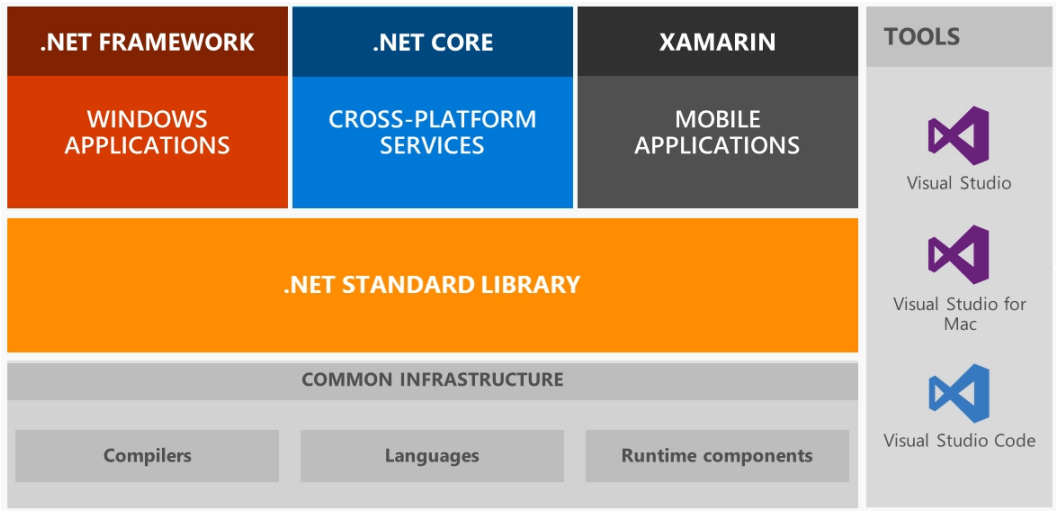
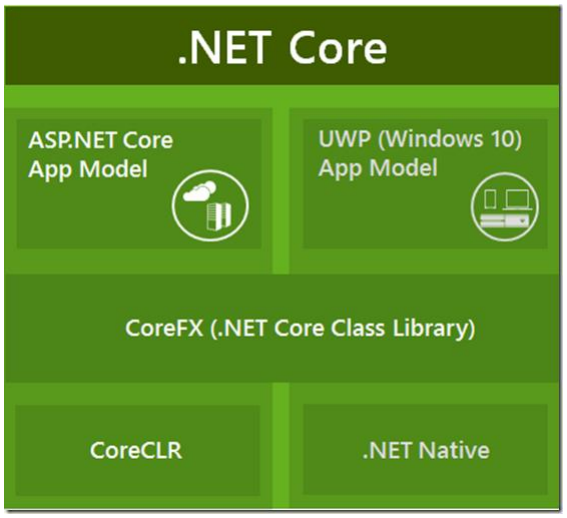
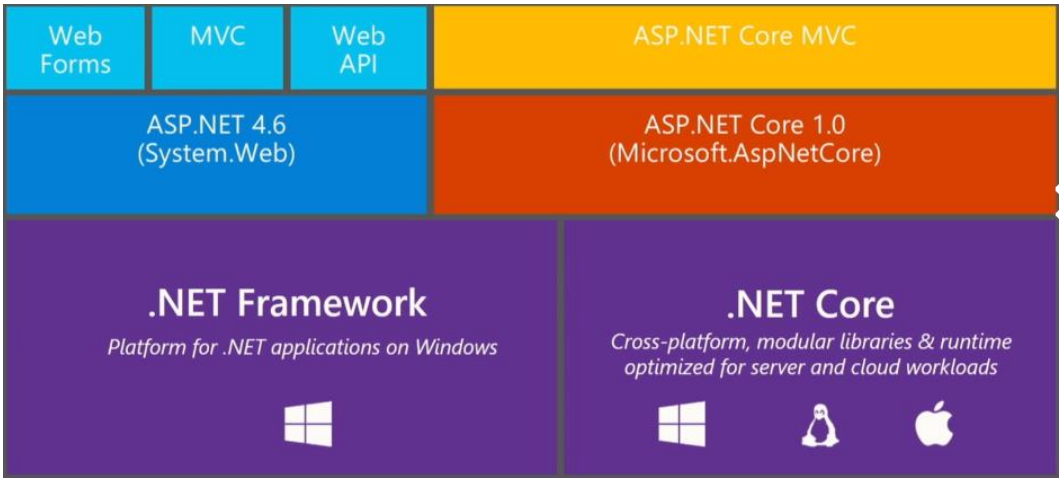
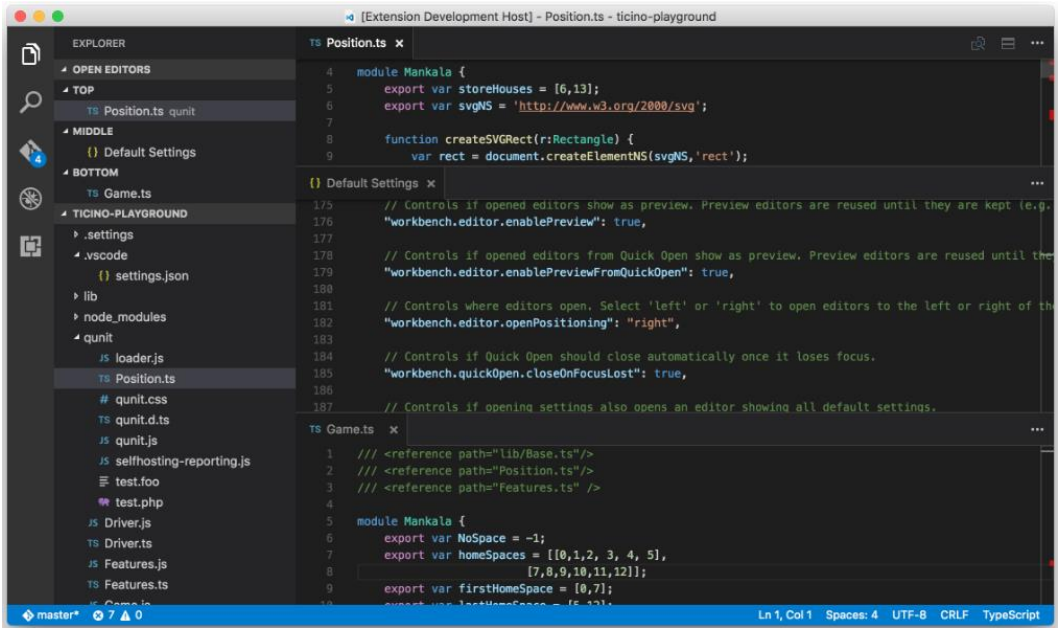
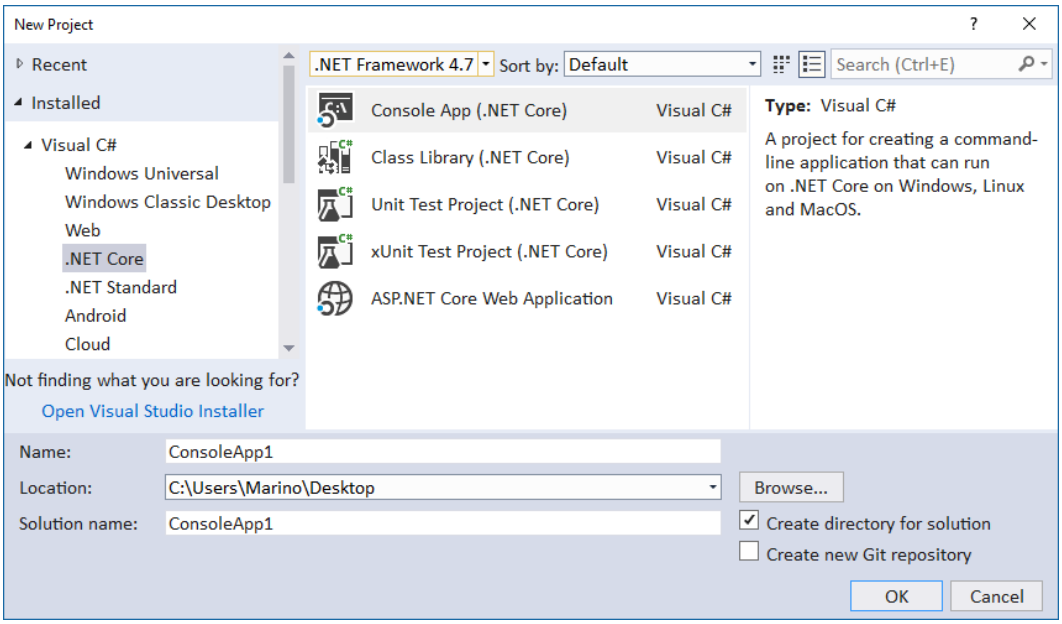
Chapter 1: The SOLID Principles of Software Design



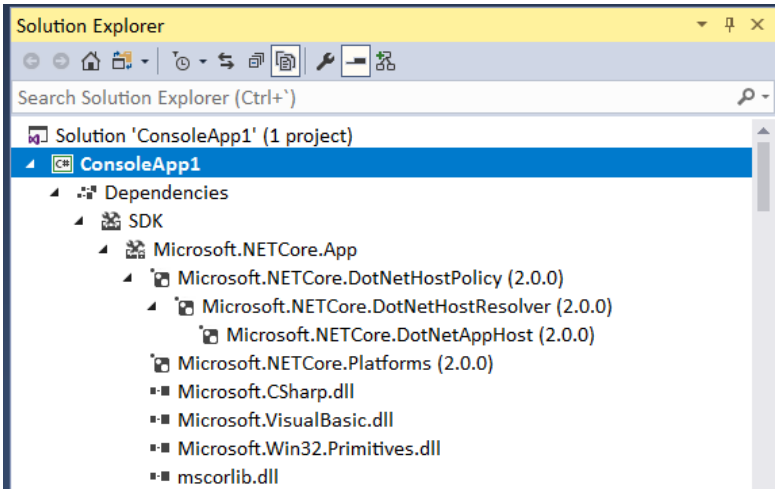


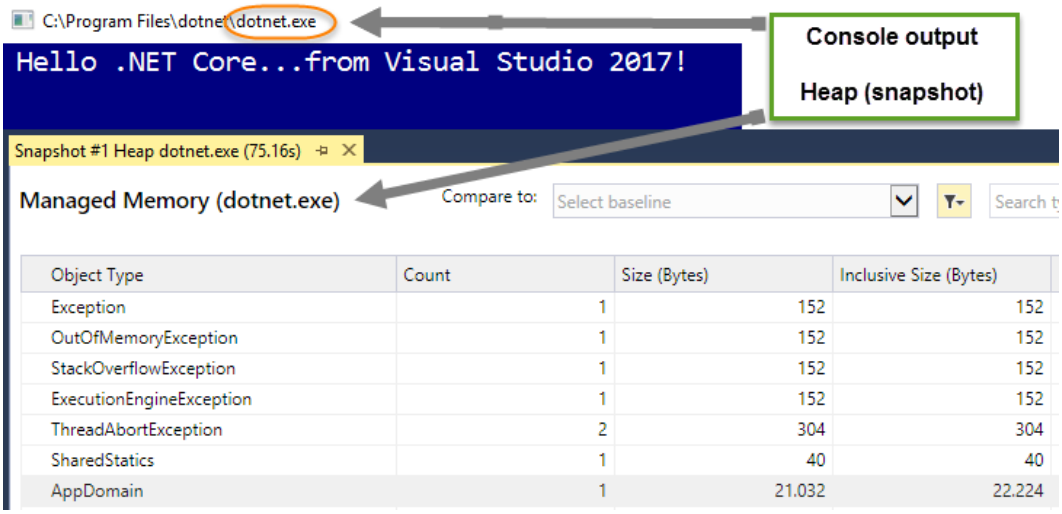

Web Forms
MVC
Web API
ASP.NET Core MVC
ASP.NET 4.6 (System.Web)
ASP.NET Core 1.0 (Microsoft.AspNetCore)
.NET Framework
Platform for .NET applications on Windows
.NET Core
Cross-platform, modular libraries & runtime optimized for server and cloud workloads

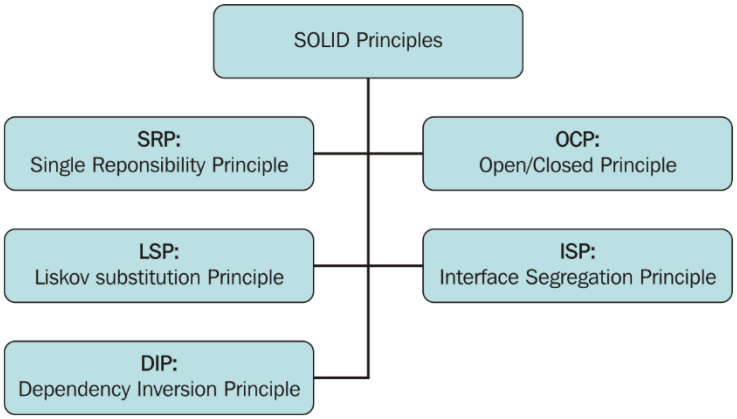

[Extension Development Host] - Position.ts - ticino-playground
EXPLORER
OPEN EDITORS
TOP
Position.ts qunit
MIDDLE
Default Settings
BOTTOM
Game.ts
TICINO-PLAYGROUND
.settings
.vscode
settings.json
lib
node_modules
qunit
loader.js
Position.ts
qunit.css
qunit.d.ts
qunit.js
selfhosting-reporting.js
test.foo
test.php
Driver.js
Driver.ts
Features.js
Features.ts
Position.ts
module Mankala {
export var storeHouses = [6,13];
export var svgNS = 'http://www.w3.org/2000/svg';
function createSVGRect(r:Rectangle) {
var rect = document.createElementNS(svgNS,'rect');
Default Settings
// Controls if opened editors show as preview. Preview editors are reused until they are kept (e.g.
"workbench.editor.enablePreview": true,
// Controls if opened editors from Quick Open show as preview. Preview editors are reused until the
"workbench.editor.enablePreviewFromQuickOpen": true,
// Controls where editors open. Select 'left' or 'right' to open editors to the left or right of th
"workbench.editor.openPositioning": "right",
// Controls if Quick Open should close automatically once it loses focus.
"workbench.quickOpen.closeOnFocusLost": true,
// Controls if opening settings also opens an editor showing all default settings.
Game.ts
/// <reference path="lib/Base.ts"/>
/// <reference path="Position.ts"/>
/// <reference path="Features.ts" />
module Mankala {
export var NoSpace = -1;
export var homeSpaces = [[0,1,2, 3, 4, 5],
[7,8,9,10,11,12]];
export var firstHomeSpace = [0,7];
master*
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
TypeScript

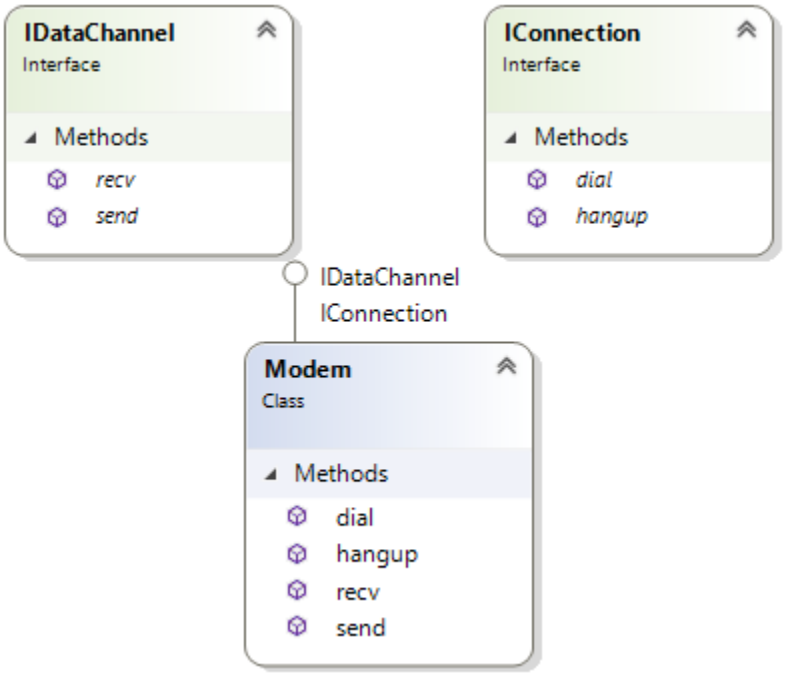


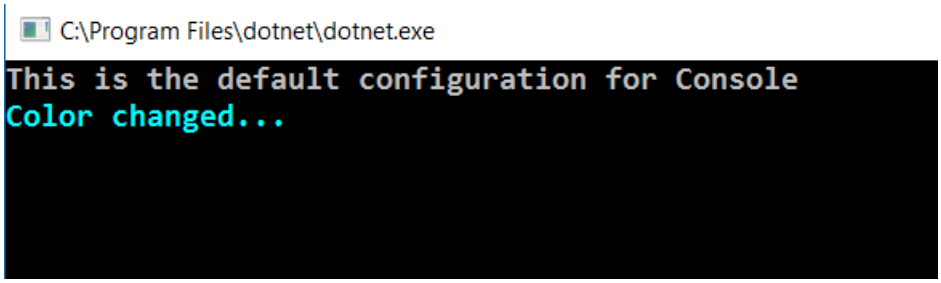

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>netcoreapp2.0</TargetFramework>
  </PropertyGroup>

</Project>
```

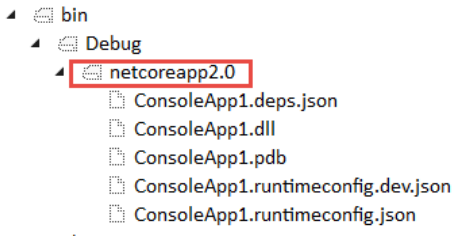

Solution Explorer
Search Solution Explorer (Ctrl+`)
Solution 'ConsoleApp1' (1 project)
ConsoleApp1
Dependencies
SDK
Microsoft.NETCore.App
Microsoft.NETCore.DotNetHostPolicy (2.0.0)
Microsoft.NETCore.DotNetHostResolver (2.0.0)
Microsoft.NETCore.DotNetAppHost (2.0.0)
Microsoft.NETCore.Platforms (2.0.0)
Microsoft.CSharp.dll
Microsoft.VisualBasic.dll
Microsoft.Win32.Primitives.dll
mscorlib.dll


C:\Program Files\dotnet\dotnet.exe
Hello .NET Core...from Visual Studio 2017!
Console output
Heap (snapshot)
Snapshot #1 Heap dotnet.exe (75.16s)
Managed Memory (dotnet.exe)
Compare to:
Select baseline
Search t
Object Type
Count
Size (Bytes)
Inclusive Size (Bytes)
Exception 1 152 152
OutOfMemoryException 1 152 152
StackOverflowException 1 152 152
ExecutionEngineException 1 152 152
ThreadAbortException 2 304 304
SharedStatics 1 40 40
AppDomain 1 21.032 22.224


SOLID Principles
SRP:
Single Reponsibility Principle
OCP:
Open/Closed Principle
LSP:
Liskov substitution Principle
ISP:
Interface Segregation Principle
DIP:
Dependency Inversion Principle

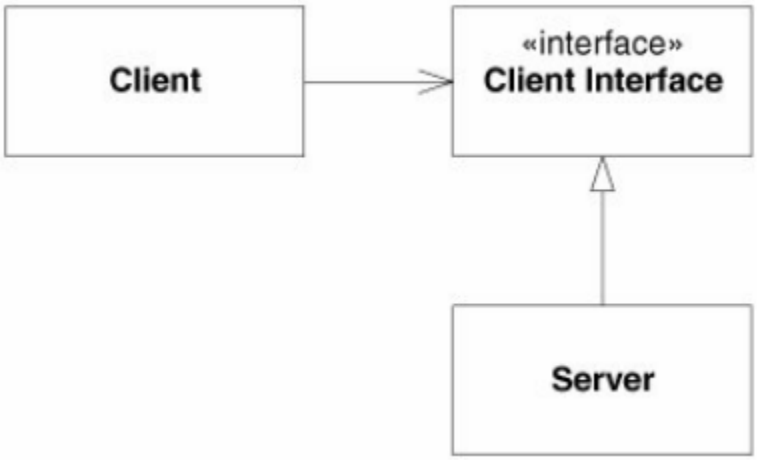

IDataChannel
Interface
Methods
recv
send
IConnection
Interface
Methods
dial
hangup
IDataChannel
IConnection
Modem
Class
Methods
dial
hangup
recv
send


C:\Program Files\dotnet\dotnet.exe
This is the default configuration for Console
Color changed...

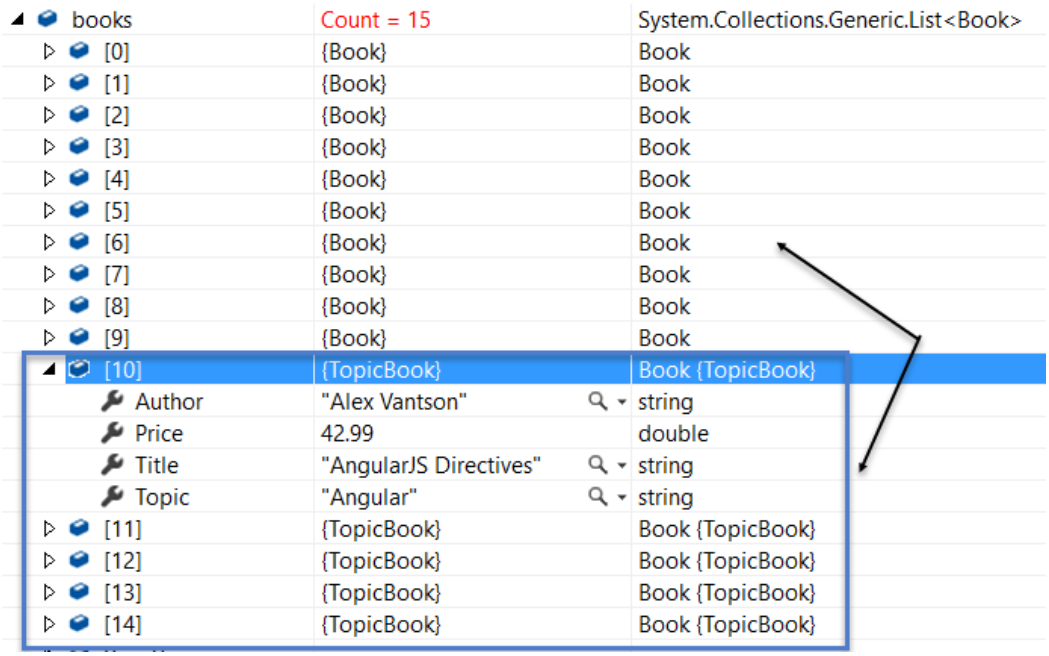

bin
Debug
netcoreapp2.0
ConsoleApp1.deps.json
ConsoleApp1.dll
ConsoleApp1.pdb
ConsoleApp1.runtimeconfig.dev.json
ConsoleApp1.runtimeconfig.json

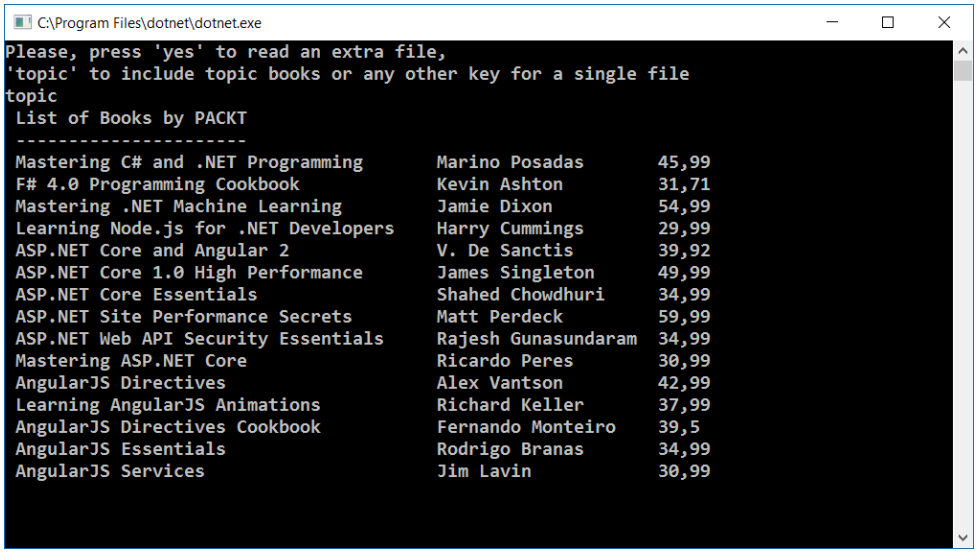
C:\Program Files\dotnet\dotnet.exe
List of Books by PACKT

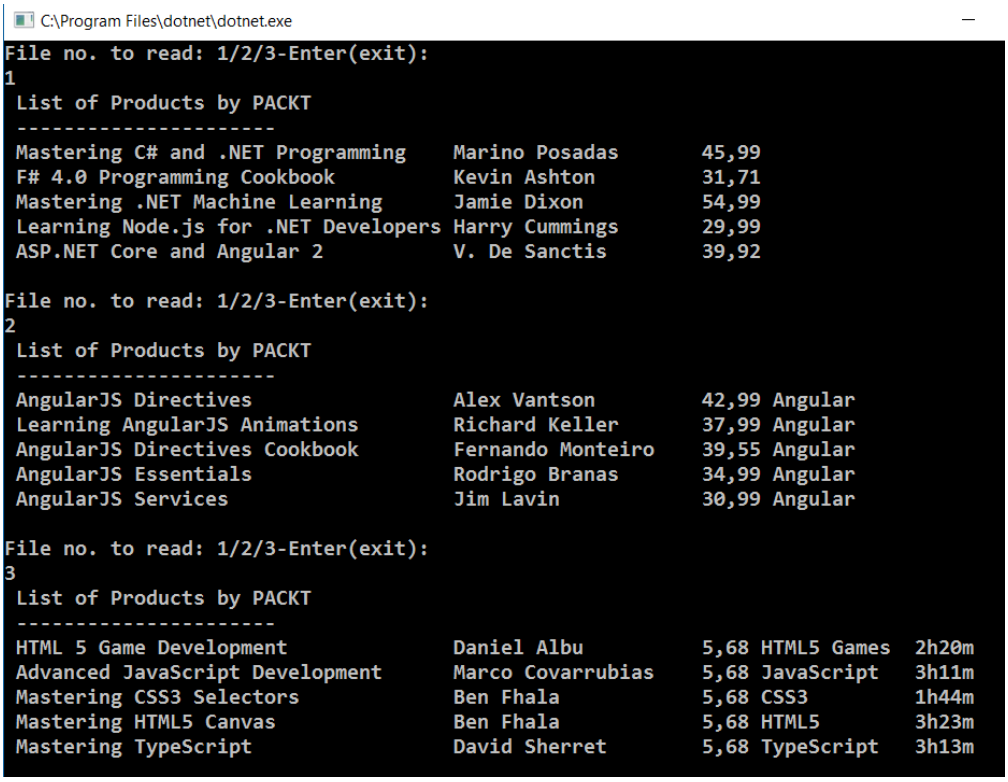
Mastering C# and .NET Programming Marino Posadas 45,99
F# 4.0 Programming Cookbook Kevin Ashton 31,71
Mastering .NET Machine Learning Jamie Dixon 54,99
Learning Node.js for .NET Developers Harry Cummings 29,99
ASP.NET Core and Angular 2 V. De Sanctis 39,92



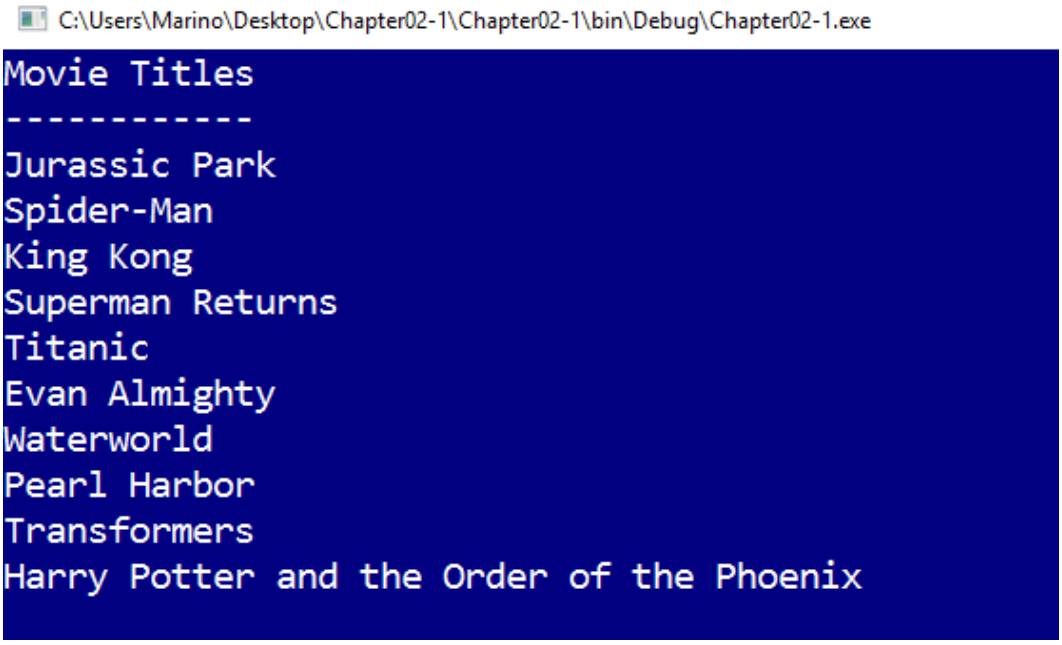

```
C:\Program Files\dotnet\dotnet.exe
Please, press 'yes' to read an extra file,
or any other key for a single file
yes
 List of Books by PACKT
 ----------------------
 Mastering C# and .NET Programming       Marino Posadas       45,99
 F# 4.0 Programming Cookbook             Kevin Ashton         31,71
 Mastering .NET Machine Learning         Jamie Dixon          54,99
 Learning Node.js for .NET Developers    Harry Cummings       29,99
 ASP.NET Core and Angular 2              V. De Sanctis        39,92
 ASP.NET Core 1.0 High Performance       James Singleton      49,99
 ASP.NET Core Essentials                 Shahed Chowdhuri     34,99
 ASP.NET Site Performance Secrets        Matt Perdeck         59,99
 ASP.NET Web API Security Essentials     Rajesh Gunasundaram  34,99
 Mastering ASP.NET Core                  Ricardo Peres        30,99
```

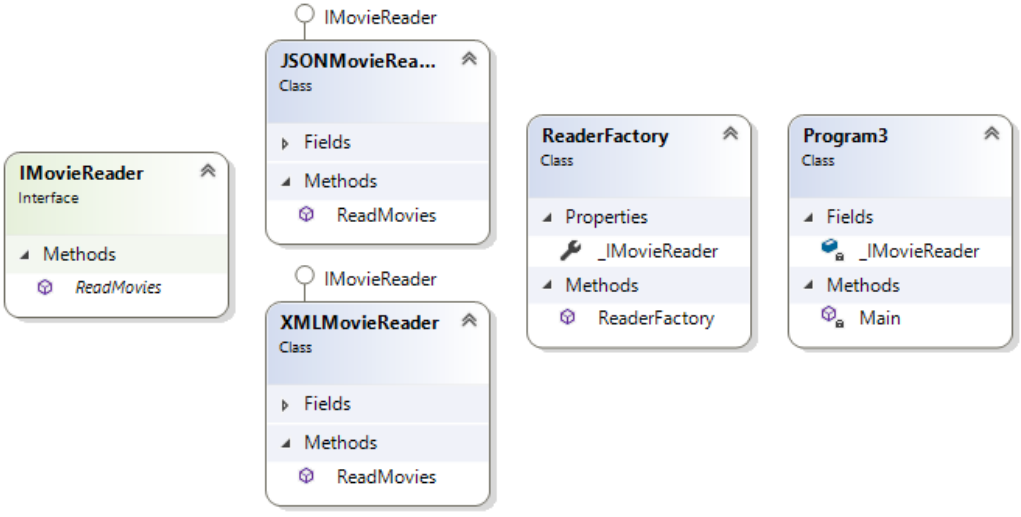

books
Count = 15
System.Collections.Generic.List<Book>
[0] {Book} Book
[1] {Book} Book
[2] {Book} Book
[3] {Book} Book
[4] {Book} Book
[5] {Book} Book
[6] {Book} Book
[7] {Book} Book
[8] {Book} Book
[9] {Book} Book
[10] {TopicBook} Book {TopicBook}
Author "Alex Vantson" string
Price 42.99 double
Title "AngularJS Directives" string
Topic "Angular" string
[11] {TopicBook} Book {TopicBook}
[12] {TopicBook} Book {TopicBook}
[13] {TopicBook} Book {TopicBook}
[14] {TopicBook} Book {TopicBook}


C:\Program Files\dotnet\dotnet.exe
Please, press 'yes' to read an extra file,
'topic' to include topic books or any other key for a single file
topic
List of Books by PACKT
Mastering C# and .NET Programming Marino Posadas 45,99
F# 4.0 Programming Cookbook Kevin Ashton 31,71
Mastering .NET Machine Learning Jamie Dixon 54,99
Learning Node.js for .NET Developers Harry Cummings 29,99
ASP.NET Core and Angular 2 V. De Sanctis 39,92
ASP.NET Core 1.0 High Performance James Singleton 49,99
ASP.NET Core Essentials Shahed Chowdhuri 34,99
ASP.NET Site Performance Secrets Matt Perdeck 59,99
ASP.NET Web API Security Essentials Rajesh Gunasundaram 34,99
Mastering ASP.NET Core Ricardo Peres 30,99
AngularJS Directives Alex Vantson 42,99
Learning AngularJS Animations Richard Keller 37,99
AngularJS Directives Cookbook Fernando Monteiro 39,5
AngularJS Essentials Rodrigo Branas 34,99
AngularJS Services Jim Lavin 30,99


C:\Program Files\dotnet\dotnet.exe
File no. to read: 1/2/3-Enter(exit):
1
List of Products by PACKT
Mastering C# and .NET Programming Marino Posadas 45,99
F# 4.0 Programming Cookbook Kevin Ashton 31,71
Mastering .NET Machine Learning Jamie Dixon 54,99
Learning Node.js for .NET Developers Harry Cummings 29,99
ASP.NET Core and Angular 2 V. De Sanctis 39,92
File no. to read: 1/2/3-Enter(exit):
2
List of Products by PACKT
AngularJS Directives Alex Vantson 42,99 Angular
Learning AngularJS Animations Richard Keller 37,99 Angular
AngularJS Directives Cookbook Fernando Monteiro 39,55 Angular
AngularJS Essentials Rodrigo Branas 34,99 Angular
AngularJS Services Jim Lavin 30,99 Angular
File no. to read: 1/2/3-Enter(exit):
3
List of Products by PACKT
HTML 5 Game Development Daniel Albu 5,68 HTML5 Games 2h20m
Advanced JavaScript Development Marco Covarrubias 5,68 JavaScript 3h11m
Mastering CSS3 Selectors Ben Fhala 5,68 CSS3 1h44m
Mastering HTML5 Canvas Ben Fhala 5,68 HTML5 3h23m
Mastering TypeScript David Sherret 5,68 TypeScript 3h13m

Chapter 2: Dependency Injection and IoC Containers

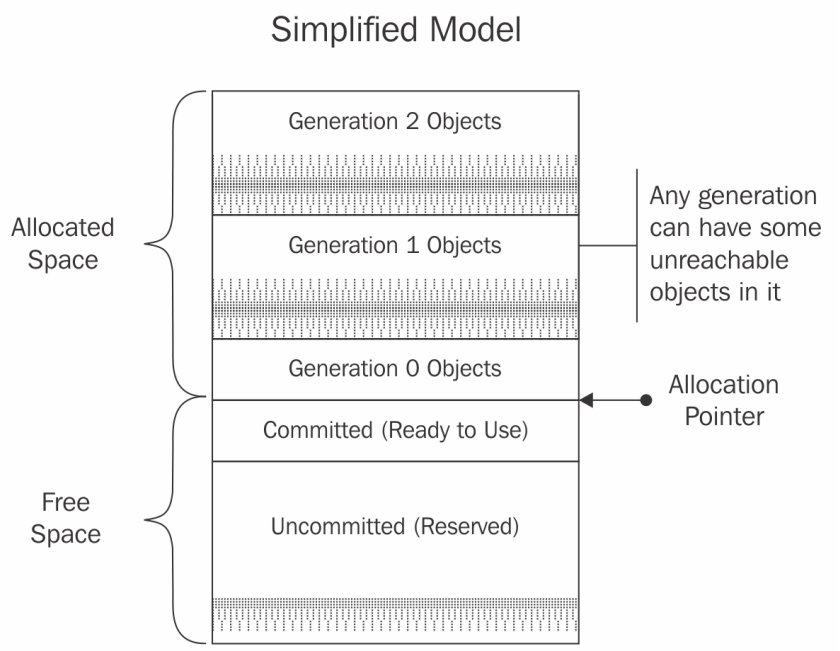


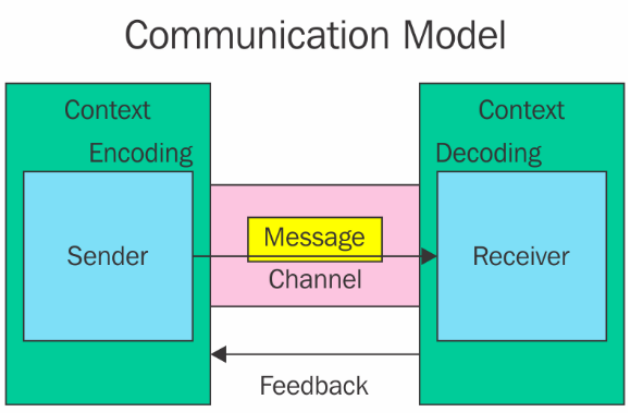


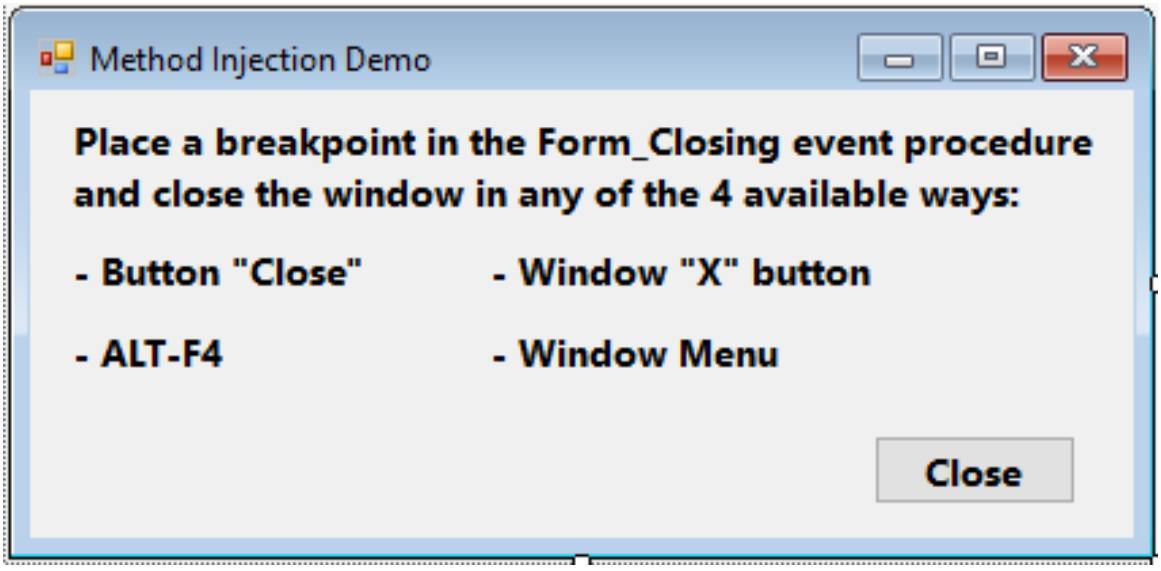
```
C:\Users\Marino\Desktop\Chapter02-1\Chapter02-1\bin\Debug\Chapter02-1.exe
Please, select the file type to read (1) XML, (2) JSON:
2
Movie Titles (JSONMovieReader)
------------
Jurassic Park
Spider-Man
King Kong
Superman Returns
Titanic
Evan Almighty
Waterworld
Pearl Harbor
Transformers
Harry Potter and the Order of the Phoenix
```

Composition: every car has an engine.

Aggregation: cars may have passengers, they come and go


Simplified Model
Generation 2 Objects
Generation 1 Objects
Generation 0 Objects
Committed (Ready to Use)
Uncommitted (Reserved)
Allocated Space
Free Space
Any generation can have some unreachable objects in it
Allocation Pointer

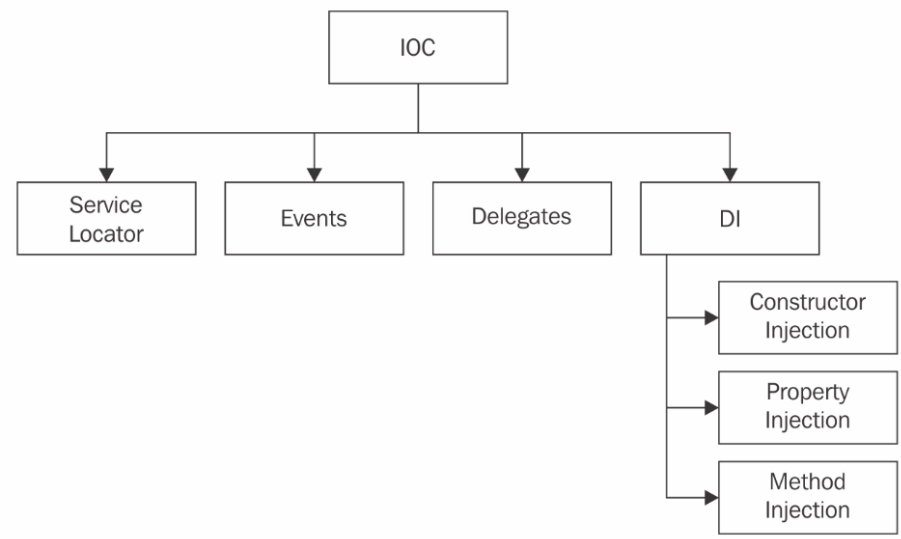

Communication Model
Context Encoding
Sender
Message
Channel
Feedback
Context Decoding
Receiver

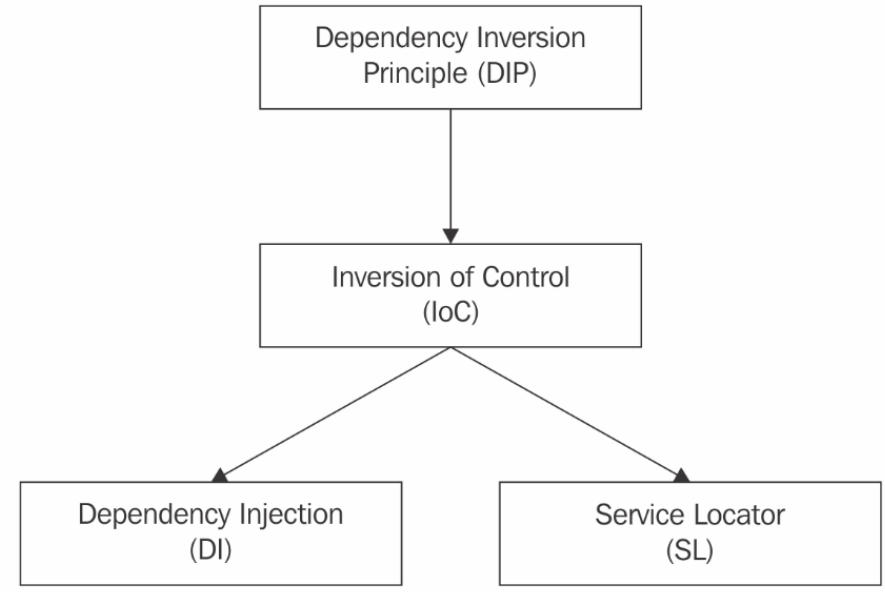


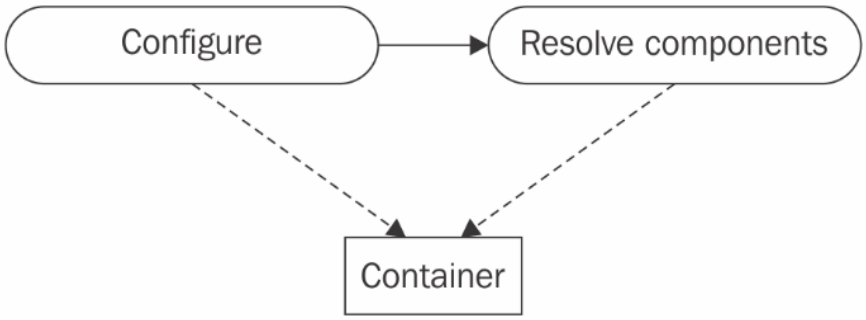

