Chapter 6: Essential Tools for MVC Notes

1. Chapter Start
   1. Main topics/tools covered in the chapter:
      1. Dependency injection (DI) container
      2. Unit test framework
      3. Mocking tool
   2. Chapter Outline
      1. **Decouple Classes**: Introduce interfaces and declare dependencies on them in the class constructor
      2. **Automatically resolve dependencies using interfaces**: Use Ninject or another dependency injection container
      3. **Integrate Ninject into an MVC application**: Create an implementation of the IDepdencyResolver interface that calls the Ninject kernel and register it as a resolver by calling the System.Web.Mvc.DepdencyResolver.SetResolver method
      4. **Inject property and constructor values into newly created objects**: Use the WithPropertyValue and WithConstructorArgument methods
      5. **Dynamically select an implementation class for an interface**: Use an Ninject conditional binding
      6. **Control the lifecycle of the objects that Ninject creates**: Set an object scope
      7. **Create a unit test**: Add a unit test project to the solution and annotate a class file with TestClass and TestMethod attributes
      8. **Check for expected outcomes in a unit test**: Use the assert class
      9. **Focus a unit test on a single feature of component**: Isolate the test target using mock objects
2. Stages to using Ninject as a dependency injection container
   1. Prepare Ninject for use
      1. Create an instance of a Ninject kernel
         1. *Object responsible for resolving dependencies and creating objects…*
         2. *When object is needed… the kernel is used instead of the new keyword*
      2. Configure Ninject Kernel
         1. *Important so that the kernel understands which implementation of objects you want to use for each interface*
      3. Create an Object
         1. *Through the kernel Get method*
3. Dependency Resolver