

Setting Up an MVC Application

Overview



- Exploring the project structure
- Configuring the site

Using VS Code and CLI

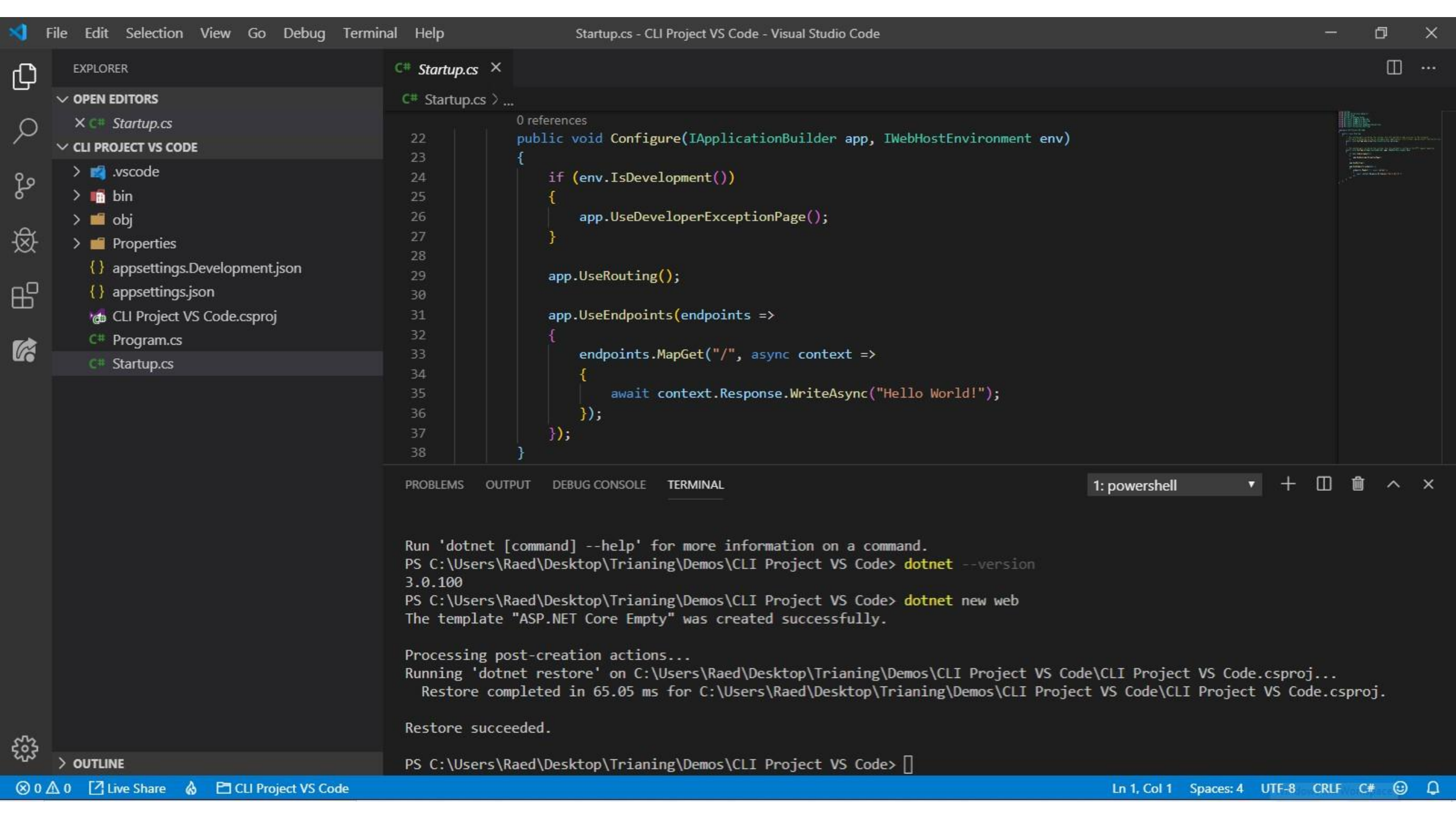
```
C:\Users\Raed\Desktop\Trianing\Demos\CLI Project>dotnet new
```

```
Usage: new [options]
```

Options:

-h, --help	Displays help for this command.
-l, --list	Lists templates containing the specified name. If no name is specified, lists all templates.
-n, --name	The name for the output being created. If no name is specified, the name of the current directory is used.
-o, --output	Location to place the generated output.
-i, --install	Installs a source or a template pack.
-u, --uninstall	Uninstalls a source or a template pack.
--nuget-source	Specifies a NuGet source to use during install.
--type	Filters templates based on available types. Predefined values are "project", "item" or "other".
--dry-run	Displays a summary of what would happen if the given command line were run if it would result in a template creation.
--force	Forces content to be generated even if it would change existing files.
-lang, --language	Filters templates based on language and specifies the language of the template to create.
--update-check	Check the currently installed template packs for updates.
--update-apply	Check the currently installed template packs for update, and install the updates.

