

Working with Services and Dependency Injection



KEVIN DOCKX

ARCHITECT

@KevinDockx <https://www.kevindockx.com>



Coming Up



Inversion of Control and Dependency Injection

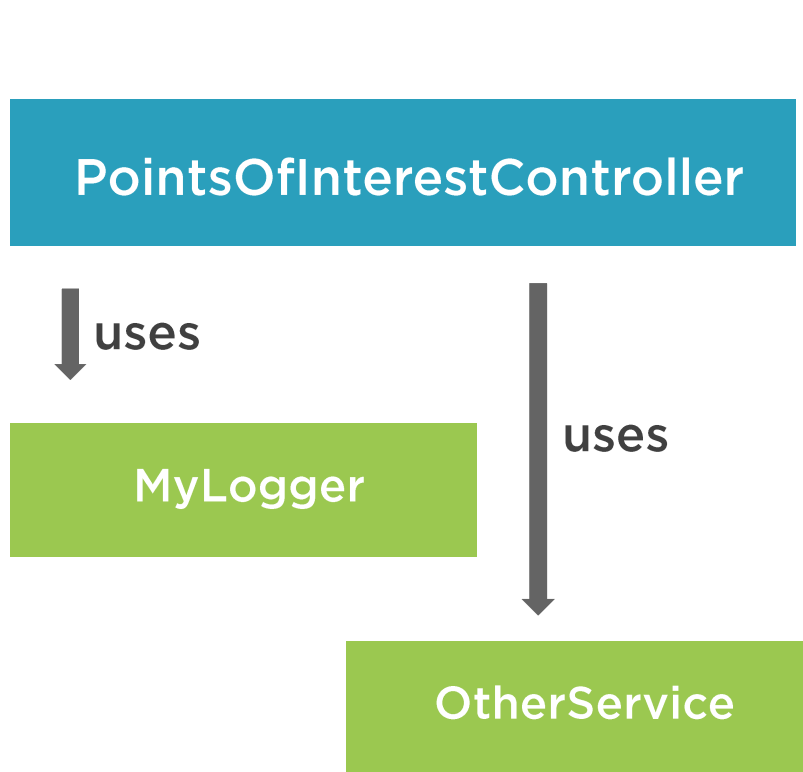
Logging

Creating and Using Custom Services

Working with Configuration Files



Inversion of Control and Dependency Injection



Class implementation has to change when a dependency changes

Difficult to test

Class manages the lifetime of the dependency

This is tight coupling



Inversion of Control

Inversion of Control delegates the function of selecting a concrete implementation type for a class's dependencies to an external component.



Dependency Injection

Dependency Injection is a specialization of the Inversion of Control pattern. The Dependency Injection pattern uses an object - the container - to initialize objects and provide the required dependencies to the object.



```
public class PointsOfInterestController :  
Controller  
{  
  
    private  
    ILogger<PointsOfInterestController>  
    _logger;  
  
    public PointsOfInterestController(  
        ILogger<PointsOfInterestController>  
        logger)  
    {  
        _logger = logger;  
    }  
  
    ...  
}
```

◀ **Interface, not concrete
implementation**

◀ **Constructor injection**



Inversion of Control and Dependency Injection

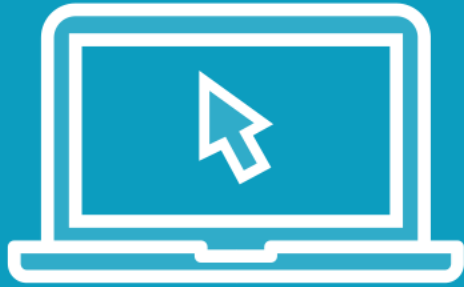


Dependency Injection is built into ASP.NET Core

ConfigureServices is used to register services with the built-in container



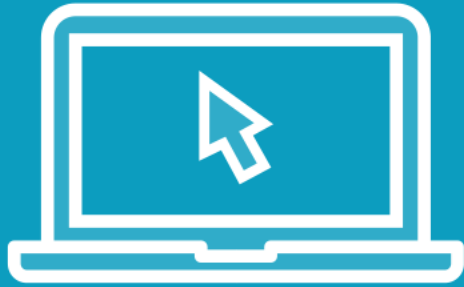
Demo



Injecting and Using a Logger



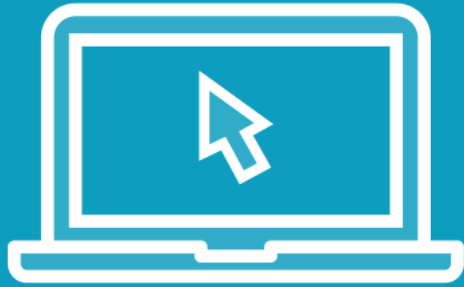
Demo



Logging to a File



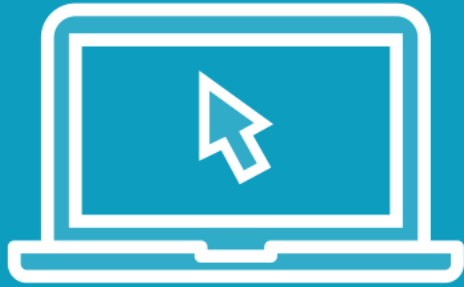
Demo



Implementing and Using a Custom Service



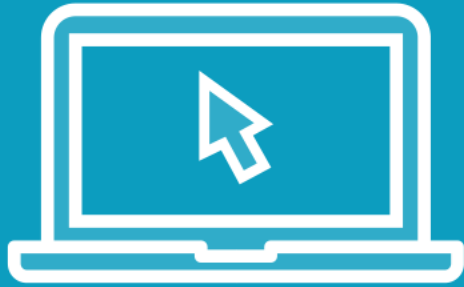
Demo



Working with Configuration Files



Demo



Scoping Configuration to Environments



Summary



Dependency injection

- Specialization of IoC
- Loose coupling, less code changes, better testability



Summary



Custom services are registered in `ConfigureServices`

- Transient
- Scoped
- Singleton

Use configuration files for configuration data, scoped to a specific environment