



Empowering Digital Skills For The Jobs Of The Future



by



Academy .NET

Dependency Injection

Sommario

- Dependency Injection pattern
- Principi di design SOLID
- Inversion of Control(IoC)

Principi di design SOLID

- **Single Responsibility:** un modulo o una funzione dovrebbe essere responsabile di un unico scopo.
- **Open-Closed:** un artefatto software dovrebbe essere aperto per l'estensione ma chiuso per la modifica.
- **Liskov Substitution:** il comportamento di un programma deve rimanere invariato quando si sostituisce uno dei suoi componenti con un altro componente che è stato definito da un supertype del primer object.

Principi di design SOLID

- **Interface Segregation:** la creazione di interfacce di grandi dimensioni causerà il verificarsi di dipendenze durante la creazione di oggetti concreti, ma questi sono dannosi per l'architettura del sistema.
- **Dependency Inversion:** i sistemi più flessibili sono quelli in cui le dipendenze degli oggetti si riferiscono solo alle astrazioni.

Dependency Injection pattern

La **Dependency Injection pattern** è una metodologia di implementazione del principio di **Inversione delle dipendenze**.

L'inversione delle dipendenze è un principio di programmazione.

Inverte il flusso di controllo

Il codice personalizzato di un programma riceve il flusso di controllo da un framework generico

Dependency Injection pattern

Invece di creare istanze degli oggetti da cui dipende il componente, è sufficiente definire le loro dipendenze, dichiarare le loro interfacce e abilitare la ricezione degli oggetti per **injection**.

Dependency Injection pattern

Metodi per eseguire la **dependency injection**:

- Utilizzare il **costruttore** della classe per ricevere gli oggetti
- Contrassegna alcune **proprietà** di classe per ricevere gli oggetti
- Definire un'**interfaccia** con un metodo per iniettare tutti i componenti necessari

Inversion of Control(IoC)

- La dependency Injection può essere implementata con un container **Inversion of Control(IoC)**
- **IoC** permette l'inserimento automatico delle dipendenze
- Presente nel namespace **Microsoft.Extensions.DependencyInjection**

Inversion of Control(IoC)

IoC container è responsabile della creazione e dell'eliminazione degli oggetti richiesti.

Inversion of Control(IoC)

Sono disponibili tre opzioni per il **ciclo di vita** del componente iniettato(**injected**):

Transient: gli oggetti vengono creati ogni volta che vengono richiesti.

Scoped: gli oggetti vengono creati per ogni ambito definito nell'applicazione.

Singleton: ogni oggetto ha la stessa durata dell'applicazione

Domande & approfondimenti

Academy .NET