go deploy

Screenshots

Module 6: Developing Models

Lab: Developing Models

Scenario

You are planning to create and code a Model-View-Controller (MVC) model that will be used in a butterflies shop application. The model includes properties that describe a butterfly. The model must enable the application to store the uploaded butterflies.

Exercise 1: Adding a Model

Scenario

In this exercise, you will:

- Add models to the web application and use them in views.
- Add a Create (GET) action.
- Add a Create (POST) action.

The main tasks for this exercise are as follows:

- Create a new model
- Use the model in a view
- Pass the model from the controller to a view
- Run the application
- Write a GET action
- Write a POST action that accepts the model

Task 1: Create a new model

1. Go to **D:\Allfiles\Mod06\Labfiles\01_ButterfliesShop_begin**, and then double-click **ButterfliesShop.sln**.

> **Note**: If a **Security Warning for ButterfliesShop** dialog box appears, verify that the **Ask me for every project in this solution** check box is cleared, and then click OK.

2. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, expand **Models**, and then click **Butterfly.cs**.

```
public int Id { get; set; }
public string CommonName { get; set; }
public Family? ButterflyFamily { get; set; }
public int? Quantity { get; set; }
public string Characteristics { get; set; }
public DateTime CreatedDate { get; set; }
public IFormFile PhotoAvatar { get; set; }
public string ImageName { get; set; }
public byte[] PhotoFile { get; set; }
public string ImageMimeType { get; set; }
```

- [ ] 4. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, right-click **Models**, point to **Add**, and then click **Class...**.

- [ ] 5. In the **Add New Item - ButterfliesShop** dialog box, in the **Name** box, type **IndexViewModel**, and then click **Add**.

- [ ] 6. In the **IndexViewModel.cs** code block, place the mouse cursor after the second **{** (opening brace) sign, press Enter, and then type the following code:

```
public List<Butterfly> Butterflies { get; set; }
```

Task 2: Use the model in a view

- [ ] 1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, expand **Views**, and then expand **Butterfly**.

- [ ] 2. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, under **Views**, under **Butterfly**, click **Index.cshtml**.

- [ ] 3. In the **Index.cshtml** code window, place the cursor at the beginning of the document, type the following code, and then press Enter.

```
@model ButterfliesShop.Models.IndexViewModel
```

- [ ] 4. In the **body** element of the **Index.cshtml** code window, type the following code:

```
<div class="container">
    <h1 class="main-title">Butterflies Shop</h1>
    <p class="into">Welcome to our Web Store, Enjoy a Wide Variety of Butterflies</
    <p class="into">Our Butterflies in the Shop</p>
    <button type="button" onclick="location.href='@Url.Action("Create", "Butterfly"
</div>
```

- [ ] 5. Place the cursor at the end of the **button** element you just created, press Enter, and then type the

```
<div class="img-container">
    @foreach (var item in Model.Butterflies)
    {
    }
</div>
```

- [ ] 6. Place the cursor in the **foreach** code block, press Enter, and then type the following code:

```
<div class="photo-index-card">
        <p>@item.CommonName</p>
        <p>@item.ButterflyFamily</p>
        <p>@item.Quantity</p>
        <p>@item.CreatedDate</p>
</div>
```

Task 3: Pass the model from the controller to a view

- [ ] 1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, expand **Controllers**, and then click **ButterflyController.cs**.

- [ ] 2. Place the cursor at the end of the **InitializeButterfliesData** method code block, press Enter two times, and then type the following code:

```
public IActionResult Index()
{
    IndexViewModel indexViewModel = new IndexViewModel();
    indexViewModel.Butterflies = _data.ButterfliesList;
    return View(indexViewModel);
}
```

Task 4: Run the application

- [ ] 1. In the **ButterfliesShop - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

- [ ] 2. In the **ButterfliesShop - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.

> **Note**: The browser displays the **Index** action result inside the **Butterfly** Controller.

- [ ] 3. In Microsoft Edge, click **Close**.

Task 5: Write a GET action

- [ ] 1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, under **Controllers**, click **ButterflyController.cs**.

