**Dependency Injection in Asp.Net Core:**

Asp.Net Core is designed from scratch to support Dependency Injection. Asp.Net Core injects objects of dependency classes (services) through Constructor, Property, Method using **built in IoC container**.

**Built in IoC container:**

ASP.NET Core framework contains simple out-of-the-box IoC container which does not have as many features as other third party IoC containers, but it has limited features, to

have more features like auto-registration, scanning, interceptors, or decorators etc., then it can be replaced with for third party IoC containers.

The Built in IoC container in .Net Core is represented by IServiceProvider implementation that supports constructor injection by default. The types (classes) managed by built in IoC container are called **Services**.

There are basically two types of services in Asp.Net Core;

**Framework Services:**

Services which are part of .Net Core framework such as IApplicationBuilder, ILoggingFactory, IHostingEnvironment etc.,

**Application Services:**

Services which we as a programmer create for our application.

In order to inject our application services, first need to let IoC know by registering it with IoC container.

Application services can be registered in **ConfigureServices** method of **Startup** class, which includes **IServiceCollection** parameter, which helps in registering the service (below screenshot).

|  |
| --- |
|  |

The Add() method of IServiceCollection adds our application service by specifying the **type Of** service and its **instance**, it is done via Service Descriptor class. So from the screenshot, can see **Servicetype** specified as **ILog** , its instance is **MyConsoleLogger**. This registers ILog service as a **Singleton** by default.

Now IoC container will create a Singleton object of MyConsoleLogger class and inject it through constructor, wherever ILog service is included as constructor or method parameter throughout the application. There are extension methods available for easy and quick registration of services which we’ll see in upcoming sections.