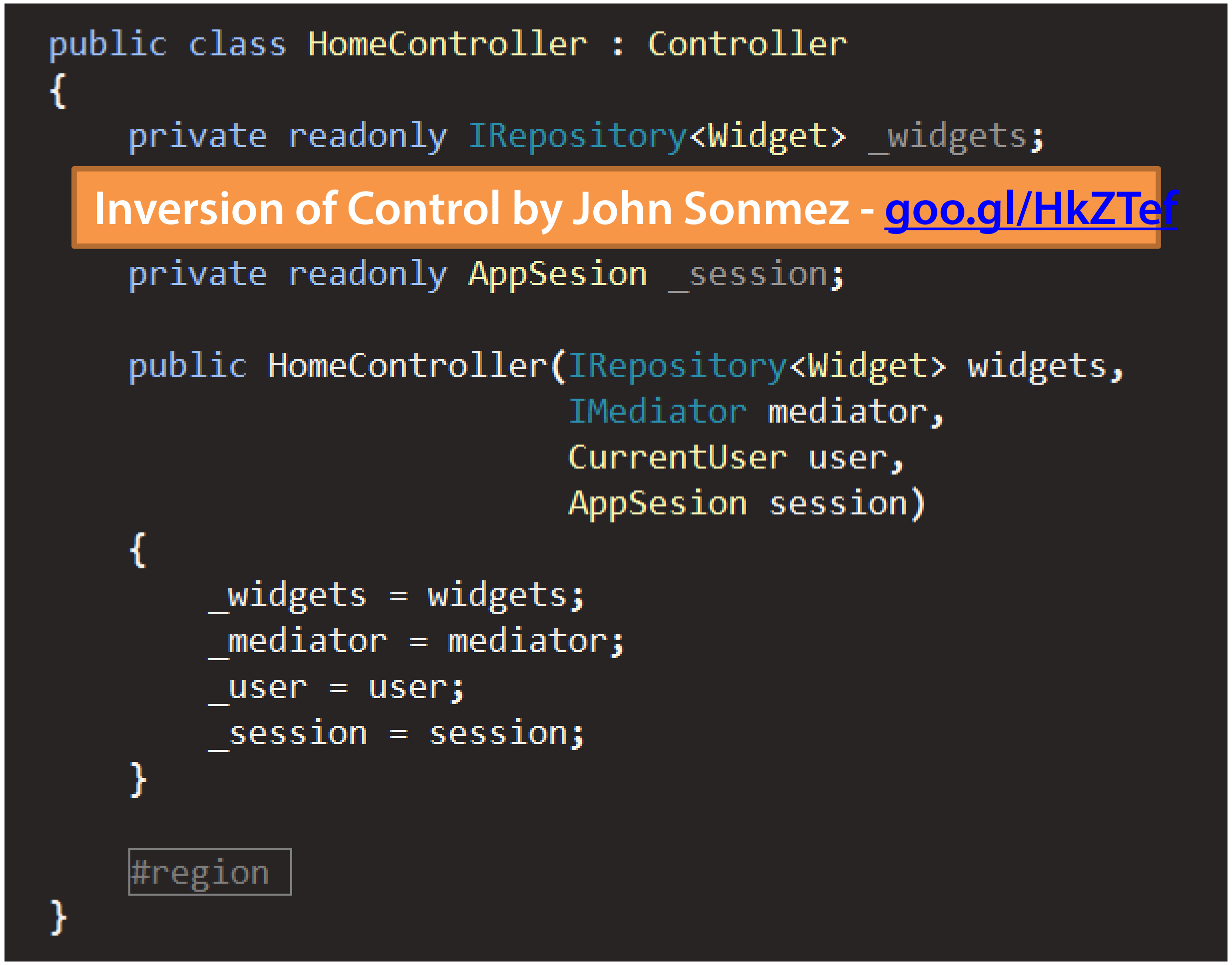
The Power of an Inversion of Control Container

