

Setting up an MVC Application



Gill Cleeren

CTO XPIRIT BELGIUM

@gillcleeren www.snowball.be



Overview



Exploring the project structure

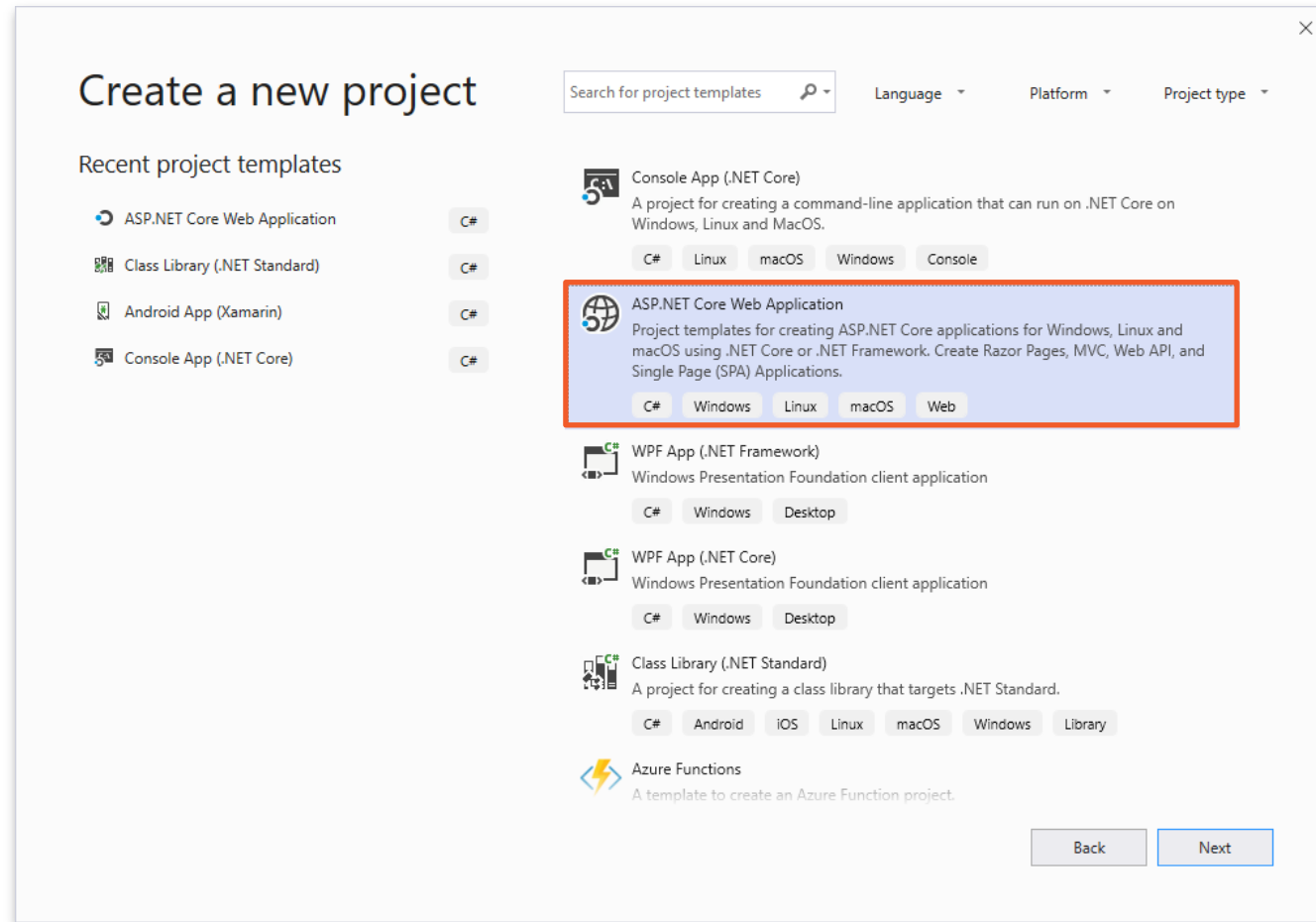
Configuring the site



Exploring the Project Structure



Creating a New Project




Available Project Templates


Create a new ASP.NET Core Web Application

.NET Core


ASP.NET Core 3.0

 **Empty**


An empty project template for creating an ASP.NET Core application. This template does not have any content in it.

