ASP.NET MVC Core 2 Homework 06

Chapter 06, Pro ASP.NET Core MVC 2

Homework

Readings

Read chapter 06, 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. Consider the following code. For each of the six items highlighted in red (sharedRepository, products, SharedRepository, SimpleRepository, Products, and AddProduct), please identify each as a field, property, costructor, or method.

```
public class SimpleRepository
{
    private static SimpleRepository sharedRepository = new SimpleRepository();
    private Dictionary<string, Product> products = new Dictionary<string, Product>();
    public static SimpleRepository SharedRepository => sharedRepository;
    public SimpleRepository()
    {
        var initialItems = new[]
        {
            new Product { Name = "Kayak", Price = 275M },
            new Product { Name = "Lifejacket", Price = 48.95M },
            new Product { Name = "Soccer␣ball", Price = 19.50M },
            new Product { Name = "Corner␣flag", Price = 34.95M }
        };
        foreach (var p in initialItems)
            AddProduct(p);
    }
    public IEnumerable<Product> Products => products.Values;
    public void AddProduct(Product p) => products.Add(p.Name, p);
}
```

2. What is *NuGet*? What is NuGet used for? Be specific.

3. What is a *package*? What is a server-side package? How does the NuGet tool keep track of the packages that have been added to a project?

4. What is a client-side package? The current version of Visual Studio deprecates Bower. What has replaced Bower? See the errata page for the book's Github repository.

5. The book mentions two tools for managing client-side libraries. What are they? The book's Github errata page describes one. The other one is not described anywhere, as far as I know. Look for it and see if you can find it.

6. When do changes to Razor views take effect during development?

7. What are the two ways changes to C# classes are handled during development? For each of them, when to changes to C# classes take effect?

8. What does the method *UseDeveloperExceptionPages()* do? What are the parameters to the method? What is the return value of the method? To what class library does the method belong?

9. How do you add a breakpoint to a statement? What effect does adding a breakpoint have when running the code?

10. Why would you want to use Browser Link?

11. What is the purpose of the \\`wwwroot`\\ directory?

12. How do you enable static content in a ASP.NET Core MVC web application? What is "static content?"

13. What is *bundling*? What is *minification*?