```
1 reference
private void frmMethodInjection_FormClosing(object sender, FormClosingEventArgs e)
{
    e.
}
```

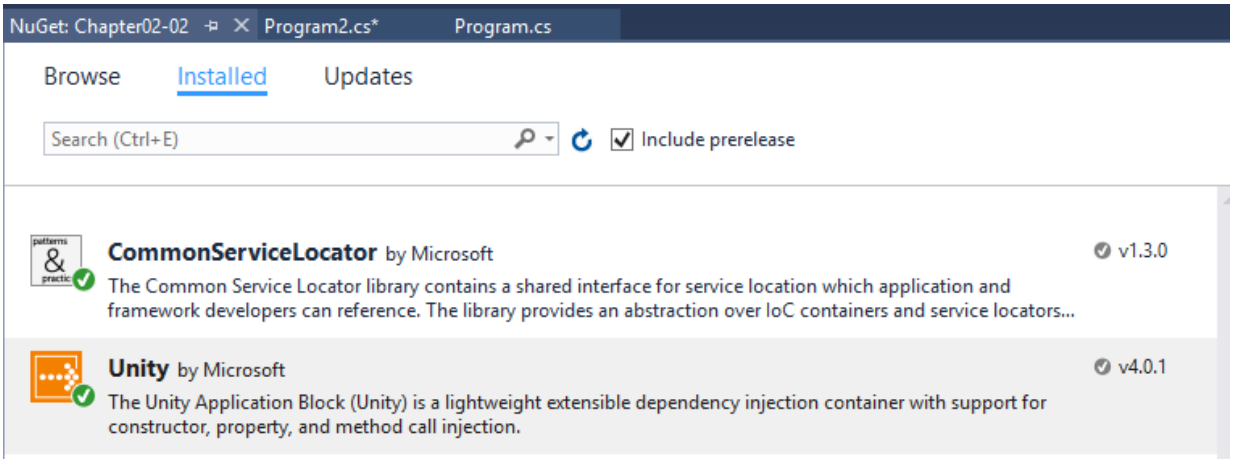
Cancel
CloseReason

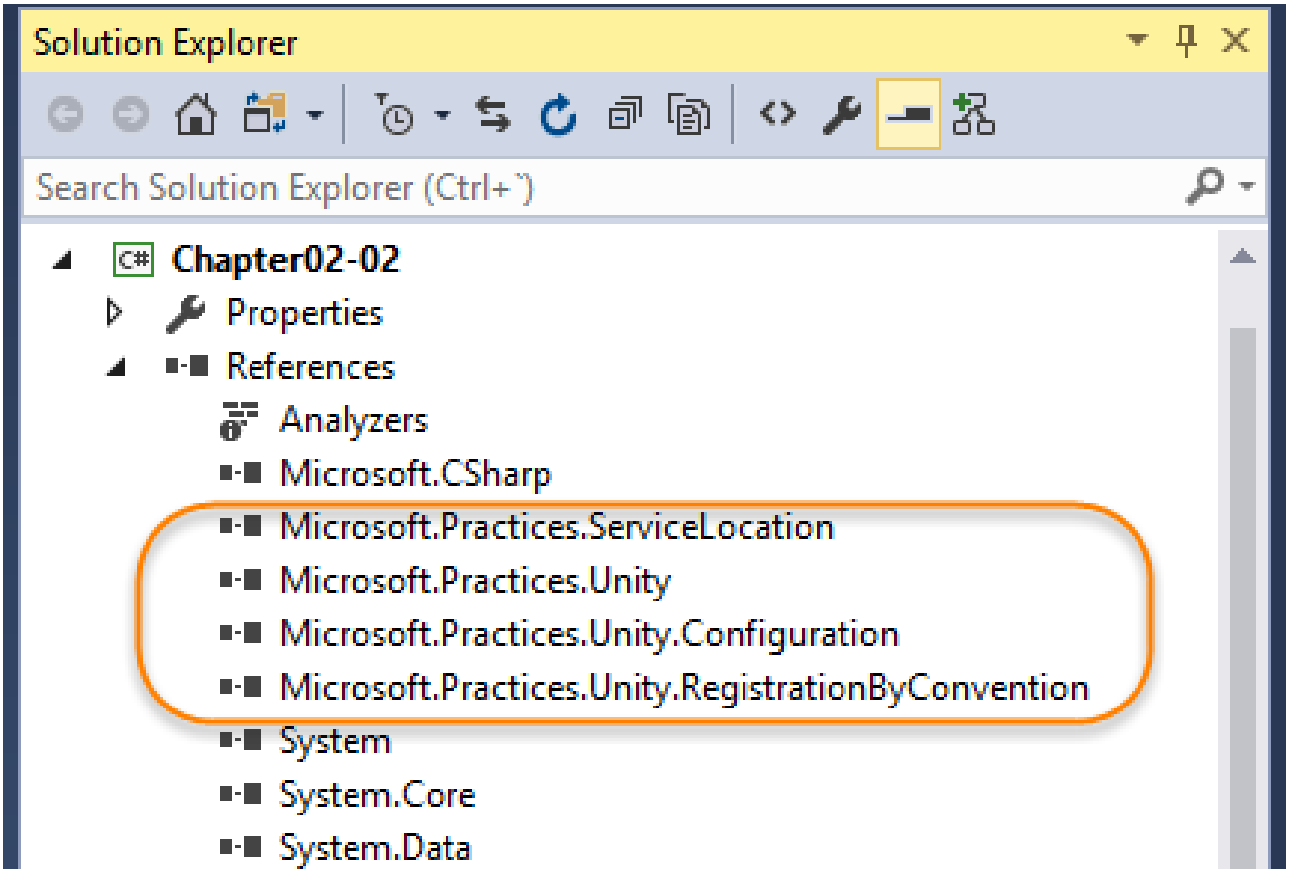
bool CancelEventArgs.Cancel { get; set; }
Gets or sets a value indicating whether the event should be canceled.

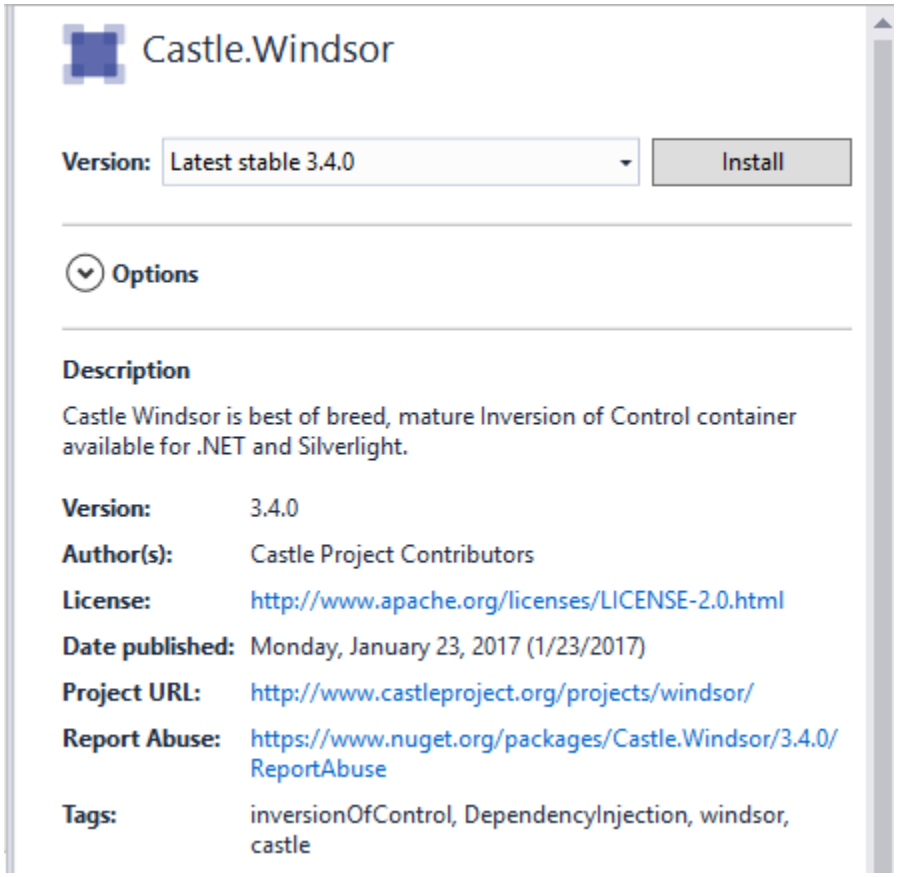

IOC
Service Locator
Events
Delegates
DI
Constructor Injection
Property Injection
Method Injection

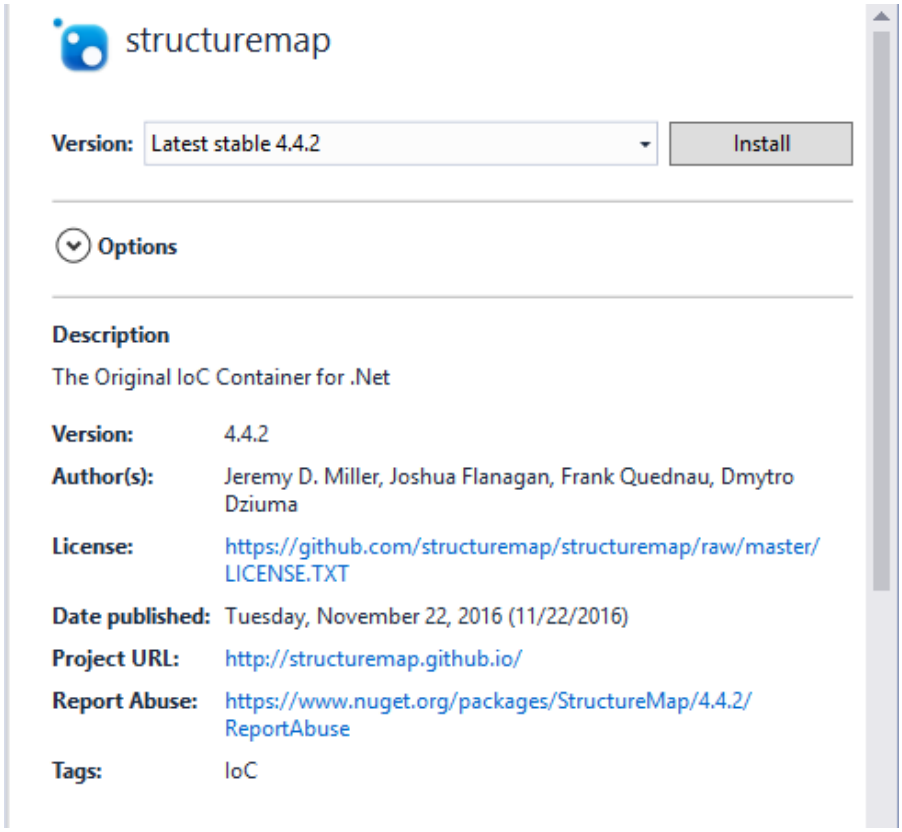

Dependency Inversion Principle (DIP)
Inversion of Control (IoC)
Dependency Injection (DI)
Service Locator (SL)

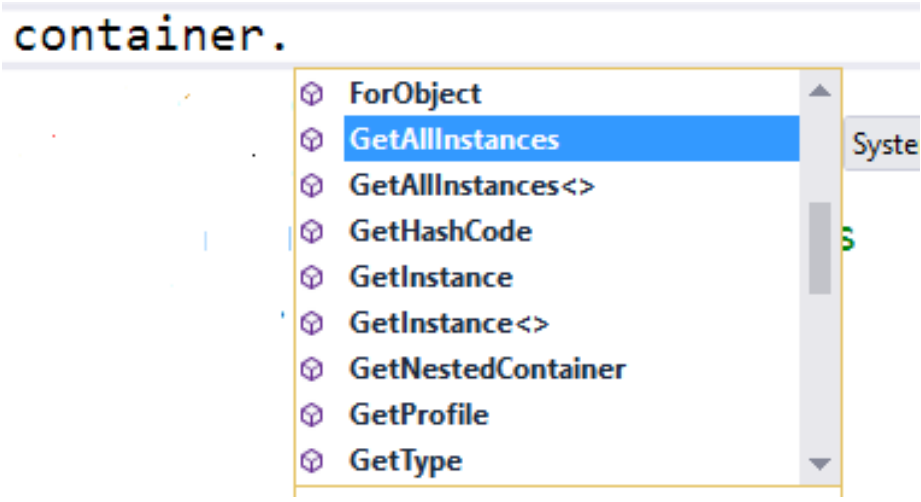

Configure
Resolve components
Container

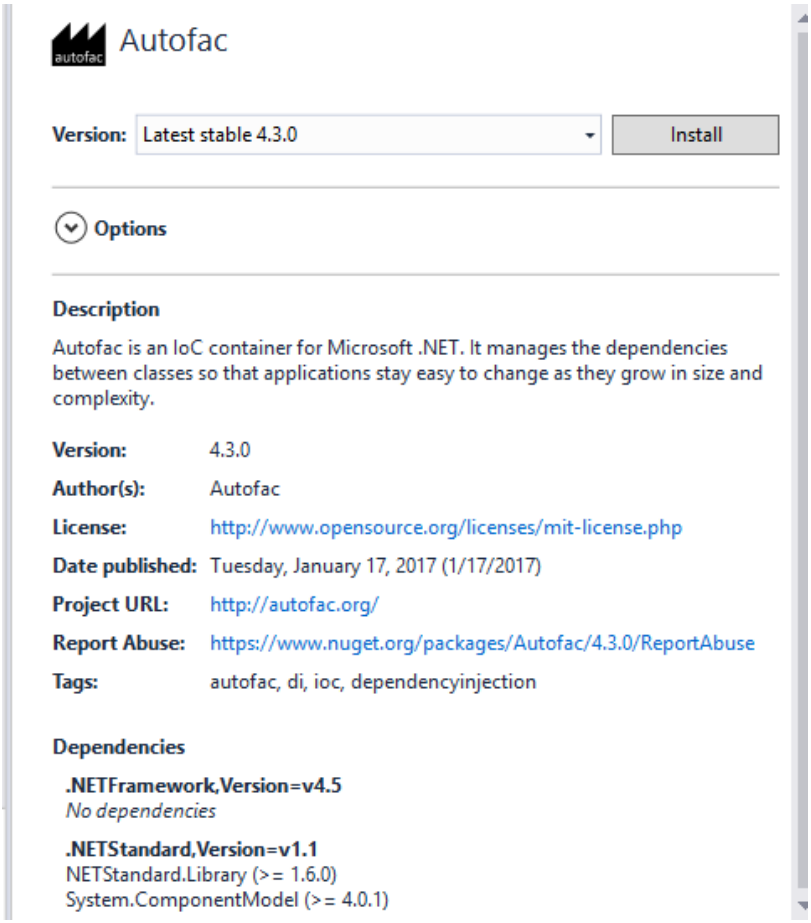

NuGet: Chapter02-02
Program2.cs*
Program.cs
Browse
Installed
Updates
Search (Ctrl+E)
Include prerelease
CommonServiceLocator by Microsoft
v1.3.0
The Common Service Locator library contains a shared interface for service location which application and framework developers can reference. The library provides an abstraction over IoC containers and service locators...
Unity by Microsoft
v4.0.1
The Unity Application Block (Unity) is a lightweight extensible dependency injection container with support for constructor, property, and method call injection.


Solution Explorer
Search Solution Explorer (Ctrl+`)
Chapter02-02
Properties
References
Analyzers
Microsoft.CSharp
Microsoft.Practices.ServiceLocation
Microsoft.Practices.Unity
Microsoft.Practices.Unity.Configuration
Microsoft.Practices.Unity.RegistrationByConvention
System
System.Core
System.Data


Castle.Windsor
Version:
Latest stable 3.4.0
Install
Options
Description
Castle Windsor is best of breed, mature Inversion of Control container available for .NET and Silverlight.
Version:
3.4.0
Author(s):
Castle Project Contributors
License:
http://www.apache.org/licenses/LICENSE-2.0.html
Date published:
Monday, January 23, 2017 (1/23/2017)
Project URL:
http://www.castleproject.org/projects/windsor/
Report Abuse:
https://www.nuget.org/packages/Castle.Windsor/3.4.0/ReportAbuse
Tags:
inversionOfControl, DependencyInjection, windsor, castle


structuremap
Version: Latest stable 4.4.2
Install
Options
Description
The Original IoC Container for .Net
Version: 4.4.2
Author(s): Jeremy D. Miller, Joshua Flanagan, Frank Quednau, Dmytro Dziuma
License: https://github.com/structuremap/structuremap/raw/master/LICENSE.TXT
Date published: Tuesday, November 22, 2016 (11/22/2016)
Project URL: http://structuremap.github.io/
Report Abuse: https://www.nuget.org/packages/StructureMap/4.4.2/ReportAbuse
Tags: IoC

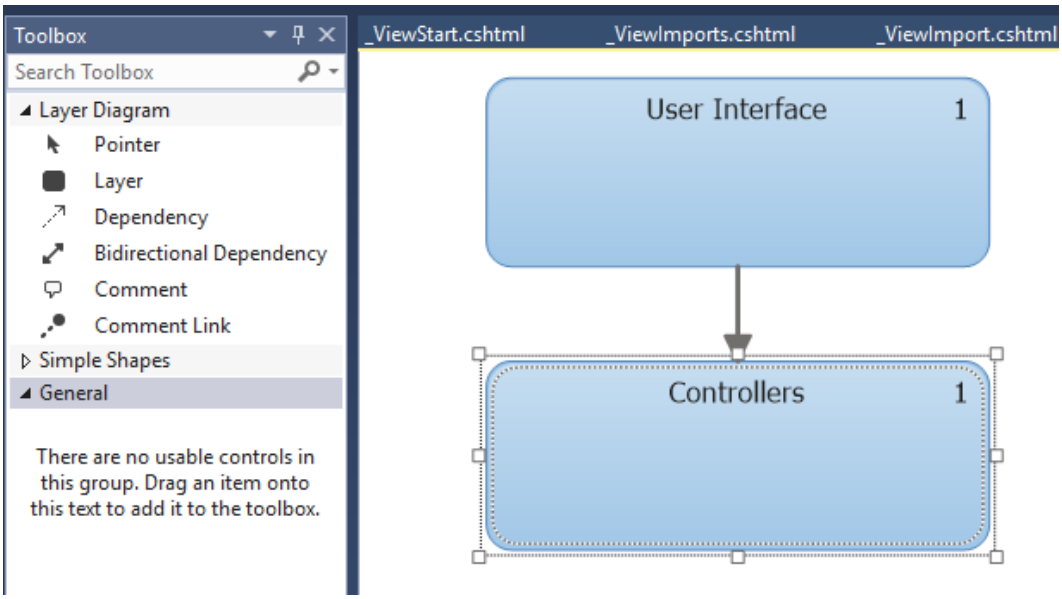

container.
ForObject
GetAllInstances
GetAllInstances<>
GetHashCode
GetInstance
GetInstance<>
GetNestedContainer
GetProfile
GetType
Syste

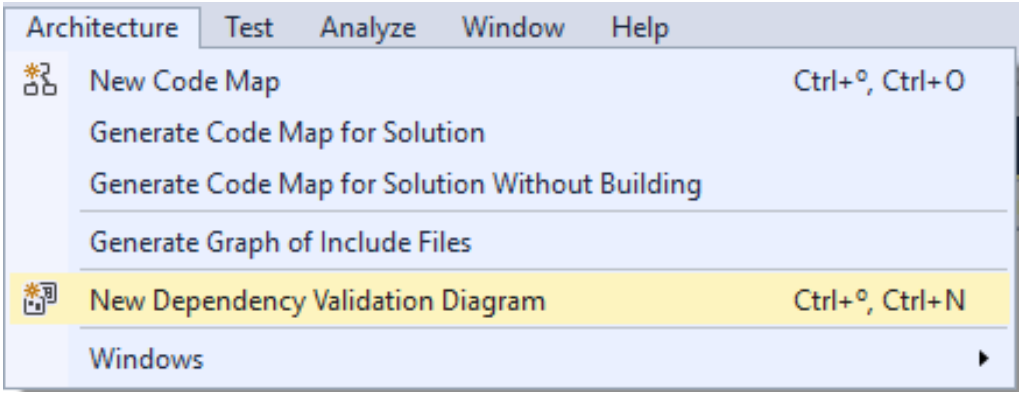

Autofac
Version:
Latest stable 4.3.0
Install
Options
Description
Autofac is an IoC container for Microsoft .NET. It manages the dependencies between classes so that applications stay easy to change as they grow in size and complexity.
Version: 4.3.0
Author(s): Autofac
License: http://www.opensource.org/licenses/mit-license.php
Date published: Tuesday, January 17, 2017 (1/17/2017)
Project URL: http://autofac.org/
Report Abuse: https://www.nuget.org/packages/Autofac/4.3.0/ReportAbuse
Tags: autofac, di, ioc, dependencyinjection
Dependencies
.NETFramework,Version=v4.5
No dependencies
.NETStandard,Version=v1.1
NETStandard.Library (>= 1.6.0)
System.ComponentModel (>= 4.0.1)

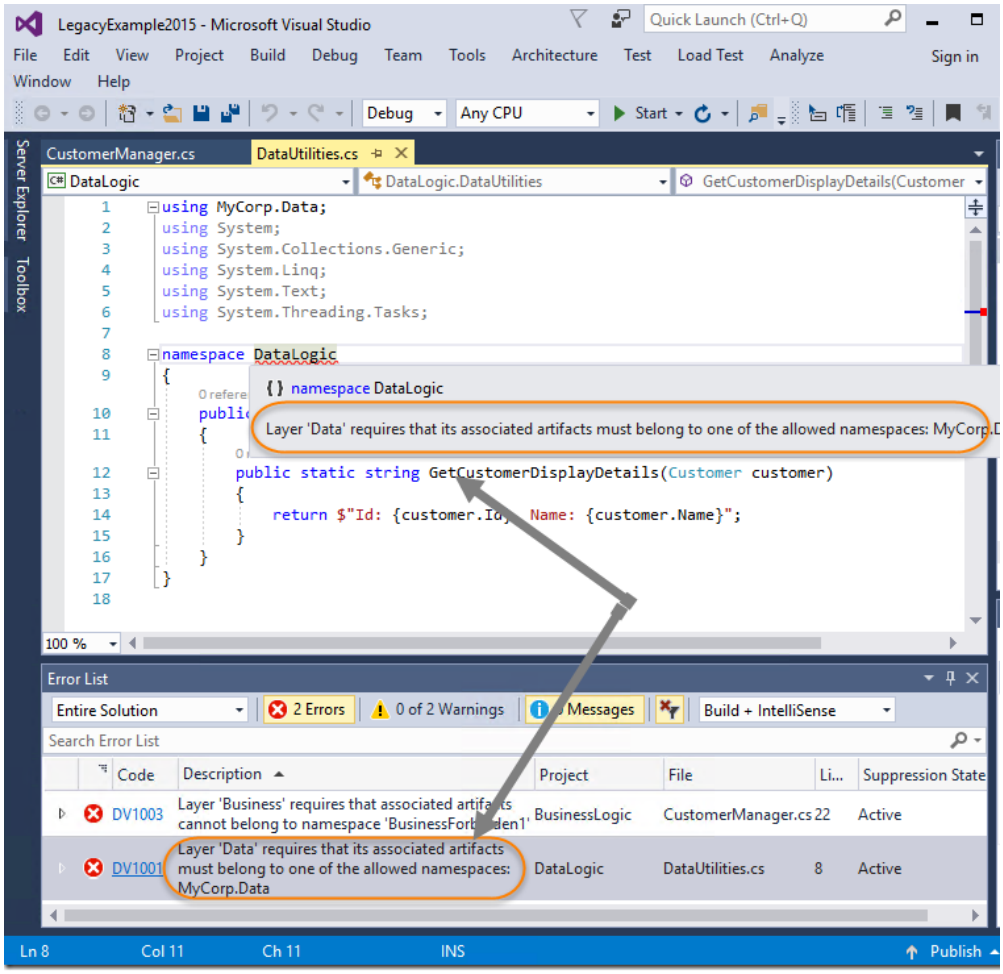
Demo Dependency Injection

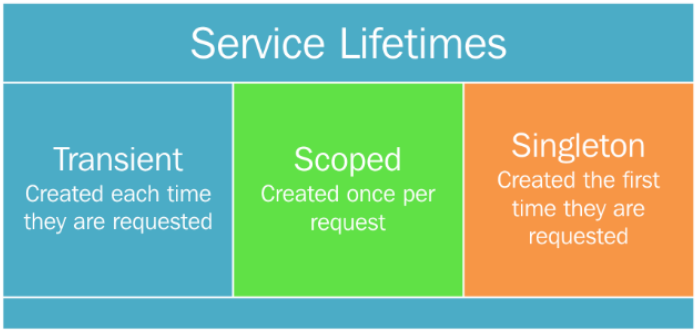
Current time: 6:38:52 PM

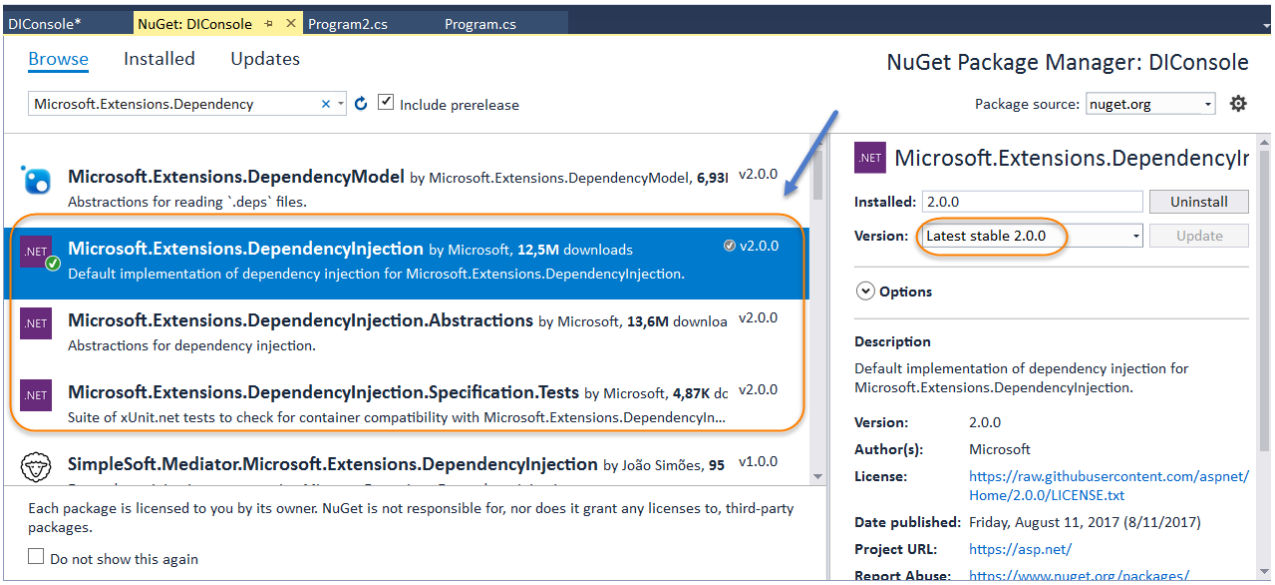
Chapter 3: Introducing Dependency Injection in .NET Core 2.0

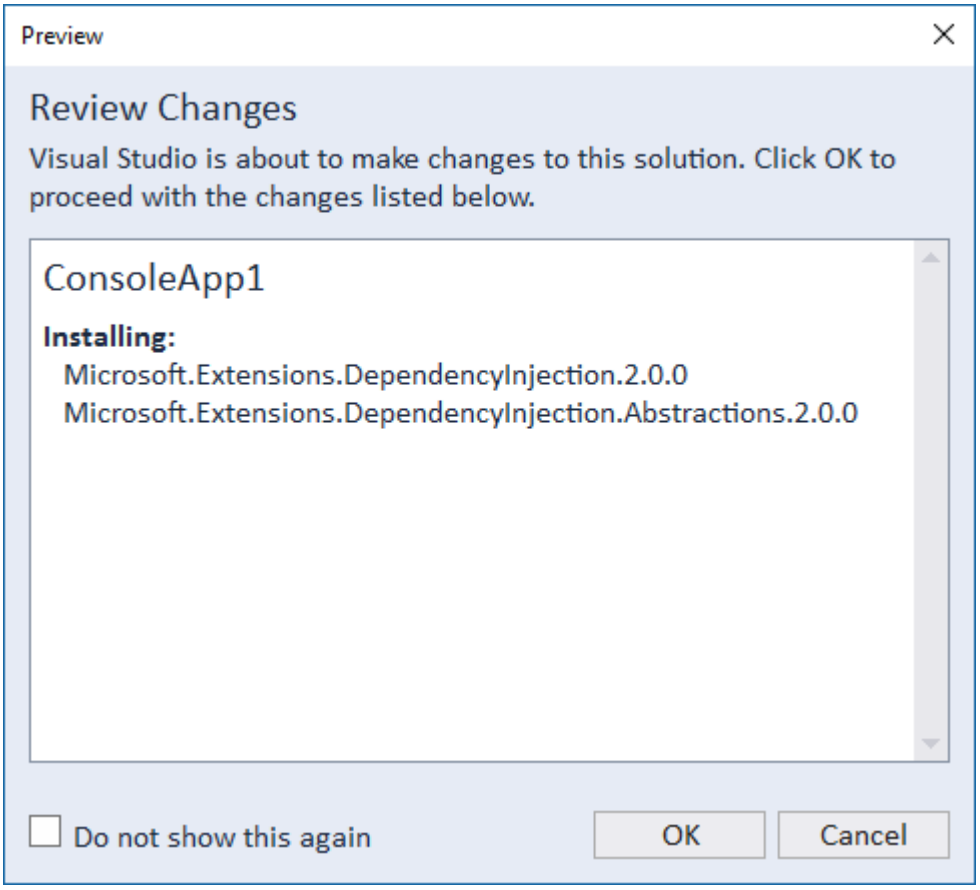





LegacyExample2015 - Microsoft Visual Studio
Quick Launch (Ctrl+Q)
File Edit View Project Build Debug Team Tools Architecture Test Load Test Analyze Sign in
Window Help
Debug
Any CPU
Start
CustomerManager.cs
DataUtilities.cs
DataLogic
DataLogic.DataUtilities
GetCustomerDisplayDetails(Customer
Server Explorer
Toolbox
using MyCorp.Data;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace DataLogic
{
namespace DataLogic
Layer 'Data' requires that its associated artifacts must belong to one of the allowed namespaces: MyCorp.D
public static string GetCustomerDisplayDetails(Customer customer)
{
return $"Id: {customer.Id}, Name: {customer.Name}";
}
}
}
100 %
Error List
Entire Solution
2 Errors
0 of 2 Warnings
Messages
Build + IntelliSense
Search Error List
Code
Description
Project
File
Li...
Suppression State
DV1003
Layer 'Business' requires that associated artifacts cannot belong to namespace 'BusinessForbidden1'
BusinessLogic
CustomerManager.cs
22
Active
DV1001
Layer 'Data' requires that its associated artifacts must belong to one of the allowed namespaces: MyCorp.Data
DataLogic
DataUtilities.cs
8
Active
Ln 8
Col 11
Ch 11
INS
Publish


Service Lifetimes
Transient
Created each time they are requested
Scoped
Created once per request
Singleton
Created the first time they are requested

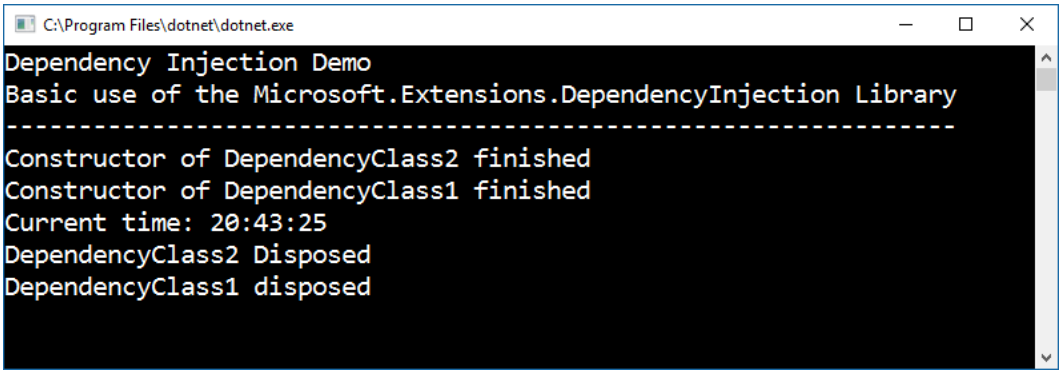

DIConsole*
NuGet: DIConsole
Program2.cs
Program.cs
Browse
Installed
Updates
NuGet Package Manager: DIConsole
Microsoft.Extensions.Dependency
Include prerelease
Package source: nuget.org
Microsoft.Extensions.DependencyModel by Microsoft.Extensions.DependencyModel, 6,93I v2.0.0
Abstractions for reading `.deps` files.
Microsoft.Extensions.DependencyInjection by Microsoft, 12,5M downloads v2.0.0
Default implementation of dependency injection for Microsoft.Extensions.DependencyInjection.
Microsoft.Extensions.DependencyInjection.Abstractions by Microsoft, 13,6M downloa v2.0.0
Abstractions for dependency injection.
Microsoft.Extensions.DependencyInjection.Specification.Tests by Microsoft, 4,87K dc v2.0.0
Suite of xUnit.net tests to check for container compatibility with Microsoft.Extensions.DependencyIn...
SimpleSoft.Mediator.Microsoft.Extensions.DependencyInjection by João Simões, 95 v1.0.0
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
Do not show this again
Microsoft.Extensions.DependencyIr
Installed: 2.0.0
Uninstall
Version: Latest stable 2.0.0
Update
Options
Description
Default implementation of dependency injection for Microsoft.Extensions.DependencyInjection.
Version: 2.0.0
Author(s): Microsoft
License: https://raw.githubusercontent.com/aspnet/Home/2.0.0/LICENSE.txt
Date published: Friday, August 11, 2017 (8/11/2017)
Project URL: https://asp.net/
Report Abuse: https://www.nuget.org/packages/

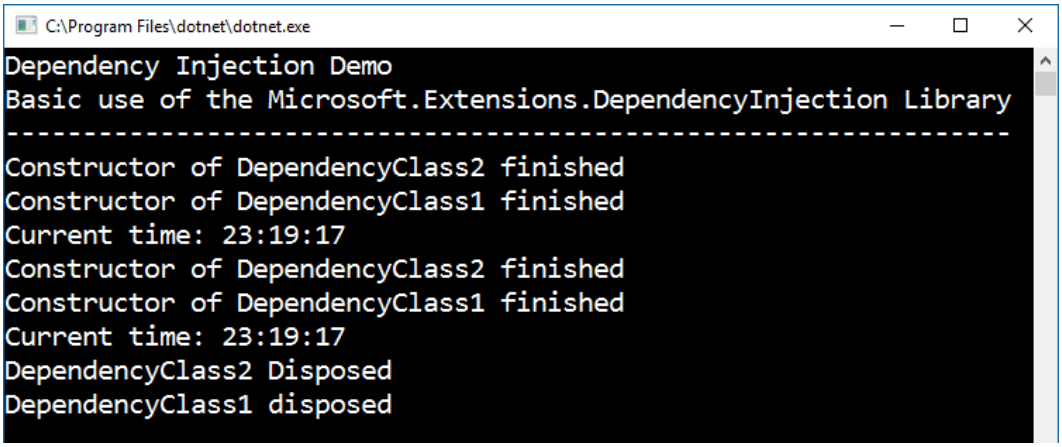


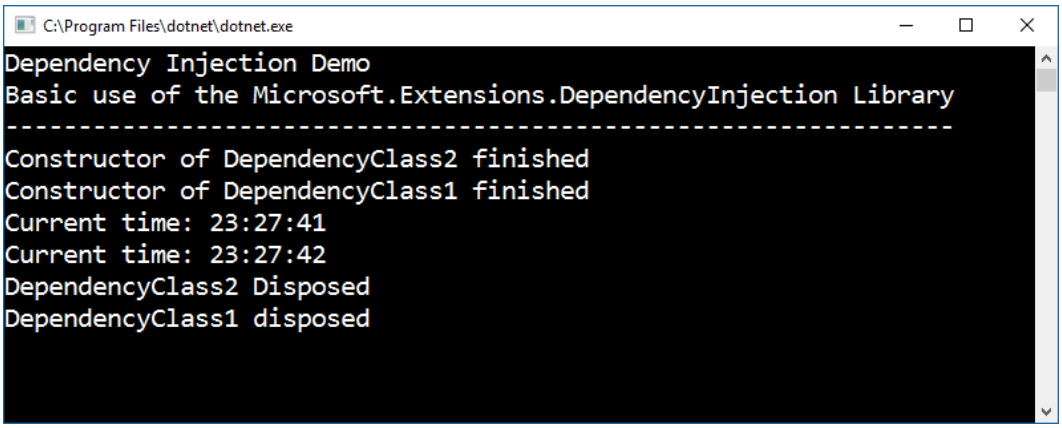