 **API**


A project template for creating an ASP.NET Core application with an example Controller for a RESTful HTTP service. This template can also be used for ASP.NET Core MVC Views and Controllers.

 **Worker Service**


An empty project template for creating a worker service.

 **Web Application**

A project template for creating an ASP.NET Core application with example ASP.NET Core Razor Pages content.

 **Web Application (Model-View-Controller)**

A project template for creating an ASP.NET Core application with example ASP.NET Core MVC Views and Controllers. This template can also be used for RESTful HTTP services.

 **gRPC Service**

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced


☒ Configure for HTTPS

☐ Enable Docker Support
(Requires Docker Desktop)

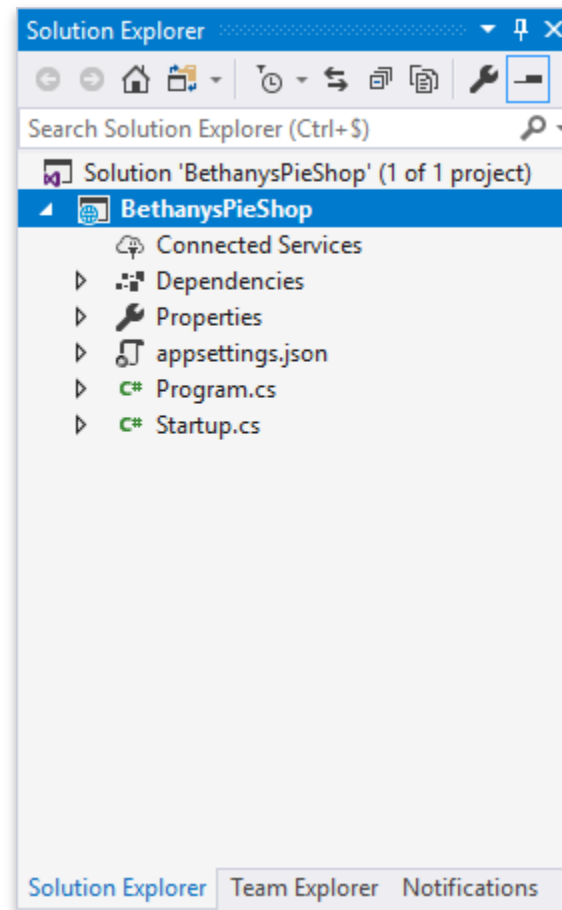
Linux

Back

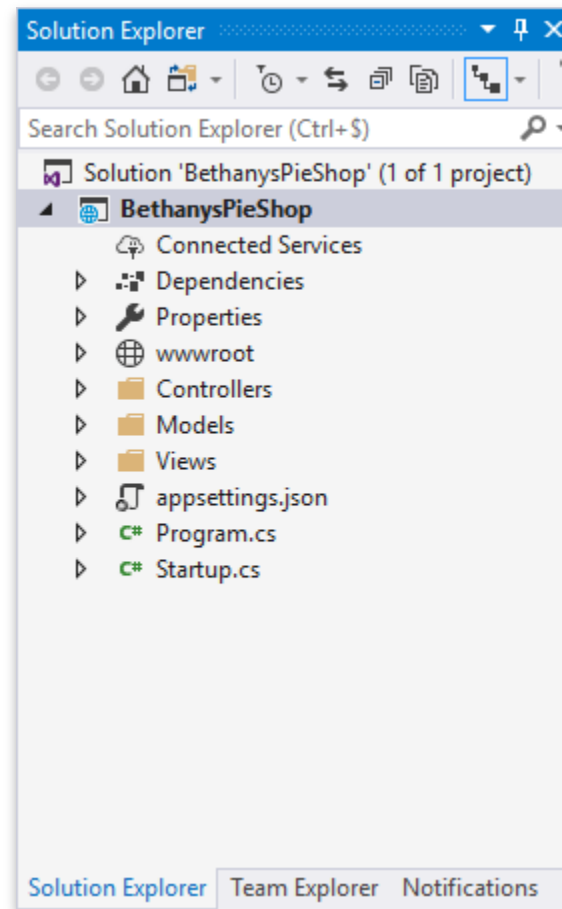
Create



Project Structure (Empty)