Matt Honeycutt
http://trycatchfail.com
mbhoneycutt@gmail.com

pluralsight
hardcore developer training

Inversion of Control Basics



```
public class HomeController : Controller
{
    private readonly IRepository<Widget> _widgets;
```

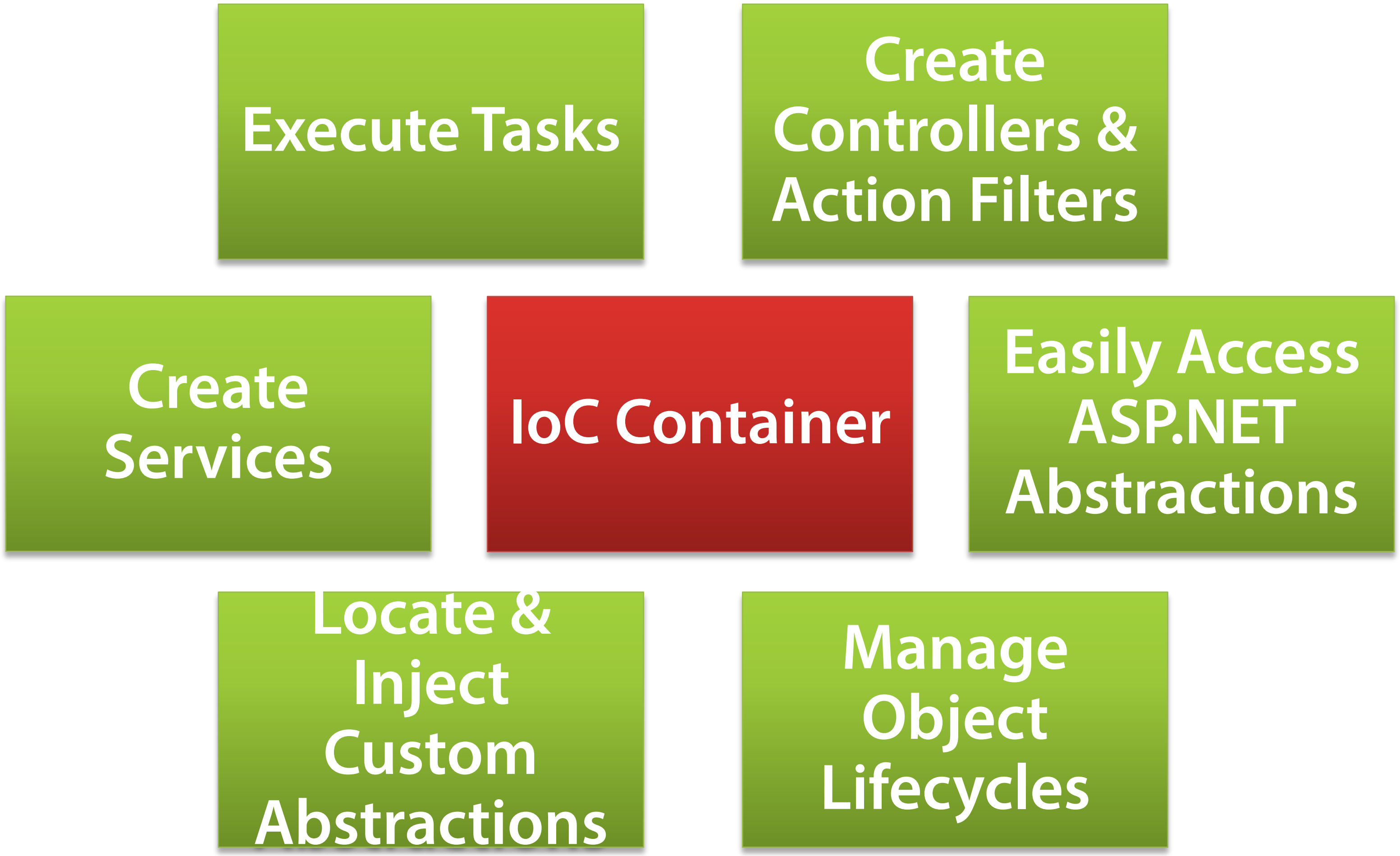
Inversion of Control by John Sonmez - goo.gl/HkZTef

```
    private readonly AppSesion _session;

    public HomeController(IRepository<Widget> widgets,
                          IMediator mediator,
                          CurrentUser user,
                          AppSesion session)
    {
        _widgets = widgets;
        _mediator = mediator;
        _user = user;
        _session = session;
    }

    #region
}
```