- [ ] 2. Place the cursor at the end of the **GetImage** action code block, press Enter two times, and then

```
[HttpGet]
public IActionResult Create()
{
return View();
}
```

Task 6: Write a POST action that accepts the model

- [ ] 1. In the **ButterflyController.cs** code window, ensure that the cursor is at the end of the **Create** action code block with **HTTP GET** verb, press Enter, and then type the following code (ignore the error on the Create method - this will be rectified in the next step):

```
[HttpPost]
public IActionResult Create(Butterfly butterfly)
{
}
```

- [ ] 2. In the **Create** action code block created with **HTTP POST** verb, type the following code, and then press Enter.

```
Butterfly lastButterfly = _data.ButterfliesList.LastOrDefault();
butterfly.CreatedDate = DateTime.Today;
if (butterfly.PhotoAvatar != null && butterfly.PhotoAvatar.Length > 0)
{
  butterfly.ImageMimeType = butterfly.PhotoAvatar.ContentType;
  butterfly.ImageName = Path.GetFileName(butterfly.PhotoAvatar.FileName);
  butterfly.Id = lastButterfly.Id + 1;
  _butterfliesQuantityService.AddButterfliesQuantityData(butterfly);
  using (var memoryStream = new MemoryStream())
  {
      butterfly.PhotoAvatar.CopyTo(memoryStream);
      butterfly.PhotoFile = memoryStream.ToArray();
  }
  _data.AddButterfly(butterfly);
  return RedirectToAction("Index");
}
```

- [ ] 3. In the **Create** action code block created with **HTTP POST** verb, immediately after the code you just added, press Enter, and then type the following code:

```
return View(butterfly);
```

✓ **Results**: After completing this exercise, you have created a model, passed the model from an action to a view, and used the model in the view.

go deploy

Scenario

In this exercise, you will:

- Add Display and Edit data annotations to a Butterfly model.
- Add Display Helpers to Index.cshtml.
- Add Form Helpers and Editor Helpers to Create.cshtml.

The main tasks for this exercise are as follows:

- Add display and edit data annotations to a model
- Update an action to return FileContentResult
- Add Display Helpers
- Add Form Helpers
- Add Editor Helpers
- Run the application

Task 1: Add display and edit data annotations to a model

1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, under **Models**, click **Butterfly.cs**.

2. In the **Butterfly.cs** code window, place the cursor at the beginning of the line of code that declares the property CommonName, press Enter, and then insert the following code:

```
[Display(Name = "Common Name:")]
```

3. In the same way, add the following attribute to the **ButterflyFamily** property:

```
[Display(Name = "Butterfly Family:")]
```

4. In the same way, add the following attribute to the **Quantity** property:

```
[Display(Name = "Butterflies Quantity:")]
```

5. Add the following attribute to the **Characteristics** property:

```
[Display(Name = "Characteristics:")]
```

6. Add the following attributes to the **CreatedDate** property:

```
[DisplayFormat(DataFormatString = "{0:dd/MM/yy}")]
```

- [ ] 7. Add the following attribute to the **PhotoAvatar** property:

```
[Display(Name = "Butterflies Picture:")]
```

Task 2: Update an action to return FileContentResult

- [ ] 1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, under **Controllers**, click **ButterflyController.cs**.

- [ ] 2. In the **GetImage** action code block, locate the following code:

```
Butterfly requestedButterfly = _data.GetButterflyById(id);
if (requestedButterfly != null)
{
    return null;
}
else
{
    return NotFound();
}
```

- [ ] 3. In the **IF** statement code block, select the following code:

```
return null;
```

- [ ] 4. Replace the selected code with the following code:

```
string fullPath = webRootpath + folderPath + requestedButterfly.ImageName;
if (System.IO.File.Exists(fullPath))
{
        FileStream fileOnDisk = new FileStream(fullPath, FileMode.Open);
        byte[] fileBytes;
        using (BinaryReader br = new BinaryReader(fileOnDisk))
        {
            fileBytes = br.ReadBytes((int)fileOnDisk.Length);
        }
        return File(fileBytes, requestedButterfly.ImageMimeType);
}
else
{
        if (requestedButterfly.PhotoFile.Length > 0)
        {
            return File(requestedButterfly.PhotoFile, requestedButterfly.ImageMimeT
        }
        else
        {
            return NotFound();
        }
}
```

Task 3: Add Display Helpers

- [ ] 1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, under **Views**, under **Butterfly**, click **Index.cshtml**.