Templates	Short Name	Language	Tags
Console Application	console	[C#], F#, VB	Common/Console
Class library	classlib	[C#], F#, VB	Common/Library
WPF Application	wpf	[C#]	Common/WPF
WPF Class library	wpflib	[C#]	Common/WPF
WPF Custom Control Library	wpfcustomcontrollib	[C#]	Common/WPF
WPF User Control Library	wpfusercontrollib	[C#]	Common/WPF
Windows Forms (WinForms) Application	winforms	[C#]	Common/WinForms
Windows Forms (WinForms) Class library	winformslib	[C#]	Common/WinForms
Worker Service	worker	[C#]	Common/Worker/Web
Unit Test Project	mstest	[C#], F#, VB	Test/MSTest
JUnit 3 Test Project	nunit	[C#], F#, VB	Test/JUnit



Exploring the Project Structure

Create a new project

Recent project templates

ASP.NET Core Web Application

C#

Search for templates (Alt+S)



Clear

C#

All Platforms

Web



ASP.NET Core Web Application

Project templates for creating ASP.NET Core web apps and web APIs for Windows, Linux and macOS using .NET Core or .NET Framework. Create web apps with Razor Pages, MVC, or Single Page Apps (SPA) using Angular, React, or React + Redux.

C#

Linux

macOS

Windows

Cloud

Service

Web



Blazor App

Project templates for creating Blazor apps that run on the server in an ASP.NET Core app or in the browser on WebAssembly. These templates can be used to build web apps with rich dynamic user interfaces (UIs).

C#

Linux

macOS

Windows

Cloud

Web

gRPC

gRPC Service

A project template for creating a gRPC ASP.NET Core service using .NET Core.

C#

Linux

macOS

Windows

Cloud

Service

Web



Razor Class Library

A project template for creating a Razor class library.

C#

Linux

macOS

Windows

Library

Web



NUnit Test Project (.NET Core)

A project that contains NUnit tests that can run on .NET Core on Windows, Linux and MacOS.

Back

Next

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.



Web Application

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



Angular

A project template for creating an ASP.NET Core application with Angular



React.js

A project template for creating an ASP.NET Core application with React.js

[Get additional project templates](#)

Authentication

No Authentication

[Change](#)

Advanced

☒ Configure for HTTPS

☐ Enable Docker Support
(Requires [Docker Desktop](#))

