

## PhotoShare

- Sign in/Sign out of your account
- Create your user profile
- Create a new album, and upload your photos to it
- Tag those photos
- Comment on photos
- Search for photos by tag, album, user

## Passing a model to a view

In your action, create a model object, and do return View(model). Then in your view, put this at the top:

```
@model <Application>.Models.<ModelName>
```

And then access the properties of this model in your view via @Model

## Unit Testing

Specify to create a unit testing project at the 'Type of Application' tab when you're creating a new asp.net mvc application

See the tests in <Application>.Tests projects, and Run tests by Ctrl+R, A

[TestClass], [TestMethod] attributes

In each test method, you need to make an instance of the controller, an instance of the action that you're testing as a ViewResult, and then do some assertions with that ViewResult

## Routing

Route table is in App\_Start/RouteConfig.cs

RouteData data structure can be used to access parts of the url. They're stored by key value pairs.

For ex: RouteData['controller'] gets you the name of the controller in the URL

To define your own routes, you wanna place them above the Default route, because the routes are handled in the order in which they are placed

## Actions

If you just wanna send a string back to the browser in an action, just do return Content("<your string>")

If you've defined some parameter in your RouteConfig for a url, then you can just access that parameter by passing an argument to your action. So for example, in the photos/search/{tag} , you can pass string tag as an argument to your action method, and its value will be available

If you define an argument in your action, it'll be available as a querystring key. So you could do photos/search?tag=vacation and the value of tag will still be available in your action

## Action Results

- If you wanna redirect your action to another action in a controller, you can use
  - `return RedirectToAction(action,controller,new {parameter1= value,...})`
- If you wanna redirect to a route from the RouteConfig, you can use `RedirectToRoute`
  - `return RedirectToAction(route name, new {controller="controller", action="Action"})`