```
var services = new ServiceCollection();
services.AddTransient<DependencyClass2>();
services.
var provi                                    uildServiceProvider();
using (va                                    s1 = provider.GetService<DependencyClass1>())
{
    // Me                                    g DependencyClass1
```

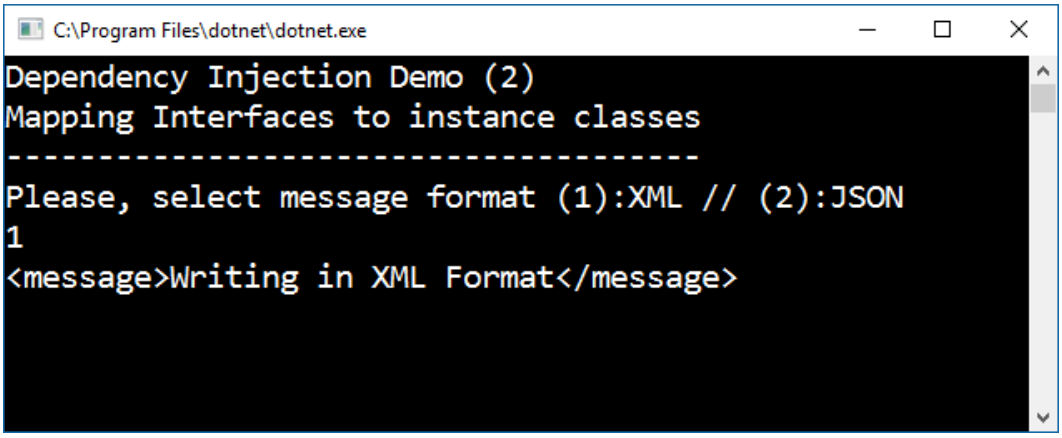
AddScoped
AddScoped<>
AddSingleton
AddSingleton<>
AddTransient
AddTransient<>
BuildServiceProvider

(extension) IServiceCollection IServiceCollection.AddScoped(Type serviceType, Type implementatio
Adds a scoped service of the type specified in serviceType with an implementation of the type speci

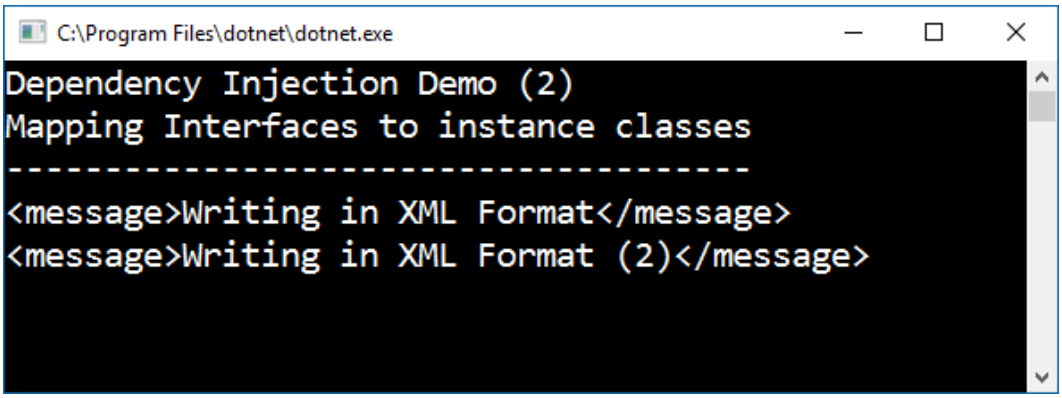

C:\Program Files\dotnet\dotnet.exe
Dependency Injection Demo
Basic use of the Microsoft.Extensions.DependencyInjection Library
----
Constructor of DependencyClass2 finished
Constructor of DependencyClass1 finished
Current time: 20:43:25
DependencyClass2 Disposed
DependencyClass1 disposed


C:\Program Files\dotnet\dotnet.exe
Dependency Injection Demo
Basic use of the Microsoft.Extensions.DependencyInjection Library
----
Constructor of DependencyClass2 finished
Constructor of DependencyClass1 finished
Current time: 23:19:17
Constructor of DependencyClass2 finished
Constructor of DependencyClass1 finished
Current time: 23:19:17
DependencyClass2 Disposed
DependencyClass1 disposed

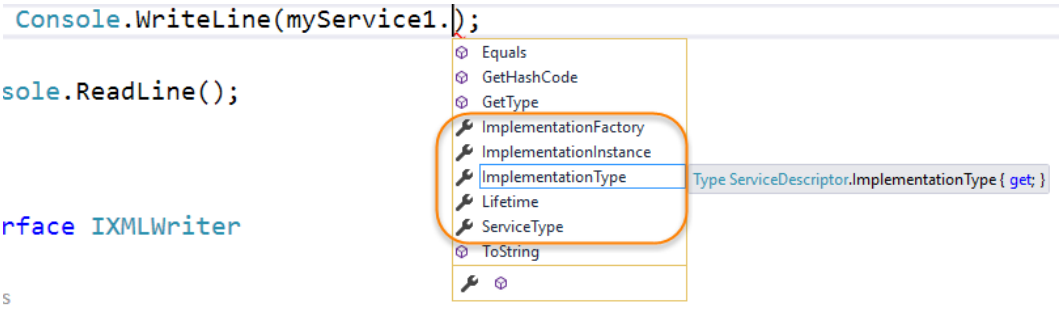

C:\Program Files\dotnet\dotnet.exe
Dependency Injection Demo
Basic use of the Microsoft.Extensions.DependencyInjection Library
----
Constructor of DependencyClass2 finished
Constructor of DependencyClass1 finished
Current time: 23:27:41
Current time: 23:27:42
DependencyClass2 Disposed
DependencyClass1 disposed

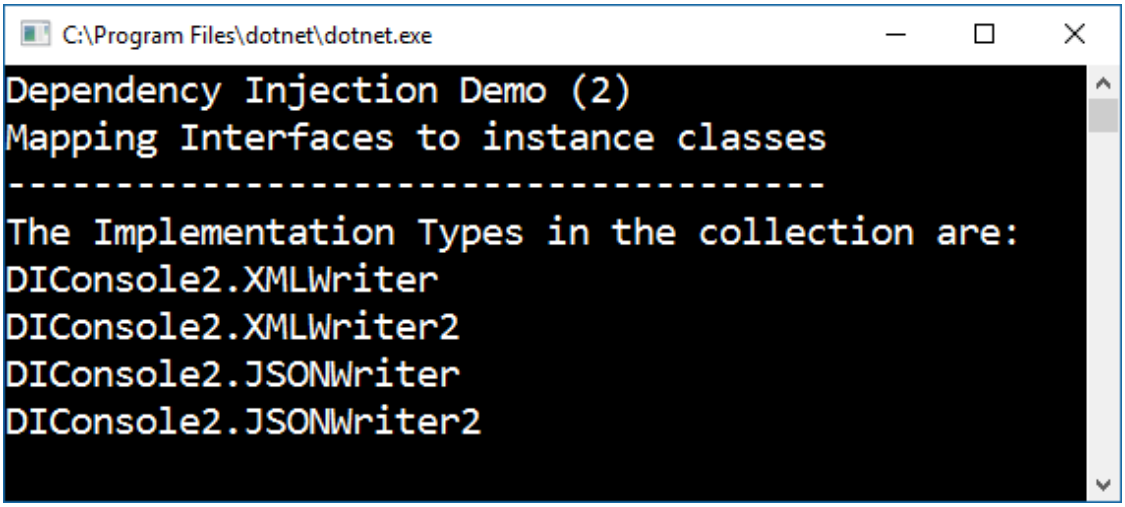

C:\Program Files\dotnet\dotnet.exe
Dependency Injection Demo (2)
Mapping Interfaces to instance classes

Please, select message format (1):XML // (2):JSON
1
<message>Writing in XML Format</message>

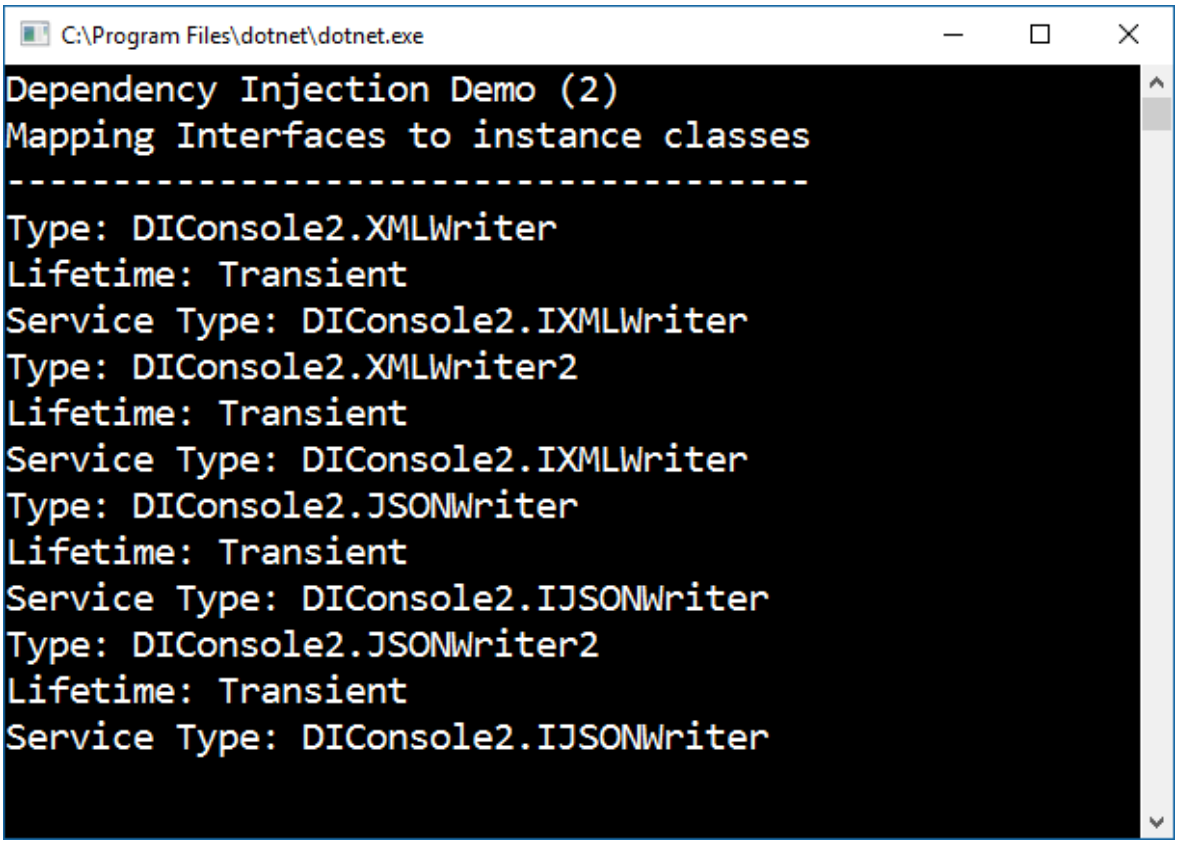

C:\Program Files\dotnet\dotnet.exe
Dependency Injection Demo (2)
Mapping Interfaces to instance classes

<message>Writing in XML Format</message>
<message>Writing in XML Format (2)</message>

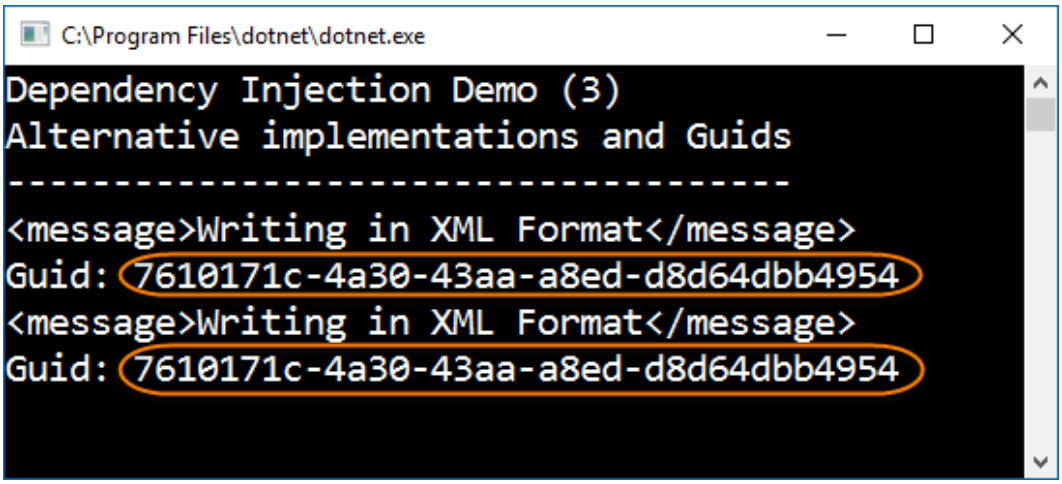

Console.WriteLine(myService1.);
sole.ReadLine();
rface IXMLWriter
s
Equals
GetHashCode
GetType
ImplementationFactory
ImplementationInstance
ImplementationType
Lifetime
ServiceType
ToString
Type ServiceDescriptor.ImplementationType { get; }

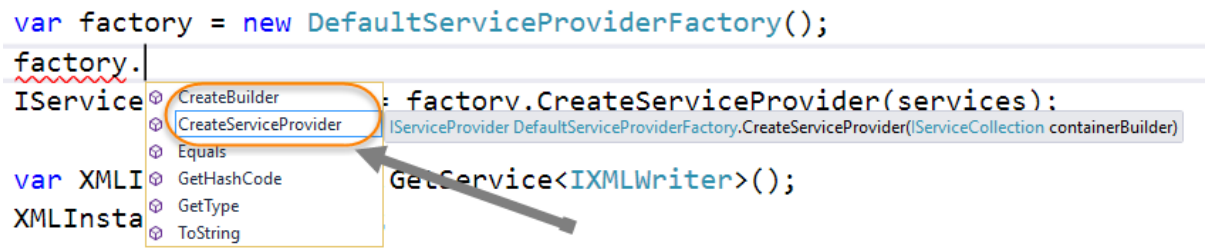

C:\Program Files\dotnet\dotnet.exe
Dependency Injection Demo (2)
Mapping Interfaces to instance classes

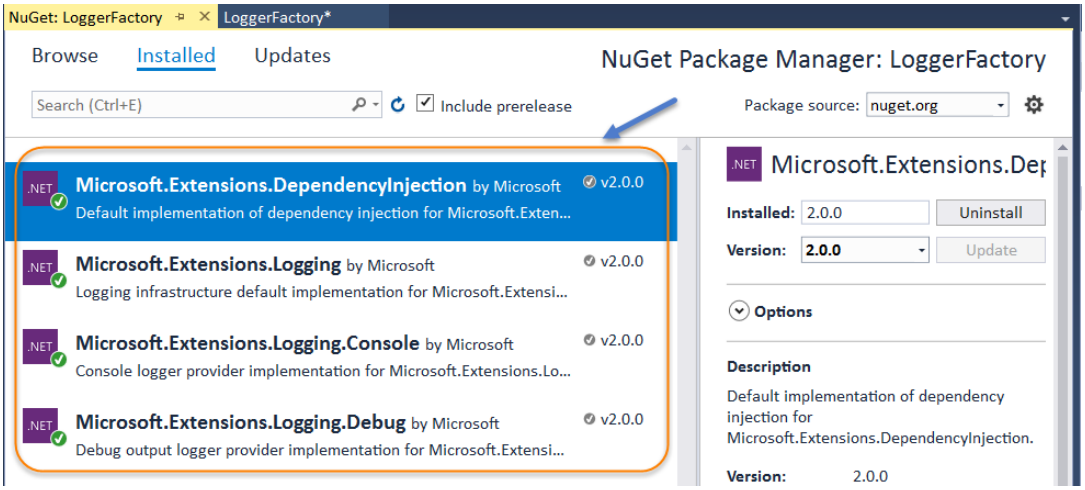
The Implementation Types in the collection are:
DIConsole2.XMLWriter
DIConsole2.XMLWriter2
DIConsole2.JSONWriter
DIConsole2.JSONWriter2

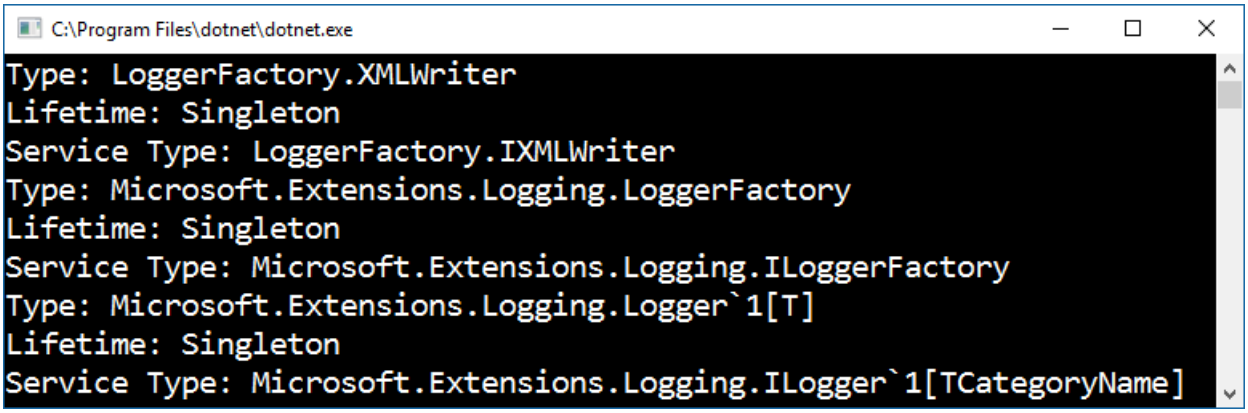

C:\Program Files\dotnet\dotnet.exe
Dependency Injection Demo (2)
Mapping Interfaces to instance classes

Type: DIConsole2.XMLWriter
Lifetime: Transient
Service Type: DIConsole2.IXMLWriter
Type: DIConsole2.XMLWriter2
Lifetime: Transient
Service Type: DIConsole2.IXMLWriter
Type: DIConsole2.JSONWriter
Lifetime: Transient
Service Type: DIConsole2.IJSONWriter
Type: DIConsole2.JSONWriter2
Lifetime: Transient
Service Type: DIConsole2.IJSONWriter

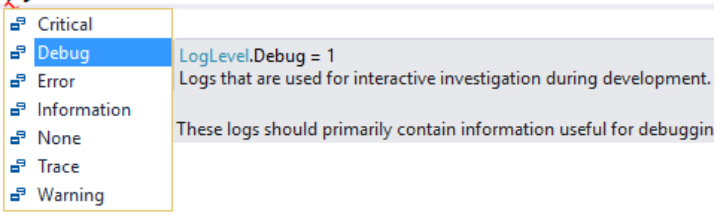

C:\Program Files\dotnet\dotnet.exe
Dependency Injection Demo (3)
Alternative implementations and Guids
<message>Writing in XML Format</message>
Guid: 7610171c-4a30-43aa-a8ed-d8d64dbb4954
<message>Writing in XML Format</message>
Guid: 7610171c-4a30-43aa-a8ed-d8d64dbb4954

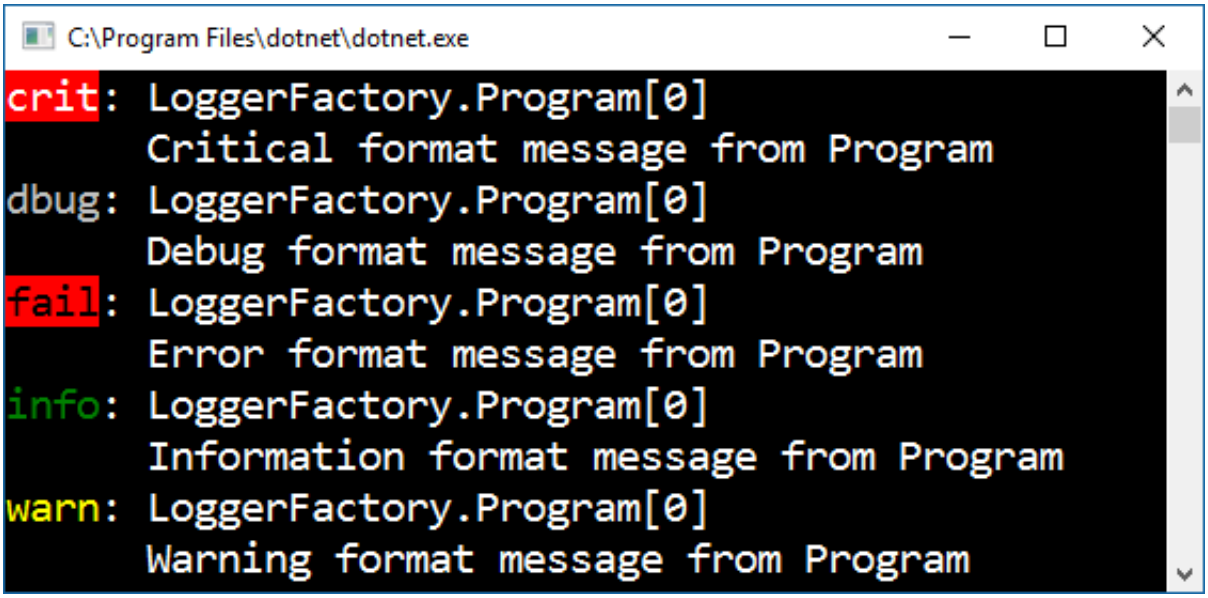

var factory = new DefaultServiceProviderFactory();
factory.
IService
factory.CreateServiceProvider(services);
var XMLI
GetService<IXMLWriter>();
XMLInsta
CreateBuilder
CreateServiceProvider
Equals
GetHashCode
GetType
ToString
IServiceProvider DefaultServiceProviderFactory.CreateServiceProvider(IServiceCollection containerBuilder)

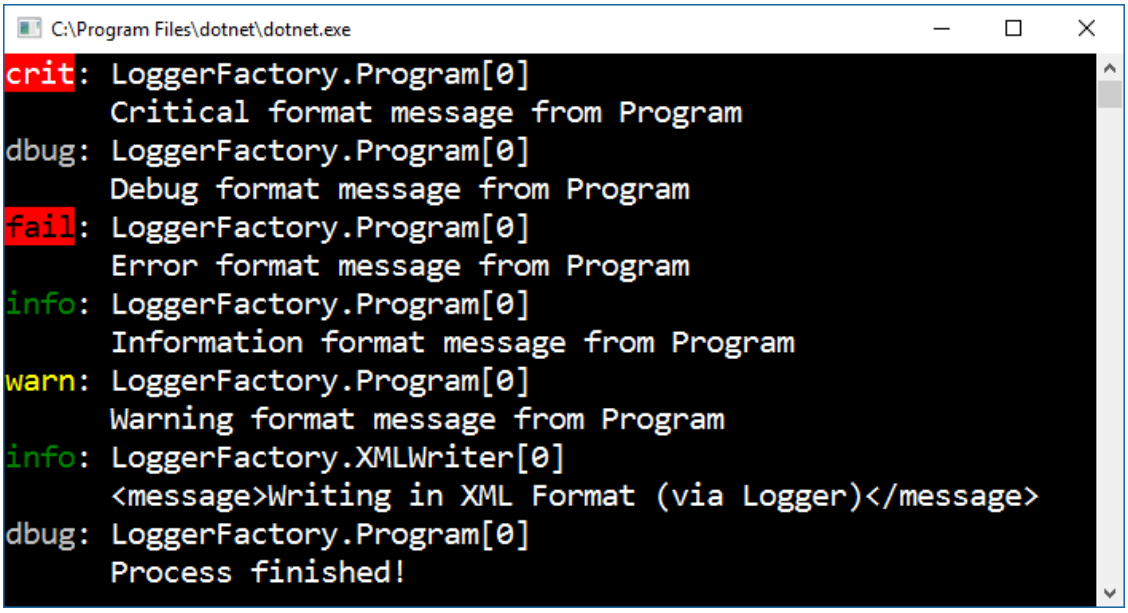

NuGet: LoggerFactory
LoggerFactory*
Browse
Installed
Updates
NuGet Package Manager: LoggerFactory
Search (Ctrl+E)
Include prerelease
Package source: nuget.org
Microsoft.Extensions.DependencyInjection by Microsoft
Default implementation of dependency injection for Microsoft.Exten...
v2.0.0
Microsoft.Extensions.Logging by Microsoft
Logging infrastructure default implementation for Microsoft.Extensi...
v2.0.0
Microsoft.Extensions.Logging.Console by Microsoft
Console logger provider implementation for Microsoft.Extensions.Lo...
v2.0.0
Microsoft.Extensions.Logging.Debug by Microsoft
Debug output logger provider implementation for Microsoft.Extensi...
v2.0.0
Microsoft.Extensions.Dep
Installed: 2.0.0
Uninstall
Version: 2.0.0
Update
Options
Description
Default implementation of dependency injection for Microsoft.Extensions.DependencyInjection.
Version: 2.0.0

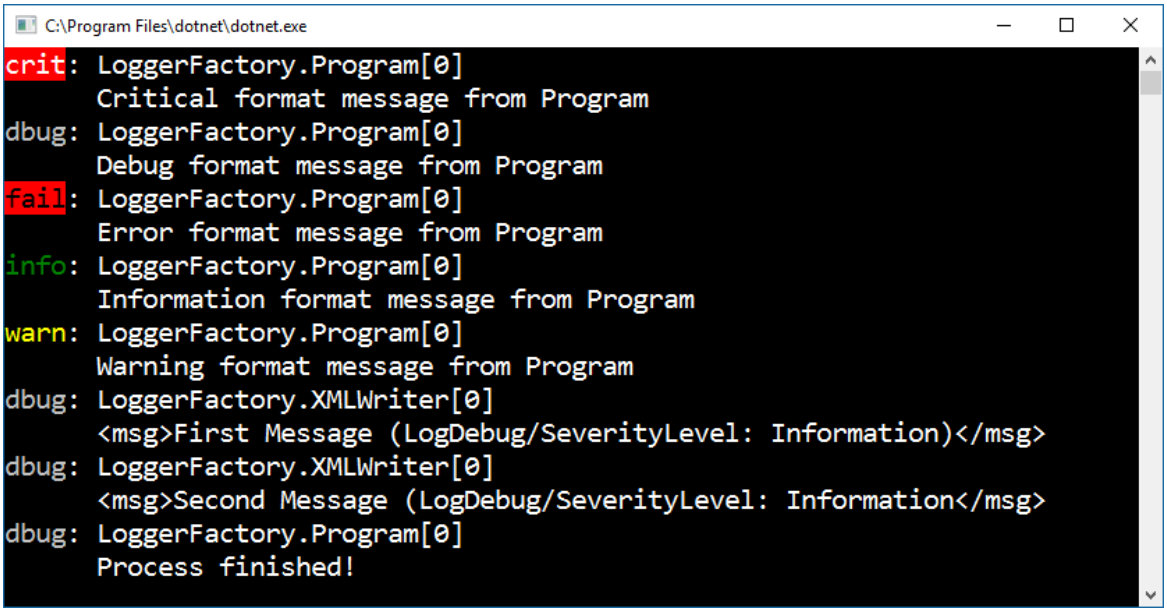

C:\Program Files\dotnet\dotnet.exe
Type: LoggerFactory.XMLWriter
Lifetime: Singleton
Service Type: LoggerFactory.IXMLWriter
Type: Microsoft.Extensions.Logging.LoggerFactory
Lifetime: Singleton
Service Type: Microsoft.Extensions.Logging.ILoggerFactory
Type: Microsoft.Extensions.Logging.Logger`1[T]
Lifetime: Singleton
Service Type: Microsoft.Extensions.Logging.ILogger`1[TCategoryName]

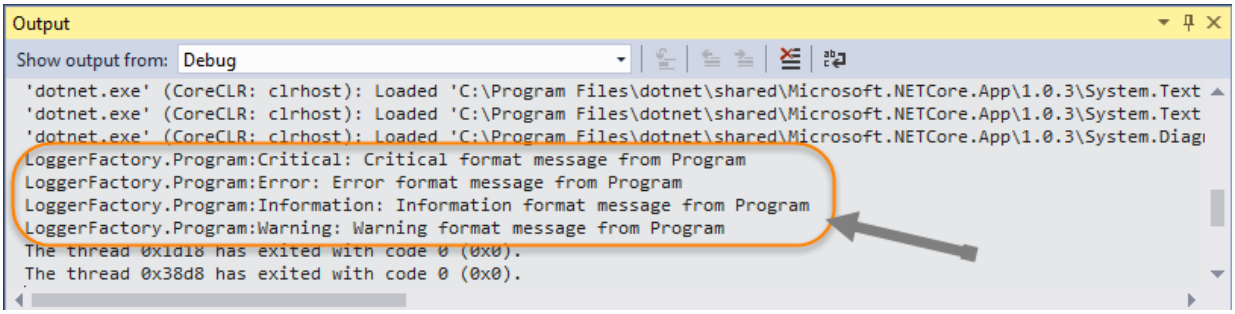
```
//Obtain service and configure logging
serviceProvider.GetService<ILoggerFactory>()
    .AddConsole(LogLevel.);
```



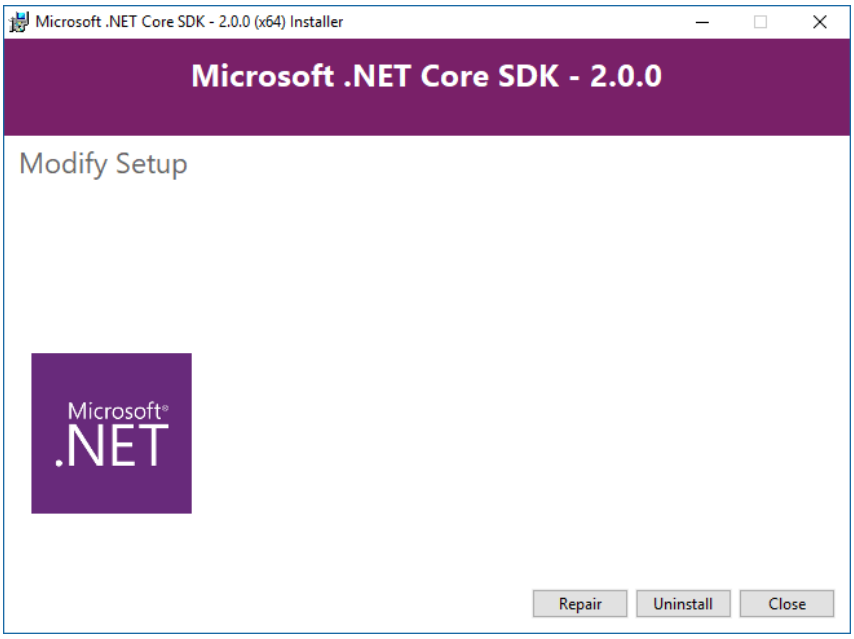


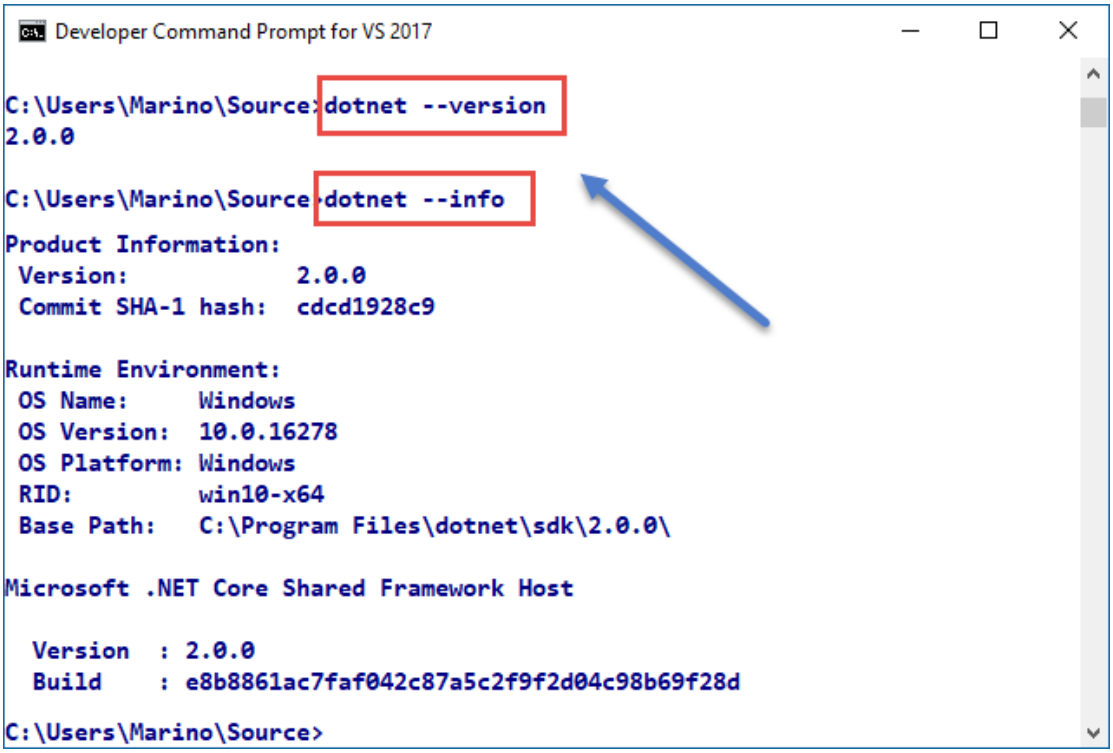

C:\Program Files\dotnet\dotnet.exe
crit: LoggerFactory.Program[0]
Critical format message from Program
dbug: LoggerFactory.Program[0]
Debug format message from Program
fail: LoggerFactory.Program[0]
Error format message from Program
info: LoggerFactory.Program[0]
Information format message from Program
warn: LoggerFactory.Program[0]
Warning format message from Program
info: LoggerFactory.XMLWriter[0]
<message>Writing in XML Format (via Logger)</message>
dbug: LoggerFactory.Program[0]
Process finished!


C:\Program Files\dotnet\dotnet.exe
crit: LoggerFactory.Program[0]
Critical format message from Program
dbug: LoggerFactory.Program[0]
Debug format message from Program
fail: LoggerFactory.Program[0]
Error format message from Program
info: LoggerFactory.Program[0]
Information format message from Program
warn: LoggerFactory.Program[0]
Warning format message from Program
dbug: LoggerFactory.XMLWriter[0]
<msg>First Message (LogDebug/SeverityLevel: Information)</msg>
dbug: LoggerFactory.XMLWriter[0]
<msg>Second Message (LogDebug/SeverityLevel: Information</msg>
dbug: LoggerFactory.Program[0]
Process finished!

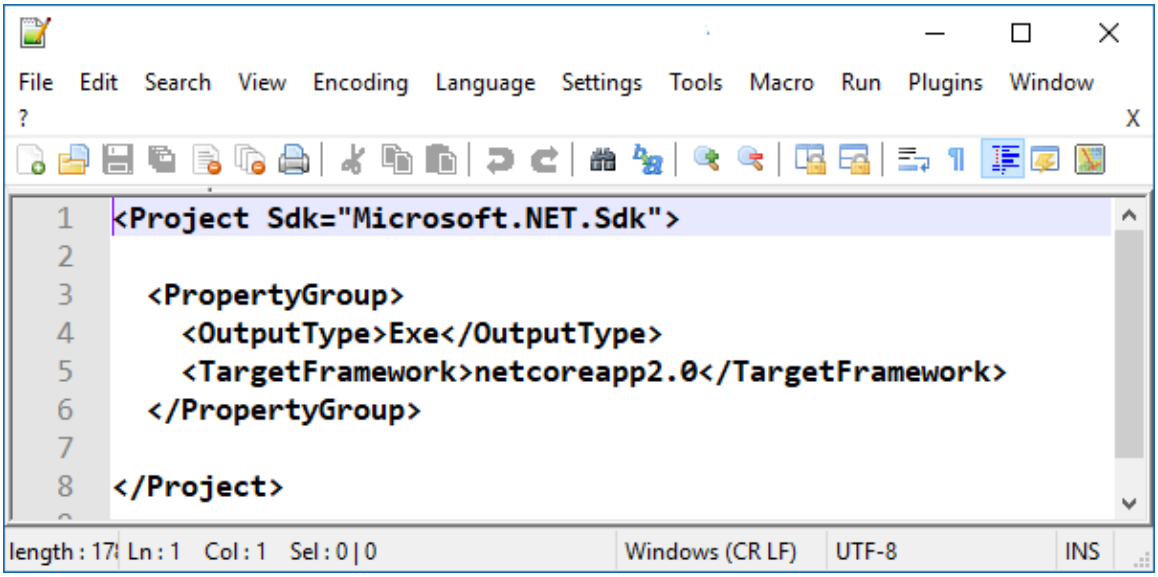

Output
Show output from: Debug
'dotnet.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\1.0.3\System.Text
'dotnet.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\1.0.3\System.Text
'dotnet.exe' (CoreCLR: clrhost): Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\1.0.3\System.Diag
LoggerFactory.Program:Critical: Critical format message from Program
LoggerFactory.Program:Error: Error format message from Program
LoggerFactory.Program:Information: Information format message from Program
LoggerFactory.Program:Warning: Warning format message from Program
The thread 0x1d18 has exited with code 0 (0x0).
The thread 0x38d8 has exited with code 0 (0x0).

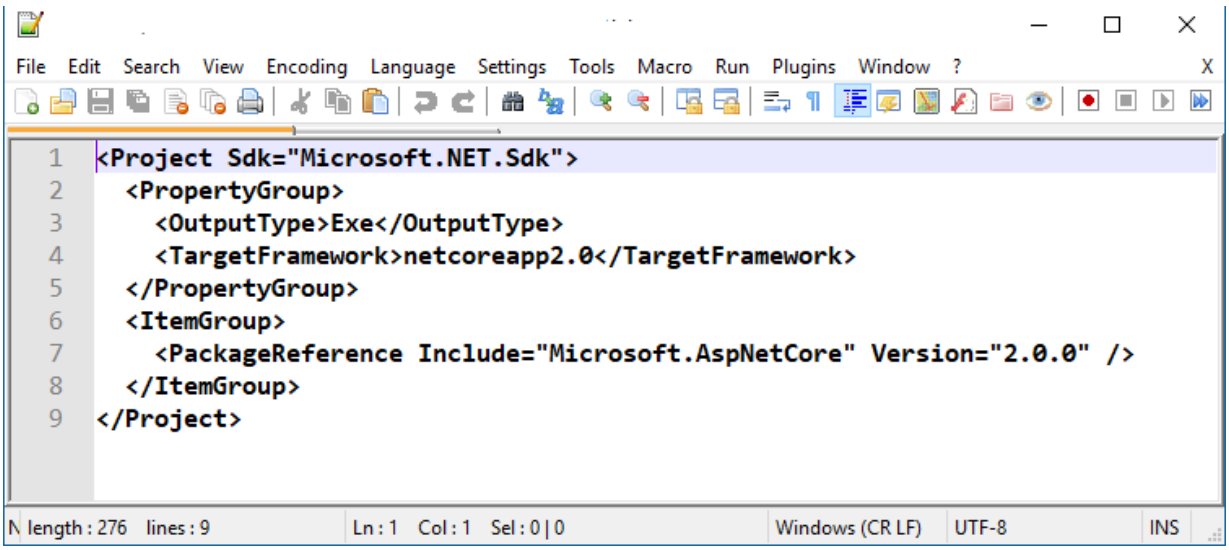
Chapter 4: Dependency Injection in ASP.NET Core

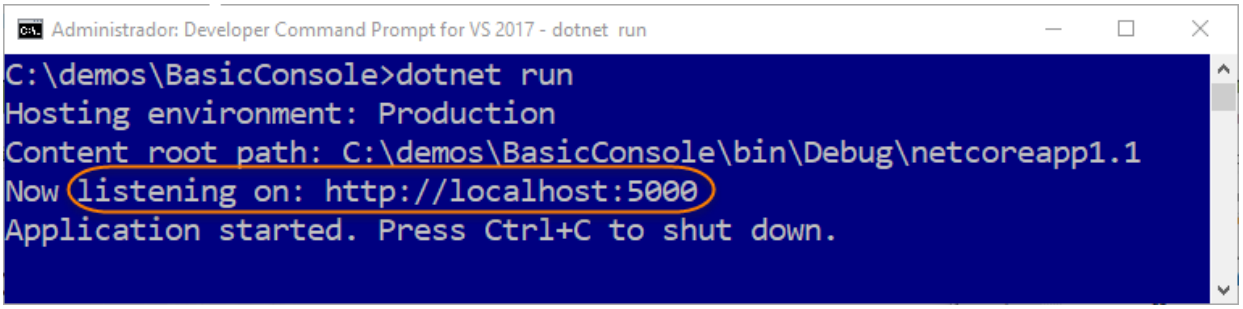


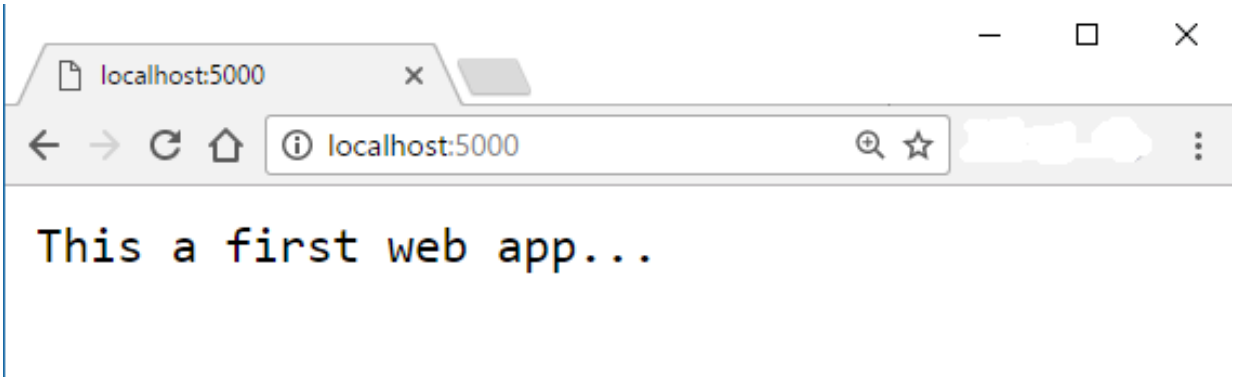


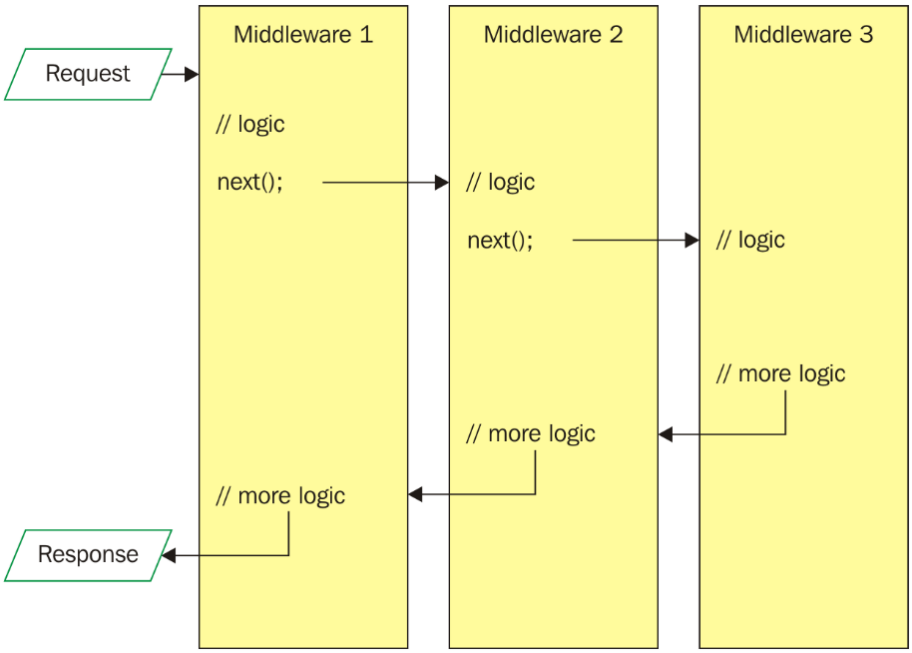
Console Application	console	[C#], F#, VB	Common/Console
Class library	classlib	[C#], F#, VB	Common/Library
Unit Test Project	mstest	[C#], F#, VB	Test/MSTest
xUnit Test Project	xunit	[C#], F#, VB	Test/xUnit
ASP.NET Core Empty	web	[C#], F#	Web/Empty
ASP.NET Core Web App (Model-View-Controller)	mvc	[C#], F#	Web/MVC
ASP.NET Core Web App	razor	[C#]	Web/MVC/Razor Pages
ASP.NET Core with Angular	angular	[C#]	Web/MVC/SPA
ASP.NET Core with React.js	react	[C#]	Web/MVC/SPA
ASP.NET Core with React.js and Redux	reactredux	[C#]	Web/MVC/SPA
ASP.NET Core Web API	webapi	[C#], F#	Web/WebAPI
global.json file	globaljson		Config
Nuget Config	nugetconfig		Config
Web Config	webconfig		Config
Solution File	sln		Solution
Razor Page	page		Web/ASP.NET
MVC ViewImports	viewimports		Web/ASP.NET
MVC ViewStart	viewstart		Web/ASP.NET

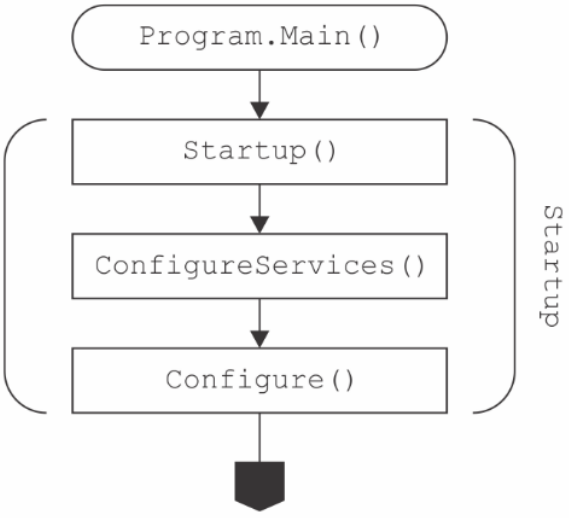

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window
?
X
1 <Project Sdk="Microsoft.NET.Sdk">
2
3 <PropertyGroup>
4 <OutputType>Exe</OutputType>
5 <TargetFramework>netcoreapp2.0</TargetFramework>
6 </PropertyGroup>
7
8 </Project>
length : 17 Ln : 1 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS


File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
X
1 <Project Sdk="Microsoft.NET.Sdk">
2 <PropertyGroup>
3 <OutputType>Exe</OutputType>
4 <TargetFramework>netcoreapp2.0</TargetFramework>
5 </PropertyGroup>
6 <ItemGroup>
7 <PackageReference Include="Microsoft.AspNetCore" Version="2.0.0" />
8 </ItemGroup>
9 </Project>
length : 276 lines : 9
Ln : 1 Col : 1 Sel : 0 | 0
Windows (CR LF)
UTF-8
INS


Administrador: Developer Command Prompt for VS 2017 - dotnet run
C:\demos\BasicConsole>dotnet run
Hosting environment: Production
Content root path: C:\demos\BasicConsole\bin\Debug\netcoreapp1.1
Now listening on: http://localhost:5000
Application started. Press Ctrl+C to shut down.


localhost:5000
localhost:5000
This a first web app...


Middleware 1
Middleware 2
Middleware 3
Request
// logic
next();
// logic
next();
// logic
// more logic
// more logic
// more logic
Response

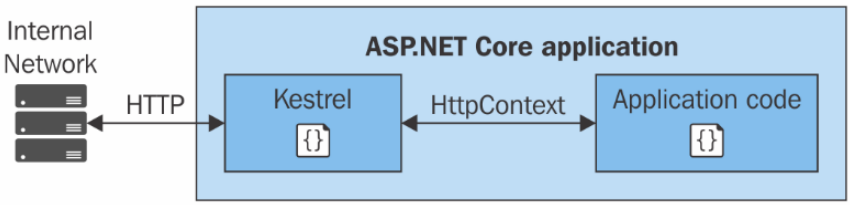

Program.Main()
Startup()
ConfigureServices()
Configure()
Startup

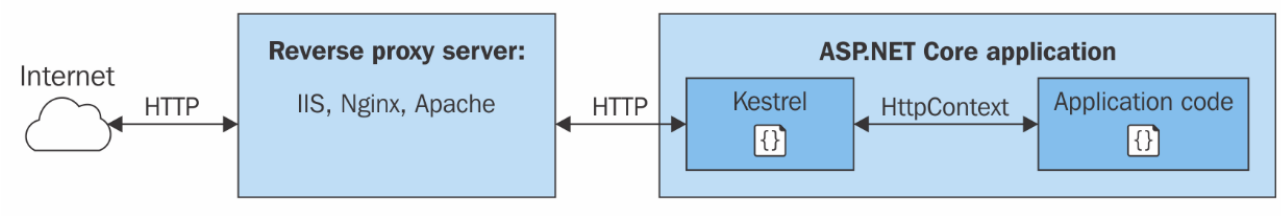
```
namespace Microsoft.AspNetCore.Builder
{
    public interface IApplicationBuilder
    {
        IServiceProvider ApplicationServices { get; set; }
        IFeatureCollection ServerFeatures { get; }
        IDictionary<string, object> Properties { get; }

