**Host in .NET Core applications**

.NET apps configure and launch a *host*. The host is responsible for app startup and lifetime management. At a minimum, the host configures a server and a request processing pipeline. Two host APIs are available for use:

* [Web Host](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/host/web-host?view=aspnetcore-2.2) – Suitable for hosting web apps.
* [Generic Host](https://docs.microsoft.com/en-us/aspnet/core/fundamentals/host/generic-host?view=aspnetcore-2.2) (ASP.NET Core 2.1 or later) – Suitable for hosting non-web apps (for example, apps that run background tasks). In a future release, the Generic Host will be suitable for hosting any kind of app, including web apps. The Generic Host will eventually replace the Web Host.

**Web Host**

ASP.NET Core Web Host (**IWebHostBuilder**) is useful for hosting web apps. This will handle your http requests. Create a host using an instance of IWebHostBuilder. This is typically performed in the app's entry point, the Main method.

public class Program

{

public static void Main(string[] args)

{

CreateWebHostBuilder(args).Build().Run();

}

public static IWebHostBuilder CreateWebHostBuilder(string[] args) =>

WebHost.CreateDefaultBuilder(args)

.UseStartup<Startup>();

}

CreateDefaultBuilder performs the following tasks:

* Configures Kestrel server as the web server using the app's hosting configuration providers.
* Sets the content root to the path returned by [Directory.GetCurrentDirectory](https://docs.microsoft.com/en-us/dotnet/api/system.io.directory.getcurrentdirectory).
* Loads host configuration from:
  + Environment variables prefixed with ASPNETCORE\_ (for example, ASPNETCORE\_ENVIRONMENT).
  + Command-line arguments.
* Loads app configuration in the following order from:
  + *appsettings.json*.
  + *appsettings.{Environment}.json*.
  + Secret Manager when the app runs in the Development environment using the entry assembly.
  + Environment variables.
  + Command-line arguments.
* Configures logging for console and debug output. Logging includes log filtering rules specified in a Logging configuration section of an *appsettings.json* or *appsettings.{Environment}.json* file.
* When running behind IIS with the ASP.NET Core Module, CreateDefaultBuilderenables IIS Integration, which configures the app's base address and port. IIS Integration also configures the app to capture startup errors.
* Sets ServiceProviderOptions.ValidateScopes to true if the app's environment is Development.

The configuration defined by **CreateDefaultBuilder** can be overridden and augmented by **ConfigureAppConfiguration**, **ConfigureLogging**, **ConfigureKestrel** and other methods and extension methods of IWebHostBuilder.

**Generic Host**

ASP.NET Core Generic Host (**HostBuilder**) is useful for hosting apps that don't process HTTP requests. The goal of the Generic Host is to decouple the HTTP pipeline from the Web Host API to enable a wider array of host scenarios. Messaging, background tasks, and other non-HTTP workloads based on the Generic Host benefit from cross-cutting capabilities, such as configuration, dependency injection (DI), and logging.

The Generic Host library is available in the Microsoft.Extensions.Hosting namespace and provided by the Microsoft.Extensions.Hosting package. The Microsoft.Extensions.Hosting package is included in the Microsoft.AspNetCore.App metapackage (ASP.NET Core 2.1 or later).

var host = new HostBuilder()

.ConfigureServices((hostContext, services) =>

{

services.Configure<HostOptions>(option =>

{

option.ShutdownTimeout = System.TimeSpan.FromSeconds(20);

});

})

.Build();

The following services are registered during host initialization:

* Environment (IHostingEnvironment)
* HostBuilderContext
* Configuration (IConfiguration)
* IApplicationLifetime (ApplicationLifetime)
* IHostLifetime (ConsoleLifetime)
* IHost
* Options (AddOptions)
* Logging (AddLogging)