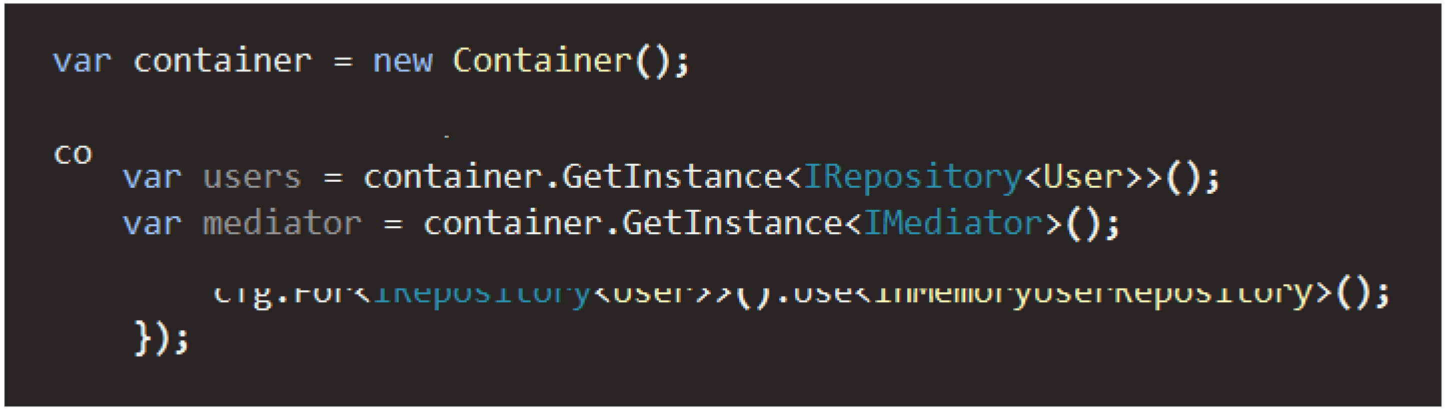
IoC in Our App Framework


Execute Tasks
Create Controllers & Action Filters
Create Services
IoC Container
Easily Access ASP.NET Abstractions
Locate & Inject Custom Abstractions
Manage Object Lifecycles

StructureMap Basics

IContainer

Configuration
Instance Creation
Lifecycle Management

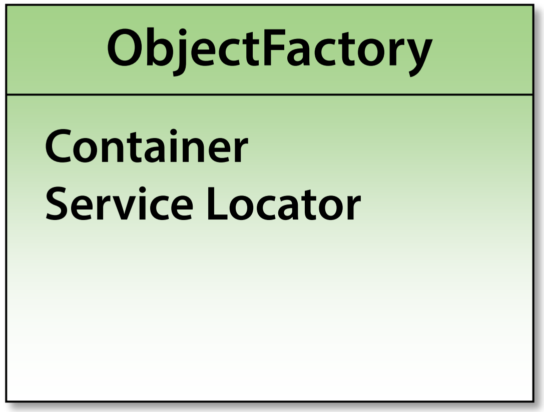


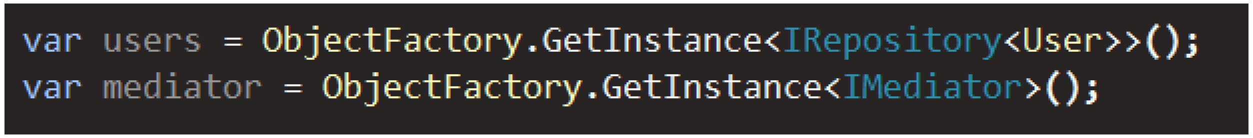
```
var container = new Container();

co
  var users = container.GetInstance<IRepository<User>>();
  var mediator = container.GetInstance<IMediator>();

     cfg.For<IRepository<User>>().Use<InMemoryUserRepository>();
  });
```

