

# Configuring and Deploying Razor Pages

---



**Alex Wolf**

[www.crywolfcode.com](http://www.crywolfcode.com)



# Application Level Configurations and Services

## Dependency Injection

Configure and manage  
app services

## Middleware

Configure and manage  
the request pipeline

## Deployment

Host the app with the  
right configurations



These topics can be advanced,  
but are easy to get started with.



# Dependency Injection Overview

---



Components should  
depend on abstractions.



```
public class MenuWidget : ViewComponent
{
    private IMenuService _menuService;

    public MenuWidget(IMenuService menuService)
    {
        _menuService = menuService
    }
}
```

## Declaring Dependencies

**Classes usually declare dependencies through a constructor**

**A component called a Container is then responsible for supplying them**



```
public void ConfigureServices(IServiceCollection services)
{
    services.AddScoped<IMenuService, MenuService>();
    services.AddScoped<IMessageService, MessageService>();
}
```

## Dependency Injection Containers

**Containers provide instances of registered dependencies**

**Containers provide various configuration options to better manage those dependencies**



# Providing Dependencies

Razor Page  
Model

```
public MenuModel(IMenuService inventoryService) { }
```

Hey, I depend  
on this contract

Okay, here is an  
implementation

Dependency  
Injection  
Container

```
services.AddScoped<IMenuService, MenuService>();
```



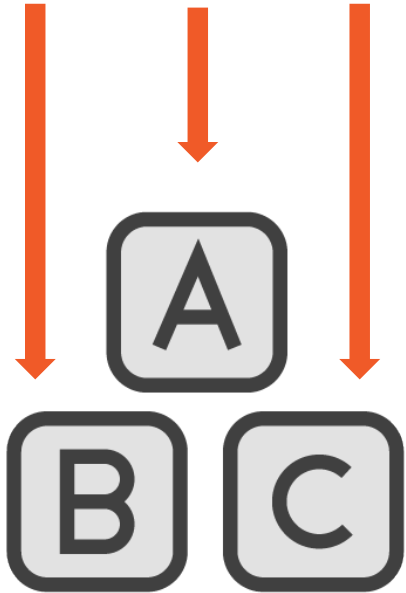


# Why dependency injection?



# Benefits of Dependency Injection

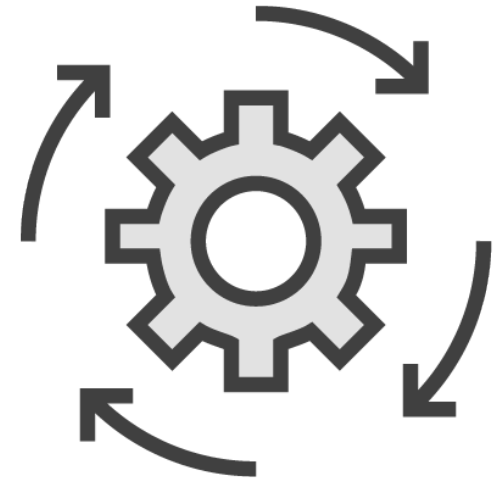
Loose Coupling



Improved Testing



Service Lifetime Management



# Demo



## Demo: Enhancing Components with Dependency Injection



# Demo



**Demo: Accessing services using  
Dependency Injection**



# Understanding Middleware

---



# Middleware

The series of components that form the application level request pipeline.



# Features Provided by Middleware

Routing

Session

Logging

Authentication

Caching



```
public void Configure(IApplicationBuilder app)
{
    app.UseStaticFiles();

    app.UseMvc();
}
```