        RequestDelegate Build();
        IApplicationBuilder New();
        IApplicationBuilder Use(Func<RequestDelegate, RequestDelegate> middleware);
    }
}
```

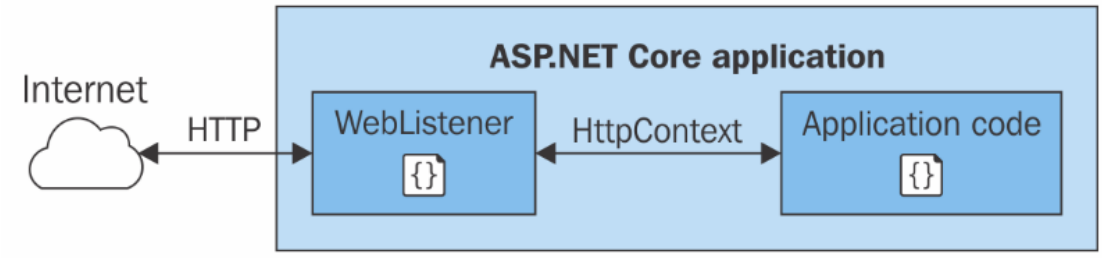
```
app.
```

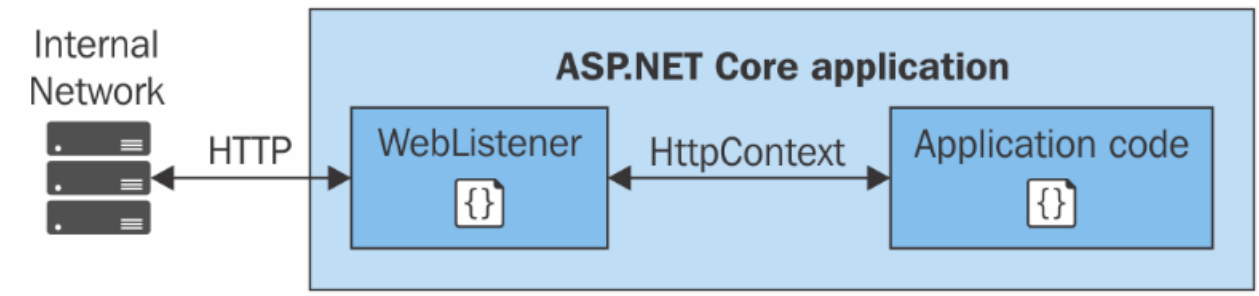
- UseDeveloperExceptionPage
- UseExceptionHandler
- UseForwardedHeaders
- UseHttpMethodOverride
- UseMiddleware
- UseMiddleware<>
- UsePathBase
- UseRouter
- UseStatusCodePages

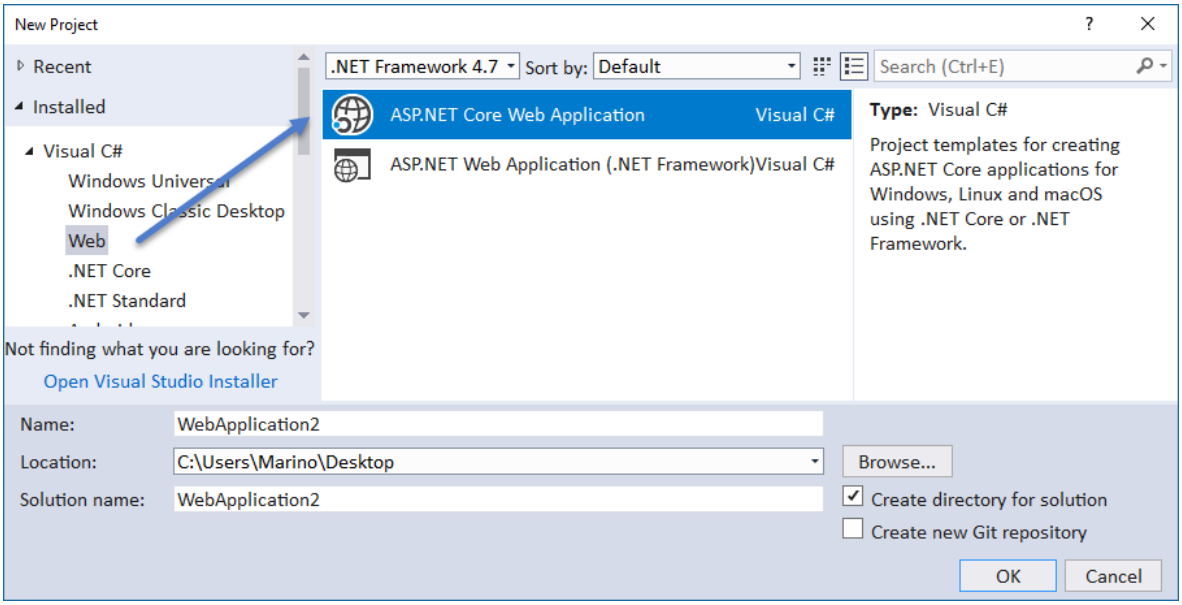
(extension) IApplicationBuilder IApplicationBuilder.Map(
Branches the request pipeline based on matches of the

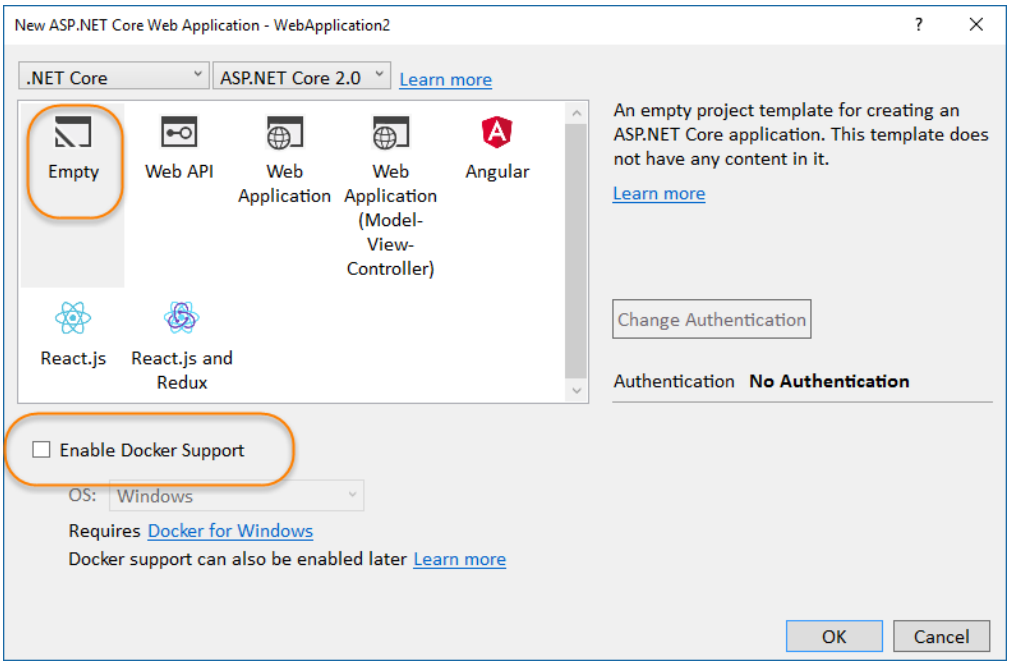


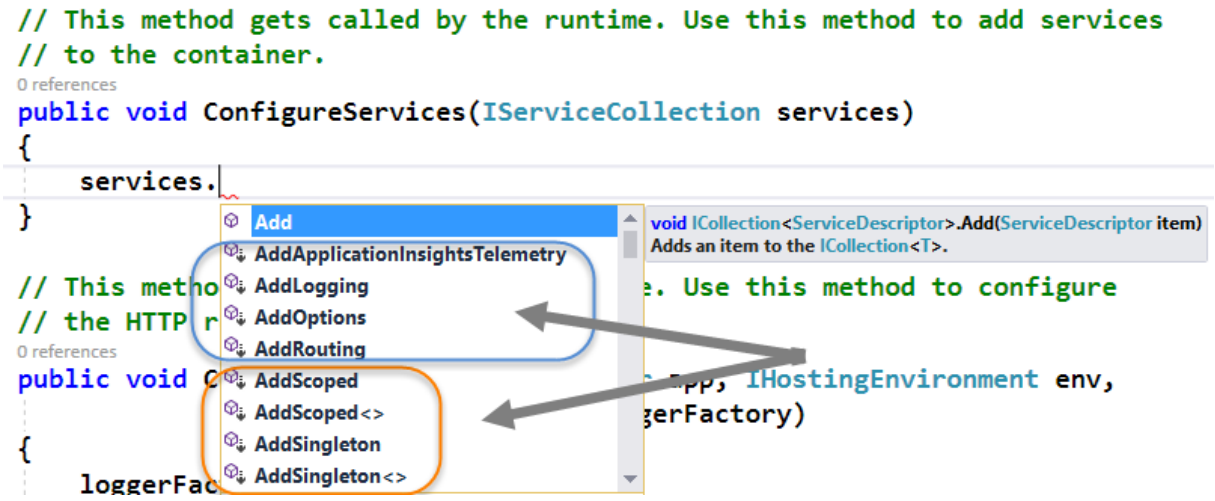

Internet
HTTP
Reverse proxy server:
IIS, Nginx, Apache
HTTP
ASP.NET Core application
Kestrel
{}
HttpContext
Application code
{}

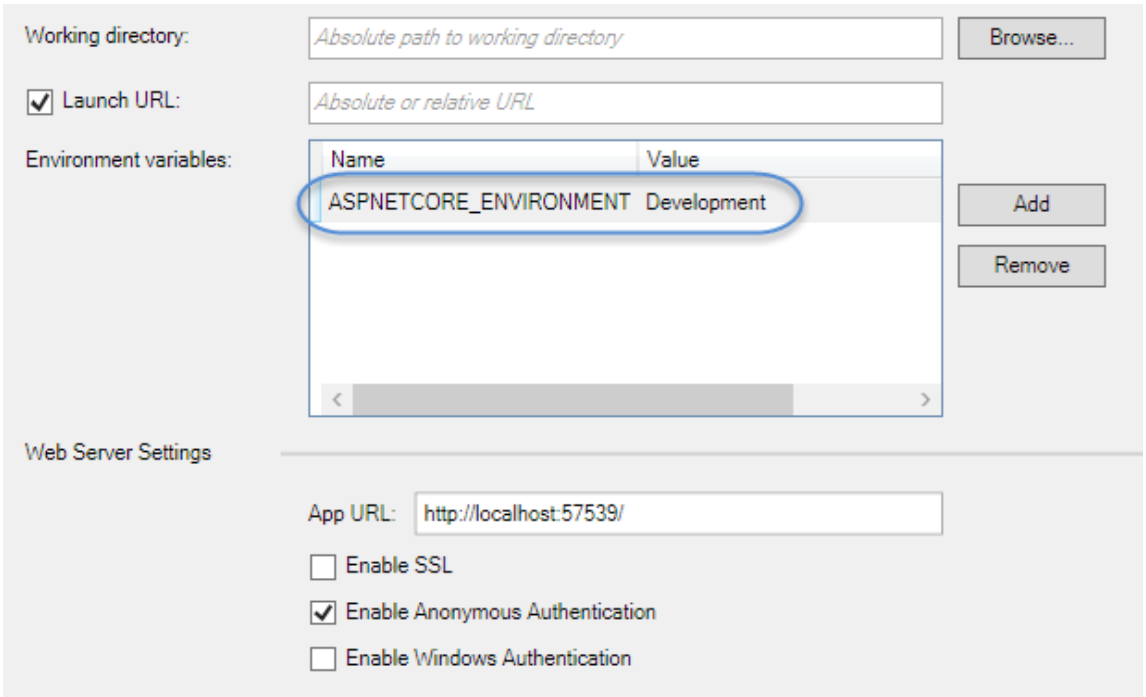

Internet
HTTP
ASP.NET Core application
WebListener
{}
HttpContext
Application code
{}

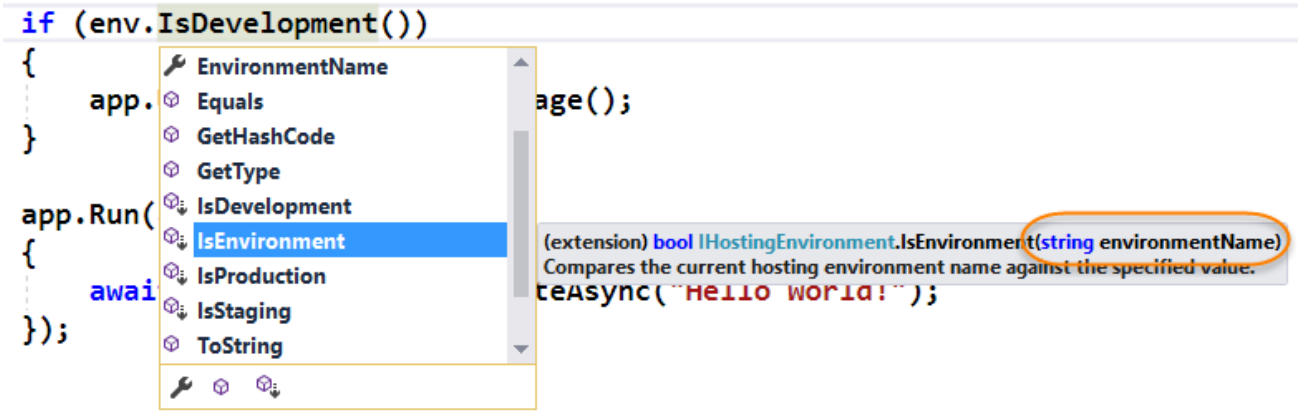

Internal
Network
HTTP
ASP.NET Core application
WebListener
{}
HttpContext
Application code
{}


New Project
Recent
Installed
Visual C#
Windows Universal
Windows Classic Desktop
Web
.NET Core
.NET Standard
Not finding what you are looking for?
Open Visual Studio Installer
.NET Framework 4.7
Sort by: Default
Search (Ctrl+E)
ASP.NET Core Web Application
Visual C#
ASP.NET Web Application (.NET Framework)Visual C#
Type: Visual C#
Project templates for creating ASP.NET Core applications for Windows, Linux and macOS using .NET Core or .NET Framework.
Name: WebApplication2
Location: C:\Users\Marino\Desktop
Browse...
Solution name: WebApplication2
Create directory for solution
Create new Git repository
OK
Cancel

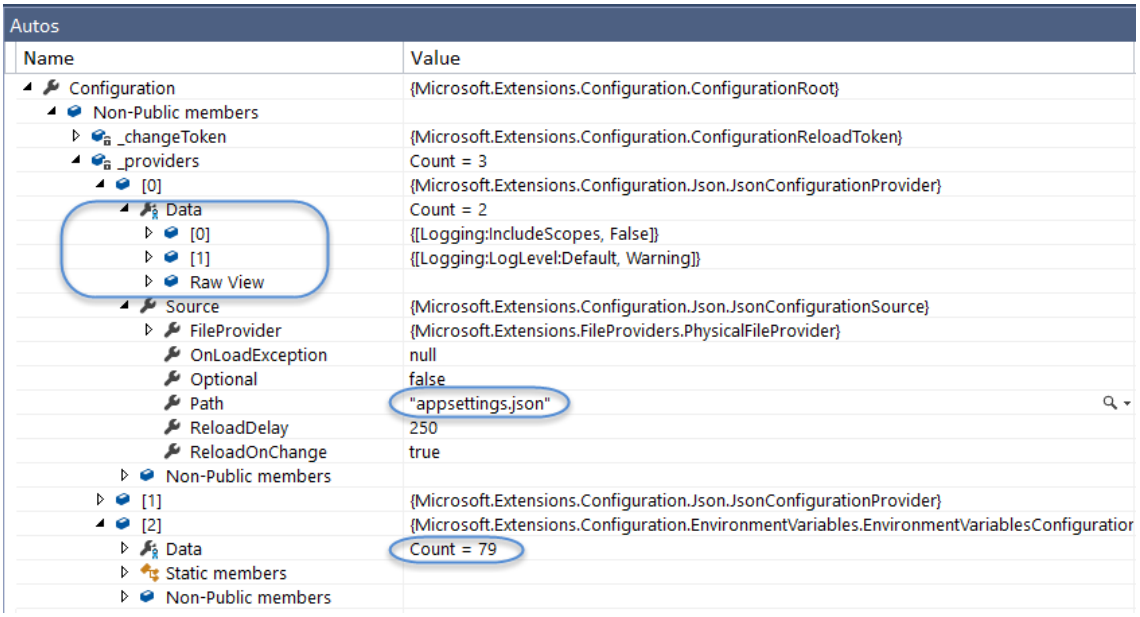

New ASP.NET Core Web Application - WebApplication2
.NET Core
ASP.NET Core 2.0
Learn more
Empty
Web API
Web Application
Web Application (Model-View-Controller)
Angular
React.js
React.js and Redux
An empty project template for creating an ASP.NET Core application. This template does not have any content in it.
Learn more
Change Authentication
Authentication No Authentication
Enable Docker Support
OS: Windows
Requires Docker for Windows
Docker support can also be enabled later Learn more
OK
Cancel

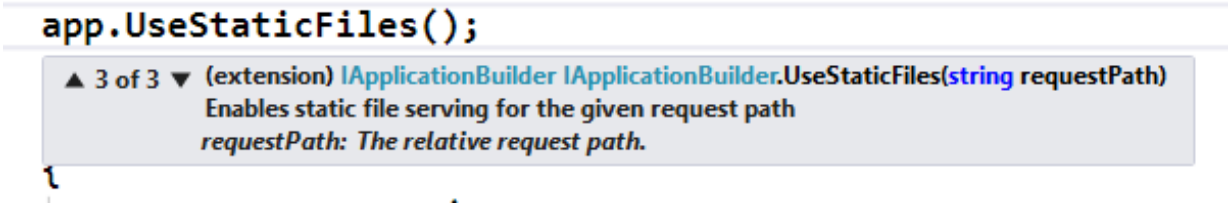

// This method gets called by the runtime. Use this method to add services
// to the container.
0 references
public void ConfigureServices(IServiceCollection services)
{
services.
}
Add
AddApplicationInsightsTelemetry
AddLogging
AddOptions
AddRouting
AddScoped
AddScoped<>
AddSingleton
AddSingleton<>
void ICollection<ServiceDescriptor>.Add(ServiceDescriptor item)
Adds an item to the ICollection<T>.
// This metho
// the HTTP r
0 references
public void C
{
loggerFac
e. Use this method to configure
pp, IHostingEnvironment env,
gerFactory)

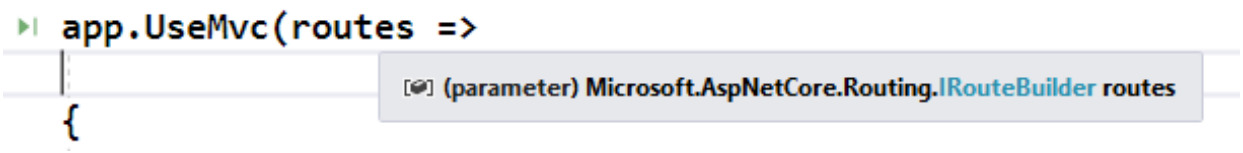

Working directory:
Absolute path to working directory
Browse...
Launch URL:
Absolute or relative URL
Environment variables:
Name
Value
ASPNETCORE_ENVIRONMENT
Development
Add
Remove
Web Server Settings
App URL:
http://localhost:57539/
Enable SSL
Enable Anonymous Authentication
Enable Windows Authentication


if (env.IsDevelopment())
{
app.
}
app.Run(
{
awai
});
EnvironmentName
Equals
GetHashCode
GetType
IsDevelopment
IsEnvironment
IsProduction
IsStaging
ToString
(extension) bool IHostingEnvironment.IsEnvironment(string environmentName)
Compares the current hosting environment name against the specified value.

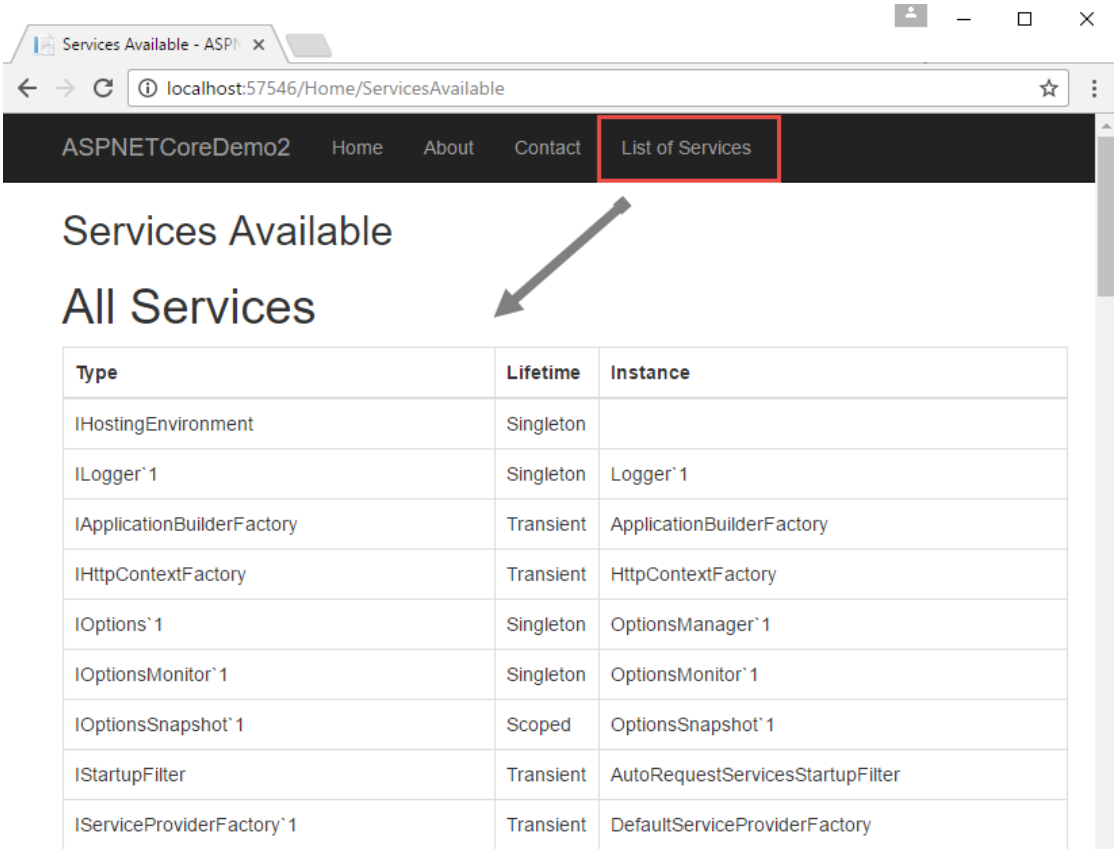
localhost:57539
localhost:57539
We're in PACKT development mode!


Autos
Name
Value
Configuration
{Microsoft.Extensions.Configuration.ConfigurationRoot}
Non-Public members
_changeToken
{Microsoft.Extensions.Configuration.ConfigurationReloadToken}
_providers
Count = 3
[0]
{Microsoft.Extensions.Configuration.Json.JsonConfigurationProvider}
Data
Count = 2
[0]
{[Logging:IncludeScopes, False]}
[1]
{[Logging:LogLevel:Default, Warning]}
Raw View
Source
{Microsoft.Extensions.Configuration.Json.JsonConfigurationSource}
FileProvider
{Microsoft.Extensions.FileProviders.PhysicalFileProvider}
OnLoadException
null
Optional
false
Path
"appsettings.json"
ReloadDelay
250
ReloadOnChange
true
Non-Public members
[1]
{Microsoft.Extensions.Configuration.Json.JsonConfigurationProvider}
[2]
{Microsoft.Extensions.Configuration.EnvironmentVariables.EnvironmentVariablesConfiguratior
Data
Count = 79
Static members
Non-Public members

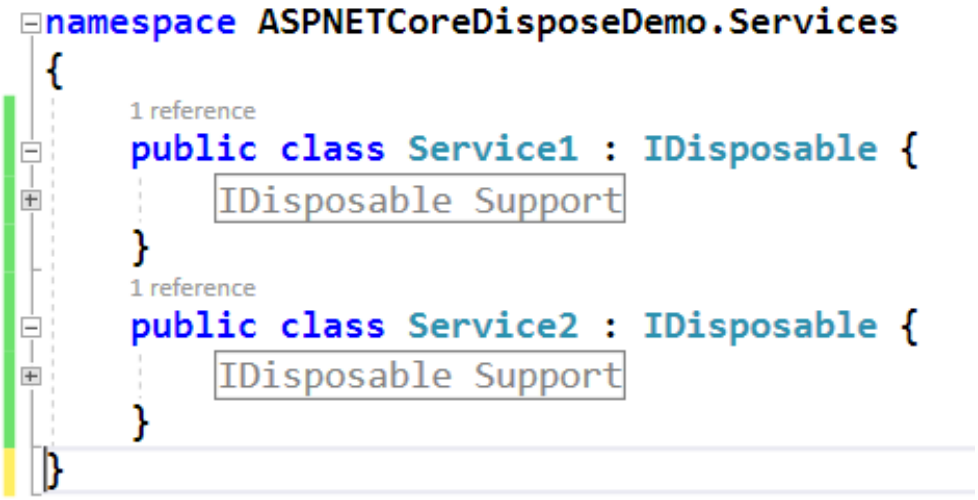

app.UseStaticFiles();
▲ 3 of 3 ▼ (extension) IApplicationBuilder IApplicationBuilder.UseStaticFiles(string requestPath)
Enables static file serving for the given request path
requestPath: The relative request path.


app.UseMvc(routes =>
(parameter) Microsoft.AspNetCore.Routing.IRouteBuilder routes
{

