

Module Review and Takeaways

In this module, you have seen the pivotal role that controllers and actions take in the construction of an MVC web application. Controllers and actions ensure that the application responds correctly to each user request, create instances of model classes, and pass the model class to a view for display. In the next module, you will see how to create and code views to implement the user interface for your application.



Best Practice: Unless you have a good reason not to, keep to the convention that each controller should be named to match the corresponding model class, with “Controller” appended. For example, for the **Photo** model class, the controller should be called **PhotoController**. By maintaining this convention, you create a logically named set of model and controller classes, and can use the default controller factory.



Best Practice: Take great care if you choose to override the built-in **AuthorizeAttribute** filter because it implements permissions and security for the web application. If you carelessly modify the security infrastructure of the MVC framework, you may introduce vulnerabilities in your application. Instead, use the built-in filter wherever possible.

Review Question(s)

Question: You want to ensure that the **CreatedDate** property of a new **Photo** object is set to today’s date when the object is created. Should this value be set in the **Photo** model class constructor method or in the **PhotoController Create** action method?