# Intro & Autofac Primer w/App Demo

**Miguel A. Castro**
PRINCIPAL CONSULTANT

@miguelcastro67 www.melvicorp.com

# Overview

Autofac is one of the more *(if not the most)* modern DI containers for .NET

ASP.NET:

- MVC
- Web API

Very little DI prerequisite knowledge

"Getting Started with Dependency Injection in .NET"
*Jeremy Clark*

# Course Agenda

## DI Primer/App Demo

- Autofac registration & resolve
- Demo "before" app
  - No DI at all

## Autofac Integration

- NuGet packages
  - MVC
  - Web API
  - Configuration

## Registering Components

- Automatic registering
- Scanning
- Modules
- Configuration

# Course Agenda

## Injection & Resolve

- Application modifications
- On-Demand
- Injecting into Views

## Advanced Patterns

- Decorators
- ASP.NET Filters

## Additional Scenarios

- OWIN
- Web Forms
- Client-side Libraries

## Each module begins with code starter project that represents the completion of the last

Other courses

- Building End-to-End Multi-Client Service Oriented Applications
  *(Knockout and AngularJS editions)*
- Developing Extensible Software
- WCF End-to-End
- WCF Power Topics
- Designing & Building Component-based AngularJS Applications

# Autofac Primer

## Step 1
## Bad, Untestable Code

- Dependency "newing" embedded into consuming classes
- Stuck with production code for testing (bad)

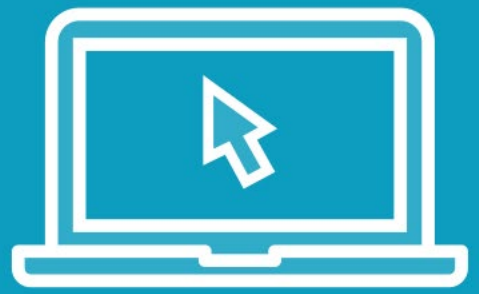
## Step 2
## Testable, but Hard to Maintain Code

- Everything gets abstracted
- Code is testable
- Hard to "new up"
- Hard to introduce new dependencies

## Step 3
## Fully DI-Aware Code

- All classes get registered in container
- Top-level class resolved from container
- Container recursively resolves everything in the object graph

# Demo



## Autofac Primer

# EasyBlog

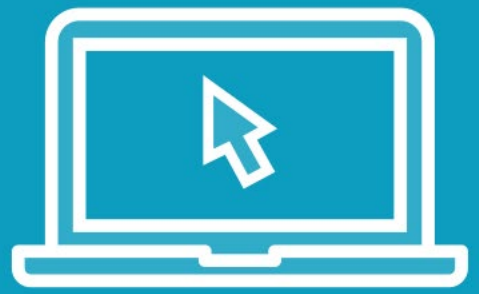
Simple Blog Engine written in ASP.NET MVC & Web API w/KnockoutJS

No DI whatsoever.
All dependencies “new’d” up in View & API Controllers

Exposes extensibility API allowing customization for multiple tenants

We’ll be DI’ing everything up, including the extensibility layer.

# Demo



## EasyBlog “the before state”