```
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li><a asp-area="" asp-controller="Home" asp-action="Index">Home</a></li>
        <li><a asp-area="" asp-controller="Home" asp-action="About">About</a></li>
        <li><a asp-area="" asp-controller="Home" asp-action="Contact">Contact</a></li>
        <li><a asp-area="" asp-controller="Home" asp-action="ServicesAvailable">List of Services</a></li>
    </ul>
</div>
```

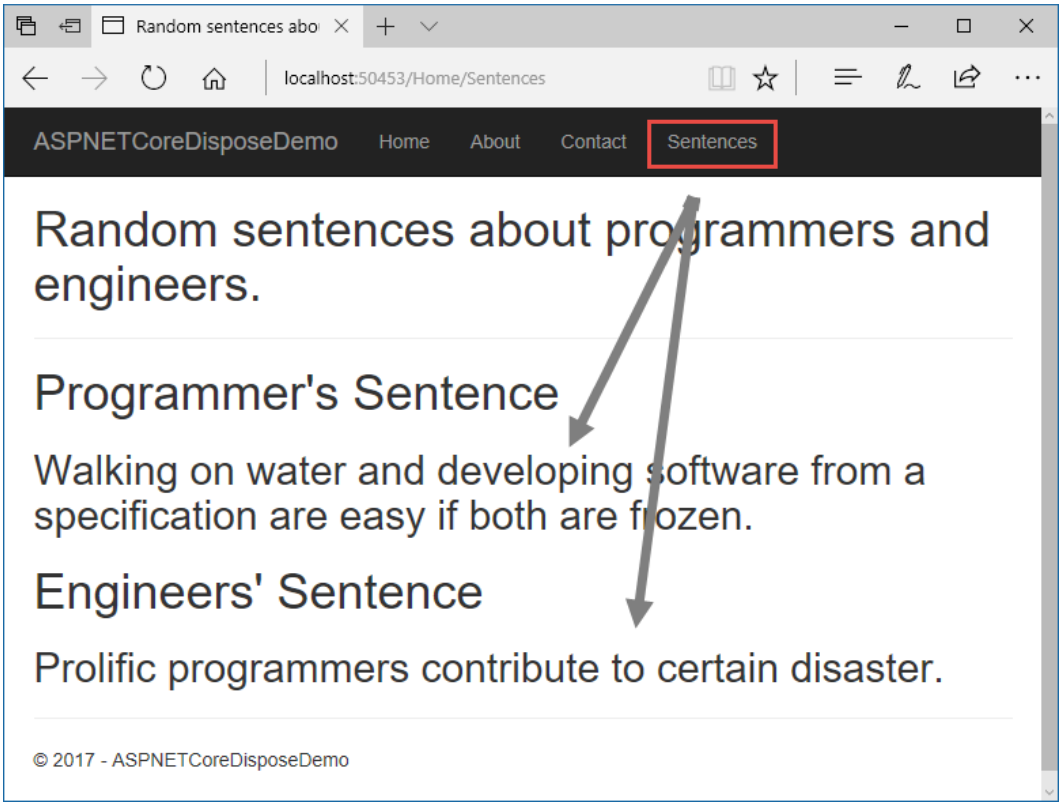


Type	Lifetime	Instance
IHostingEnvironment	Singleton	
ILogger`1	Singleton	Logger`1
IApplicationBuilderFactory	Transient	ApplicationBuilderFactory
IHttpContextFactory	Transient	HttpContextFactory
IOptions`1	Singleton	OptionsManager`1
IOptionsMonitor`1	Singleton	OptionsMonitor`1
IOptionsSnapshot`1	Scoped	OptionsSnapshot`1
IStartupFilter	Transient	AutoRequestServicesStartupFilter
IServiceProviderFactory`1	Transient	DefaultServiceProviderFactory



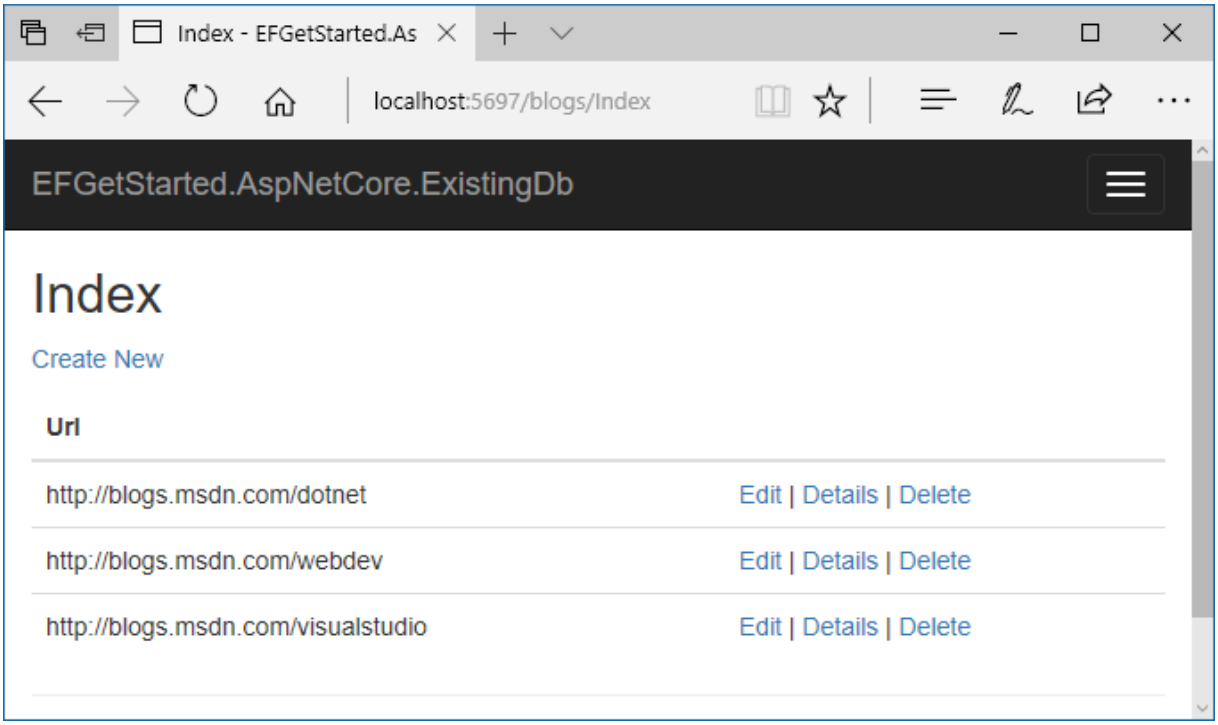

```
public class ProgrammerSentenceSvc : IDisposable {
    public List<string> programmersSentences = new List<string>()
    {
        "The trouble with programmers is that you can never tell what" +
        " a programmer is doing until it's too late",
        "Walking on water and developing software from a specification" +
        " are easy if both are frozen.",
        "A programming language is low level when its programs" + "" +
        " require attention to the irrelevant.",
        "In theory, theory and practice are the same. In practice, they're not.",
        "Perl - The only language that looks the same before" +
        " and after RSA encryption."
    };

    IDisposable Support
}
```

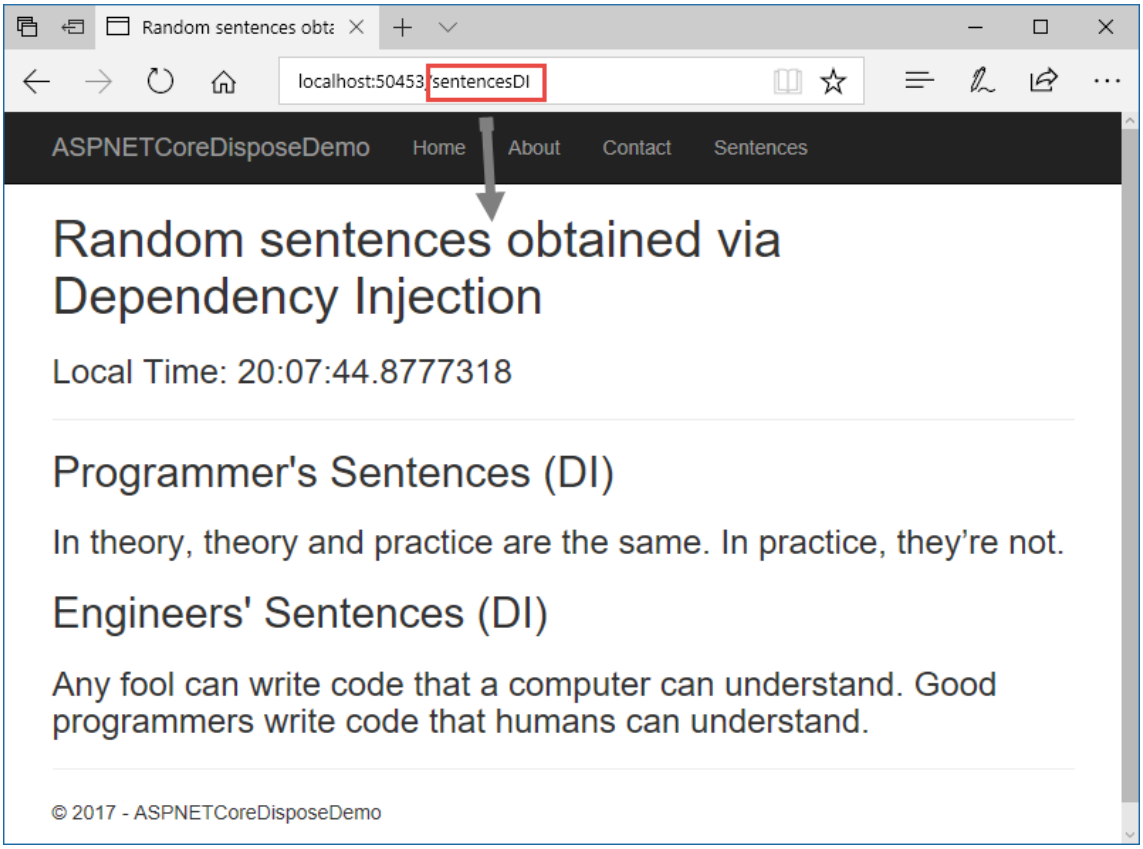
```
public class EngineerSentenceSvc : IDisposable {
    public List<string> engineersSentences = new List<string>()
    {
        "The best way to get a project done faster is to start sooner",
        "Even the best planning is not so omniscient" +
        " as to get it right the first time.",
        "Any fool can write code that a computer can " +
        "understand. Good programmers write code that" +
        "humans can understand.",
        "Program testing can be used to show the presence" +
        " of bugs, but never to show their absence!",
        "Prolific programmers contribute to certain disaster."
    };
    IDisposable Support
}
```



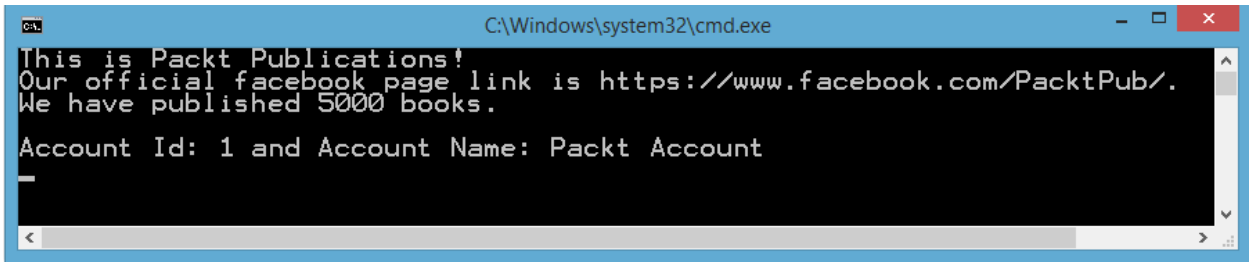
```
public void ConfigureServices(IServiceCollection services)
{
    // Add framework services.
    services.AddMvc();

    var connection = @"Server=(localdb)\mssqllocaldb;Database=Blogging;Trusted_Connection=True;";
    services.AddDbContext<BloggingContext>(options => options.UseSqlServer(connection));
}
```

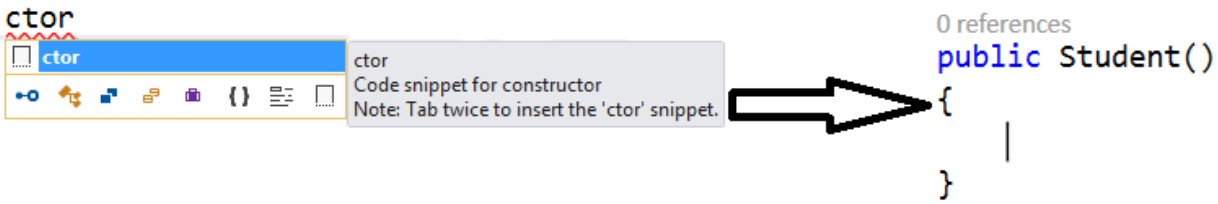
```
public class BlogsController : Controller
{
    private readonly BloggingContext _context;

    0 references | 0 exceptions
    public BlogsController(BloggingContext context)
    {
        _context = context;
    }
```

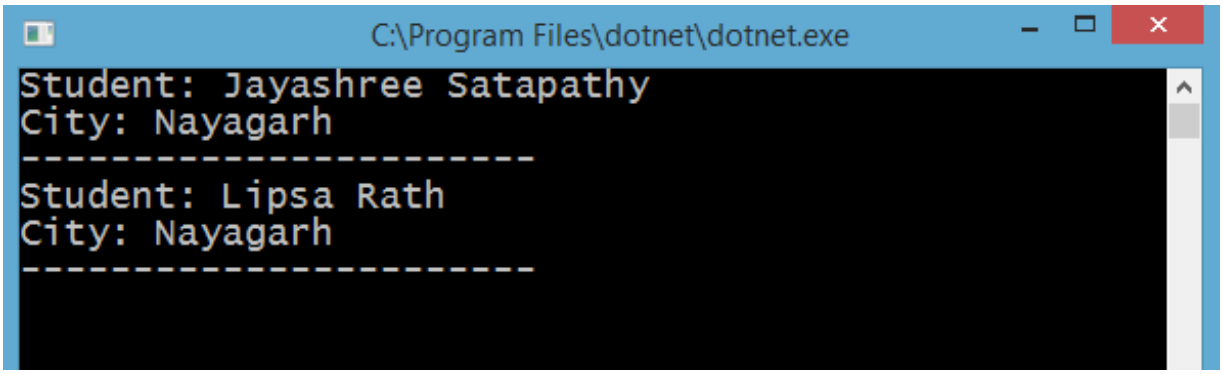

Index - EFGetStarted.As
localhost:5697/blogs/Index
EFGetStarted.AspNetCore.ExistingDb
Index
Create New
Url
http://blogs.msdn.com/dotnet
Edit | Details | Delete
http://blogs.msdn.com/webdev
Edit | Details | Delete
http://blogs.msdn.com/visualstudio
Edit | Details | Delete

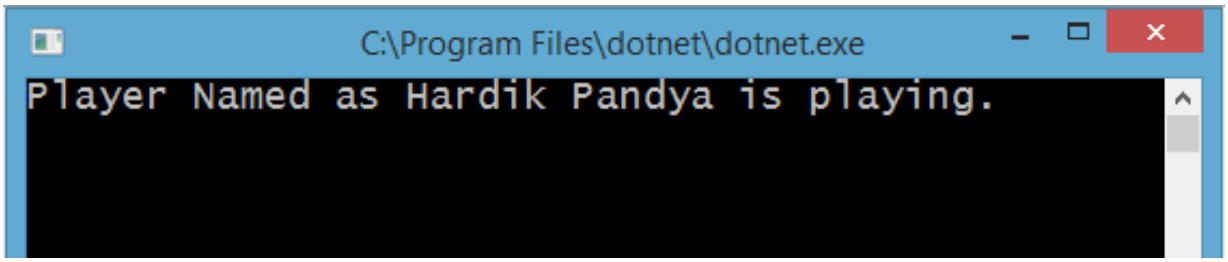

Random sentences obta
localhost:50453/sentencesDI
ASPNETCoreDisposeDemo
Home
About
Contact
Sentences
Random sentences obtained via Dependency Injection
Local Time: 20:07:44.8777318
Programmer's Sentences (DI)
In theory, theory and practice are the same. In practice, they're not.
Engineers' Sentences (DI)
Any fool can write code that a computer can understand. Good programmers write code that humans can understand.
© 2017 - ASPNETCoreDisposeDemo

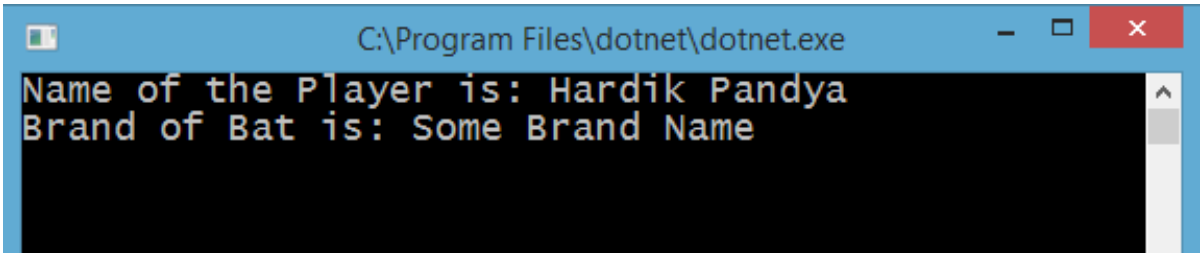
Chapter 5: Object Composition

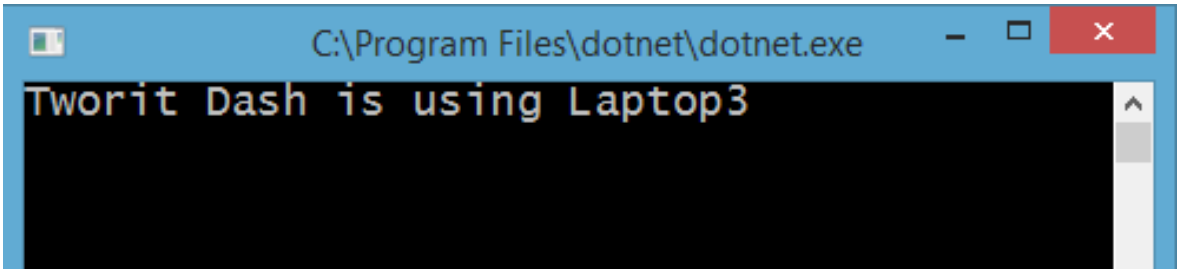


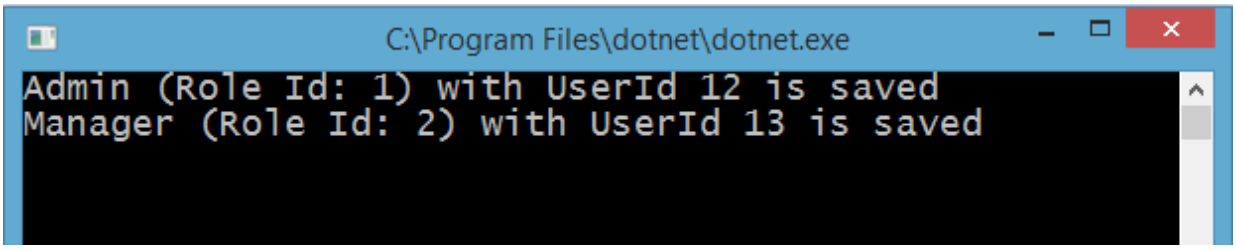





C:\Program Files\dotnet\dotnet.exe
Player Named as Hardik Pandya is playing.


C:\Program Files\dotnet\dotnet.exe
Name of the Player is: Hardik Pandya
Brand of Bat is: Some Brand Name

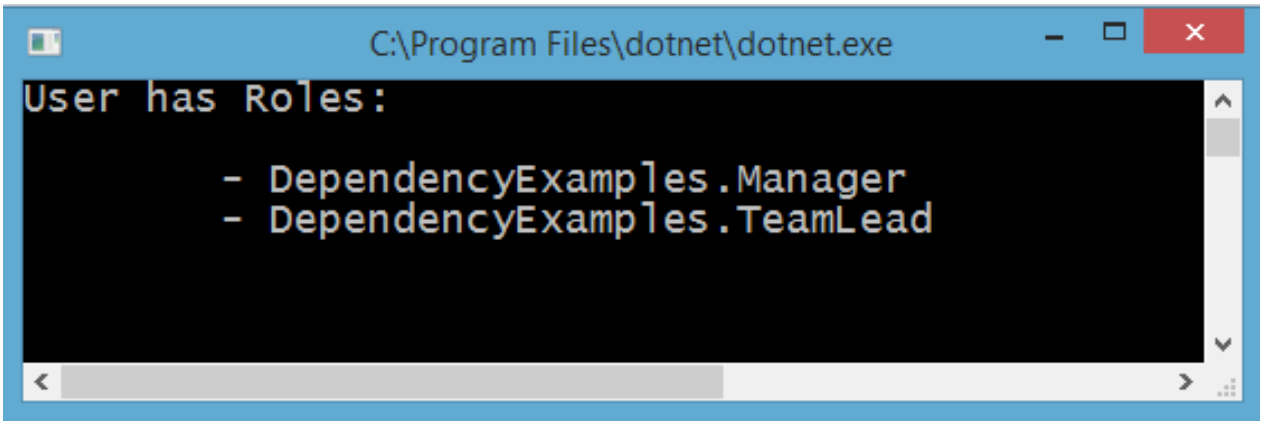

C:\Program Files\dotnet\dotnet.exe
Tworit Dash is using Laptop3




```
public class DeliveryHead : Manager, TeamLead
{

}
```

class DependencyExamples.TeamLead

Class 'DeliveryHead' cannot have multiple base classes: 'Manager' and 'TeamLead'



Main Method

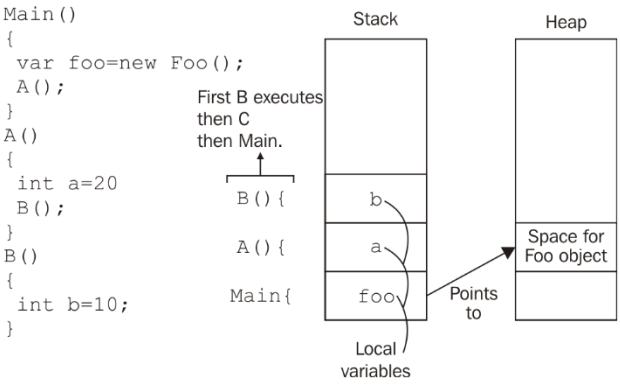
```
namespace PacktCoreWebApp
{
    0 references
    public class Program
    {
        0 references | 0 exceptions
        public static void Main(string[] args)
        {
            BuildWebHost(args).Run();
        }

        1 reference | 0 exceptions
        public static IWebHost BuildWebHost(string[] args) =>
            WebHost.CreateDefaultBuilder(args)
                .UseStartup<Startup>()
                .Build();
    }
}
```

Startup Class

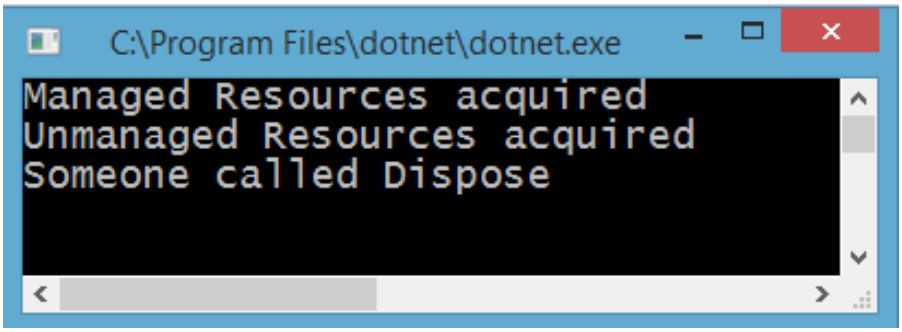
```
namespace PacktCoreWebApp
{
    2 references
    public class Startup
    {
        0 references | 0 exceptions
        public Startup(IConfiguration configuration)
        {
            Configuration = configuration;
        }

        1 reference | 0 exceptions
        public IConfiguration Configuration { get; }

        // This method gets called by the runtime.
        // Use this method to add services to the container.
        0 references | 0 exceptions
        public void ConfigureServices(IServiceCollection services)
        {
            services.AddMvc();
        }

        // This method gets called by the runtime.
        // Use this method to configure the HTTP request pipeline.
        0 references | 0 exceptions
        public void Configure(IApplicationBuilder app,
            IHostingEnvironment env)...
    }
}
```

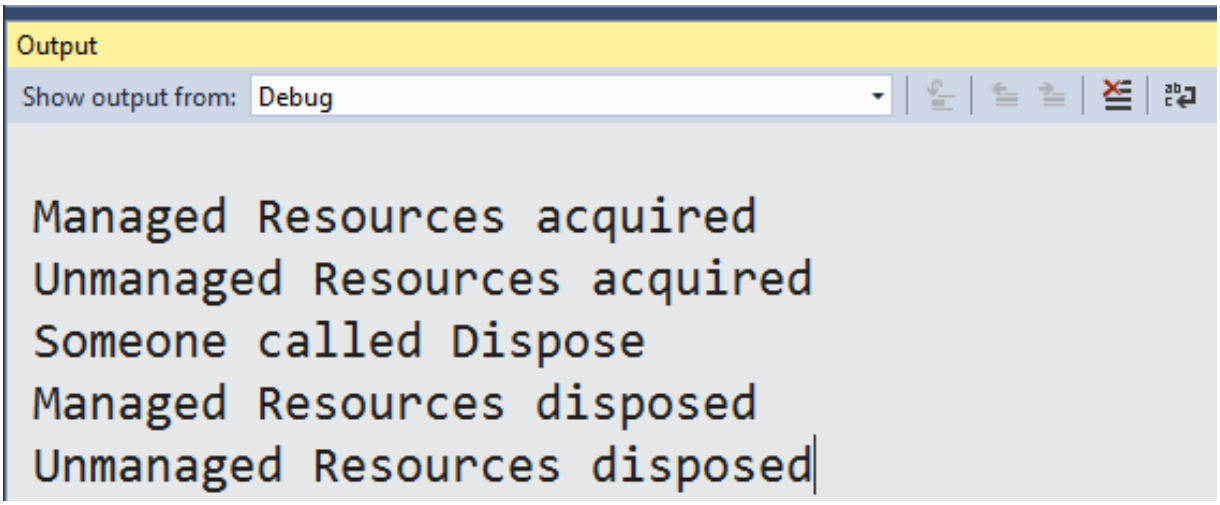
```
//
// Controller Factory
//
// This has a cache, so it needs to be a singleton
services.TryAddSingleton<IControllerFactory, DefaultControllerFactory>();

// Will be cached by the DefaultControllerFactory
services.TryAddTransient<IControllerActivator, DefaultControllerActivator>();

services.TryAddSingleton<IControllerFactoryProvider, ControllerFactoryProvider>();
services.TryAddSingleton<IControllerActivatorProvider, ControllerActivatorProvider>();
services.TryAddEnumerable(
    ServiceDescriptor.Transient<IControllerPropertyActivator, DefaultControllerPropertyActivator>());
```

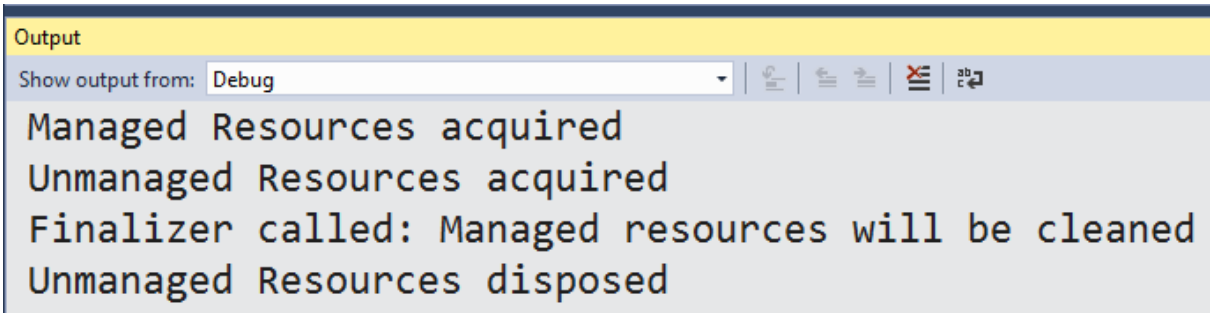
Chapter 6: Object Lifetime




```
DeriveClass2's constructor is called.
DeriveClass2's destructor is called.
DeriveClass1's destructor is called.
BaseClass's destructor is called.
```



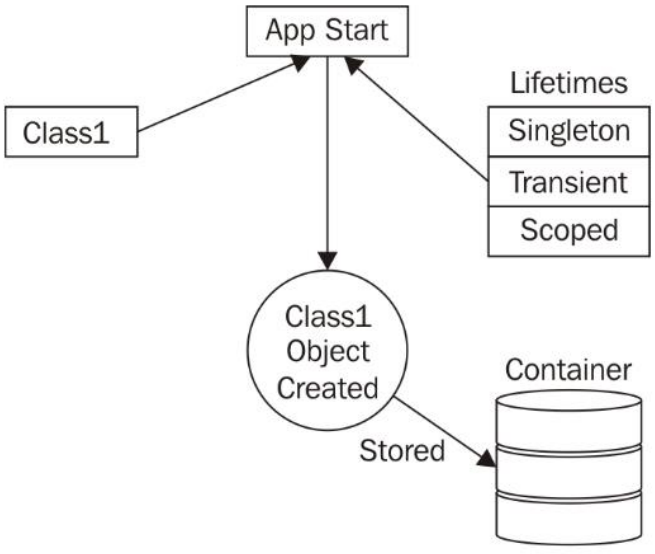

Output
Show output from: Debug
Managed Resources acquired
Unmanaged Resources acquired
Someone called Dispose
Managed Resources disposed
Unmanaged Resources disposed

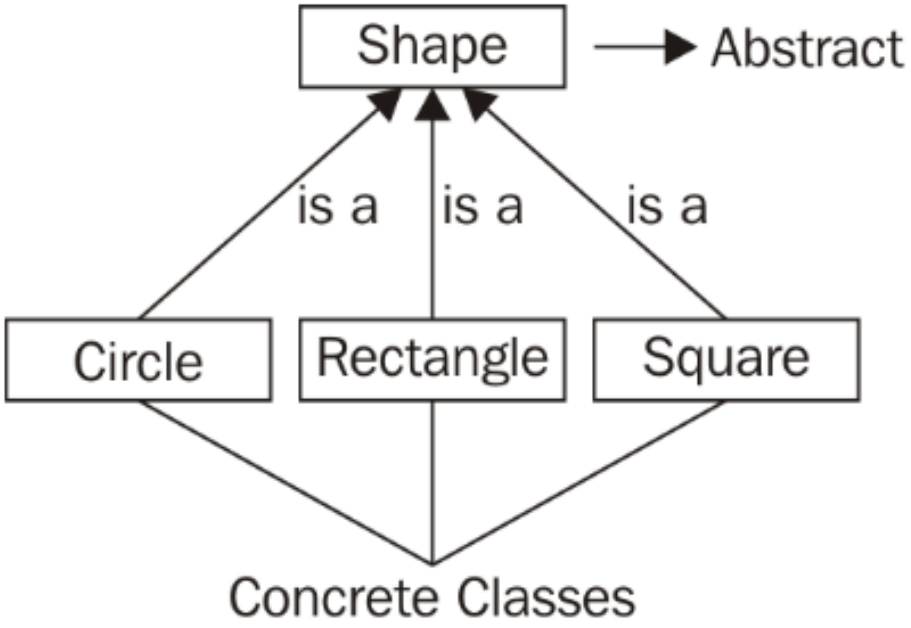

Output
Show output from: Debug
Managed Resources acquired
Unmanaged Resources acquired
Finalizer called: Managed resources will be cleaned
Unmanaged Resources disposed

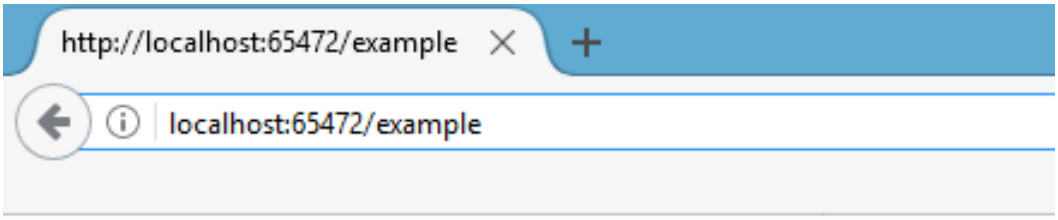
Output

Show output from: Debug

