‹ Previous Next ›

IoC Container

In the previous chapter, we learned how to implement the Dependency Injection pattern to achieve loose coupled classes. IoC Container (a.k.a. DI Container) is a framework for implementing automatic dependency injection. It manages object creation and it's life-time, and also injects dependencies to the class.

The IoC container creates an object of the specified class and also injects all the dependency objects through a constructor, a property or a method at run time and disposes it at the appropriate time. This is done so that we don't have to create and manage objects manually.

All the containers must provide easy support for the following DI lifecycle.

- **Register:** The container must know which dependency to instantiate when it encounters a particular type. This process is called registration. Basically, it must include some way to register type-mapping.
- **Resolve:** When using the IoC container, we don't need to create objects manually. The container does it for us. This is called resolution. The container must include some methods to resolve the specified type; the container creates an object of the specified type, injects the required dependencies if any and returns the object.
- **Dispose:** The container must manage the lifetime of the dependent objects. Most IoC containers include different lifetimemanagers to manage an object's lifecycle and dispose it.

There are many open source or commercial containers available for .NET. Some are listed below.

- Unity
- StructureMap
- Castle Windsor
- Ninject
- Autofac
- DryIoc
- Simple Injector
- Light Inject

Learn about Unity Container in the next chapters.

Share Tweet Share Whatsapp

‹ Previous Next ›

TUTORIALS

- ASP.NET Core
- ASP.NET MVC
- IoC
- Web API
- C#
- LINQ
- AngularJS 1
- Node.js
- D3.js
- JavaScript
- jQuery
- Sass

› Entity Framework › Https

HOME PRIVACY POLICY TERMS OF USE ADVERTISE WITH US