Project Structure (Web Application MVC)



Demo



File → New Project

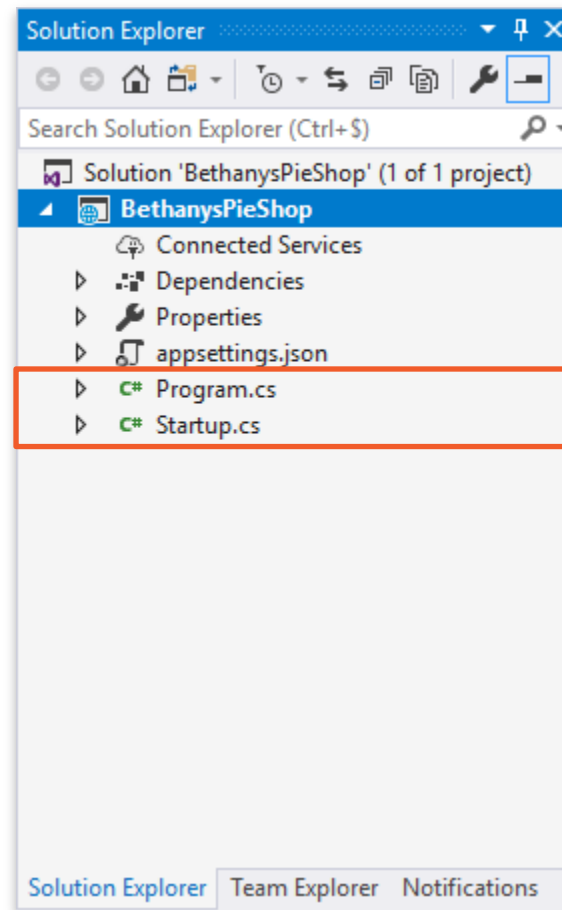
Exploring the new project



Site Configuration



Project Structure



Program.cs

```
public class Program
{
    public static void Main(string[] args)
    {
        CreateHostBuilder(args).Build().Run();
    }

    public static IHostBuilder CreateHostBuilder(string[] args) =>
        Host.CreateDefaultBuilder(args)
            .ConfigureWebHostDefaults(webBuilder =>
            {
                webBuilder.UseStartup<Startup>();
            });
}
```



Program.cs

```
public class Program
{
    public static void Main(string[] args)
    {
        CreateWebHostBuilder(args).Build().Run();
    }

    public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
        WebHost.CreateDefaultBuilder(args)
            .UseStartup<Startup>();
}
```



Program.cs

```
public class Program
{
    public static void Main(string[] args)
    {
        CreateWebHostBuilder(args).Build().Run();
    }

    public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>
        WebHost.CreateDefaultBuilder(args)
            .UseStartup<Startup>();
}
```



Startup.cs

```
public class Startup
```

```
{
```

```
    public void ConfigureServices(IServiceCollection services)
```

```
    {
```

```
        ...
```

```
    }
```

```
    public void Configure(IApplicationBuilder app,  
        IHostingEnvironment env)
```

```
    {
```

```
        ...
```

```
    }
```

```
}
```



```
public void ConfigureServices(IServiceCollection services)
{
    //register services here through Dependency Injection
}
```