```
Managed Resources acquired
Unmanaged Resources acquired
Someone called Dispose
Managed Resources disposed
Unmanaged Resources disposed
Someone called Dispose
Dispose(bool) already called
Someone called Dispose
Dispose(bool) already called
```




Shape
Abstract
is a
is a
is a
Circle
Rectangle
Square
Concrete Classes



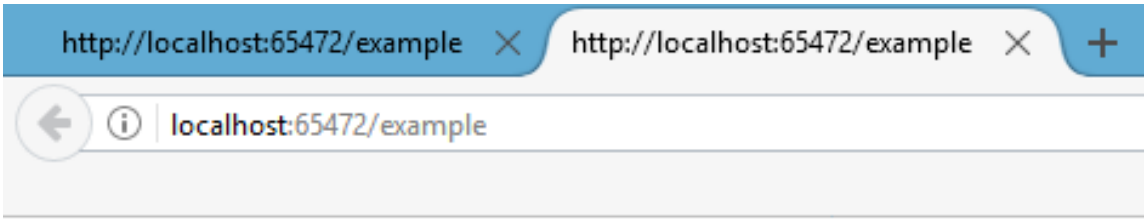
Lifetimes

ExampleController Dependencies

Lifestyle	Guid Value
Transient	9a18e7f1-a37d-4f18-9404-9d4ccd5a9584
Scoped	0af905dc-832e-4de2-b596-6b83e7ecda11
Singleton	f1f1b06b-3f6b-4e07-8cf6-9ead785401c1
Instance	00000000-0000-0000-0000-000000000000

ExampleService Dependencies

Lifestyle	Guid Value
Transient	092cf7e8-07c9-488f-977b-d570fcc72ba1
Scoped	0af905dc-832e-4de2-b596-6b83e7ecda11
Singleton	f1f1b06b-3f6b-4e07-8cf6-9ead785401c1
Instance	00000000-0000-0000-0000-000000000000



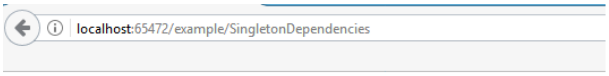
Lifetimes

ExampleController Dependencies

Lifestyle	Guid Value
Transient	fd33c6ff-cabb-4555-840b-6cb33b05813a
Scoped	6b9f0c9c-33b6-4cef-985e-a69171db7f48
Singleton	f1f1b06b-3f6b-4e07-8cf6-9ead785401c1
Instance	00000000-0000-0000-0000-000000000000

ExampleService Dependencies

Lifestyle	Guid Value
Transient	0bf9a17a-e5ae-4f96-a410-eb35b751f962
Scoped	6b9f0c9c-33b6-4cef-985e-a69171db7f48
Singleton	f1f1b06b-3f6b-4e07-8cf6-9ead785401c1
Instance	00000000-0000-0000-0000-000000000000



Singleton Lifetime Dependencies

ExampleController

Request 1

Singleton ExampleId: 3506d3ac-3948-4047-b6dd-d0f7eed0cdbf

Dependencies	Guid Value
Scoped Dependency	4173e92d-430a-4fa0-b47c-81f54fbc8630
Transient Dependency	a4fa8b4f-d6d2-4928-8e43-df62ea0d3cd2

ExampleService

Singleton ExampleId: 3506d3ac-3948-4047-b6dd-d0f7eed0cdbf

Dependencies	Guid Value
Scoped Dependency	4173e92d-430a-4fa0-b47c-81f54fbc8630
Transient Dependency	a4fa8b4f-d6d2-4928-8e43-df62ea0d3cd2

Singleton Lifetime Dependencies

ExampleController

Request 2

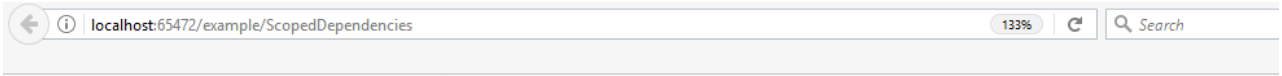
Singleton ExampleId: 3506d3ac-3948-4047-b6dd-d0f7eed0cdbf

Dependencies	Guid Value
Scoped Dependency	4173e92d-430a-4fa0-b47c-81f54fbc8630
Transient Dependency	a4fa8b4f-d6d2-4928-8e43-df62ea0d3cd2

ExampleService

Singleton ExampleId: 3506d3ac-3948-4047-b6dd-d0f7eed0cdbf

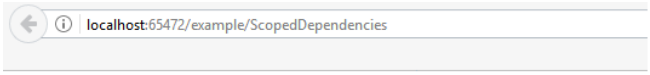
Dependencies	Guid Value
Scoped Dependency	4173e92d-430a-4fa0-b47c-81f54fbc8630
Transient Dependency	a4fa8b4f-d6d2-4928-8e43-df62ea0d3cd2



An unhandled exception occurred while processing the request.

InvalidOperationException: A circular dependency was detected for the service of type 'LifetimesExample.IExampleScoped'.

Microsoft.Extensions.DependencyInjection.ServiceProvider.GetServiceCallSite(Type serviceType, ISet<Type> callSiteChain)



Scoped Lifetime Dependencies

ExampleController Request 1

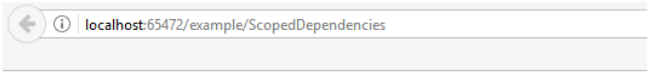
Scoped ExampleId: 4c2857f5-ae27-417b-b7b3-4a619231ddfc

Dependencies	Guid Value
Singleton Dependency	2081d240-2915-4558-9356-16c19777a43f
Transient Dependency	e67aee53-331f-4a79-8128-3046d0c8edda

ExampleService

Scoped ExampleId: 4c2857f5-ae27-417b-b7b3-4a619231ddfc

Dependencies	Guid Value
Singleton Dependency	2081d240-2915-4558-9356-16c19777a43f
Transient Dependency	e67aee53-331f-4a79-8128-3046d0c8edda



Scoped Lifetime Dependencies

ExampleController Request 2

Scoped ExampleId: 65e2c65c-281b-4fda-adb3-e390420b84e3

Dependencies	Guid Value
Singleton Dependency	2081d240-2915-4558-9356-16c19777a43f
Transient Dependency	1d0ffa38-c857-4edf-8b72-78b467950968

ExampleService

Scoped ExampleId: 65e2c65c-281b-4fda-adb3-e390420b84e3

Dependencies	Guid Value
Singleton Dependency	2081d240-2915-4558-9356-16c19777a43f
Transient Dependency	1d0ffa38-c857-4edf-8b72-78b467950968

localhost:65472/example/TransientDependencies

Transient Lifetime Dependencies

ExampleController Request 1

Transient ExampleId: b56ea918-5baf-458f-b5cd-2d3fa40a0b8f

Dependencies	Guid Value
Singleton Dependency	6ad2cff8-dabb-4ca1-8a75-ae69328f5819
Scoped Dependency	8a69f773-508f-4610-9041-987b9f4ada0e

ExampleService

Transient ExampleId: df3a3deb-b365-48a6-a326-1ad7d36acd8b

Dependencies	Guid Value
Singleton Dependency	6ad2cff8-dabb-4ca1-8a75-ae69328f5819
Scoped Dependency	8a69f773-508f-4610-9041-987b9f4ada0e

localhost:65472/example/TransientDependencies

Transient Lifetime Dependencies

ExampleController Request 2

Transient ExampleId: 8c20b689-0f72-4204-b947-fe671e2e9a78

Dependencies	Guid Value
Singleton Dependency	6ad2cff8-dabb-4ca1-8a75-ae69328f5819
Scoped Dependency	2dd6c085-6d26-412a-8f93-2935f25c3ff8

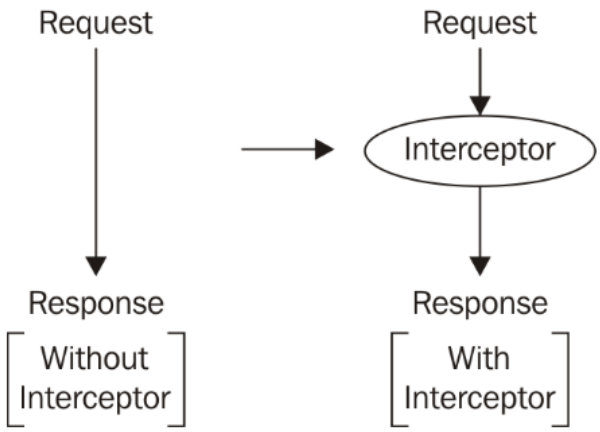
ExampleService

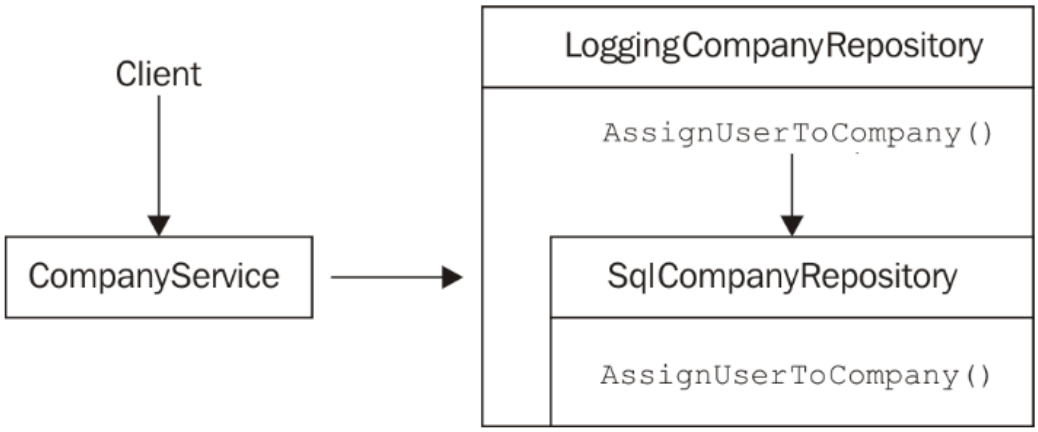
Transient ExampleId: 43f0415c-c8c1-473d-a5a6-563001e4318b

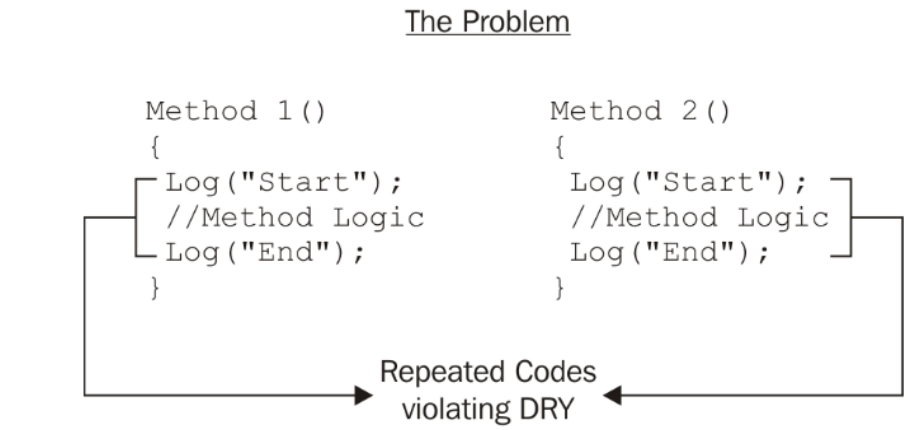
Dependencies	Guid Value
Singleton Dependency	6ad2cff8-dabb-4ca1-8a75-ae69328f5819
Scoped Dependency	2dd6c085-6d26-412a-8f93-2935f25c3ff8

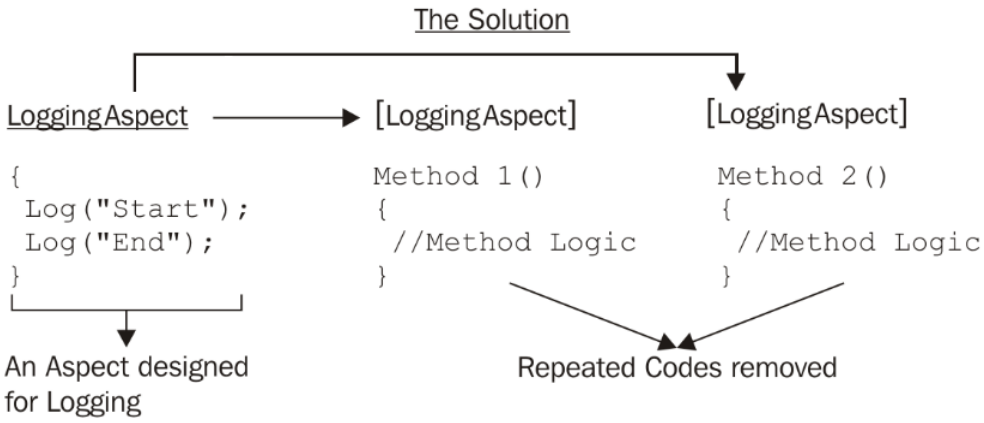
Lifestyles	Injected Lifestyle		
	Transient	Scoped	Singleton
Transient	✓	✓	✓
Scoped	✗	✓	✓
Singleton	✗	✗	✓

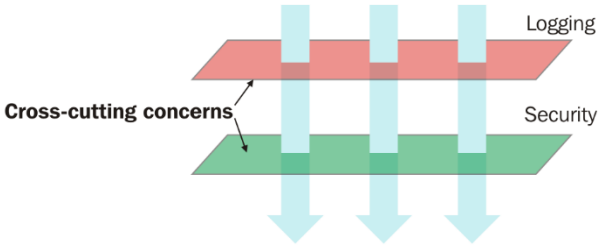
Chapter 7: Interception

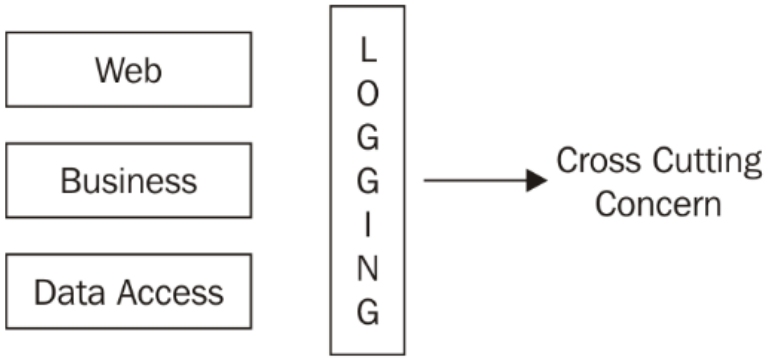


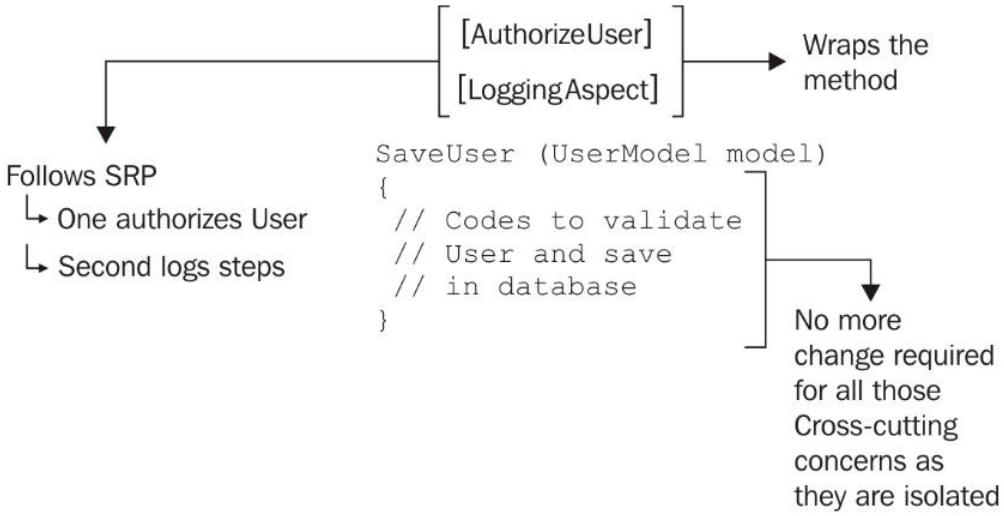


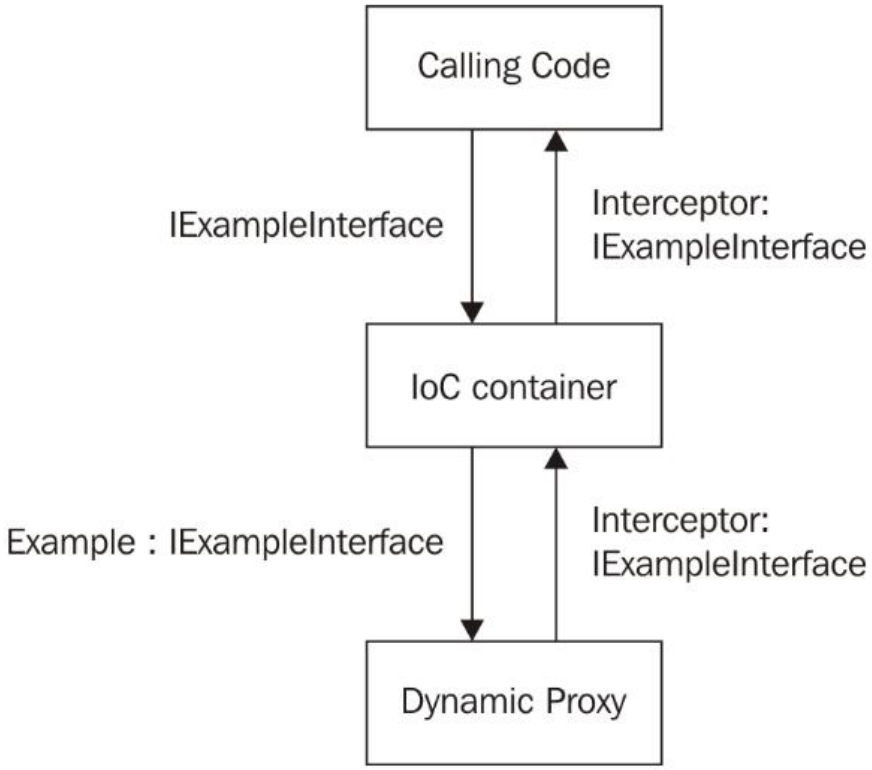

The Problem
Method 1()
{
Log("Start");
//Method Logic
Log("End");
}
Method 2()
{
Log("Start");
//Method Logic
Log("End");
}
Repeated Codes
violating DRY

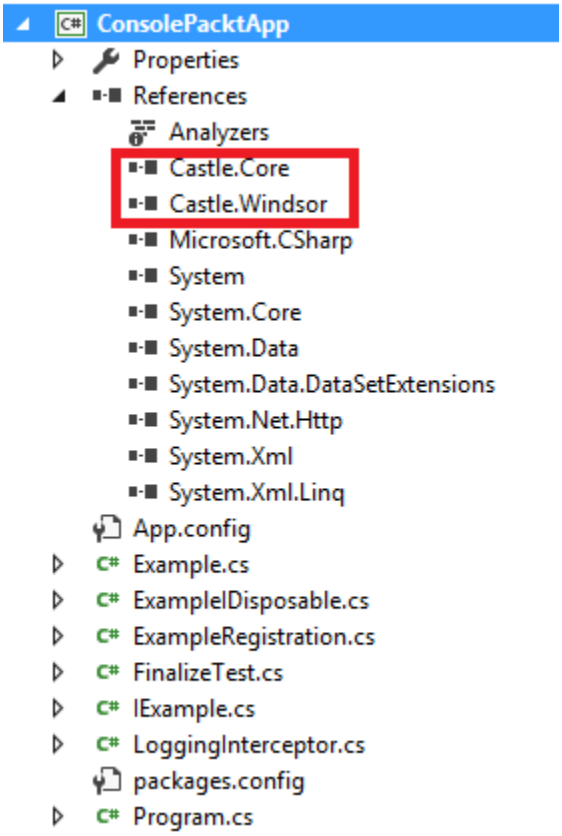

The Solution
LoggingAspect
[LoggingAspect]
[LoggingAspect]
{
Log("Start");
Log("End");
}
Method 1()
{
//Method Logic
}
Method 2()
{
//Method Logic
}
An Aspect designed
for Logging
Repeated Codes removed

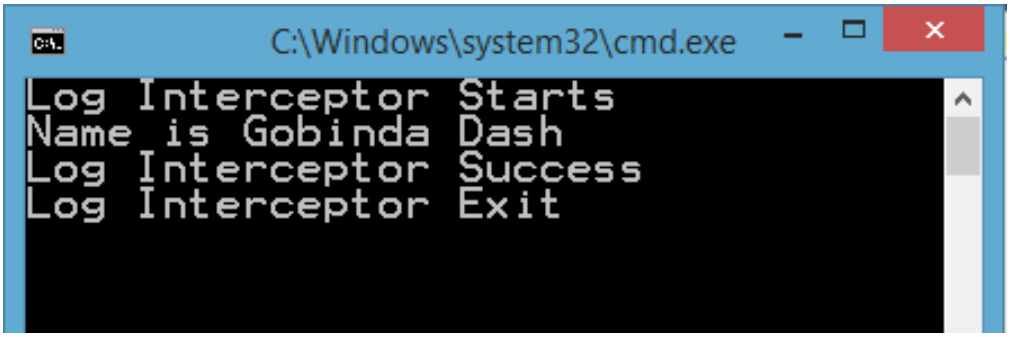

Logging
Cross-cutting concerns
Security

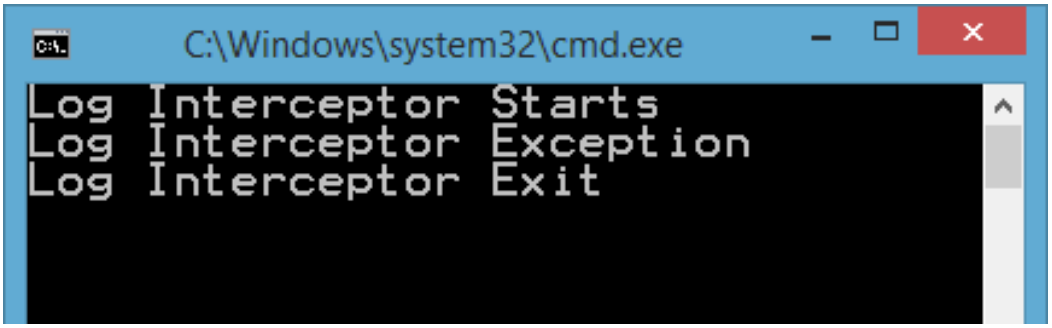

Web
Business
Data Access
LOGGING
Cross Cutting Concern

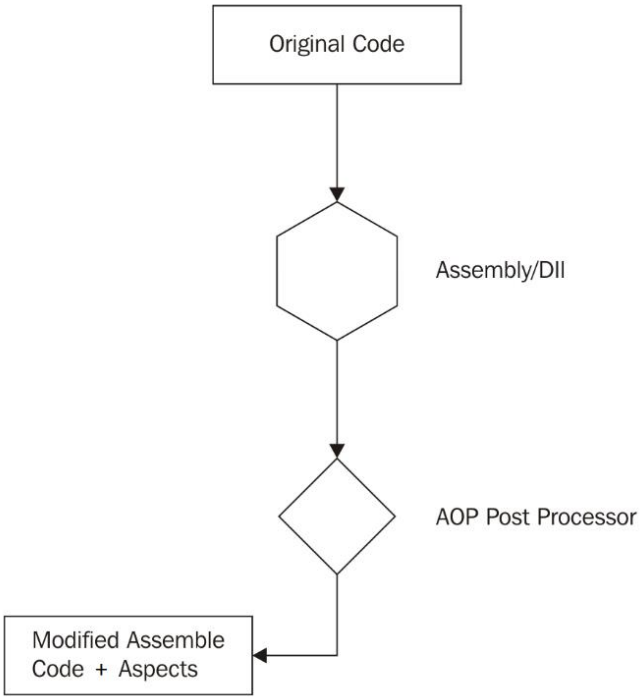

[AuthorizeUser]
[LoggingAspect]
Wraps the method
Follows SRP
One authorizes User
Second logs steps
SaveUser (UserModel model)
{
// Codes to validate
// User and save
// in database
}
No more change required for all those Cross-cutting concerns as they are isolated

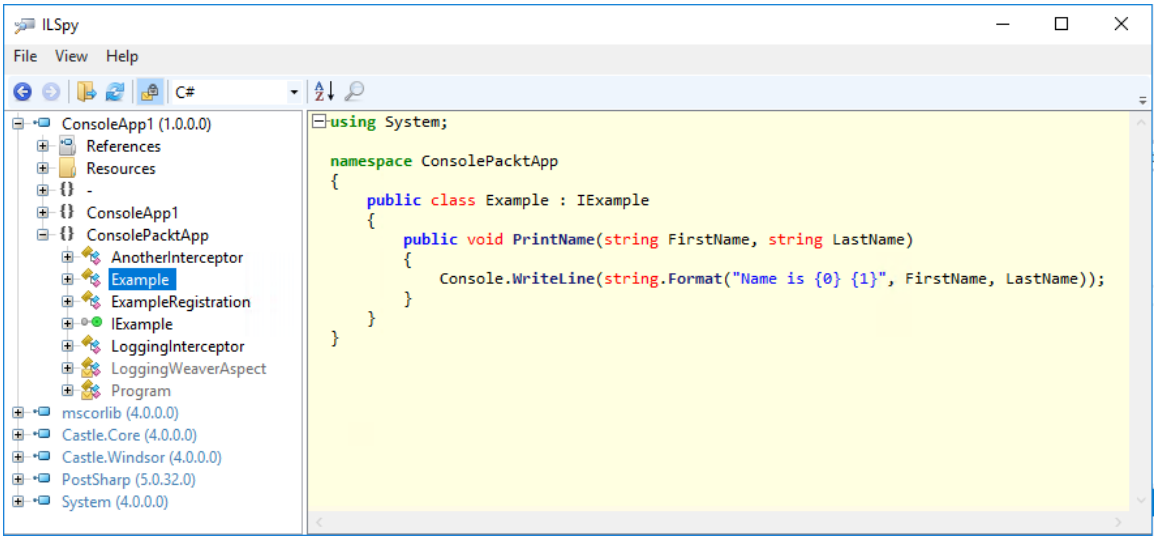

Calling Code
IExampleInterface
Interceptor:
IExampleInterface
IoC container
Example : IExampleInterface
Interceptor:
IExampleInterface
Dynamic Proxy

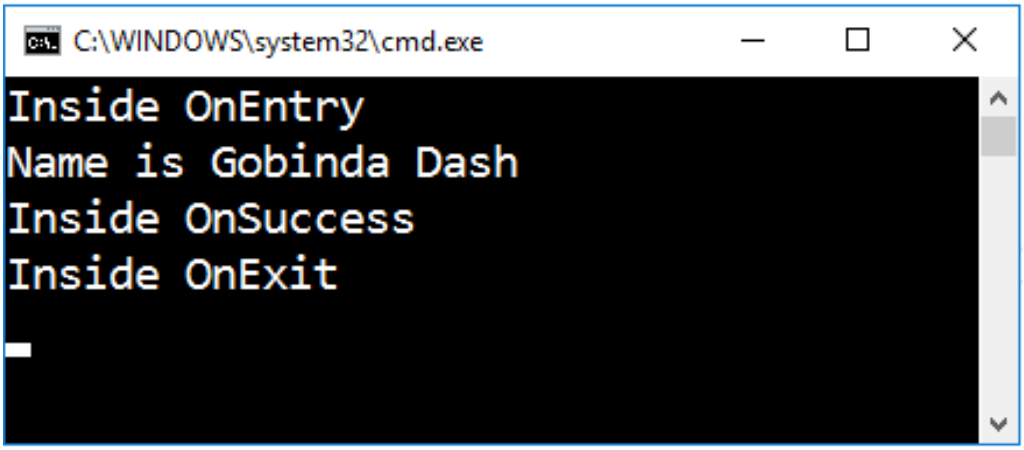

ConsolePacktApp
Properties
References
Analyzers
Castle.Core
Castle.Windsor
Microsoft.CSharp
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Net.Http
System.Xml
System.Xml.Linq
App.config
Example.cs
ExampleIDisposable.cs
ExampleRegistration.cs
FinalizeTest.cs
IExample.cs
LoggingInterceptor.cs
packages.config
Program.cs

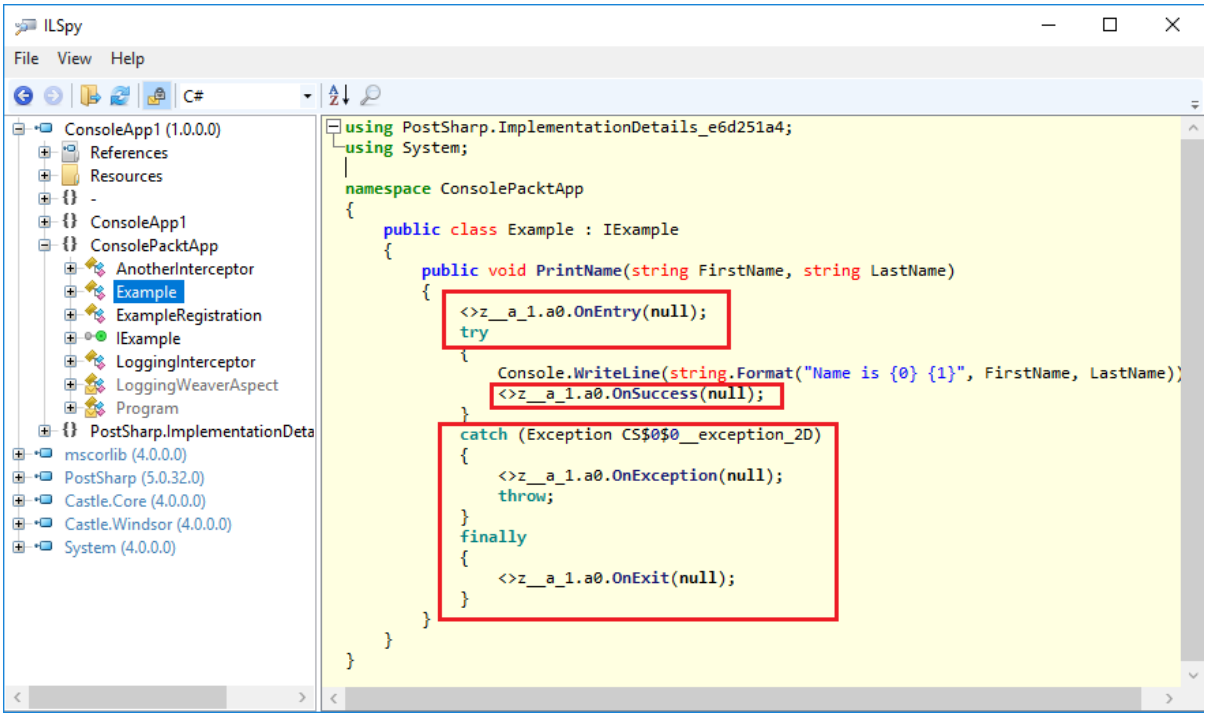

C:\Windows\system32\cmd.exe
Log Interceptor Starts
Name is Gobinda Dash
Log Interceptor Success
Log Interceptor Exit

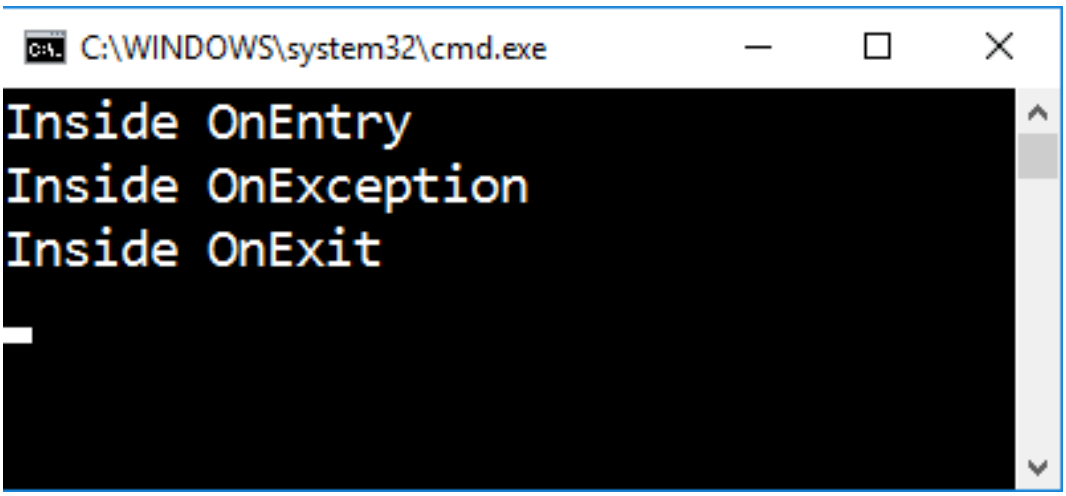

C:\Windows\system32\cmd.exe
Log Interceptor Starts
Log Interceptor Exception
Log Interceptor Exit


Original Code
Assembly/Dll
AOP Post Processor
Modified Assemble Code + Aspects


ILSpy
File View Help
C#
ConsoleApp1 (1.0.0.0)
References
Resources
-
ConsoleApp1
ConsolePacktApp
AnotherInterceptor
Example
ExampleRegistration
IExample
LoggingInterceptor
LoggingWeaverAspect
Program
mscorlib (4.0.0.0)
Castle.Core (4.0.0.0)
Castle.Windsor (4.0.0.0)
PostSharp (5.0.32.0)
System (4.0.0.0)
using System;
namespace ConsolePacktApp
{
public class Example : IExample
{
public void PrintName(string FirstName, string LastName)
{
Console.WriteLine(string.Format("Name is {0} {1}", FirstName, LastName));
}
}
}


C:\WINDOWS\system32\cmd.exe
Inside OnEntry
Name is Gobinda Dash
Inside OnSuccess
Inside OnExit

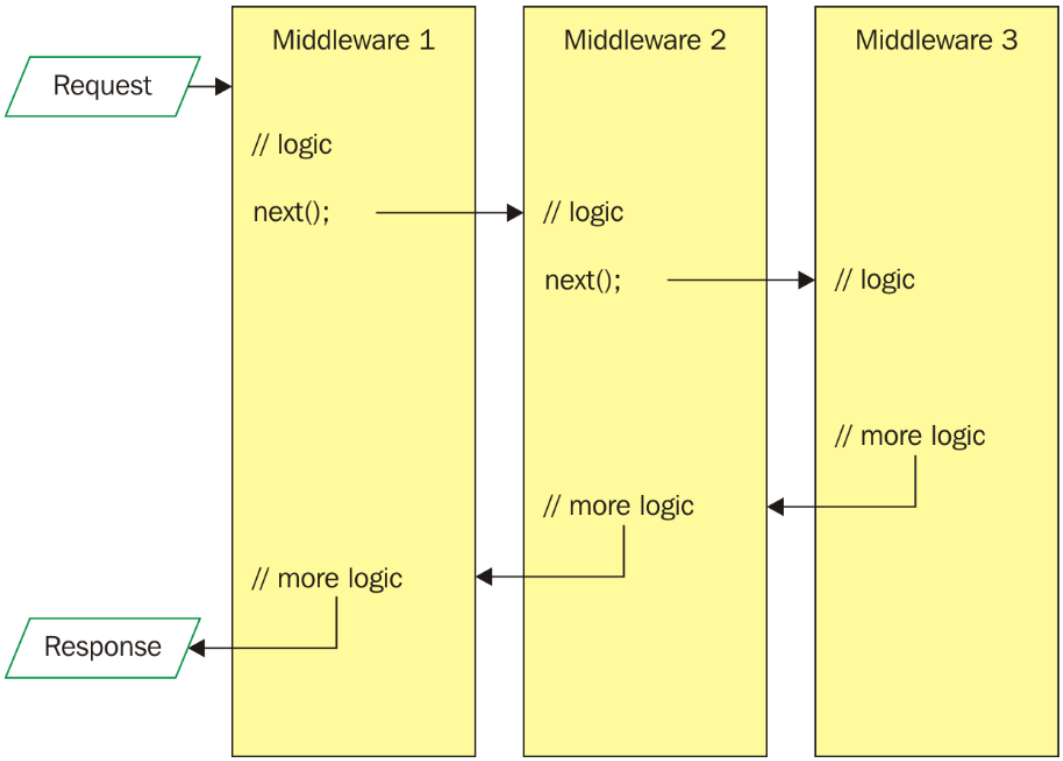

ILSpy
File View Help
C#
ConsoleApp1 (1.0.0.0)
References
Resources
-
ConsoleApp1
ConsolePacktApp
AnotherInterceptor
Example
ExampleRegistration
IExample
LoggingInterceptor
LoggingWeaverAspect
Program
PostSharp.ImplementationDeta
mscorlib (4.0.0.0)
PostSharp (5.0.32.0)
Castle.Core (4.0.0.0)
Castle.Windsor (4.0.0.0)
System (4.0.0.0)
using PostSharp.ImplementationDetails_e6d251a4;
using System;
namespace ConsolePacktApp
{
public class Example : IExample
{
public void PrintName(string FirstName, string LastName)
{
<>z__a_1.a0.OnEntry(null);
try
{
Console.WriteLine(string.Format("Name is {0} {1}", FirstName, LastName));
<>z__a_1.a0.OnSuccess(null);
}
catch (Exception CS$0$0__exception_2D)
{
<>z__a_1.a0.OnException(null);
throw;
}
finally
{
<>z__a_1.a0.OnExit(null);
}
}
}
}

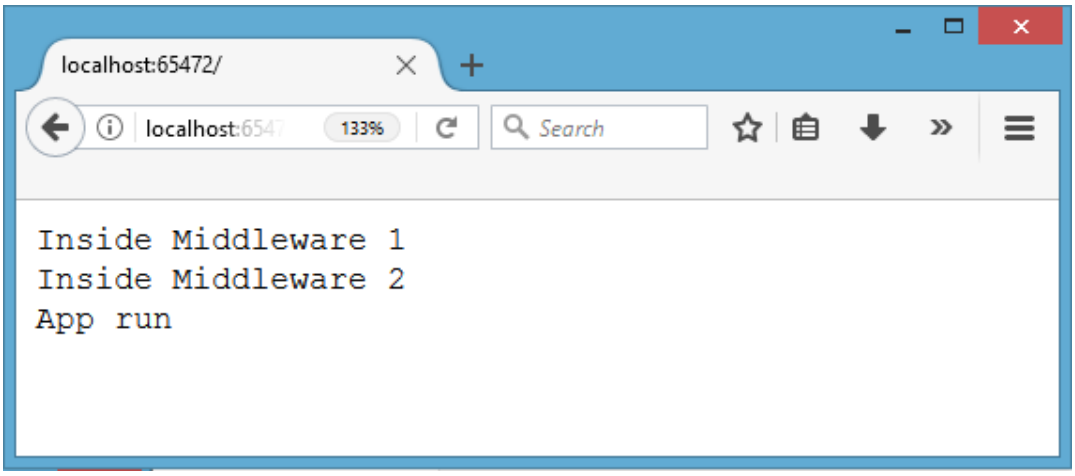

C:\WINDOWS\system32\cmd.exe
Inside OnEntry
Inside OnException
Inside OnExit

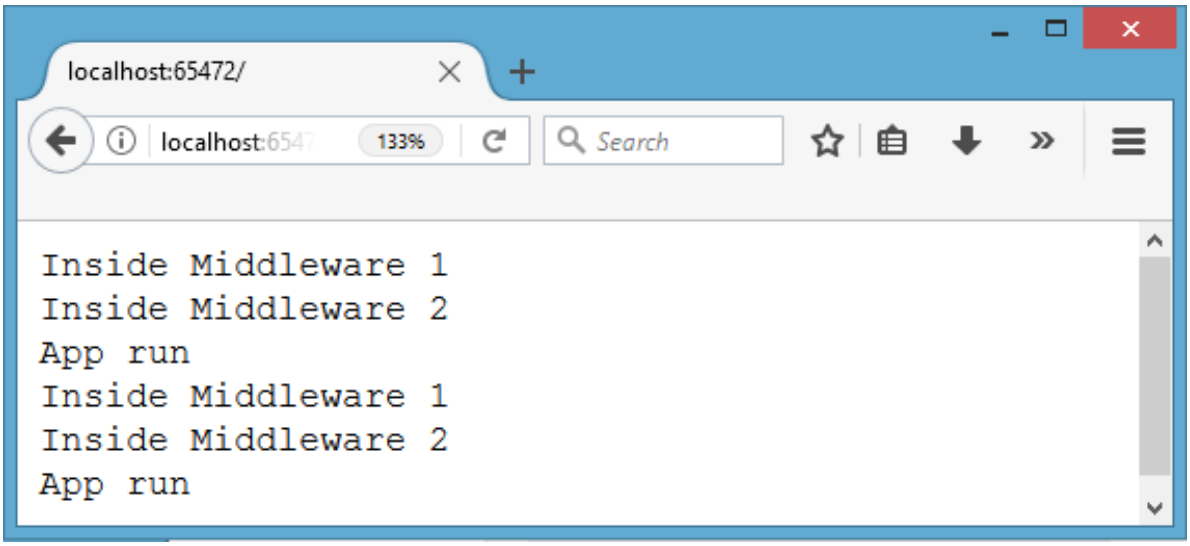
```
[SomeFilter]
0 references
public IActionResult About()
{
    ViewData["Message"] = "Your application description page.";

    return View();
}
```

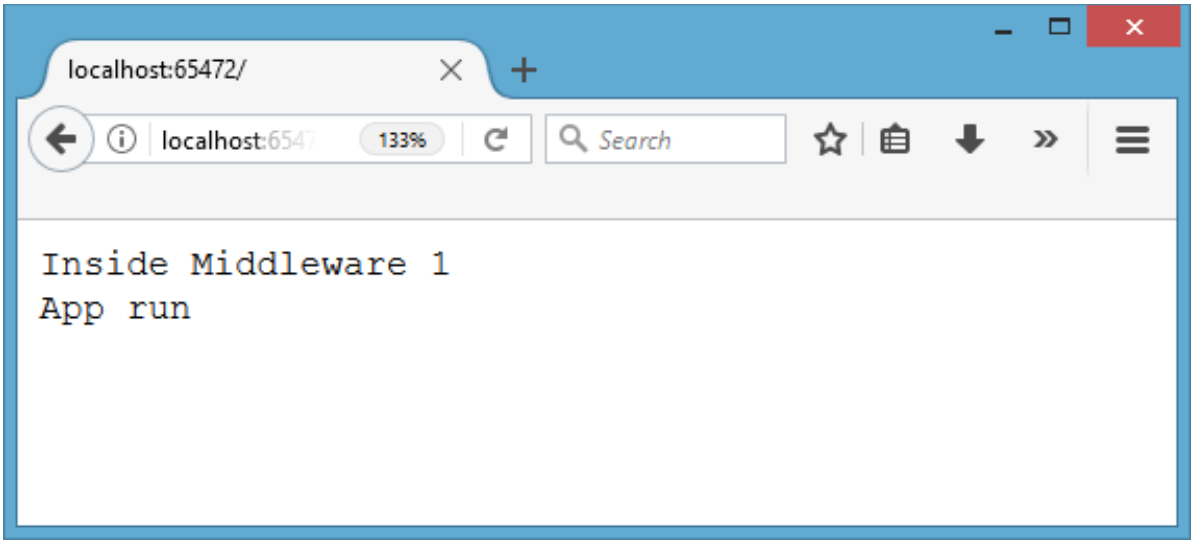
InvalidOperationException: No service for type 'PacktDIExamples.Filters.SomeFilter' has been registered.

Microsoft.Extensions.DependencyInjection.ServiceProviderServiceExtensions.GetRequiredService(IServiceProvider provider, Type serviceType)

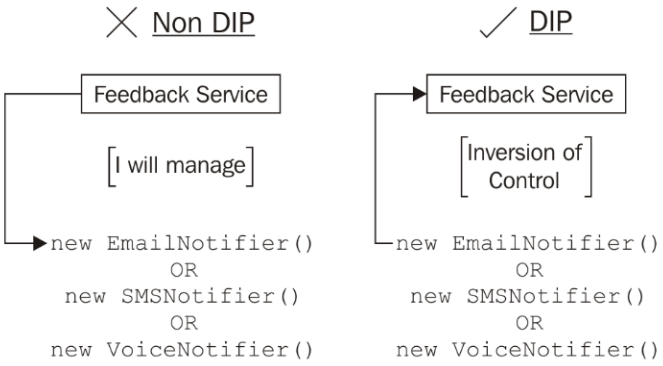

Middleware 1
Middleware 2
Middleware 3
Request
// logic
next();
// logic
next();
// logic
// more logic
// more logic
// more logic
Response

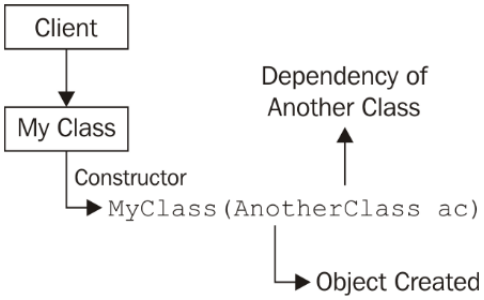

localhost:65472/
localhost:6547
133%
Search
Inside Middleware 1
Inside Middleware 2
App run

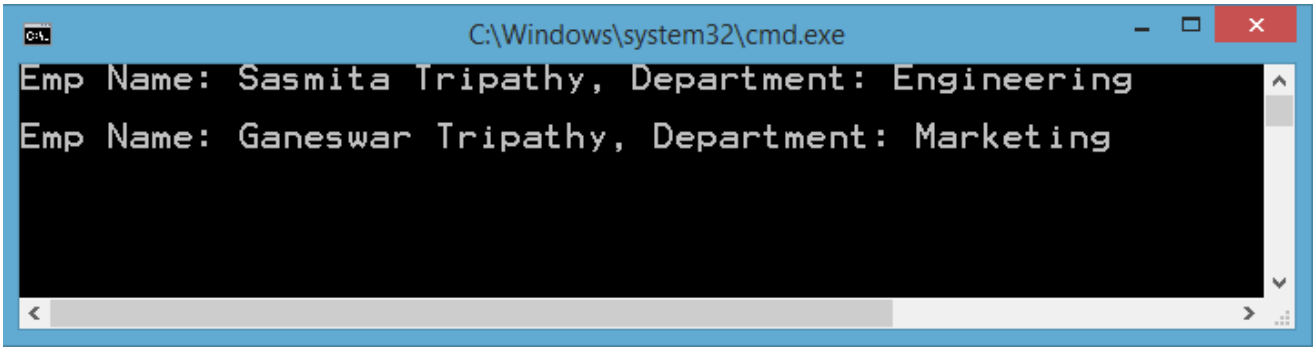

localhost:65472/
localhost:6547
133%
Search
Inside Middleware 1
Inside Middleware 2
App run
Inside Middleware 1
Inside Middleware 2
App run


localhost:65472/
localhost:6547
133%
Search
Inside Middleware 1
App run

Chapter 8: Patterns - Dependency Injection

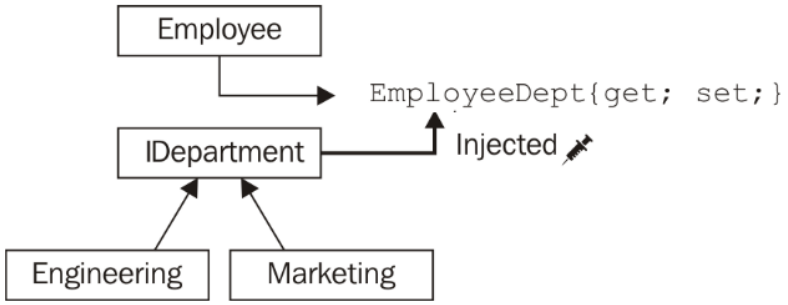






