

# Refactoring Legacy Code to Follow DIP and ISP



**Steve Smith**

Founder and Principal Architect, NimblePros

@ardalis | ardalisc.com

# Overview



## Inverting Dependencies

- Introduce ISP interfaces
- Invert and inject dependencies





## Refactoring

- Inverting Dependencies (DIP)
- Introducing Interfaces (ISP)



# Review After Refactoring to Apply DIP and ISP

**Leveraging  
Dependency  
Injection**

**More Flexible Design**

**Improving Testability**



# Summary



**Identify and Extract Interfaces**

**Move Behavior into New Classes**

**Inject Dependencies**

- Constructor (preferably)
- Possibly Methods or Properties

**Configure DI in Program.cs**



**Up Next:**

# **Refactoring Legacy Code to Follow OCP and LSP**

---