ConfigureServices Method



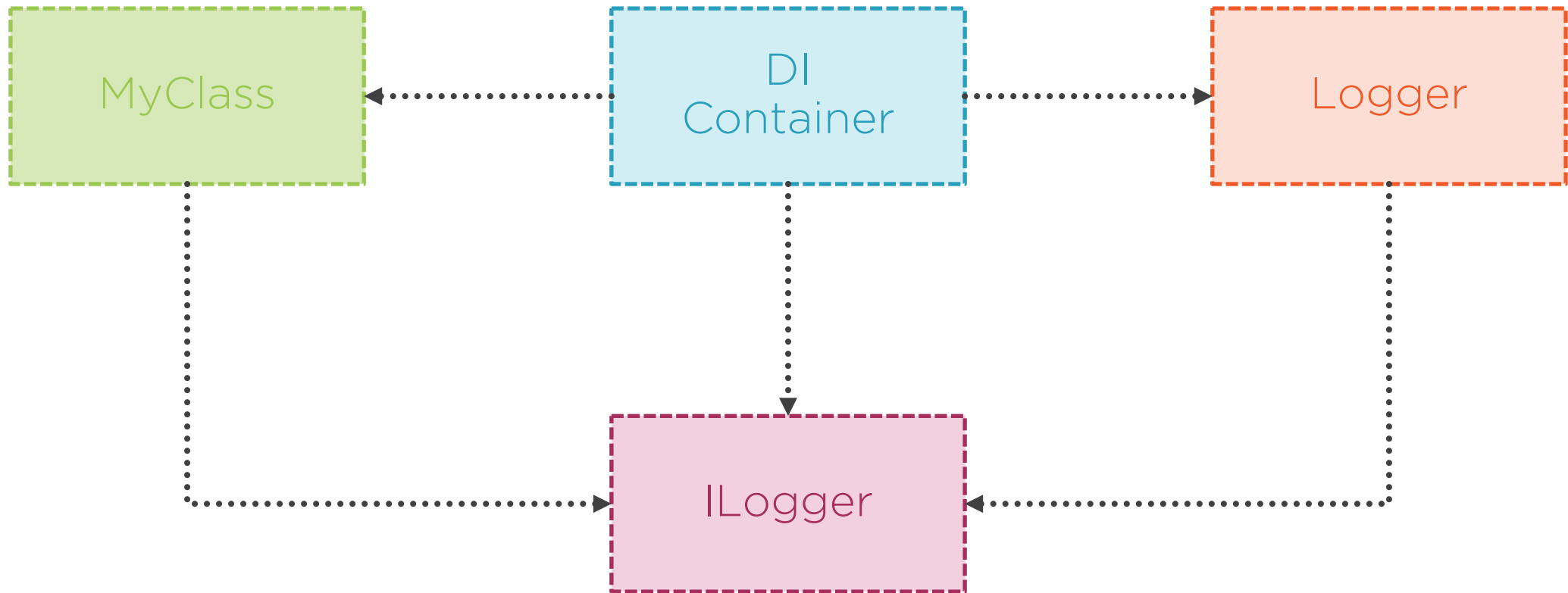
Tight Coupling



Adding Interfaces to the Mix



Adding Interfaces to the Mix



```
public void ConfigureServices(IServiceCollection services)
{
    //register framework services
    services.AddControllersWithViews();

    //register our own services (more later)
}
```