## Registering Middleware

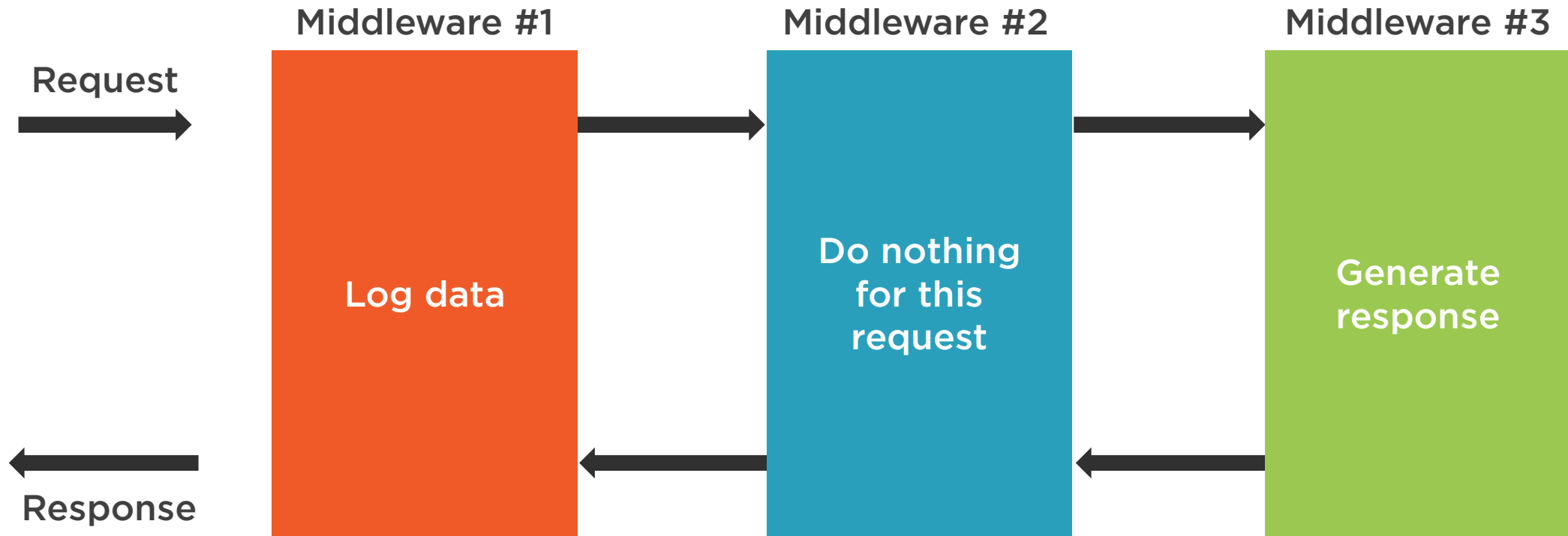
**Middleware is added in the Configure method of Startup.cs**