- [ ] 2. In the **Index.cshtml** code window, select the following code:

```
<p>@item.CommonName</p>
<p>@item.ButterflyFamily</p>
<p>@item.Quantity</p>
<p>@item.CreatedDate</p>
```

- [ ] 3. Replace the selected code with the following code (you may want to adjust the pasted code to align with the rest of the code using the tab key):

```
<h3 class="display-picture-title">
    "@Html.DisplayFor(modelItem => item.CommonName)"
</h3>
@if (item.ImageName != null)
{
    <div class="photo-display">
        <img class="photo-display-img" src="@Url.Action("GetImage", "Butterfly", ne
    </div>
}
```

go deploy

```
<div>
        <p class="display-label">
            @Html.DisplayNameFor(model => item.ButterflyFamily)
        </p>
        <br />
        <p class="display-field">
            @Html.DisplayFor(model => item.ButterflyFamily)
        </p>
</div>
<div class="display-info">
        <p class="display-label">
            @Html.DisplayNameFor(model => item.Characteristics)
        </p>
        <p class="display-field">
            @Html.DisplayFor(model => item.Characteristics)
        </p>
</div>
<div>
        <p class="display-label">
            @Html.DisplayNameFor(model => item.Quantity)
        </p>
        <p class="display-field">
            @Html.DisplayFor(model => item.Quantity)
        </p>
</div>
<div>
        <p class="display-label">
            @Html.DisplayNameFor(model => item.CreatedDate)
        </p>
        <p class="display-field">
            @Html.DisplayFor(model => item.CreatedDate)
        </p>
</div>
```

Task 4: Add Form Helpers

- [ ] 1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, under **Views**, under **Butterfly**, click **Create.cshtml**.

- [ ] 2. In the **Create.cshtml** code window, place the cursor at the beginning of the document, type the following code, and then press Enter.

```
@using ButterfliesShop.Models
@model ButterfliesShop.Models.Butterfly
```

- [ ] 3. In the **BODY** element of the **Create.cshtml** code window, type the following code:

go deploy

```
        <form method="post" enctype="multipart/form-data" asp-action="Create">
        </form>
    </div>
```

Task 5: Add Editor Helpers

1. In the **FORM** element of the **Create.cshtml** code window, type the following code:

```
<div class="form-field">
        <label asp-for="CommonName"></label>
        <input asp-for="CommonName" />
    </div>
    <div class="form-field">
        <label asp-for="ButterflyFamily"></label>
        <select asp-for="ButterflyFamily" asp-items="Html.GetEnumSelectList<Family>
            <option selected="selected" value="">Select</option>
        </select>
    </div>
    <div class="form-field">
        <label asp-for="Characteristics"></label>
        <textarea asp-for="Characteristics"></textarea>
    </div>
    <div class="form-field">
        <label asp-for="Quantity"></label>
        <input asp-for="Quantity" />
    </div>
    <div class="form-field">
        <label asp-for="PhotoAvatar"></label>
        <input asp-for="PhotoAvatar" type="file" />
    </div>
    <div class="form-field">
        <input type="submit" value="Submit" />
    </div>
```

Task 6: Run the application

1. In the **ButterfliesShop - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
2. In the **ButterfliesShop - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.

> **Note**: The browser displays the **Index** action result inside the **Butterfly** Controller.

3. In Microsoft Edge, click **Add Butterflies**.

> **Note**: The browser displays the **Create** action with **HTTP GET** verb result inside the

go deploy

- [ ] 4. On the **Add Butterflies to The Shop** page, in the **Common Name** box, type Red Admiral.
- [ ] 5. On the **Add Butterflies to The Shop** page, in the **Butterfly Family** box, select **Nymphalidae**.
- [ ] 6. On the **Add Butterflies to The Shop** page, in the **Characteristics** box, type the following:

```
Common throughout Europe, Asia, North Africa, and North America in open ground, mar
```

- [ ] 7. On the **Add Butterflies to The Shop** page, in the **Butterflies Quantity** box, type **25**.
- [ ] 8. On the **Add Butterflies to The Shop** page, in the **Butterflies Picture** box, click on **Choose File**, and select the image at **D:\AllFiles\Mod06\Labfiles\Images\red-admiral.jpg**, and then click **Submit**.

> **Note**: Verify that the submitted butterfly details are valid and that a new butterfly was added to the homepage.

- [ ] 9. In Microsoft Edge, click **Close**.

✓ **Results**: After completing this exercise, you will be able to add data annotations to the **Butterfly** model class in the application and add tag helpers to views.