Linux

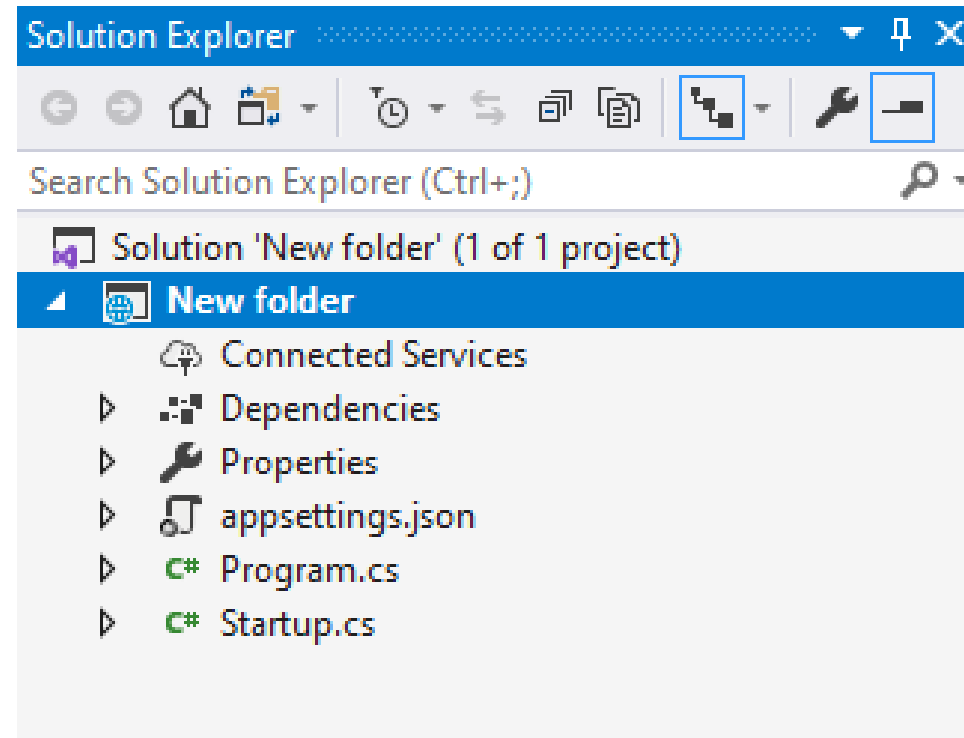
Author: Microsoft

Source: .NET Core 3.0.0

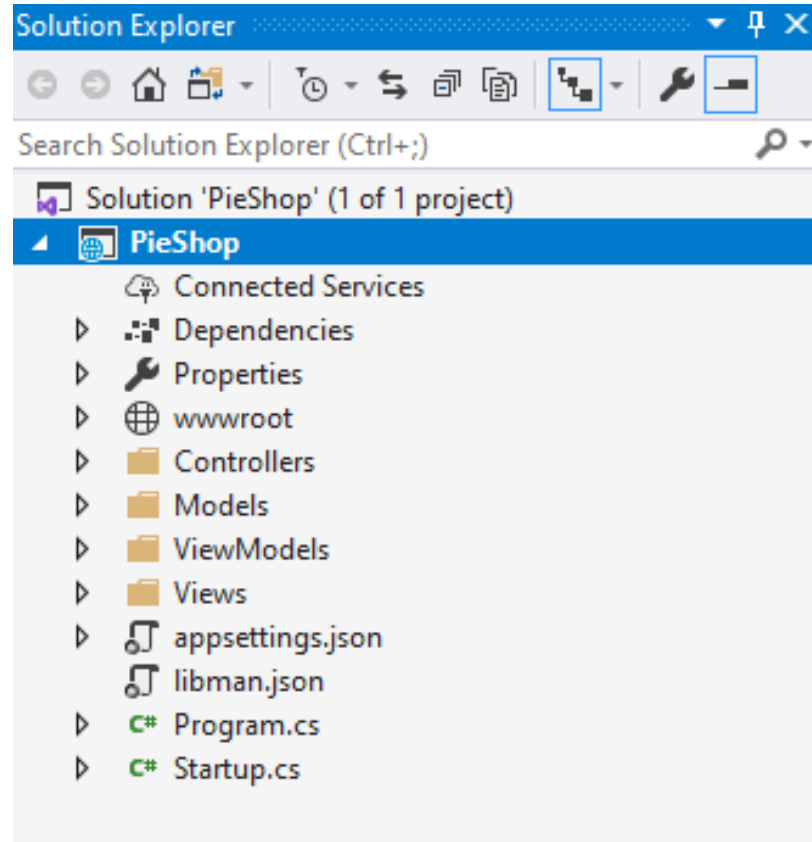
Back

Create

Project Structure (Empty)



Project Structure (Web Application MVC)



```
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>();
            });
}
```