**Generally applied through wrapper helper methods**

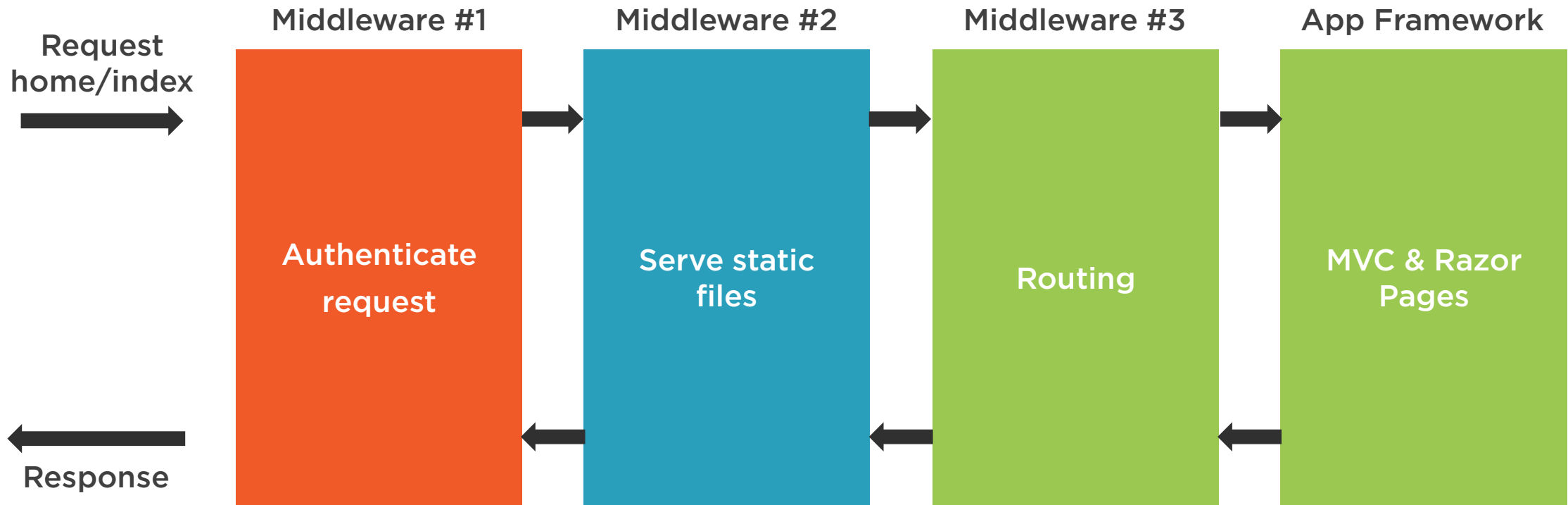




# A Simple Middleware Pipeline



# A Real World Middleware Pipeline



# A Case for Middleware



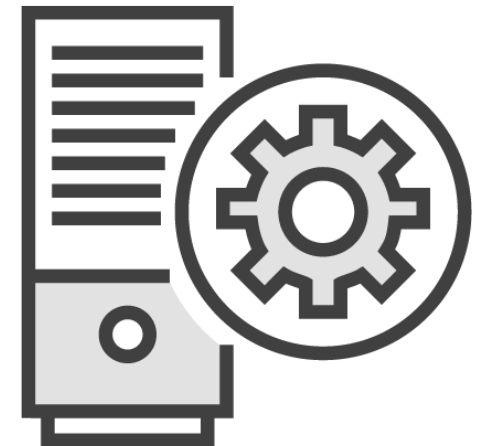
Browser

HTTP Request

Are you there?