```
/// <summary>
/// Provides methods to create an MVC controller.
/// </summary>
public class ControllerActivatorProvider : IControllerActivatorProvider
{
    private static readonly Action<ControllerContext, object> _dispose = Dispose;
    private readonly Func<ControllerContext, object> _controllerActivatorCreate;
    private readonly Action<ControllerContext, object> _controllerActivatorRelease;

    public ControllerActivatorProvider(IControllerActivator controllerActivator)
    {
        if (controllerActivator == null)
        {
            throw new ArgumentNullException(nameof(controllerActivator));
        }

        // Compat: Delegate to controllerActivator if it's not the default implementation.
        if (controllerActivator.GetType() != typeof(DefaultControllerActivator))
        {
            _controllerActivatorCreate = controllerActivator.Create;
            _controllerActivatorRelease = controllerActivator.Release;
        }
    }
```

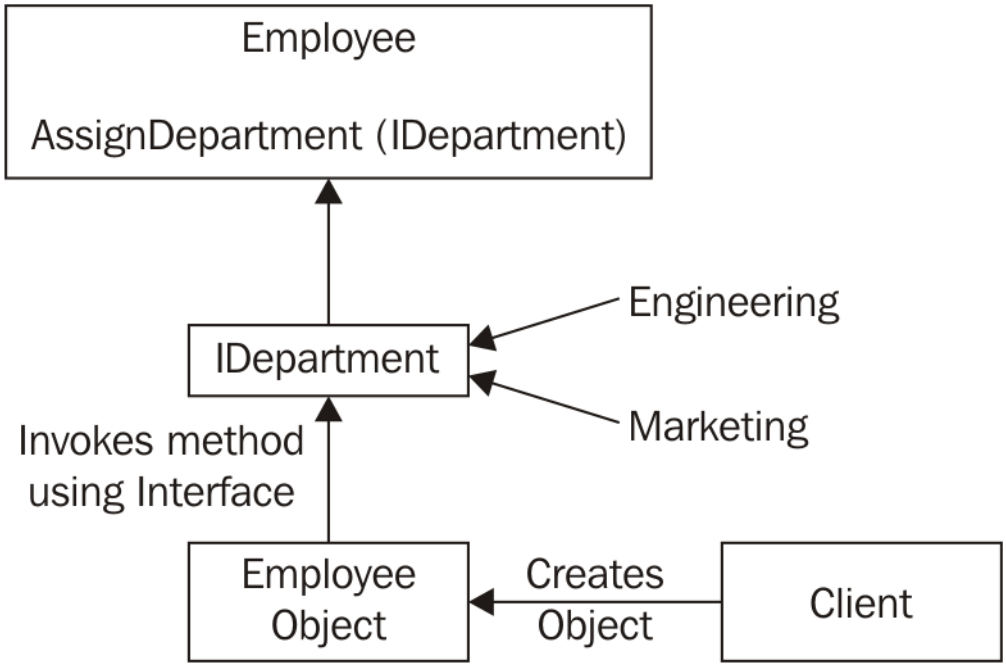
```
        //
        // Controller Factory
        //
        // This has a cache, so it needs to be a singleton
        services.TryAddSingleton<IControllerFactory, DefaultControllerFactory>();

        // Will be cached by the DefaultControllerFactory
        services.TryAddTransient<IControllerActivator, DefaultControllerActivator>();

        services.TryAddSingleton<IControllerFactoryProvider, ControllerFactoryProvider>();
        services.TryAddSingleton<IControllerActivatorProvider, ControllerActivatorProvider>();
```

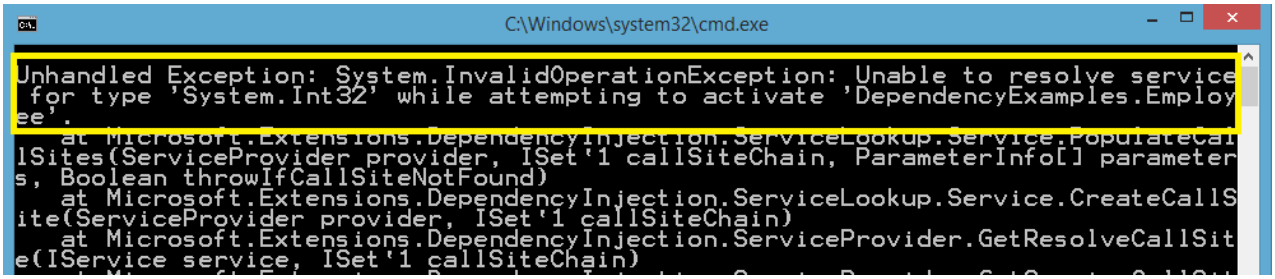



```
Console.WriteLine("Emp Name: " + emp.EmployeeName + ", Department: " + emp.EmployeeDept.DeptName);
Console.WriteLine();
Console.WriteLine("Emp Name: " + emp1.EmployeeName + ", Department: " + emp1.EmployeeDept.DeptName);
```

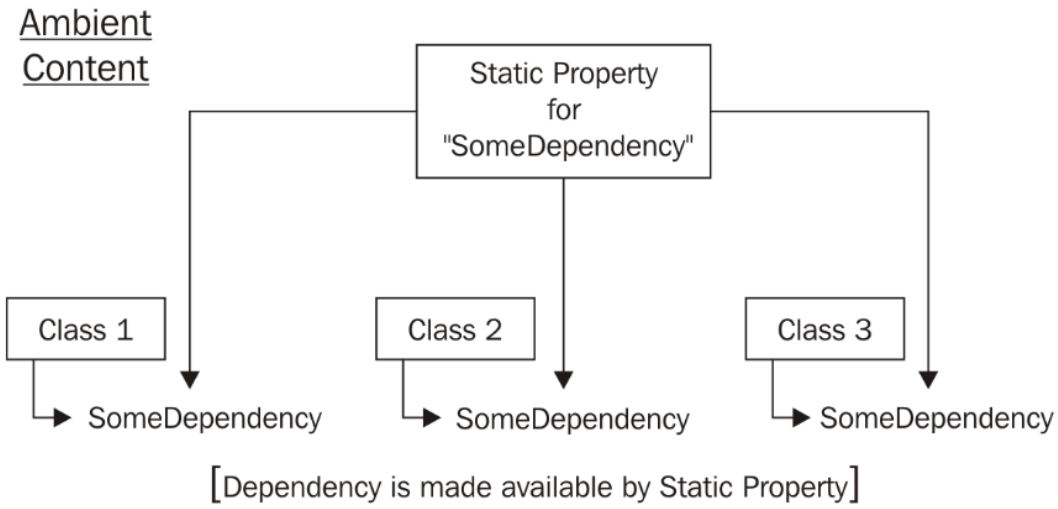
IDepartment Employee.EmployeeDept { set; }

The property or indexer 'Employee.EmployeeDept' cannot be used in this context because it lacks the get accessor



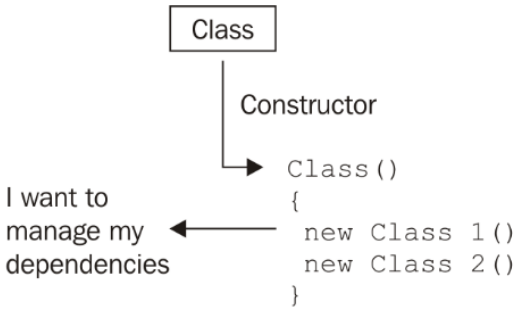


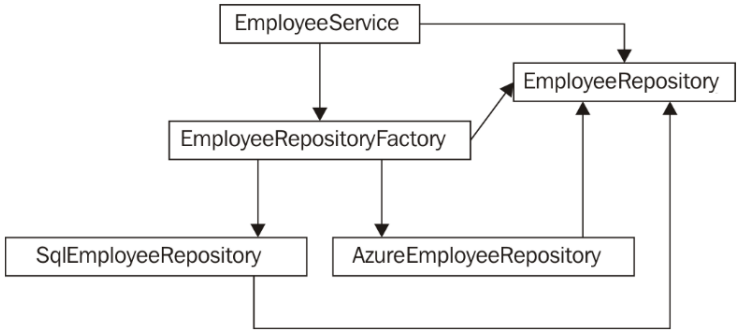

```
/// <summary>
/// Adds essential MVC services to the specified <see cref="IServiceCollection" />.
/// </summary>
/// <param name="services">The <see cref="IServiceCollection" /> to add services to.</param>
/// <returns>An <see cref="IMvcCoreBuilder"/> that can be used to further configure the MVC 
public static IMvcCoreBuilder AddMvcCore(this IServiceCollection services)
{
    if (services == null)
    {
        throw new ArgumentNullException(nameof(services));
    }

    var partManager = GetApplicationPartManager(services);
    services.TryAddSingleton(partManager);

    ConfigureDefaultFeatureProviders(partManager);
    ConfigureDefaultServices(services);
    AddMvcCoreServices(services);

    var builder = new MvcCoreBuilder(services, partManager);

    return builder;
}
```

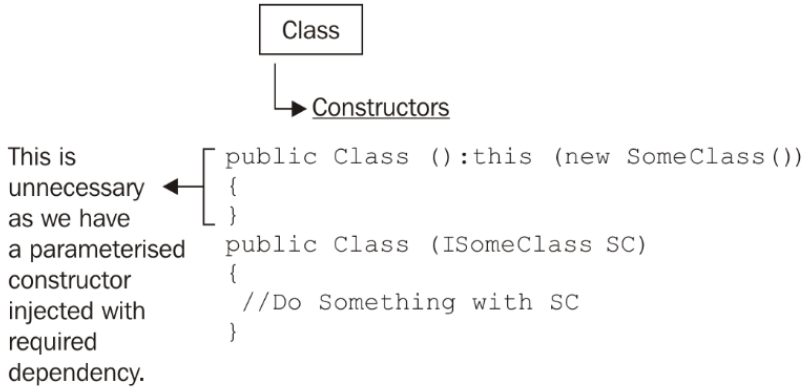



```
public void Demand()
{
    IPrincipal principal = Thread.CurrentPrincipal;
    if (principal == null)
        throw new SecurityException(SR.Security_PrincipalPermission);
    if (_idArray == null)
        return;
```

Chapter 9: Anti-Patterns and Misconceptions on Dependency Injection





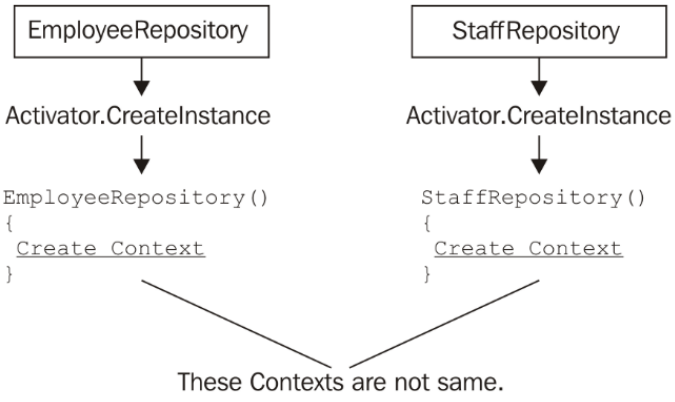


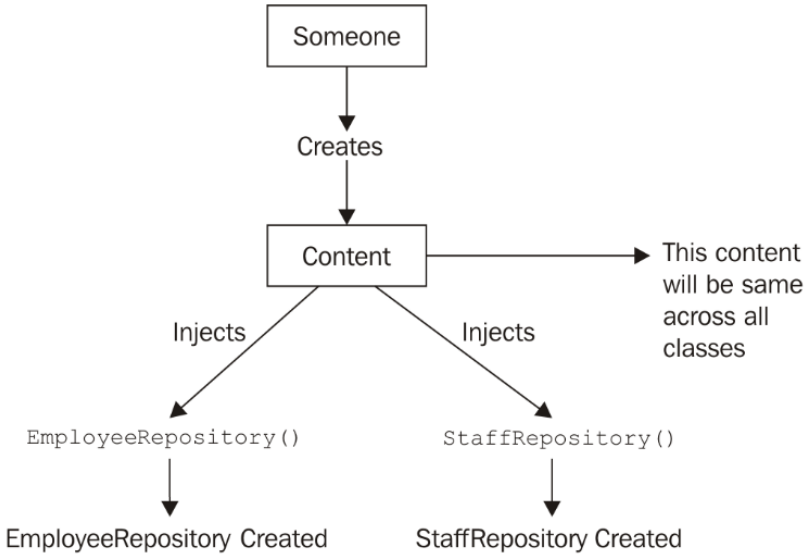
EmployeeRepository

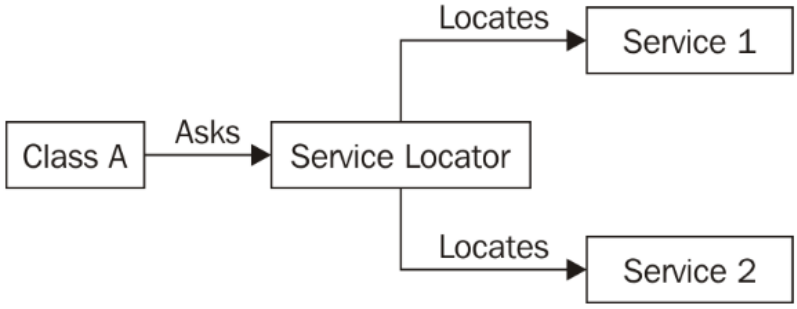
```
EmployeeRepository()
{
 Read connectionString from config
 Create (connectionString)
}
```

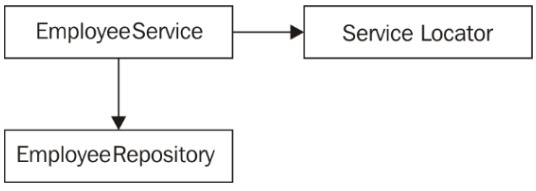
StaffRepository

```
StaffRepository()
{
 Read connectionString from config
 Create (connectionString)
}
```

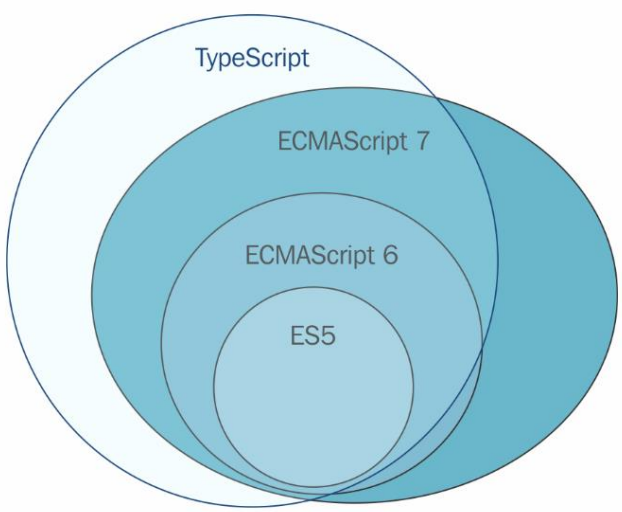



Someone
Creates
Content
This content will be same across all classes
Injects
Injects
EmployeeRepository()
StaffRepository()
EmployeeRepository Created
StaffRepository Created


Locates
Service 1
Class A
Asks
Service Locator
Locates
Service 2


EmployeeService
Service Locator
EmployeeRepository

Chapter 10: Dependency Injection in Other JavaScript Frameworks



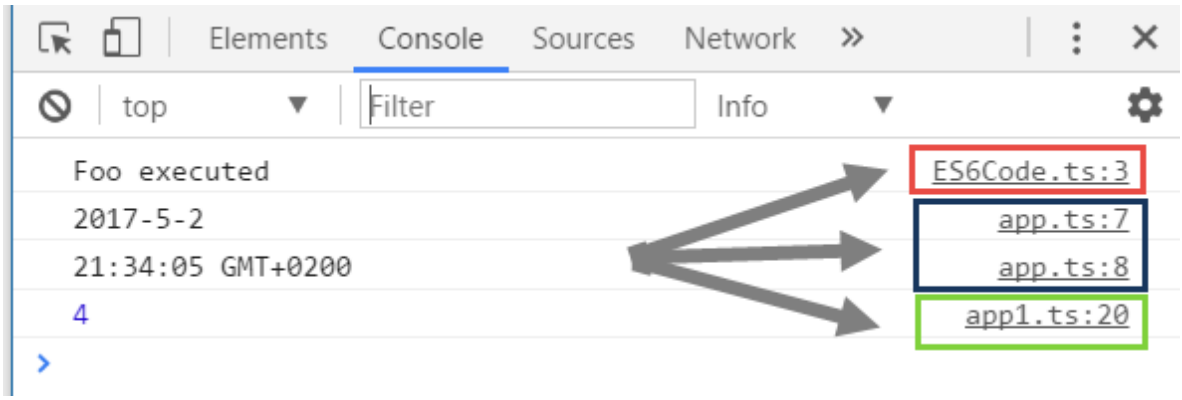

```
module MyClasses {
    export class ClassA {
        private x = "The String";
        public y = 4;
    };
    var ic = new ClassA();
}
ic.y = 9;  // Not recognized
```

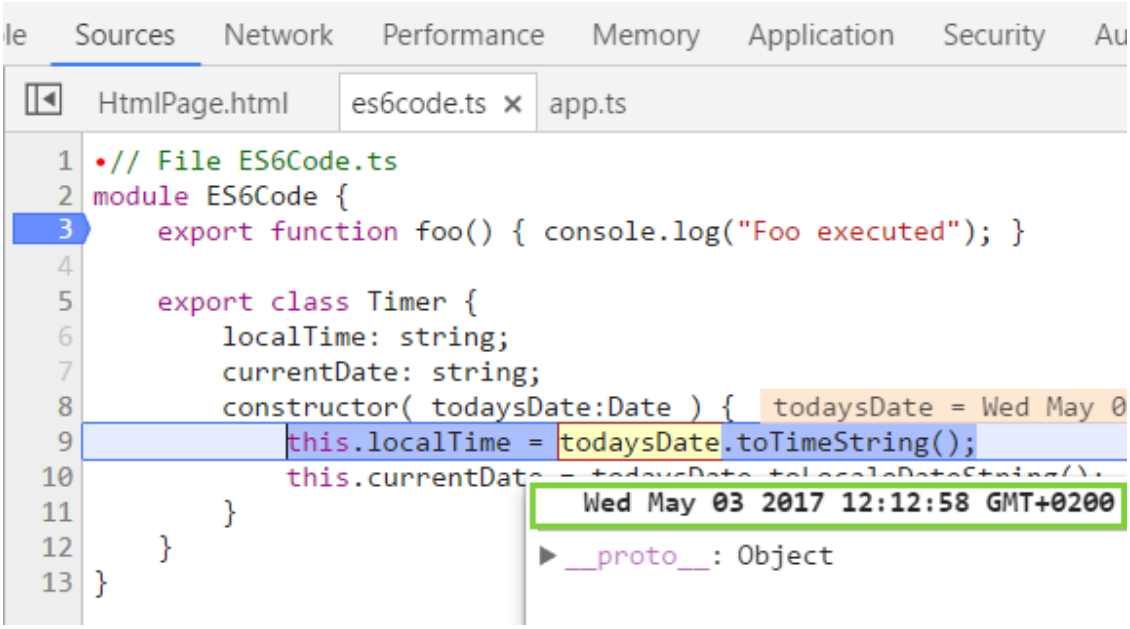
any

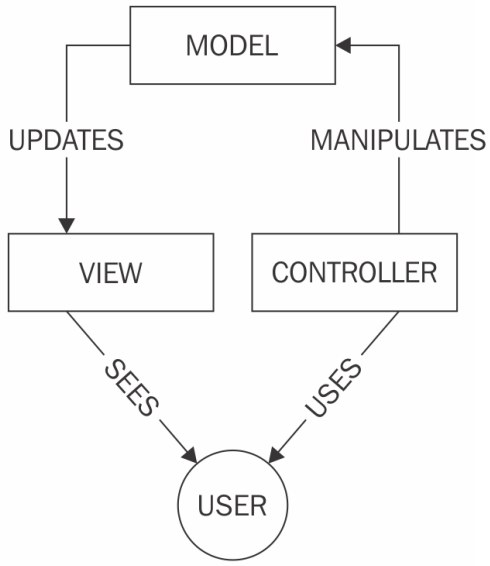
Cannot find name 'ic'.

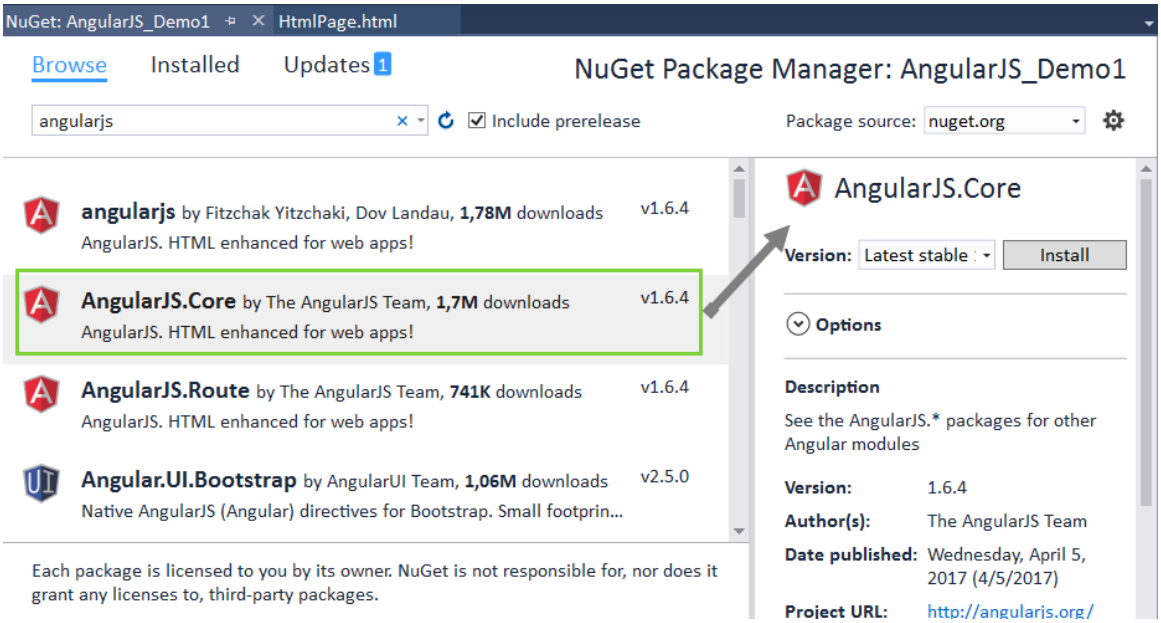
```
va                asses.ClassA();
co                ; // Logs 4 in the cons
```

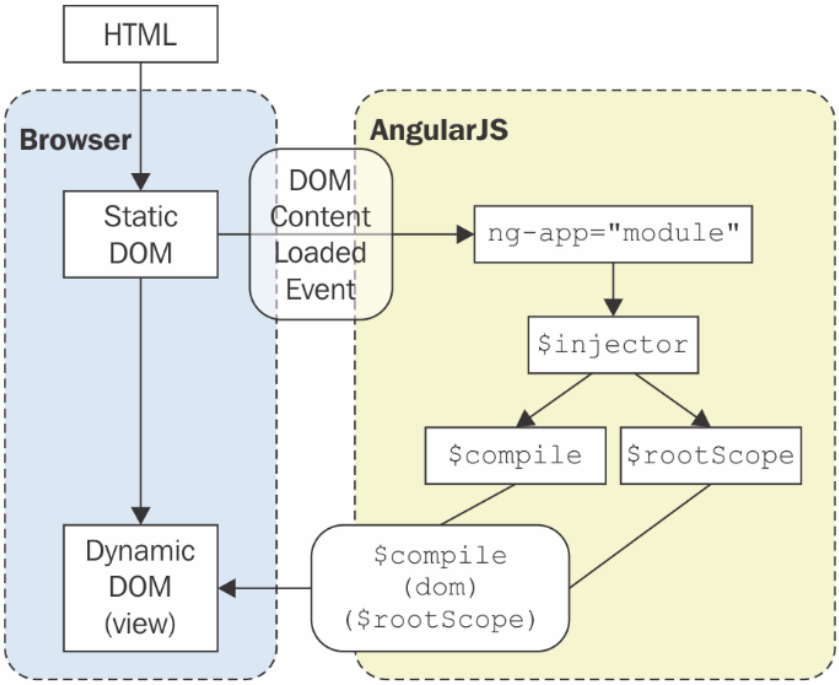
Elements
Console
Sources
top
Filter
Info
Foo executed
ES6Code.ts:3
2017-5-2
app.js:6
13:32:57 GMT+0200
app.js:7

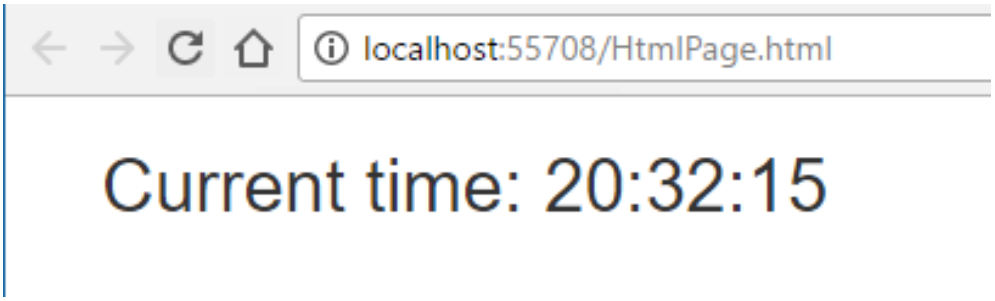

Elements
Console
Sources
Network
top
Filter
Info
Foo executed
ES6Code.ts:3
2017-5-2
app.ts:7
21:34:05 GMT+0200
app.ts:8
4
app1.ts:20

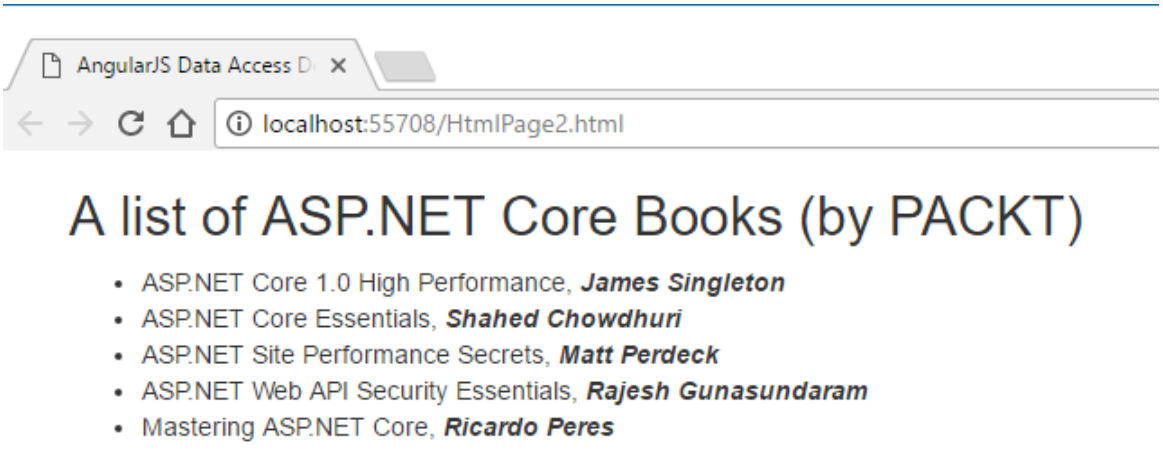

Sources
Network
Performance
Memory
Application
Security
HtmlPage.html
es6code.ts
app.ts
1 •// File ES6Code.ts
2 module ES6Code {
3 export function foo() { console.log("Foo executed"); }
4
5 export class Timer {
6 localTime: string;
7 currentDate: string;
8 constructor(todaysDate:Date) { todaysDate = Wed May 0
9 this.localTime = todaysDate.toTimeString();
10 this.currentDat
11 }
12 }
13 }
Wed May 03 2017 12:12:58 GMT+0200
▶ __proto__: Object

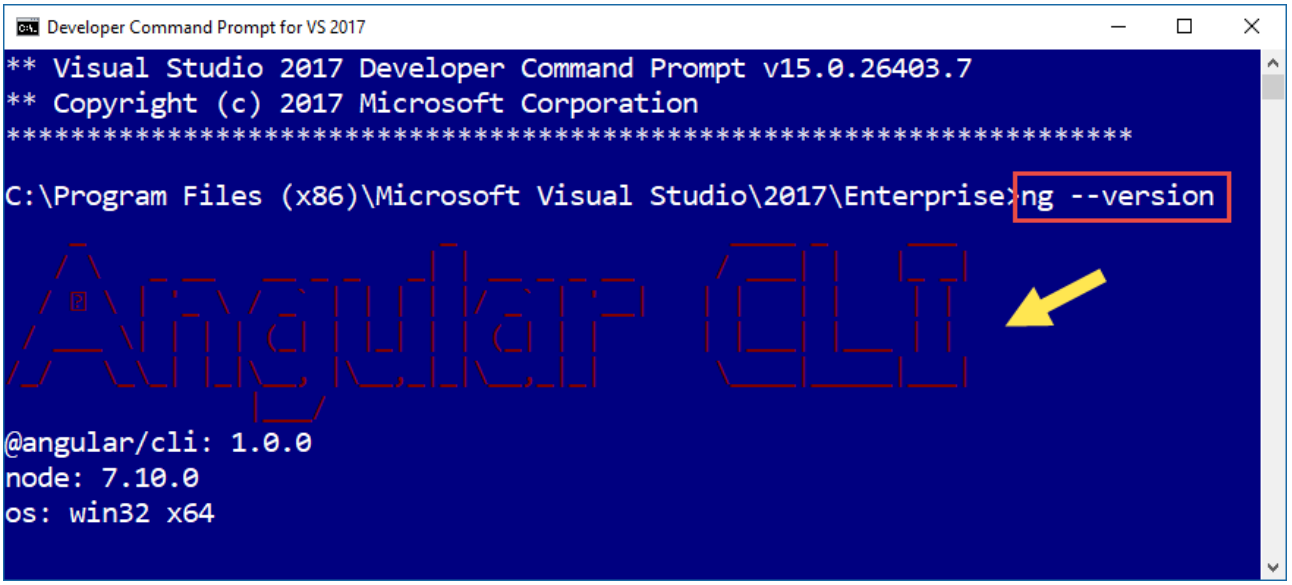

MODEL
UPDATES
MANIPULATES
VIEW
CONTROLLER
SEES
USES
USER

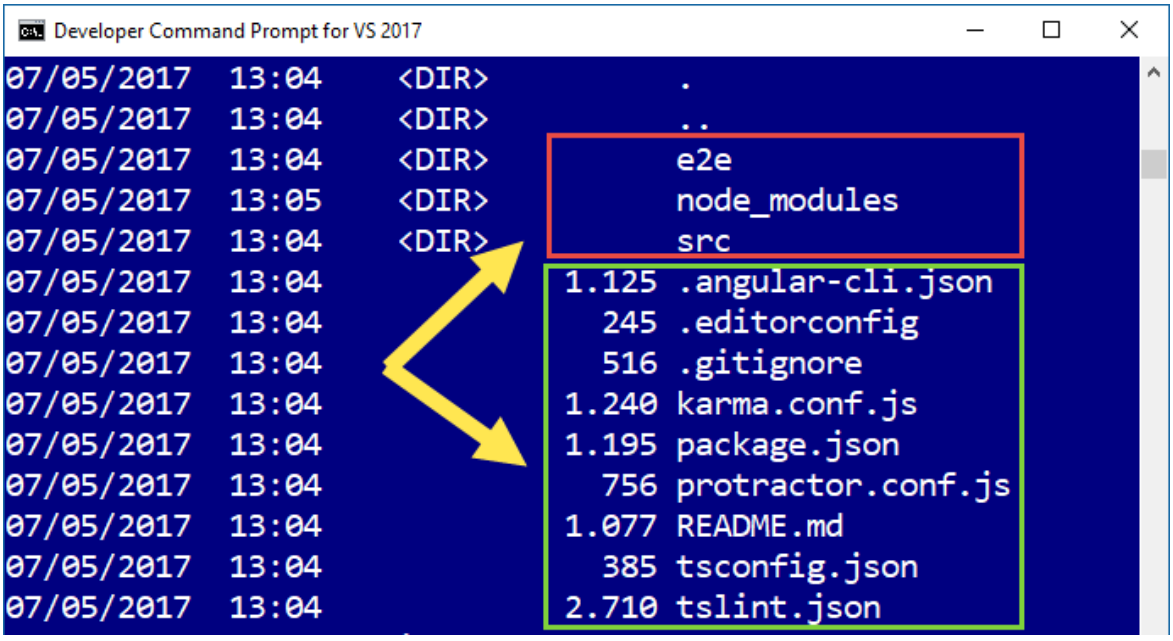

NuGet: AngularJS_Demo1
HtmlPage.html
Browse
Installed
Updates 1
NuGet Package Manager: AngularJS_Demo1
angularjs
Include prerelease
Package source: nuget.org
angularjs by Fitzchak Yitzchaki, Dov Landau, 1,78M downloads
v1.6.4
AngularJS. HTML enhanced for web apps!
AngularJS.Core by The AngularJS Team, 1,7M downloads
v1.6.4
AngularJS. HTML enhanced for web apps!
AngularJS.Route by The AngularJS Team, 741K downloads
v1.6.4
AngularJS. HTML enhanced for web apps!
Angular.UI.Bootstrap by AngularUI Team, 1,06M downloads
v2.5.0
Native AngularJS (Angular) directives for Bootstrap. Small footprin...
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
AngularJS.Core
Version: Latest stable
Install
Options
Description
See the AngularJS.* packages for other Angular modules
Version: 1.6.4
Author(s): The AngularJS Team
Date published: Wednesday, April 5, 2017 (4/5/2017)
Project URL: http://angularjs.org/


HTML
Browser
Static DOM
DOM Content Loaded Event
AngularJS
ng-app="module"
$injector
$compile
$rootScope
Dynamic DOM (view)
$compile (dom) ($rootScope)


localhost:55708/HtmlPage.html
Current time: 20:32:15


AngularJS Data Access D
localhost:55708/HtmlPage2.html
A list of ASP.NET Core Books (by PACKT)
ASP.NET Core 1.0 High Performance, James Singleton
ASP.NET Core Essentials, Shahed Chowdhuri
ASP.NET Site Performance Secrets, Matt Perdeck
ASP.NET Web API Security Essentials, Rajesh Gunasundaram
Mastering ASP.NET Core, Ricardo Peres


Developer Command Prompt for VS 2017
** Visual Studio 2017 Developer Command Prompt v15.0.26403.7
** Copyright (c) 2017 Microsoft Corporation
C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise>ng --version
@angular/cli: 1.0.0
node: 7.10.0
os: win32 x64


Developer Command Prompt for VS 2017
07/05/2017 13:04 <DIR> .
07/05/2017 13:04 <DIR> ..
07/05/2017 13:04 <DIR> e2e
07/05/2017 13:05 <DIR> node_modules
07/05/2017 13:04 <DIR> src
07/05/2017 13:04 1.125 .angular-cli.json
07/05/2017 13:04 245 .editorconfig
07/05/2017 13:04 516 .gitignore
07/05/2017 13:04 1.240 karma.conf.js
07/05/2017 13:04 1.195 package.json
07/05/2017 13:04 756 protractor.conf.js
07/05/2017 13:04 1.077 README.md
07/05/2017 13:04 385 tsconfig.json
07/05/2017 13:04 2.710 tslint.json

app.module.ts .angular-cli.json ×