Exercise 3: Adding Validation

In this exercise, you will:

- Add validation data annotations to a butterfly model.
- Add validation Helpers to the Create view.
- Add ModelState.IsValid property in ButterflyController.
- Add MaxButterflyQuantityValidation custom validation.

The main tasks for this exercise are as follows:

- Add validation data annotations to a model
- Add validation helpers to a view
- Using ModelState.IsValid property in a controller
- Run the application
- Add custom validation
- Run the application

Task 1: Add validation data annotations to a model

1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, under **Models**, click **Butterfly.cs**.

2. In the **Butterfly.cs** code window, in the **CommonName** property code, place the cursor at the end of the **Display** data attribute, press Enter, and then type the following code:

```
[Required(ErrorMessage = "Please enter the butterfly name")]
```

3. In the same way, add another data attribute to the **ButterflyFamily** property:

```
[Required(ErrorMessage = "Please select the butterfly family")]
```

4. Add another data attribute to the **Quantity** property:

```
[Required(ErrorMessage = "Please select the butterfly quantity")]
```

5. Add further data attributes to the **Characteristics** property:

```
[Required(ErrorMessage = "Please type the characteristics")]
[StringLength(50)]
```

6. Add another data attribute to the **PhotoAvatar** property:

```
[Required(ErrorMessage = "Please select the butterflies picture")]
```

under **Butterfly**, click **Create.cshtml**.

2. In the **Create.cshtml** code window, locate the following code:

```
<form method="post" enctype="multipart/form-data" asp-action="Create">
```

3. Place the cursor at the end of the located code, press Enter, and then type the following code:

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

4. In the **Create.cshtml** code window, locate the following code:

```
<input asp-for="CommonName" />
```

5. In the **Create.cshtml** code window, place the cursor at the end of the located code, press Enter, and then type the following code:

```
<span asp-validation-for="CommonName"></span>
```

6. In the **Create.cshtml** code window, locate the following code:

```
<select asp-for="ButterflyFamily" asp-items="Html.GetEnumSelectList<Family>()">
    <option selected="selected" value="">Select</option>
</select>
```

7. In the **Create.cshtml** code window, place the cursor at the end of the located code, press Enter, and then type the following code:

```
<span asp-validation-for="ButterflyFamily"></span>
```

8. In the **Create.cshtml** code window, locate the following code:

```
<textarea asp-for="Characteristics"></textarea>
```

9. In the **Create.cshtml** code window, place the cursor at the end of the located code, press Enter, and then type the following code:

```
<span asp-validation-for="Characteristics"></span>
```

10. In the **Create.cshtml** code window, locate the following code:

```
<input asp-for="Quantity" />
```

```
<span asp-validation-for="Quantity"></span>
```

12. In the **Create.cshtml** code window, locate the following code:

```
<input asp-for="PhotoAvatar" type="file" />
```

13. In the **Create.cshtml** code window, place the cursor at the end of the located code, press Enter, and then type the following code:

```
<span asp-validation-for="PhotoAvatar"></span>
```

Task 3: Using the ModelState.IsValid property in a controller

1. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, under **Controllers**, click **ButterflyController.cs**.
2. In the **Create** action code block created with **HTTP POST** verb, locate the following code:

```
[HttpPost]
public IActionResult Create(Butterfly butterfly)
{
```

3. Place the cursor after the **{** (opening brace) sign, press Enter, and then type the following code:

```
if (ModelState.IsValid)
{
```

4. In the **Create** action code block created with **HTTP POST** verb, locate the following code:

```
return View(butterfly);
```

5. In the **Create** action code block created with **HTTP POST** verb, immediately after the located code, press Enter, and then type the following code:

```
}
return View(butterfly);
```

Task 4: Run the application

1. In the **ButterfliesShop - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.
2. In the **ButterfliesShop - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.

3. In Microsoft Edge, click **Add Butterflies**.

> **Note**: The browser displays the **Create** action with **HTTP GET** verb result inside the **Butterfly** Controller.

4. On the **Add Butterflies to The Shop** page, leave all the boxes blank, and then click **Submit**.

> **Note**: The browser window displays the **Create.cshtml** view with validation messages below all the boxes.