HTTP Response

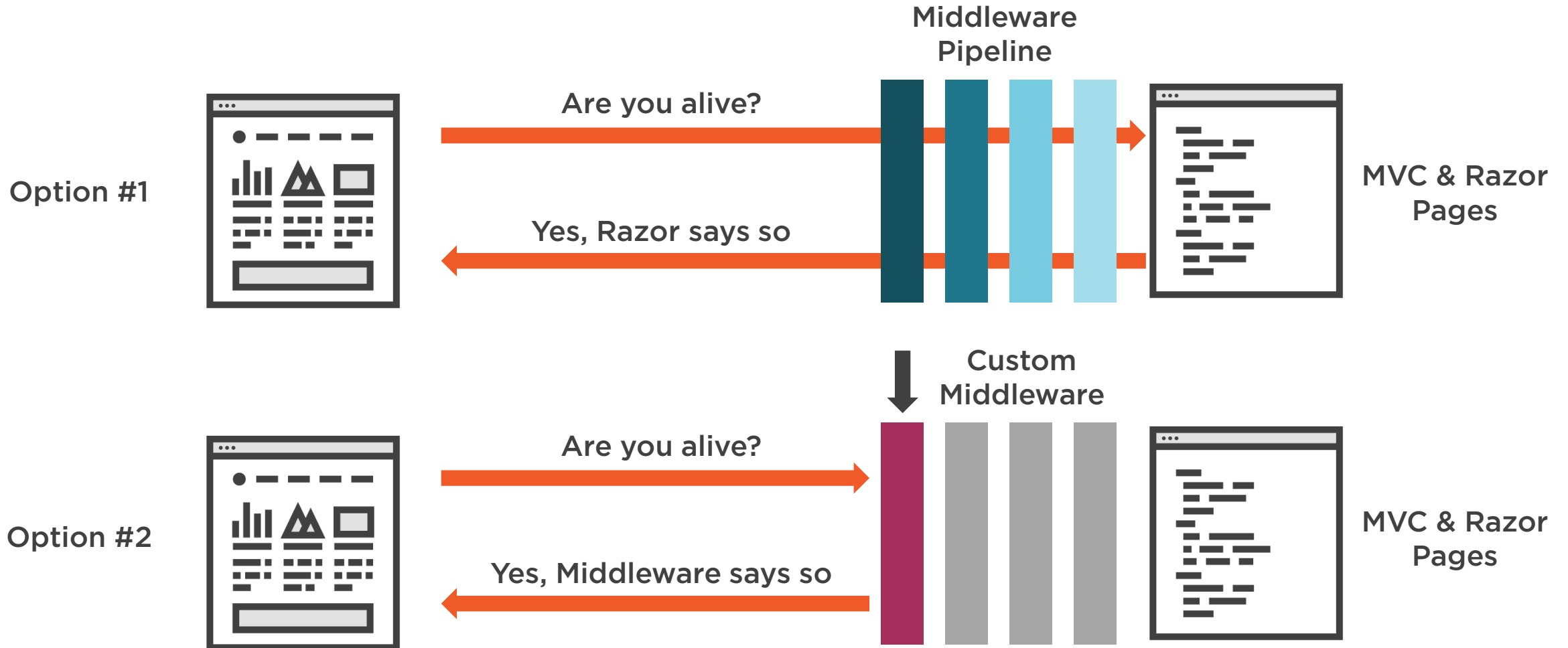
Yes, I'm here



Web Application



# Optimizing Responses with Middleware



# Demo



## Demo: Working with Middleware



# Application Deployment Basics with Azure

---



# Why Azure?



# Example Azure Services

Cloud Storage

Virtual Machines

Function Apps

Databases

Active Directory



App Services

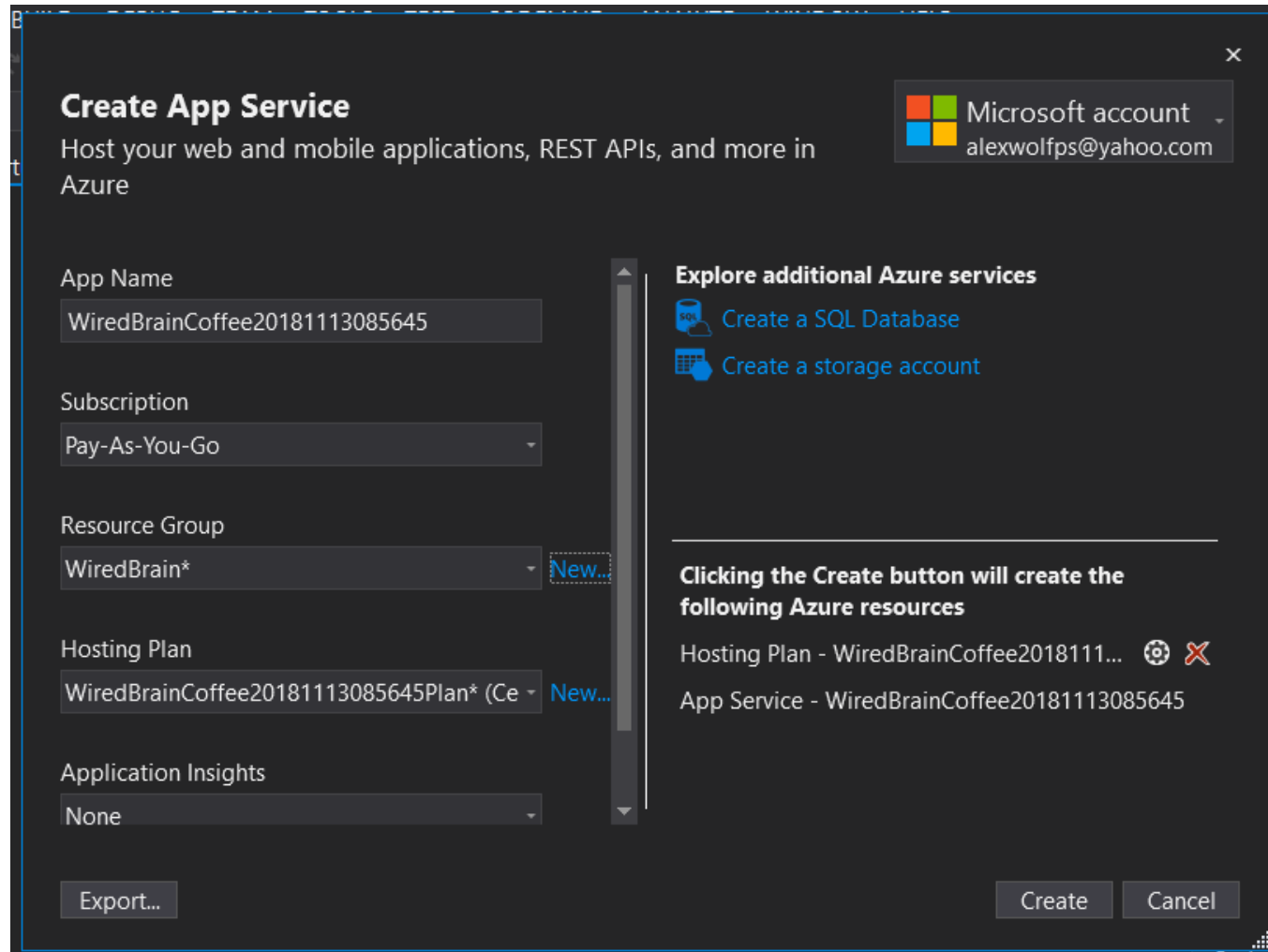




# A Basic Azure Deployment Workflow



# Visual Studio Integration with Azure



The screenshot shows the 'Create App Service' dialog box in Visual Studio. The dialog has a dark theme. At the top left, the title 'Create App Service' is followed by the text 'Host your web and mobile applications, REST APIs, and more in Azure'. In the top right corner, there is a Microsoft account dropdown menu showing 'alexwolfps@yahoo.com'. The main area is divided into two columns. The left column contains five dropdown menus: 'App Name' (set to 'WiredBrainCoffee20181113085645'), 'Subscription' (set to 'Pay-As-You-Go'), 'Resource Group' (set to 'WiredBrain\*' with a 'New...' link), 'Hosting Plan' (set to 'WiredBrainCoffee20181113085645Plan\* (Ce...' with a 'New...' link), and 'Application Insights' (set to 'None'). The right column has a section titled 'Explore additional Azure services' with two links: 'Create a SQL Database' and 'Create a storage account'. Below this is a section titled 'Clicking the Create button will create the following Azure resources' which lists 'Hosting Plan - WiredBrainCoffee2018111...' and 'App Service - WiredBrainCoffee20181113085645'. At the bottom, there are three buttons: 'Export...', 'Create', and 'Cancel'.

