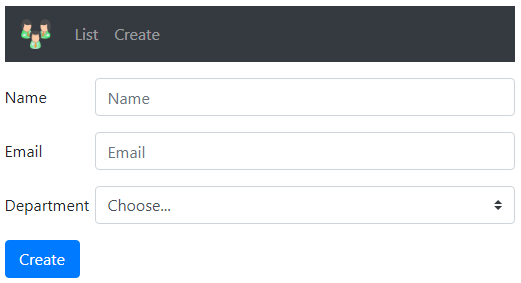
**Tag Helpers to create forms**  
  
We use the following common tag helpers to create a form in ASP.NET Core

* Form Tag Helper
* Label Tag Helper
* Input Tag Helper
* Select Tag Helper
* Textarea Tag Helper

we want to create a form using the form tag helpers and style it using Bootstrap 4. The ***"Create Employee Form"*** should be as shown below.  
  
  
  
**Form Tag Helper**  
  
To create a form we use the ***<form>*** tag helper  
  
Notice, we are using ***asp-controller*** and ***asp-action*** tag helpers. These 2 tag helpers specify the controller and the action method to which the form data must be posted when the form is submitted. We want to issue a POST request when the form is submitted, so we have set the ***method*** attribute to ***post***

<**form** **asp-controller**="home" **asp-action**="create" method="post">

</**form**>

The above code produces the following HTML when the form is rendered on the client browser. As you can see from the generated HTML when the form is submitted it will be posted to the ***index()*** action of the ***HomeController.***

<**form** method="post" action="/home/create"></**form**>

**Please note :** By default, when a form is submitted, it will be posted to the same action of the controller that rendered the form. So this means, even if we did not specify the controller and action using the ***asp-controller***and ***asp-action***tag helpers, the form will still be posted to the ***index()*** action of the ***HomeController.***  
  
**Input Tag Helper**  
  
The Input Tag Helper binds an HTML **<input>** element to a model expression in your razor view.  
  
In our case, we want a form to create a new employee. So the model for our ***Create.cshtml*** view is **Employee**class. We specify that using the ***model***directive. 

**@model Employee**

To b able to capture the ***employee name***we want a text box. We want the text box to bind to the ***Name***property of the ***Employee***model class. We do this by using ***asp-for***tag helper and setting it's value to the ***Name*** property of the ***Employee***model class. Notice we also have intellisense. Later if we change the property name form ***Name*** to ***FullName*** on the ***Employee***class, and if we do not change the value assigned to the tag helper, we get a compiler error.

<**input** **asp-for**="Name">

The above code generates an ***input***element with ***id***and ***name***attributes. Notice both of them are set to a value of ***Name***. 

<input type="text" id="Name" name="Name" value="">

The ***name*** attribute is required and it is used to map the value of the input element to the corresponding property of the model class when the form is submitted. This is done by a process called model binding in ASP.NET Core. We will discuss model binding in our next video.  
  
**Label Tag Helper**  
  
The Label Tag Helper generates a label with ***for***attribute. The ***for***attribute links the label with it's associated input element. Consider the following example.

<**label** **asp-for**="Name"></**label**>

<**input** **asp-for**="Name">

The above code generates the following HTML. 

<label for="Name">Name</label>

<input type="text" id="Name" name="Name" value="">

The label is linked to the input element, because both the label ***for***attribute and the input element ***id***attribute have the same value ***(Name)***. So this means when we click on the label, the corresponding input element receives the focus.  
  
Similarly, the following code generates a label and an input element to capture employee email.

<**label** **asp-for**="Email"></**label**>

<**input** **asp-for**="Email">

**Select Tag Helper**  
  
Generates ***select***element and it's associated option elements. In our case we want a select element to display the list of departments.  
  
Ultimately we want a label and a select element with list of department options as shown below.

<label for="Department">Department</label>

<select id="Department" name="Department">

    <**option** **value**="0">None</**option**>

    <**option** **value**="1">HR</**option**>

    <**option** **value**="2">Payroll</**option**>

    <**option** **value**="3">IT</**option**>

</select>

The options for the ***department***select element can be hard-coded like in the example above, or they can come from an ***enum***or a database table. We do not have a database hooked up yet. So for our example, let's get the options from an ***enum***.  
  
Place the following ***enum***in ***Dept.cs***file in the ***Models***folder

namespace EmployeeManagement.Models

{

    public enum Dept

    {

        None,

        HR,

        Payroll,

        IT

    }

}

Update the Employee class in *Employee.cs*file in the *Models*folder  
  
*Department*property data type is ***Dept***enum.

namespace EmployeeManagement.Models

{

    public class Employee

    {

        public int Id { get; set; }

        public string Name { get; set; }

        public string Email { get; set; }

        public Dept Department { get; set; }

    }

}

In ***Create.cshtml*** view, include the following code

<**label** **asp-for**="Department"></**label**>

<**select** **asp-for**="Department"

**asp-items**="Html.GetEnumSelectList<Dept>()"></**select**>

Notice, we are using ***asp-items***tag helper and ***Html.GetEnumSelectList<Dept>()*** to get the options for the ***select***element.  
  
The above code produces the following HTML

<label for="Department">Department</label>

<select id="Department" name="Department">

    <**option** **value**="0">None</**option**>

    <**option** **value**="1">HR</**option**>

    <**option** **value**="2">Payroll</**option**>

    <**option** **value**="3">IT</**option**>

</select>

**Create.cshtml - Complete Code without Bootstrap**

@model Employee

@{

    ViewBag.Title = "Create Employee";

}

<**form** **asp-controller**="home" **asp-action**="create" method="post">

    <div>

        <**label** **asp-for**="Name"></**label**>

        <**input** **asp-for**="Name" />

    </div>

    <div>

        <**label** **asp-for**="Email"></**label**>

        <**input** **asp-for**="Email">

    </div>

    <div>

        <**label** **asp-for**="Department"></**label**>

        <**select** **asp-for**="Department"

**asp-items**="Html.GetEnumSelectList<Dept>()"></**select**>

    </div>

    <button type="submit">Create</button>

</**form**>

The above code generates the following HTML

<form method="post" action="/home/create">

    <div>

        <label for="Name">Name</label>

        <input type="text" id="Name" name="Name" value="" />

    </div>

    <div>

        <label for="Email">Email</label>

        <input type="text" id="Email" name="Email" value="">

    </div>

    <div>

        <label for="Department">Department</label>

        <select id="Department" name="Department">

            <option value="0">None</option>

            <option value="1">HR</option>

            <option value="2">Payroll</option>

            <option value="3">IT</option>

        </select>

    </div>

    <button type="submit">Create</button>

</form>

**Create.cshtml - Complete Code with Bootstrap**

@model Employee

@{

    ViewBag.Title = "Create Employee";

}

<**form** **asp-controller**="home" **asp-action**="create" method="post" class="mt-3">

    <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">

        </div>

    </div>

    <div class="form-group row">

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

        <div class="col-sm-10">

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

        </div>

    </div>

    <div class="form-group row">

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

        <div class="col-sm-10">

            <**select** **asp-for**="Department" class="custom-select mr-sm-2"

**asp-items**="Html.GetEnumSelectList<Dept>()"></**select**>

        </div>

    </div>

    <div class="form-group row">

        <div class="col-sm-10">

            <button type="submit" class="btn btn-primary">Create</button>

        </div>

    </div>

</**form**>