ASP.NET MVC Core 2 Homework 04

Chapter 04, Pro ASP.NET Core MVC 2

Homework

Readings

Read chapter 04, in the *Pro ASP.NET Core MVC 2* book.

Discussion Questions

Be prepared to discuss each of the following questions is class.

Written Deliverable

Your written homework is due the Monday following our class discussion *unless otherwise directed.* Format your homework in MarkDown. Copy your written homework to your clipboard and paste it into the text input box in Canvas for this assignment. Add and commit it to your local git repository, and push the homework up to your Github repository.

Homework Questions

1. The `AddMvc()` method belongs to the `MvcServiceCollectionExtensions` namespace. What does this method do? What parameters does this method take? What is its return value?
2. The `UseMvcWithDefaultRoute()` method belongs to the `MvcApplicationBuilderExtensions` namespace. What does this method do? What parameters does this method take? What is its return value?
3. Is the *null conditioal operator* used on value types or reference types? Explain why or why not.
4. The book states that a read only property cannot be changed. Is this a true statement? If not, how can you change a read only property? Please be careful when you answer this question, the answer might not be obvious.
5. What is *object initializer syntax*? What is the most unusual feature of object initializer syntax?
6. Describe the *is* keyword. What is it used for? Why do we use it? Give an example of the use of the `is` operator.
7. What is an *extension method*? Why do we need extension methods?
8. How do you apply extension methods to an interface?
9. What is a *lambda expression*? (not in book) See `https://en.wikipedia.org/wiki/Lambda_expression` and `https://en.wikipedia.org/wiki/Anonymous_function`. Given this definition, why do you think that lambda expressions are useful?
10. This is not in the book, and is an optional question. The `>=` operator is overloaded in C# to make *expression bodied methods* and *lambda expressions.* There is a very significant difference between expression bodied methods and lambda expressions. What is this difference?

11. Describe *implicit typing*. How do you use an implicitly typed variable?

12. What is an *asynchronous method*? Why would you want to use asynchronous methods?

13. How do you tell .NET that work will be done asynchronously? In other words, what is the return value of an asynchronous method?

14. Describe the `async` and `await` keywords. What are they used for? How do you use them?