Lab: Structuring ASP.NET MVC Core Web Applications

# Scenario

An important design priority for the Photo Sharing application is that the visitors should be able to easily and logically locate photographs. Additionally, photo galleries and photos need to appear high in search engine results. To implement these priorities, you have been asked to configure routes that enable the entry of user-friendly URLs to access photos.

You have been asked to ensure that the URLs of the following forms work to display a photo:

* ~/photos/details/id. In this form of URL, id is the database ID of the photo object. This form of URL already works because it matches the default route.
* ~/photos/id. In this form of URL, id is the database ID of the photo object. This is the logical URL to enter when you know the ID of the photo that you want to access.
* ~/photos/title/PhotoTitle. In this form of URL, PhotoTitle is the title of the photo object. This is the logical URL to enter when you know the title of the photo that you want to access.

# Objectives

After completing this lab, you will be able to:

* Add routes to the ASP.NET Routing Engine in an ASP.NET MVC application.

**Estimated Time**: 20 minutes

# Exercise 1: Using Attributes Routing

## Scenario

In this exercise, you will:

* Add routes to the application using Attributes Routing.
* Try out routes by typing URLs in the browser Address bar.

The main tasks for this exercise are as follows:

1. Add the Photo ID route.
2. Add the Photo Title route.
3. Try out the new routes.

### Task 1: Add the Photo ID route.

https://docs.microsoft.com/en-us/aspnet/core/mvc/controllers/routing?view=aspnetcore-2.1

1. Open the PhotosController.cs file in the PhotoSharingApplication project.
2. Add a new route to the Photo Sharing application by using the following information. Add the new route as an attribute, before the Details action:

* URL: photos/{id}
* Constraints: id: int

### Task 2: Add the Photo Title route.

1. Open the IPhotosRepository interface and add a FindByTitle method that accepts a string title and returns a Task<Photo>
2. Open the PhotosRepository class and implement the FindByTitle method by returning the first photo whose title is equal to the title parameter
3. Open the PhotosController.cs file in the PhotoSharingApplication project.
4. Add a DisplayByTitle Action that receives a title parameter and passes to the Details view the first photo with a matching title
5. Add a new route to the Photo Sharing application by using the following information. Add the new route as an attribute, before the DisplayByTitle action:

* URL: photos/title/{title}

### Task 3: Try out the new routes.

1. Start the PhotoSharingApplication project with debugging.
2. Click on the Display link of any image on the home page and note the route that has been used to formulate the link.
3. Access the following relative URL:

* /photos/title/Me standing on top of a mountain

1. Stop debugging.