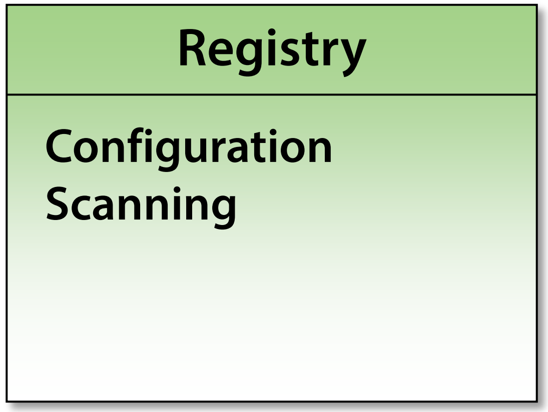
StructureMap Basics

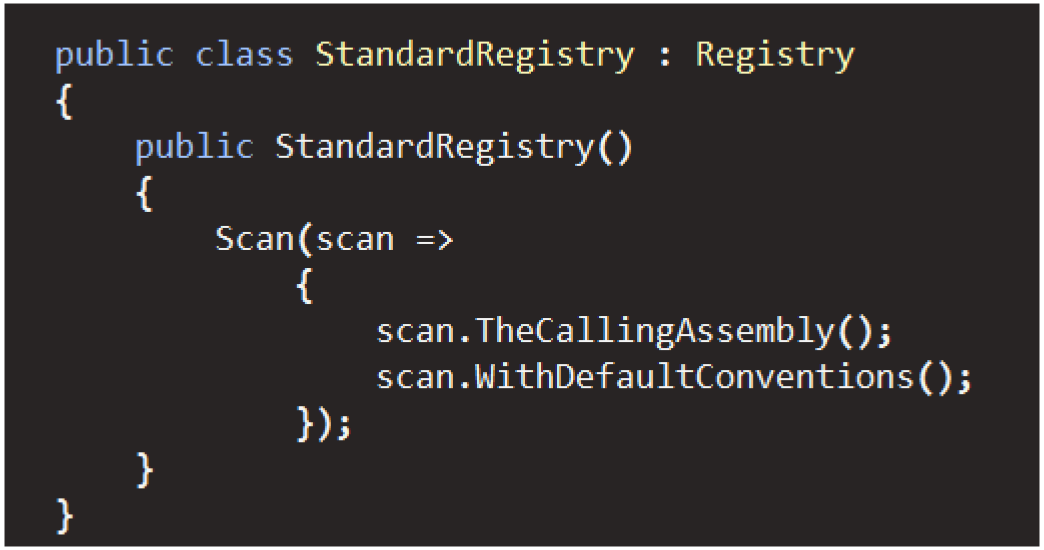




```
var users = ObjectFactory.GetInstance<IRepository<User>>();
var mediator = ObjectFactory.GetInstance<IMediator>();
```