5. On the **Add Butterflies to The Shop** page, in the **Common Name** box, type Red Admiral.
6. On the **Add Butterflies to The Shop** page, in the **Butterfly Family** box, select **Nymphalidae**.
7. On the **Add Butterflies to The Shop** page, in the **Characteristics** box, type the following:

```
Common throughout Europe, Asia, North Africa, and North America in open ground, mar
```

8. On the **Add Butterflies to The Shop** page, in the **Butterflies Quantity** box, type **25**.
9. On the **Add Butterflies to The Shop** page, in the **Butterflies Picture** box, click on **Choose File**, and select the image at **D:\AllFiles\Mod06\Labfiles\Images\red-admiral.jpg**, and then click **Submit**.

> **Note**: Verify that the submitted butterfly details are valid and that a new butterfly was added to the homepage.

10. In Microsoft Edge, click **Close**.

Task 5: Add custom validation

1. In **Solution Explorer**, right-click **ButterfliesShop**, point to **Add**, and then click **New Folder**.
2. In the **NewFolder** box, type Validators, and then press Enter.
3. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, right-click **Validators**, point to **Add**, and then click **Class...**.
4. In the **Add New Item - ButterfliesShop** dialog box, in the **Name** box, type MaxButterflyQuantityValidation, and then click **Add**.

```
public class MaxButterflyQuantityValidation
```

6. Add a **ValidationAttribute** base class so that it looks like the following:

```
public class MaxButterflyQuantityValidation : ValidationAttribute
```

7. In the **MaxButterflyQuantityValidation.cs** constructor code block, insert the following code:

```
private int _maxAmount;
public MaxButterflyQuantityValidation(int maxAmount)
{
    _maxAmount = maxAmount;
}
```

8. Place the cursor at the end of the constructor code block, press Enter, and add the following method:

```
protected override ValidationResult IsValid(object value, ValidationContext validat
{
    var service = (IButterfliesQuantityService)validationContext.GetService(typeof(
    Butterfly butterfly = (Butterfly)validationContext.ObjectInstance;
    if (butterfly.ButterflyFamily != null)
    {
        int? quantity = service.GetButterflyFamilyQuantity(butterfly.ButterflyFamil
        int? sumQuantity = quantity + butterfly.Quantity;
        if (sumQuantity > _maxAmount)
        {
            return new ValidationResult(string.Format("Limit of butterflies from th
        }
        return ValidationResult.Success;
    }
    return ValidationResult.Success;
}
```

9. In the **ButterfliesShop - Microsoft Visual Studio** window, in **Solution Explorer**, expand **Models**, and then click **Butterfly.cs**.

10. In the **Butterfly.cs** code window, locate the Quantity property:

```
public int? Quantity { get; set; }
```

11. Add a new annotation using the custom validator attribute:

```
[MaxButterflyQuantityValidation(50)]
```

2. In the **ButterfliesShop - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Without Debugging**.

> **Note**: The browser displays the **Index** action result inside the **Butterfly** Controller.

3. In Microsoft Edge, click **Add Butterflies**.

> **Note**: The browser displays the **Create** action with **HTTP GET** verb result inside the **Butterfly** Controller.

4. On the **Add Butterflies to The Shop** page, in the **Common Name** box, type Red Admiral.

5. On the **Add Butterflies to The Shop** page, in the **Butterfly Family** box, select **Nymphalidae**.

6. On the **Add Butterflies to The Shop** page, in the **Characteristics** box, type the following:

```
Common throughout Europe, Asia, North Africa, and North America in open ground, mar
```

7. On the **Add Butterflies to The Shop** page, in the **Butterflies Quantity** box, type **60**.

8. On the **Add Butterflies to The Shop** page, in the **Butterflies Picture** box, click on **Choose File**, and select the image at **D:\AllFiles\Mod06\Labfiles\Images\red-admiral.jpg**, and then click **Submit**.

> **Note**: The browser window displays the **Create.cshtml** view with custom validation message below the **Butterflies Quantity** box.

9. On the **Add Butterflies to The Shop** page, in the **Butterflies Quantity** box, type **2**.

10. On the **Add Butterflies to The Shop** page, in the **Butterflies Picture** box, click on **Choose File**, and select the image at **D:\AllFiles\Mod06\Labfiles\Images\red-admiral.jpg**, and then click **Submit**.

> **Note**: Verify that the submitted butterfly details are valid and that a new butterfly was added to the homepage.

11. In Microsoft Edge, click **Close**.

12. In the **ButterfliesShop - Microsoft Visual Studio** window, on the **File** menu, click **Exit**.

go deploy

users can submit a variety of butterflies to the shop.