

```
public static IHostBuilder CreateHostBuilder(string[] args) =>
```

```
1. // new HostBuilder().ConfigureLogging(configureLogging => { });
```

2.Or

```
Host.CreateDefaultBuilder(args)//init new instance of HostBuilder with pre-configured defaults
```

```
.ConfigureWebHostDefaults(webBuilder =>
{
    webBuilder.UseStartup<Startup>();

    //confiure host builder

    webBuilder.ConfigureServices(service =>
    {
        service.AddControllersWithViews();
        service.AddTransient<IStartupFilter, AlekFilter>();

        //service container
    });

    webBuilder.Configure(app =>
    {
        //middleware pipeline
    });

    //.Configure(app =>
    //{
    //    //middleware pipeline
    //});

});

////////////////////////////////////
```

3.Or

```
.ConfigureWebHost(configure =>
{
    //confiure host builder

    configure.ConfigureServices(services =>
    {
        //service container
    });

    configure.Configure(configureApp =>
    {
```

```

        //middleware pipeline

    });
}
)

//configure host builder

.ConfigureAppConfiguration((context, builder) =>
{
    //za conf files
});

```

Configure the Defaults of Host Builder

```

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>();

                webBuilder.ConfigureAppConfiguration(config =>
                    config.AddJsonFile("appsettings.json"); // Loads configuration values from
appsettings.json

                webBuilder.ConfigureLogging((ctx, config) => {
                    config.AddConfiguration(
                        ctx.Configuration.GetSection("Logging")); //Loads the log filtering configuration from
the Logging section

                    config.AddConsole();//Adds a console provider to the app
                });

            });
}

```