StructureMap Basics



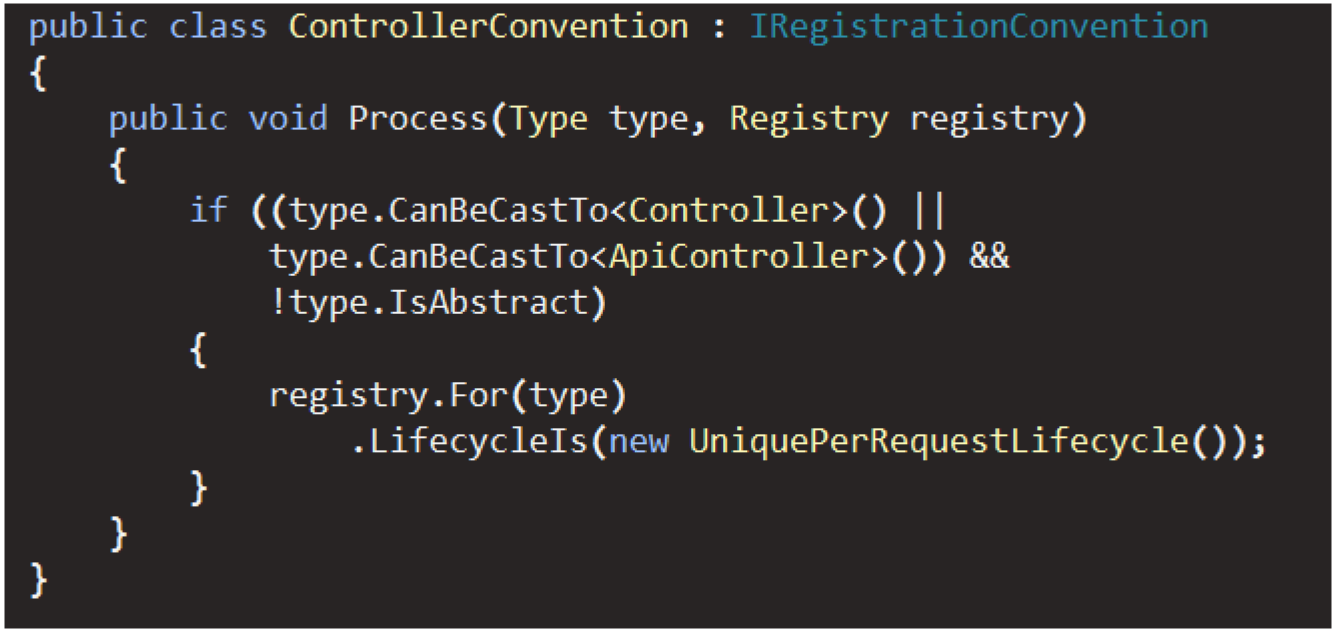


```
public class StandardRegistry : Registry
{
    public StandardRegistry()
    {
        Scan(scan =>
            {
                scan.TheCallingAssembly();
                scan.WithDefaultConventions();
            });
    }
}
```

StructureMap Basics

IRegistrationConvention

Custom registration conventions

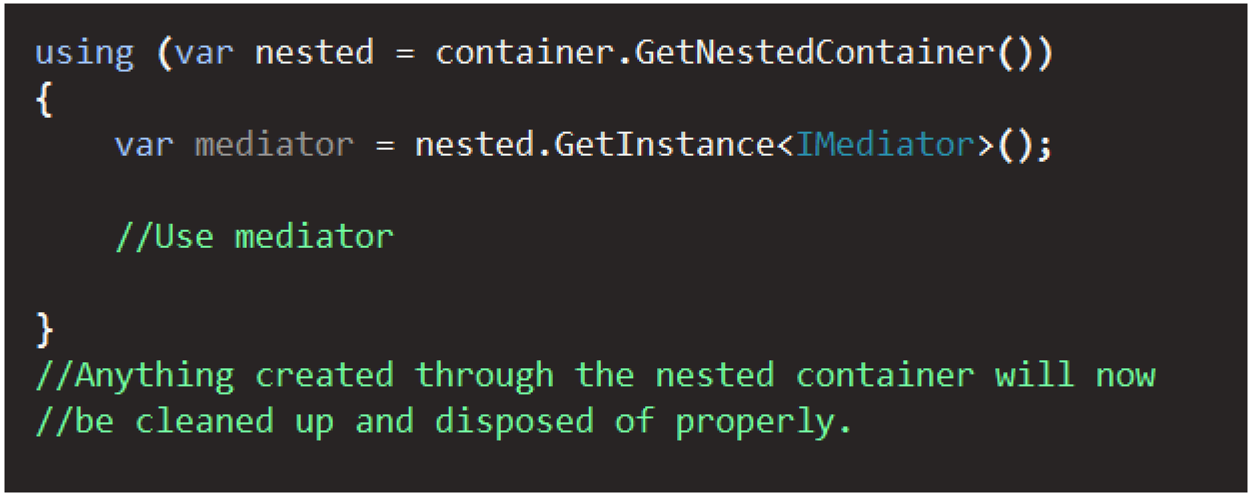


```
public class ControllerConvention : IRegistrationConvention
{
    public void Process(Type type, Registry registry)
    {
        if ((type.CanBeCastTo<Controller>() ||
            type.CanBeCastTo<ApiController>()) &&
            !type.IsAbstract)
        {
            registry.For(type)
                .LifecycleIs(new UniquePerRequestLifecycle());
        }
    }
}
```

