the posted form values to the respective properties of the Employee class
* With the *Required*attribute on the *Name* property of the Employee class, if a value for the *Name*property is not present, the validation fails
* Use *ModelState.IsValid* property to check if validation has failed or succeeded
* If validation has failed we return the same view so the user can provide the required data and resubmit the form.

[HttpPost]

public IActionResult Create(Employee employee)

{

    if (ModelState.IsValid)

    {

        Employee newEmployee = \_employeeRepository.Add(employee);

        return RedirectToAction("details", new { id = newEmployee.Id });

    }

    return View();

}

**Displaying Model Validation Errors**  
  
To display validation errors use *asp-validation-for*and *asp-validation-summary*tag helpers. *asp-validation-for*tag helper displays a validation message for a single property of our model class. *asp-validation-summary*tag helper displays a summary of validation errors.  
  
To display the validation error associated with the *Name*property of the *Employee*class use *asp-validation-for* tag helper on a <span> element as shown below.

<div class="form-group row">

    <**label** **asp-for**="Name" class="col-sm-2 col-form-label"></**label**>

    <div class="col-sm-10">

        <**input** **asp-for**="Name" class="form-control" placeholder="Name">

        <**span** **asp-validation-for**="Name"></**span**>

    </div>

</div>

To display a summary of all validation errors use *asp-validation-summary*tag helper on a <div> element as shown below.

<**div** **asp-validation-summary**="All">

</**div**>

The value for *asp-validation-summary*tag helper can be any of the following 

* All
* ModelOnly
* None

We will discuss the difference between these values and what they do in our upcoming videos. For now set it to a value of *All*.  
  
**Customising Model Validation Error Message**

public class Employee

{

    public int Id { get; set; }

    [Required]

    public string Name { get; set; }

    public string Email { get; set; }

    public Dept Department { get; set; }

}

By default the *Required*attribute on the *Name*property displays the following validation error message.  
The Name field is required.  
  
If you want to change the validation error message to "Please provide a value for the Name field" you can do so using the *ErrorMessage*property of the *Required*attribute as shown below.

public class Employee

{

    public int Id { get; set; }

    [Required(ErrorMessage = "Please provide a value for Name field")]

    public string Name { get; set; }

    public string Email { get; set; }

    public Dept Department { get; set; }

}

**ASP.NET Core Built-in Model Validation Attributes**  
  
The following are some of the common built-in validation attributes in ASP.NET Core

|  |  |
| --- | --- |
| **Attribute** | **Purpose** |
| Required | Specifies the field is required |
| Range | Specifies the minimum and maximum value allowed |
| MinLength | Specifies the minimum length of a string |
| MaxLength | Specifies the maximum length of a string |
| Compare | Compares 2 properties of a model. For example compare Email and ConfirmEmail properties |
| RegularExpression | Validates if the provided value matches the pattern specified by the regular expression |

**Display Attribute**  
  
This is not a validation attribute. It is commonly used for display purpose in the UI.   
  
For example, in the UI by default, the label for *Email*field displays the text *Email*, because the property name is *Email*. 

public class Employee

{

    public int Id { get; set; }

    public string Name { get; set; }

    public string Email { get; set; }

    public Dept Department { get; set; }

}

If you want the label to display *Office Email*instead, use the *Display*attribute

public class Employee

{

    public int Id { get; set; }

    public string Name { get; set; }

    [Display(Name = "Office Email")]

    public string Email { get; set; }

    public Dept Department { get; set; }

}

**Using Multiple Model Validation Attributes**  
  
Multiple validation attributes can be applied on a property by separating them with a comma as shown on the *Name*property or we stack them on top of each other as shown on the *Email*property.

public class Employee

{

    public int Id { get; set; }

    [Required, MaxLength(50, ErrorMessage = "Name cannot exceed 50 characters")]

    public string Name { get; set; }

    [Display(Name = "Office Email")]

    [RegularExpression(@"^[a-zA-Z0-9\_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$",

        ErrorMessage = "Invalid email format")]

    [Required]

    public string Email { get; set; }

    public Dept Department { get; set; }

}

**Model Validation Errors Colour**  
  
To change the colour of the model validation errors on the UI, use Bootstrap *text-danger*class on the <span> and <div> elements that have *asp-validation-for* and *asp-validation-summary* tag helpers

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

<**span** **asp-validation-for**="Name" class="text-danger"></**span**>