```
  },
  "apps": [
    {
      "root": "src",
      "outDir": "dist",
      "assets": [
        "assets",
        "favicon.ico"
      ],
      "index": "index.html",
      "main": "main.ts",
      "polyfills": "polyfills.ts",
      "test": "test.ts",
      "tsconfig": "tsconfig.app.json",
      "testTsconfig": "tsconfig.spec.json",
```

app.module.ts main.ts ×

```
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule);
```

app.module.ts × app.component.ts

```ts
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { HttpModule } from '@angular/http';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [ AppComponent ],
  imports: [ BrowserModule, FormsModule, HttpModule ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

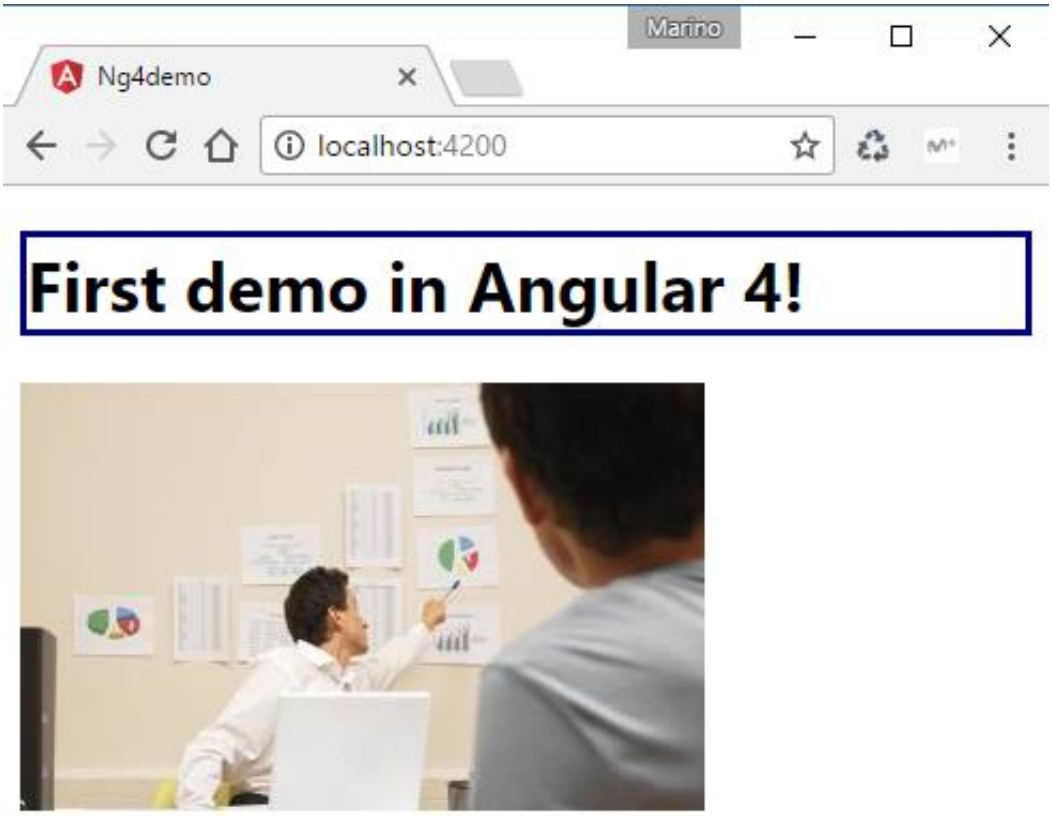
app.module.ts app.component.ts ×

```ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app works!';
}
```

Directive (links to Index.html)

HTML Template

CSS files or templates

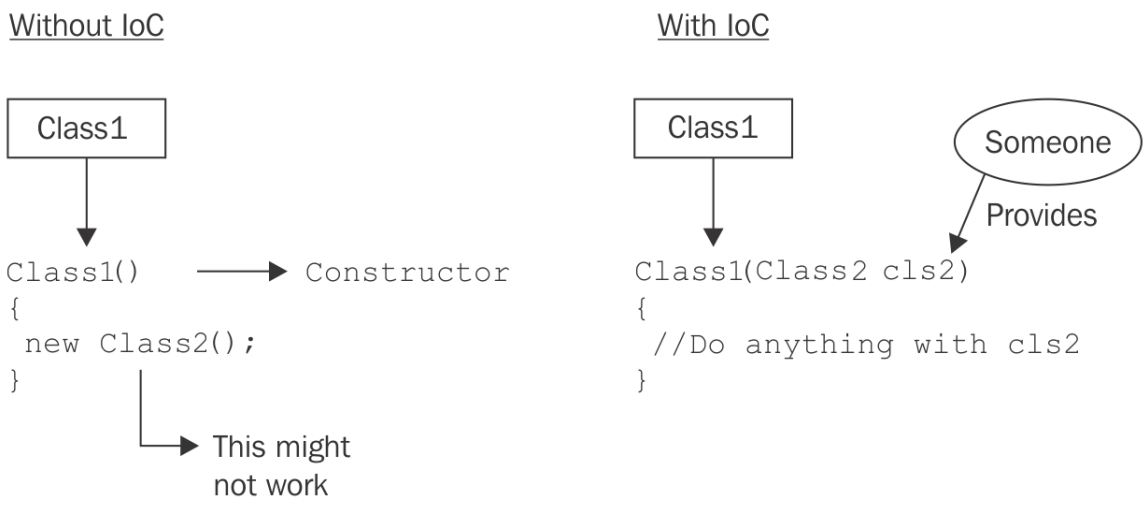


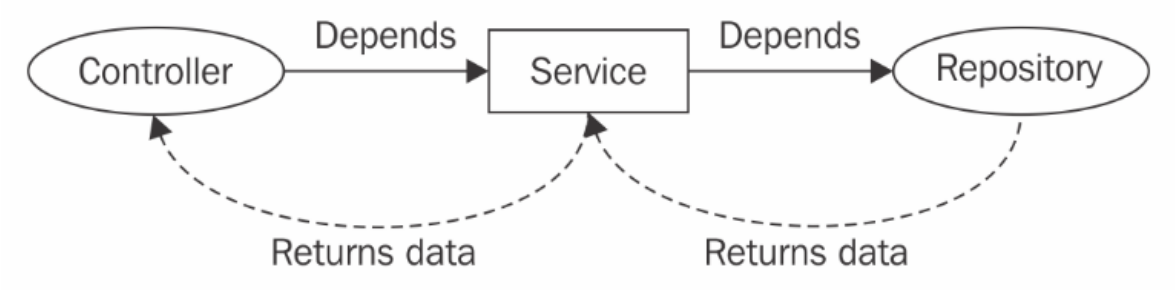

```typescript
import { Injectable } from "@angular/Core";

@Injectable()
export class DIClass {
    public DIProperty: string = "This block comes from an injected Class";
    public DIPicture: string = 'http://lorempixel.com/320/200/business';
}
```

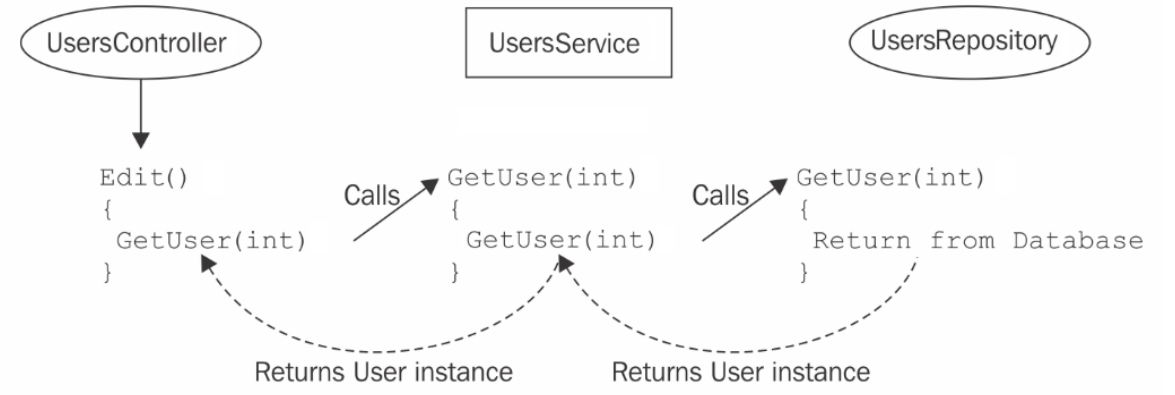
import { Component } from '@angular/core';
import { DIClass } from "./DIComponent/DIClass";
@Component({
selector: 'app-root',
providers: [DIClass],
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {
title = 'First demo in Angular 4!';
subtitle: string = "";
fotosource : string = "";
constructor(injectedValues: DIClass) {
this.subtitle = injectedValues.DIProperty;
this.fotosource = injectedValues.DIPicture;
}
}
3 steps for Injection

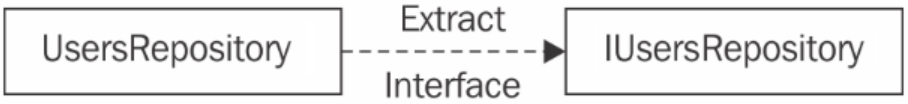
Ng4demo
localhost:4200
First demo in Angular 4!
This block comes from an injected Class

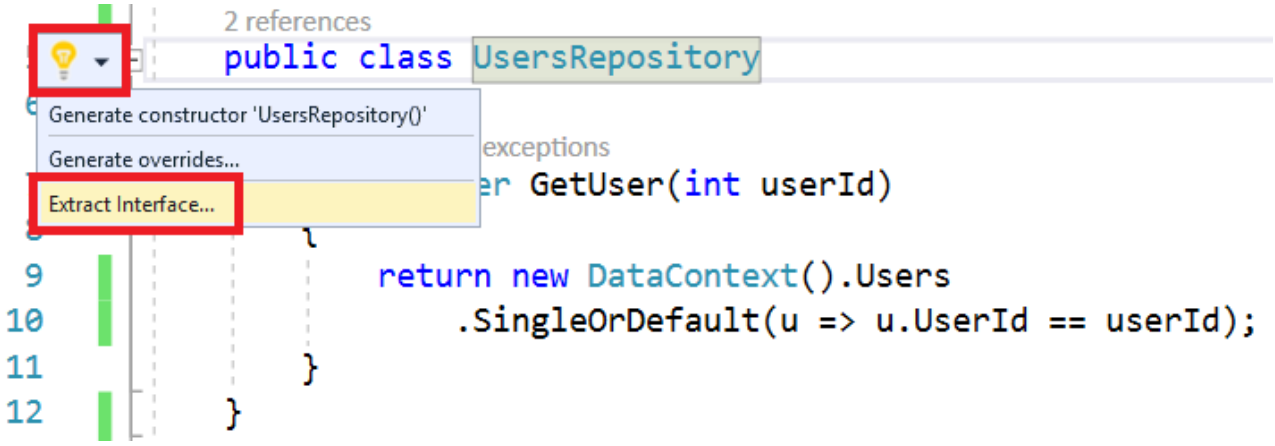
Chapter 11: Best Practices and Other Related Techniques

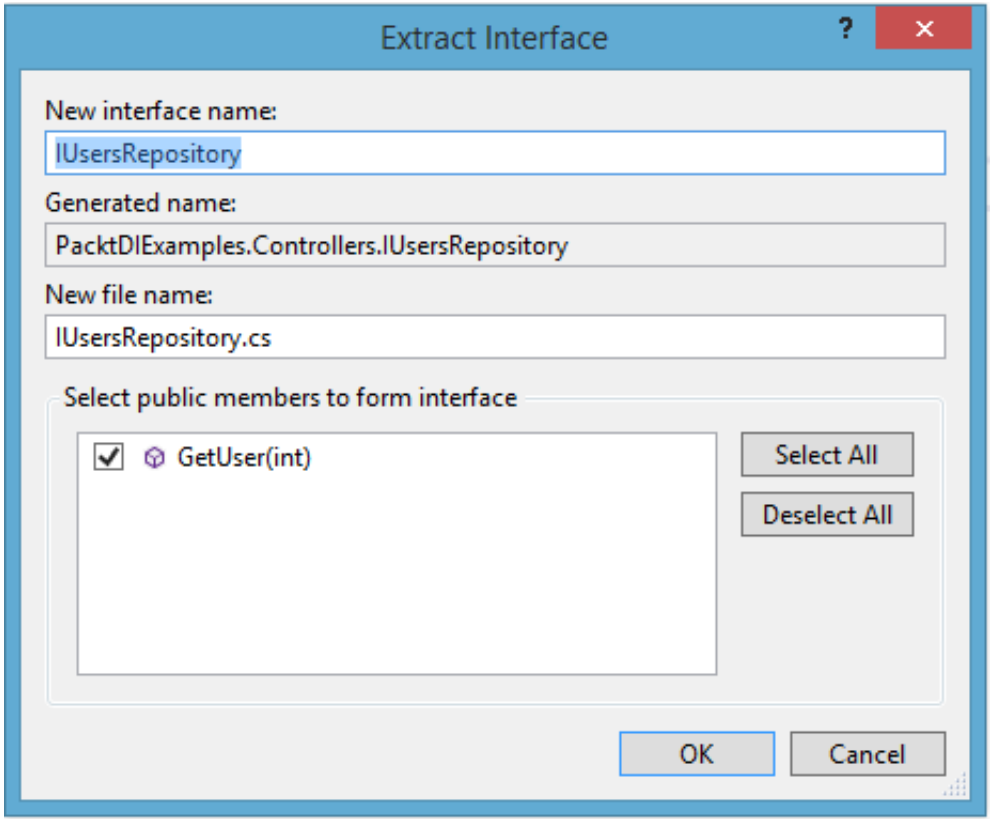


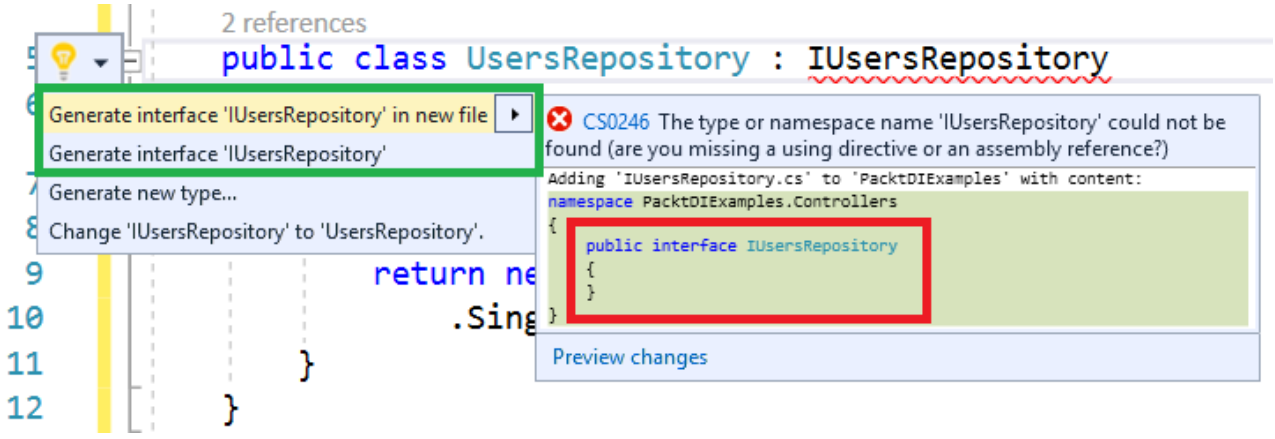

Controller
Depends
Service
Depends
Repository
Returns data
Returns data

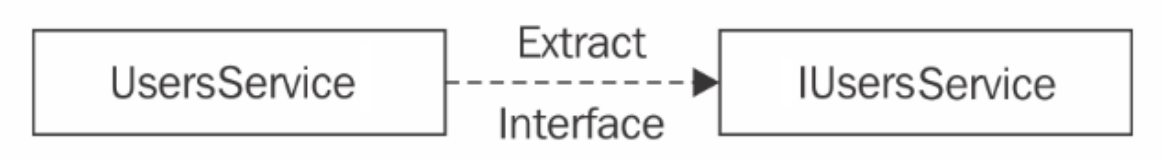

UsersController
UsersService
UsersRepository
Edit()
{
GetUser(int)
}
Calls
GetUser(int)
{
GetUser(int)
}
Calls
GetUser(int)
{
Return from Database
}
Returns User instance
Returns User instance


UsersRepository
Extract
Interface
IUsersRepository


2 references
public class UsersRepository
Generate constructor 'UsersRepository()'
Generate overrides...
Extract Interface...
exceptions
er GetUser(int userId)
{
return new DataContext().Users
.SingleOrDefault(u => u.UserId == userId);
}
}
9
10
11
12

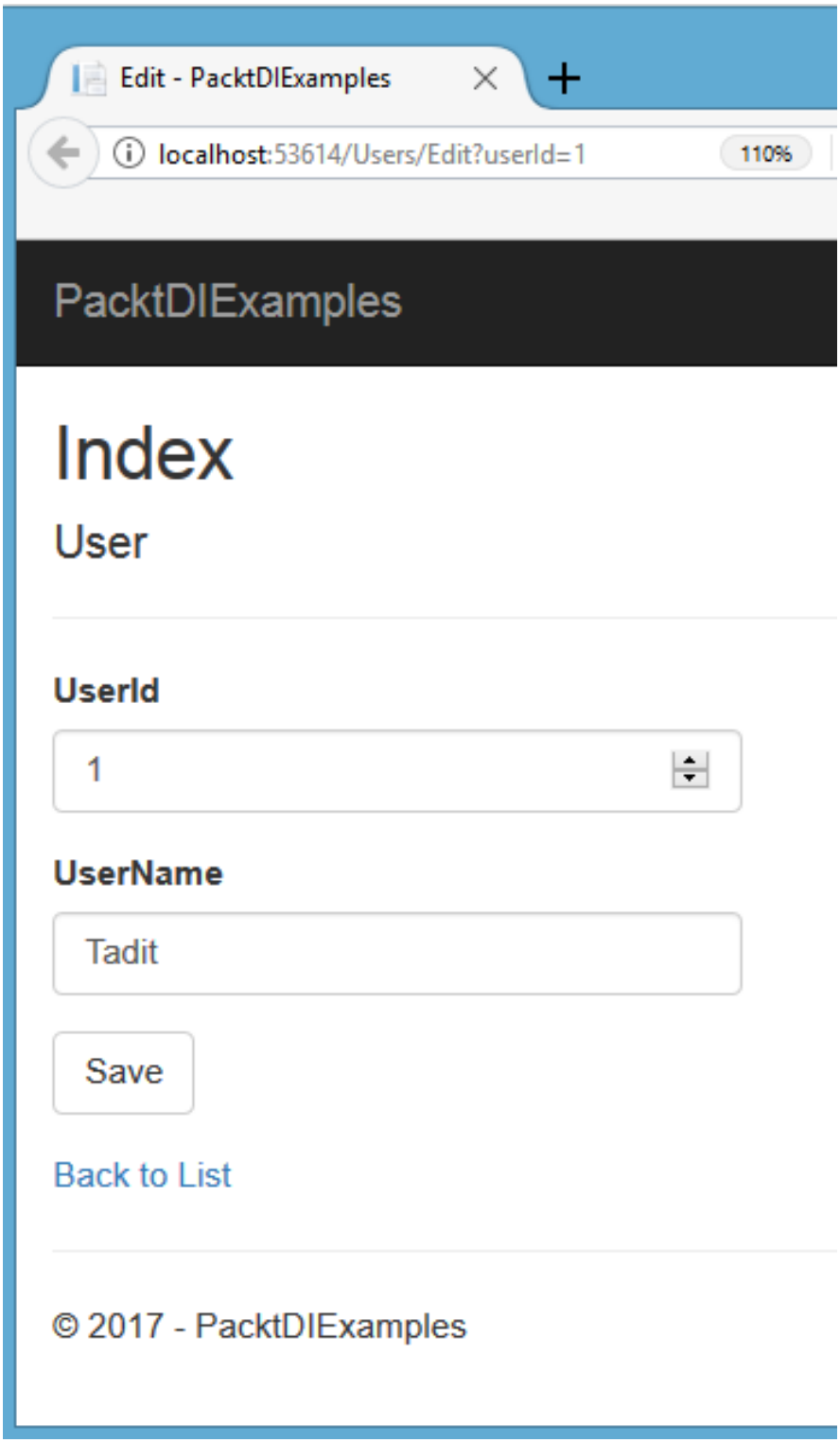

Extract Interface
New interface name:
IUsersRepository
Generated name:
PacktDIExamples.Controllers.IUsersRepository
New file name:
IUsersRepository.cs
Select public members to form interface
GetUser(int)
Select All
Deselect All
OK
Cancel


2 references
public class UsersRepository : IUsersRepository
Generate interface 'IUsersRepository' in new file
Generate interface 'IUsersRepository'
Generate new type...
Change 'IUsersRepository' to 'UsersRepository'.
CS0246 The type or namespace name 'IUsersRepository' could not be found (are you missing a using directive or an assembly reference?)
Adding 'IUsersRepository.cs' to 'PacktDIExamples' with content:
namespace PacktDIExamples.Controllers
{
public interface IUsersRepository
{
}
}
Preview changes
return ne
.Sing
}
}

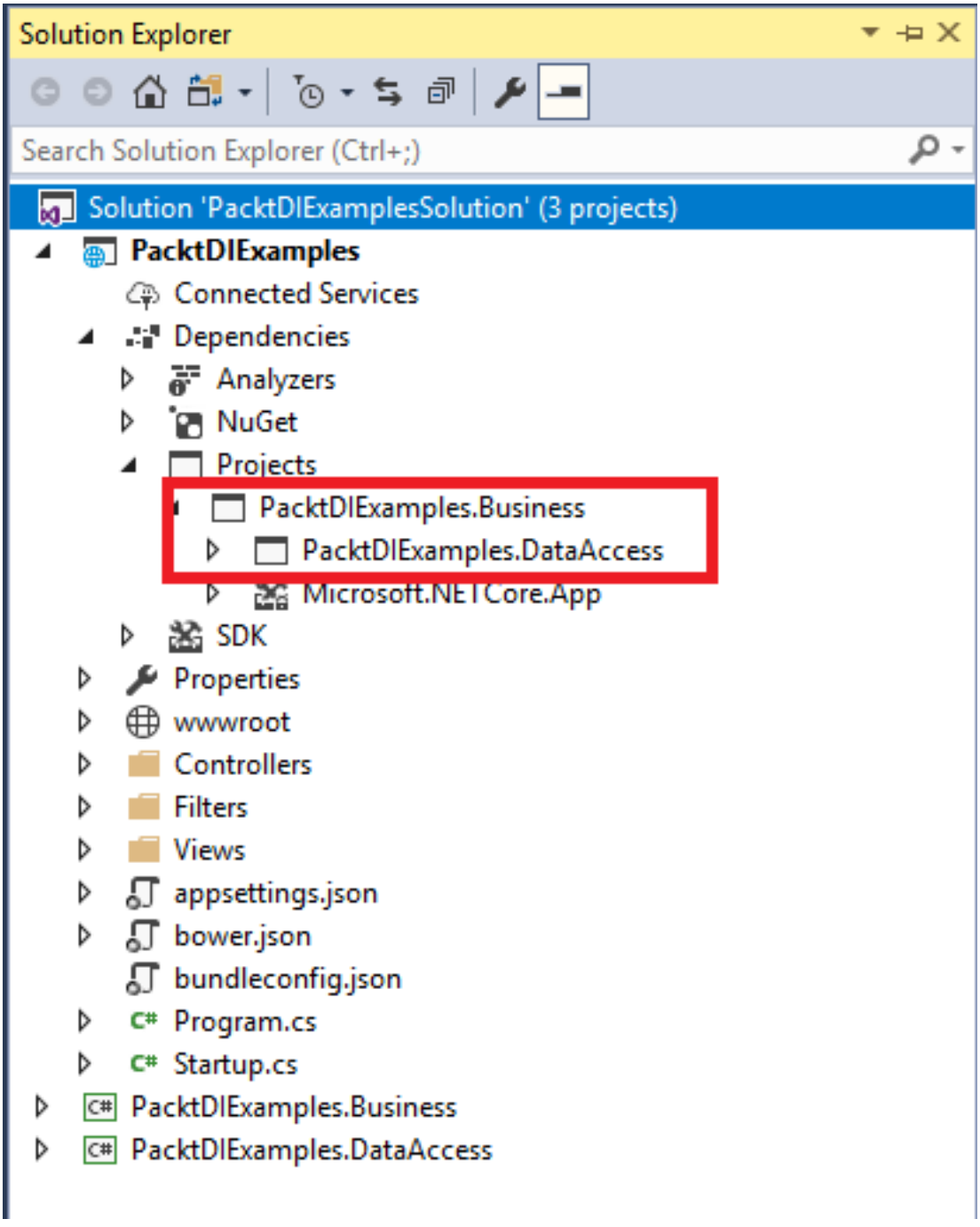


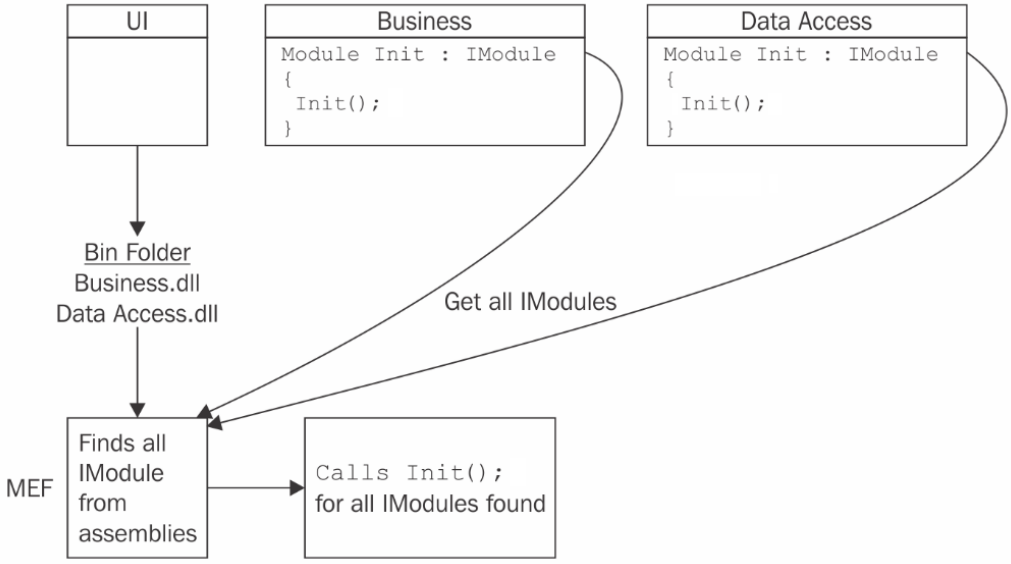
InvalidOperationException: Unable to resolve service for type 'PacktDIExamples.Controllers.UsersService' while attempting to activate 'PacktDIExamples.Controllers.UsersController'.

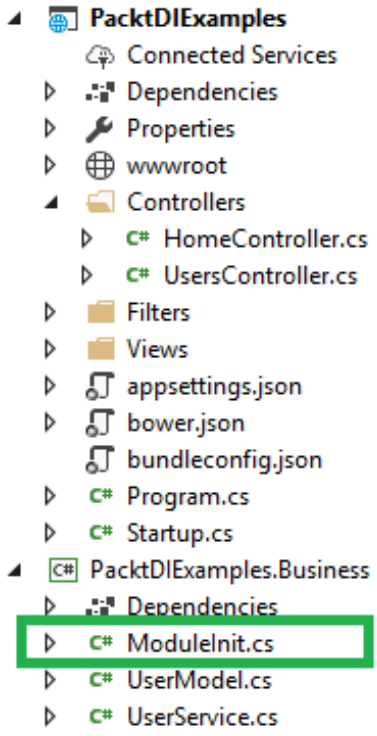
InvalidOperationException: Multiple constructors accepting all given argument types have been found in type 'PacktDIExamples.Controllers.UsersController'. There should only be one applicable constructor.

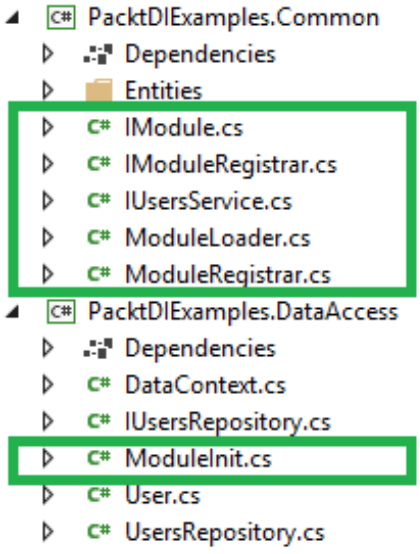

Edit - PacktDIExamples
localhost:53614/Users/Edit?userId=1
110%
PacktDIExamples
Index
User
UserId
1
UserName
Tadit
Save
Back to List
© 2017 - PacktDIExamples

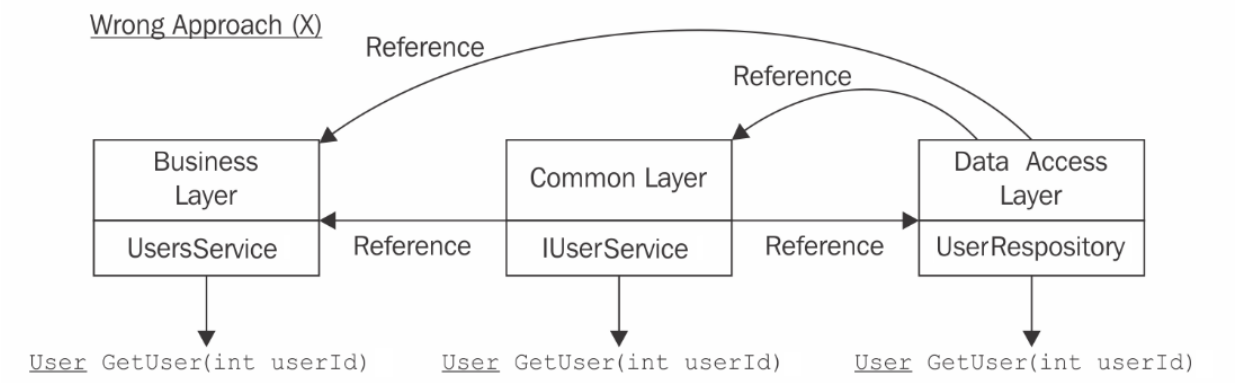
ASP.NET
MVC
Web App
Depends
on
Service/
Business
Layer
Depends
on
Data
Access
Layer

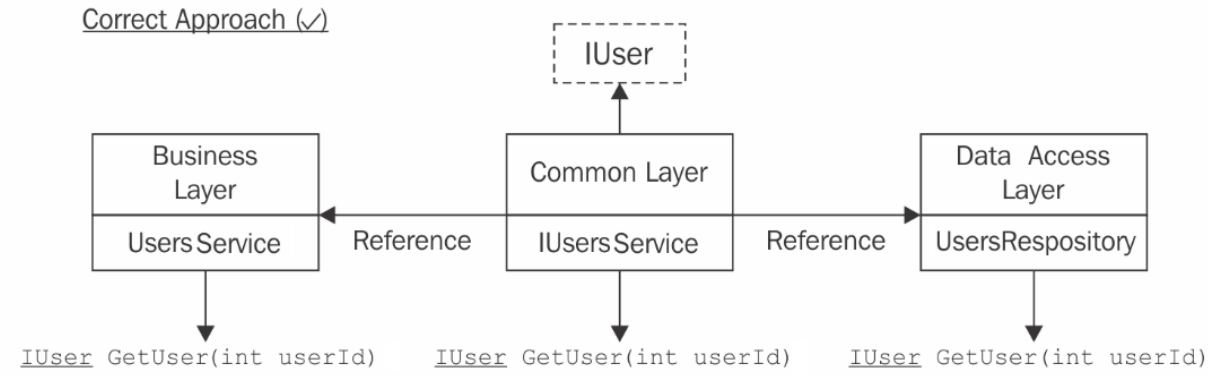

Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'PacktDIExamplesSolution' (3 projects)
PacktDIExamples
Connected Services
Dependencies
Analyzers
NuGet
Projects
PacktDIExamples.Business
PacktDIExamples.DataAccess
Microsoft.NETCore.App
SDK
Properties
wwwroot
Controllers
Filters
Views
appsettings.json
bower.json
bundleconfig.json
Program.cs
Startup.cs
PacktDIExamples.Business
PacktDIExamples.DataAccess

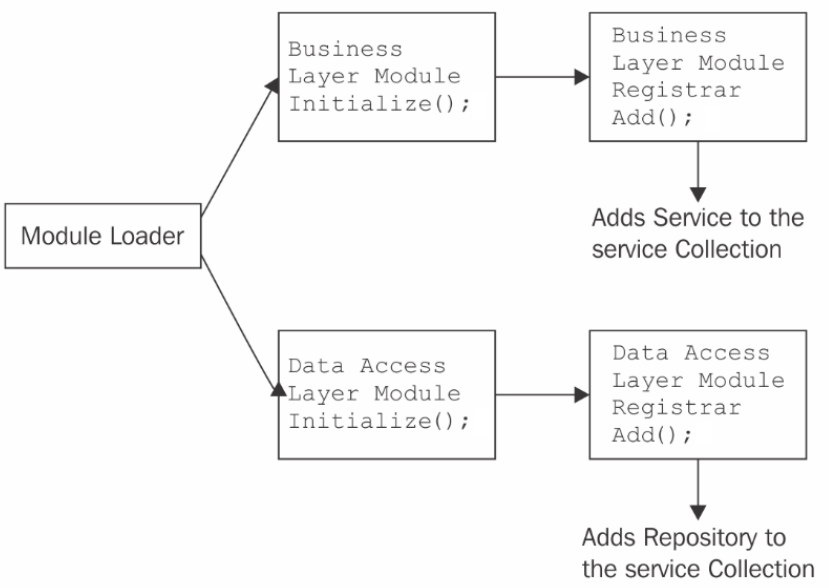

UI
Business
Module Init : IModule
{
Init();
}
Data Access
Module Init : IModule
{
Init();
}
Bin Folder
Business.dll
Data Access.dll
Get all IModules
MEF
Finds all IModule from assemblies
Calls Init(); for all IModules found

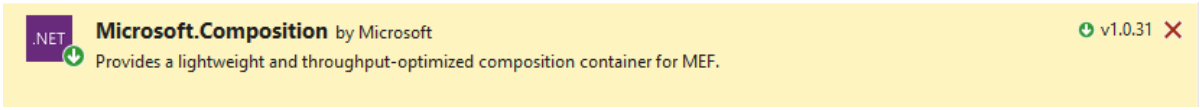

PacktDIExamples
Connected Services
Dependencies
Properties
wwwroot
Controllers
HomeController.cs
UsersController.cs
Filters
Views
appsettings.json
bower.json
bundleconfig.json
Program.cs
Startup.cs
PacktDIExamples.Business
Dependencies
ModuleInit.cs
UserModel.cs
UserService.cs


PacktDIExamples.Common
Dependencies
Entities
IModule.cs
IModuleRegistrar.cs
IUsersService.cs
ModuleLoader.cs
ModuleRegistrar.cs
PacktDIExamples.DataAccess
Dependencies
DataContext.cs
IUsersRepository.cs
ModuleInit.cs
User.cs
UsersRepository.cs


Wrong Approach (X)
Reference
Reference
Business Layer
UsersService
Reference
Common Layer
IUserService
Reference
Data Access Layer
UserRespository
User GetUser(int userId)
User GetUser(int userId)
User GetUser(int userId)

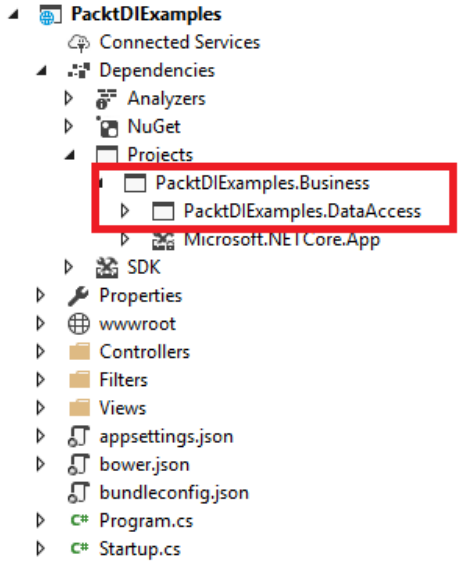

Correct Approach (✓)
IUser
Business Layer
Users Service
Reference
Common Layer
IUsers Service
Reference
Data Access Layer
UsersRespository
IUser GetUser(int userId)
IUser GetUser(int userId)
IUser GetUser(int userId)



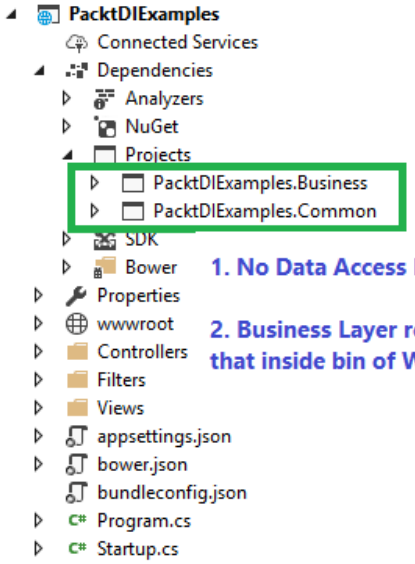



```
using (var container = configuration.CreateContainer())
{
    IEnumerable<IModule> modules = container.GetExports<IModule>();

    var registrar = new Module
    foreach (IModule module in
    {
        module.Initialize(registrar);
    }
}
```

Before MEF



After MEF



1. No Data Access Layer reference.

2. Business Layer reference is just to have that inside bin of Web Project.

An unhandled exception occurred while processing the request.

InvalidOperationException: Unable to resolve service for type 'PacktDIExamples.Common.IUsersService' while attempting to activate 'PacktDIExamples.Controllers.UsersController'.

Microsoft.Extensions.Internal.ActivatorUtilities.GetService(IServiceProvider sp, Type type, Type requiredBy, bool isDefaultParameterRequired)

Business Layer

```
2 references
internal class UsersService : IUsersService
```

Data Access Layer

```
2 references
internal class User : IUser

1 reference
internal class UsersRepository : IUsersRepository
```

```
public IActionResult Edit(int userId)
{
    UsersService service = new UsersService(new UsersRepository());
    re
}
```

The type or namespace name 'UsersService' could not be found (are you missing a using directive or an assembly reference?)

Show potential fixes (Alt+Enter or Ctrl+.)

```
0 references | 0 requests | 0 exceptions
public IActionResult Edit(int userId)
{
    UsersRepository repo = new UsersRepository();
    return View(repo.GetUser(userId));
}
```

```
0 references | 0 requests | 0 exceptions
public IActionResult Edit(int userId)
{
    return View(
                new DataContext()
                .Users
                .SingleOrDefault(u => u.UserId == userId)
            );
}
```