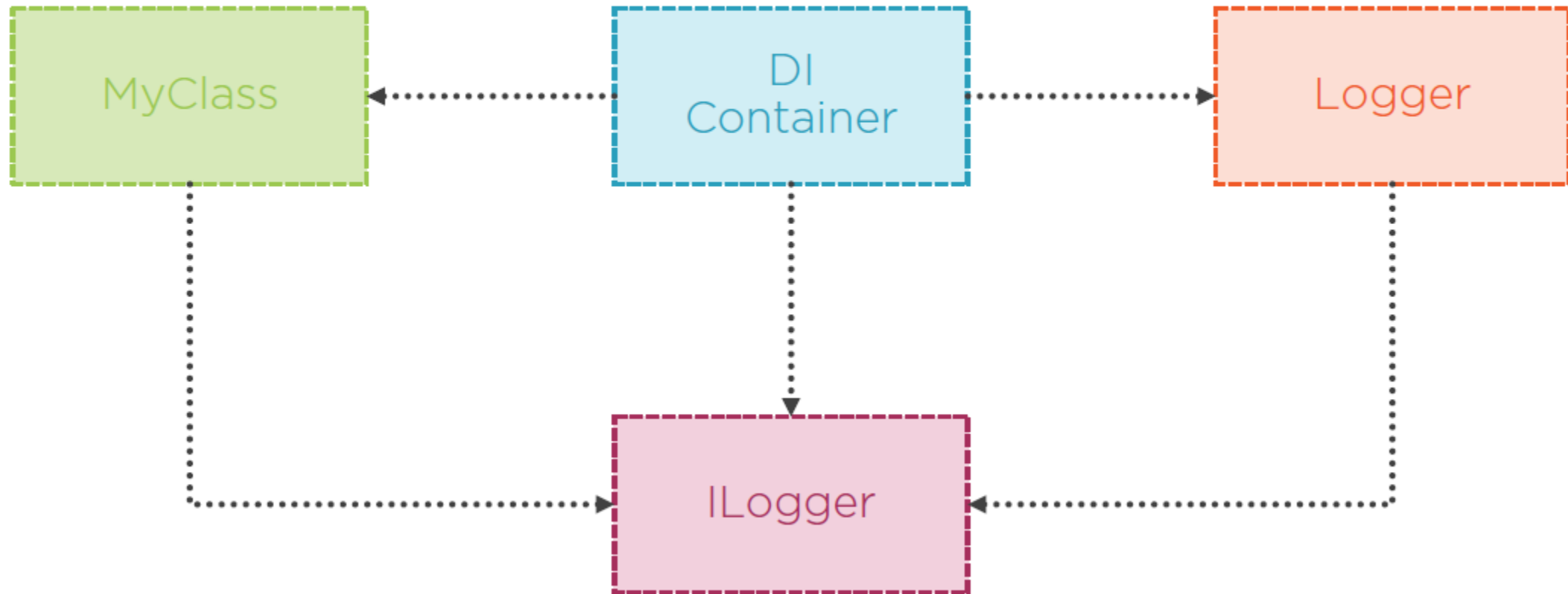
```
public class Startup
{
    // This method gets called by the runtime. Use this method to add services to the container.
    // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?
    public void ConfigureServices(IServiceCollection services)
    {
    }

    // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
    public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
    {
        if (env.IsDevelopment())
        {
            app.UseDeveloperExceptionPage();
        }

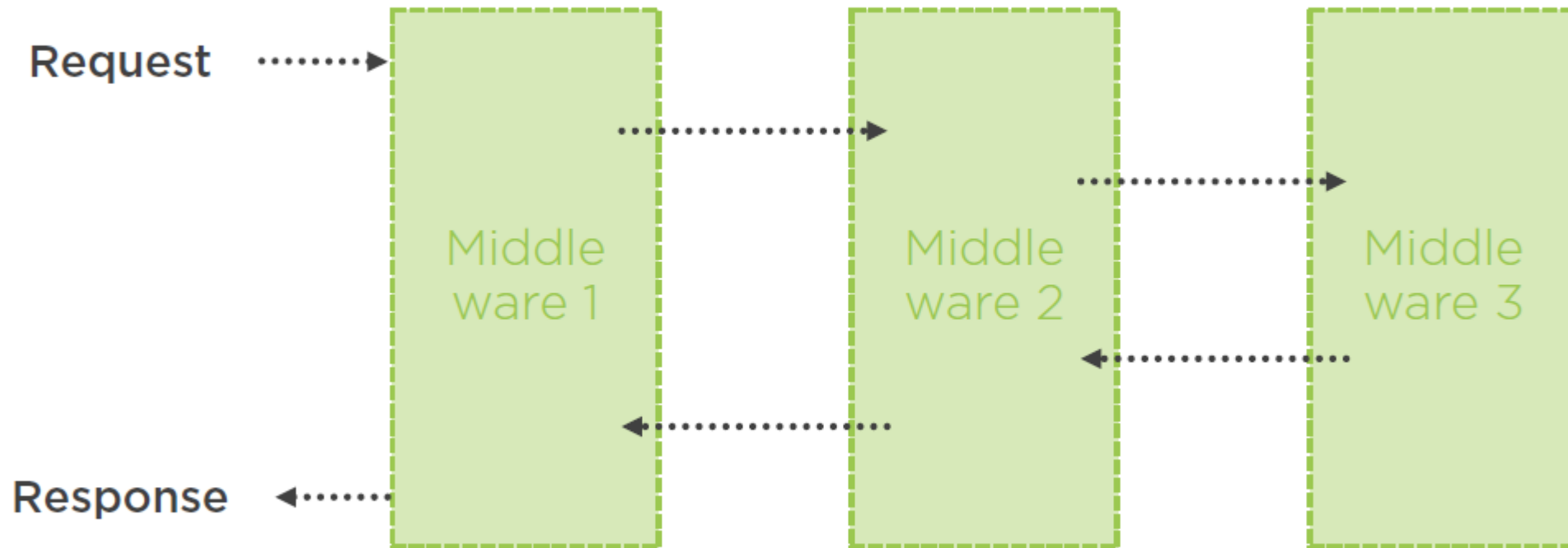
        app.UseRouting();

        app.UseEndpoints(endpoints =>
        {
            endpoints.MapGet("/", async context =>
            {
                await context.Response.WriteAsync("Hello World!");
            });
        });
    }
}
```

Dependency Injection (DI)



Middleware Request Pipeline



The Startup of the Application