ConfigureServices Method

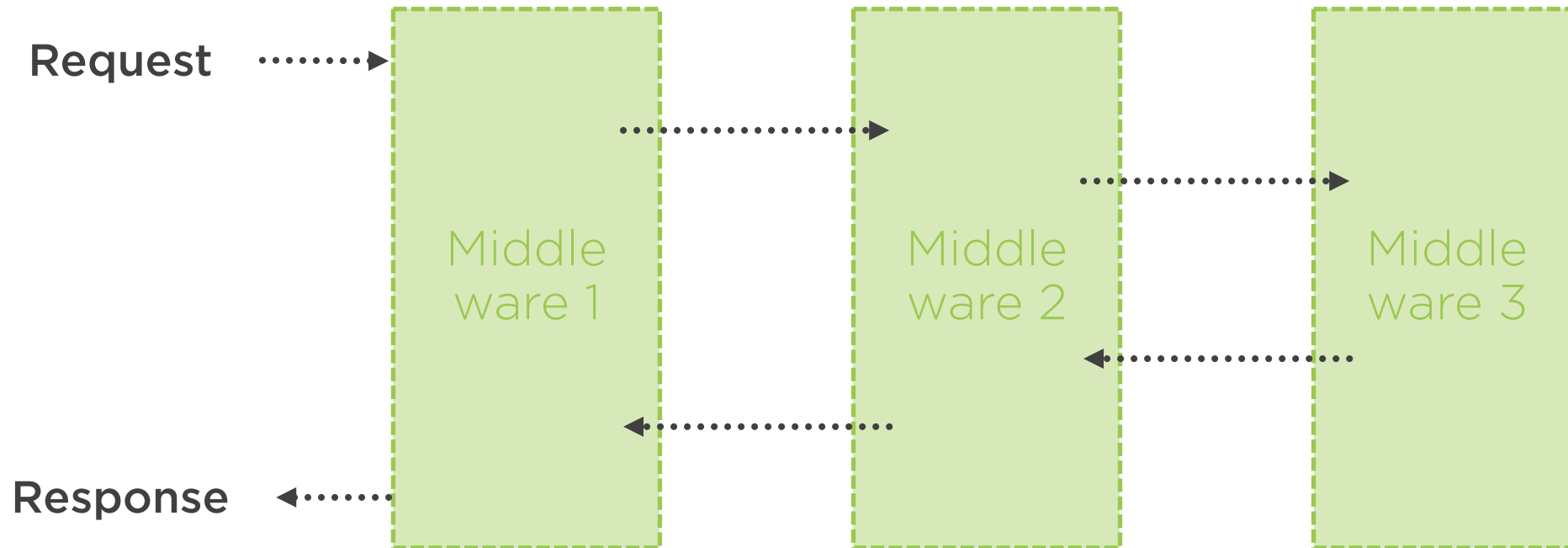


Configure Method

```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    //add middleware components here
}
```