**Create App Service**

Host your web and mobile applications, REST APIs, and more in Azure

Microsoft account  
alexwolfps@yahoo.com

App Name  
WiredBrainCoffee20181113085645

Subscription  
Pay-As-You-Go

Resource Group  
WiredBrain\* [New...](#)

Hosting Plan  
WiredBrainCoffee20181113085645Plan\* (Ce... [New...](#)

Application Insights  
None

[Export...](#) [Create](#) [Cancel](#)

**Explore additional Azure services**

- [Create a SQL Database](#)
- [Create a storage account](#)

**Clicking the Create button will create the following Azure resources**

- Hosting Plan - WiredBrainCoffee2018111...
- App Service - WiredBrainCoffee20181113085645

Many Azure features can be utilized through the Visual Studio UI



# Hosting and Deployment Project Files

## Publishing Profile

Configures publishing and deployment location settings

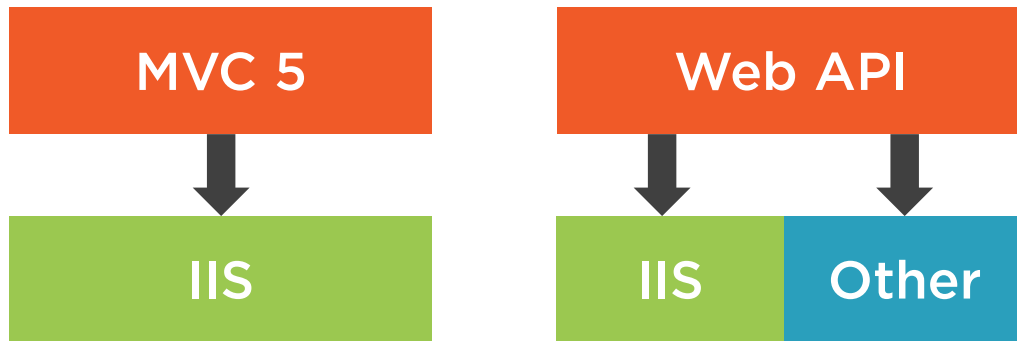
## Launch Settings

Configures environment and web server specific settings

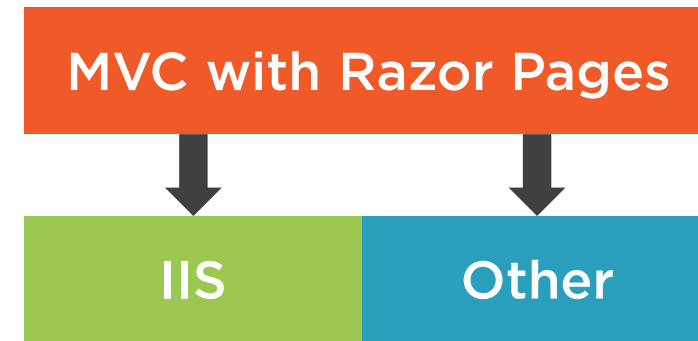


# Razor Pages Hosting Options

## ASP.NET 4.x (pre-Razor Pages)



## ASP.NET Core



# Demo



## Demo: Deploying the Site to Azure



## Summary



Dependency injection allows for inversion of control between components

Inversion of control allows for more loosely coupled, better managed components

Middleware forms the application request pipeline around Razor Pages

Middleware runs before and after Razor Pages, in the order they are registered

Visual Studio features strong Azure integration to make deployments easy

Azure hosts apps inside of App Services, which belong to a Resource Group



Thank you, and good luck!