StructureMap Basics

Nested Container

Scoped Container
Handles IDisposable
Controlled Lifecycle



```
using (var nested = container.GetNestedContainer())
{
    var mediator = nested.GetInstance<IMediator>();

    //Use mediator

}
//Anything created through the nested container will now
//be cleaned up and disposed of properly.
```

Demos

Setup

- Add StructureMap Package
- Implement Dependency Resolver
- Register Standard Conventions
- Controller Injection

Container-Per-Request

- Add Infrastructure
- Demonstrate Session-Per-Request

Action Filters

- Action Filter Injection Woes!
- Setter Injection for Action Filters

Injecting Abstractions

- Common ASP.NET Abstractions
- Current User

Tasks

- Startup Tasks
- Per-Request Tasks & On-Error Tasks

Demos

Setup	• Add StructureMap Package • Implement Dependency Resolver • Register Standard Conventions • Controller Injection
Container-Per-Request	• Add Infrastructure • Demonstrate Session-Per-Request
Action Filters	• Action Filter Injection Woes! • Setter Injection for Action Filters
Injecting Abstractions	• Common ASP.NET Abstractions • Current User
Tasks	• Startup Tasks • Per-Request Tasks & On-Error Tasks

Nested Container

Transient objects are singleton

Disposes objects recursively

Demos

Setup	• Add StructureMap Package • Implement Dependency Resolver • Register Standard Conventions • Controller Injection
Container-Per-Request	• Add Infrastructure • Demonstrate Session-Per-Request
Action Filters	• Action Filter Injection Woes! • Setter Injection for Action Filters
Injecting Abstractions	• Common ASP.NET Abstractions • Current User
Tasks	• Startup Tasks • Per-Request Tasks & On-Error Tasks

Demos

- Setup
  - Add StructureMap Package
  - Implement Dependency Resolver
  - Register Standard Conventions
  - Controller Injection
- Container-Per-Request
  - Add Infrastructure
  - Demonstrate Session-Per-Request
- Action Filters
  - Action Filter Injection Woes!
  - Setter Injection for Action Filters
- **Injecting Abstractions**
  - Common ASP.NET Abstractions
  - Current User
- Tasks
  - Startup Tasks
  - Per-Request Tasks & On-Error Tasks

Demos

Setup

- Add StructureMap Package
- Implement Dependency Resolver
- Register Standard Conventions
- Controller Injection

Container-Per-Request

- Add Infrastructure
- Demonstrate Session-Per-Request

Action Filters

- Action Filter Injection Woes!
- Setter Injection for Action Filters

Injecting Abstractions

- Common ASP.NET Abstractions
- Current User

Tasks

- Startup Tasks
- Per-Request Tasks & On-Error Tasks

Summary

- ✓ StructureMap Integration
- ✓ Custom Dependency Resolver
- ✓ Container-Per-Request
- ✓ Session-Per-Request
- ✓ Custom Action Filter Provider
- ✓ Common Abstractions
- ✓ Custom Abstractions (Current User)
- ✓ Tasks
- ✓ Transaction-Per-Request

Summary

- Tailor to <u>your</u> needs!
- Pick and choose!

Up next: Streamline your controlle