Middleware Request Pipeline

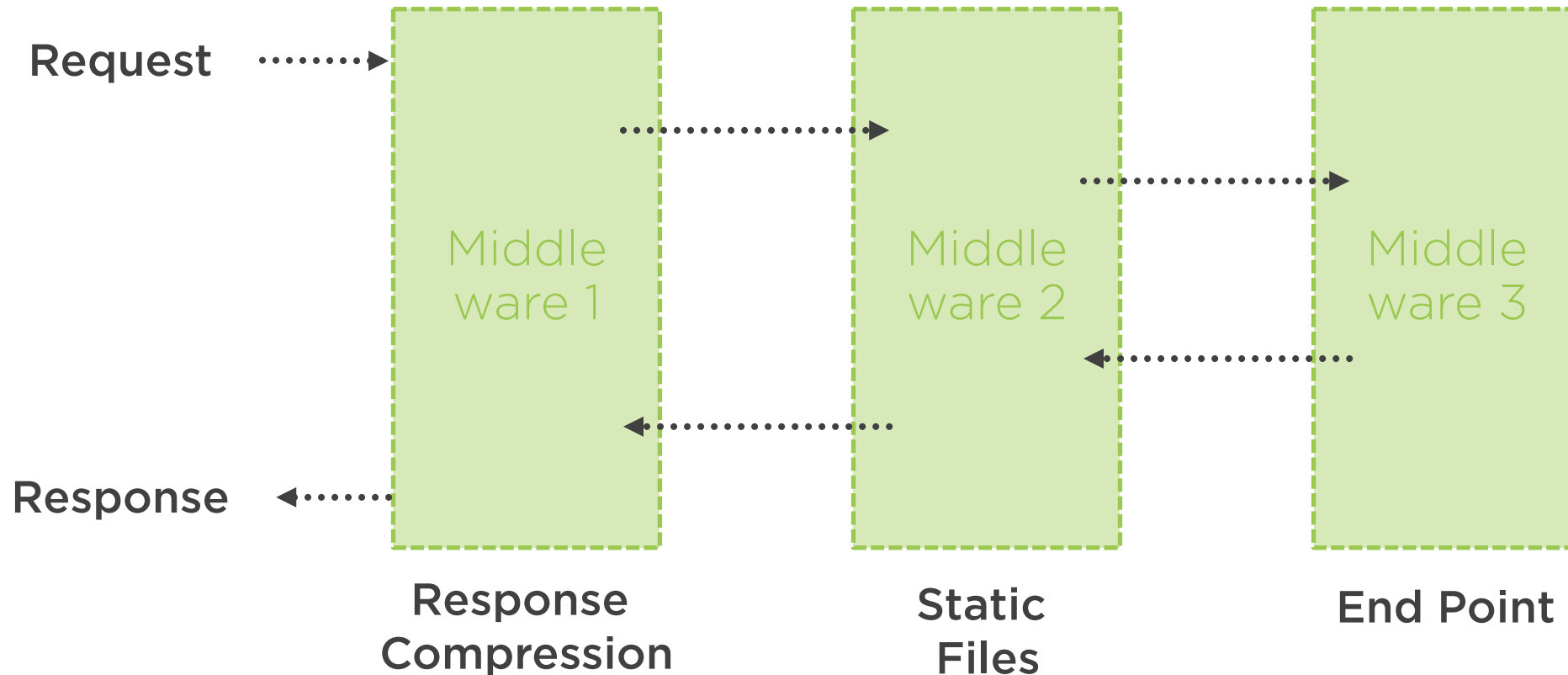


Configure Method

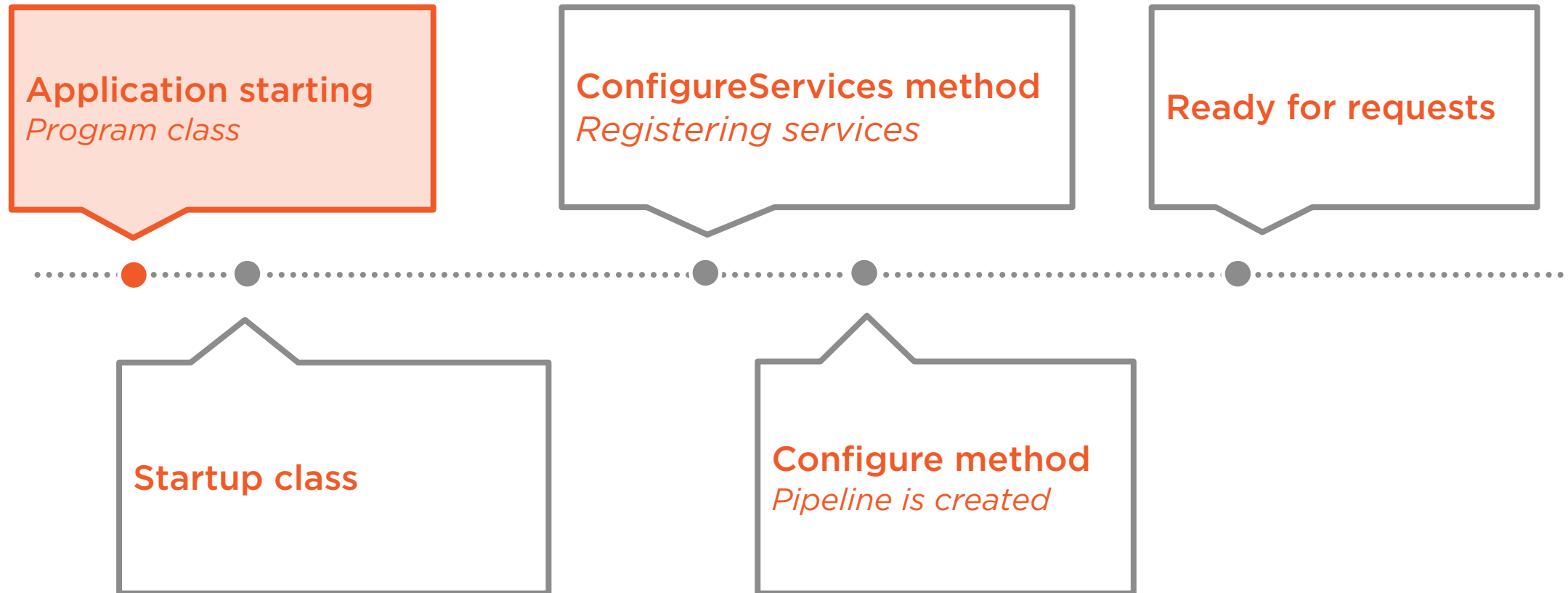
```
public void Configure(IApplicationBuilder app, IHostingEnvironment env)
{
    app.UseDeveloperExceptionPage();
    app.UseStatusCodePages();
    app.UseStaticFiles();
    app.UseEndpoints(endpoints =>
    { ... });
}
```



Ordering Middleware Components



The Startup of the Application



Demo



Configuring the site



Summary



ASP.NET Core MVC applications have a new project structure

Dependency injection is built-in

Middleware components





Up next:
Creating the first page using MVC

