



EBook Gratuito

APPENDIMENTO unity-container

Free unaffiliated eBook created from
Stack Overflow contributors.

#unity-
container

Sommario

Di.....	1
Capitolo 1: Iniziare con unità-contenitore.....	2
Osservazioni.....	2
Versioni.....	2
Examples.....	2
Installazione.....	2
Ciao mondo.....	2
Costruttore di iniezione.....	3
Capitolo 2: Unity WebAPI.....	5
Examples.....	5
Impostazione di Unity con Web API.....	5
Titoli di coda.....	7

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [unity-container](#)

It is an unofficial and free unity-container ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official unity-container.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capitolo 1: Iniziare con unità-contenitore

Osservazioni

Unity Container (Unity) è un contenitore di iniezione di dipendenza leggero ed estensibile. Facilita la creazione di applicazioni liberamente accoppiate e offre agli sviluppatori i seguenti vantaggi: Creazione semplificata di oggetti, in particolare per strutture gerarchiche di oggetti e dipendenze. [<https://msdn.microsoft.com/en-us/library/ff647202.aspx>]

Dovrebbe anche menzionare eventuali soggetti di grandi dimensioni all'interno dell'unità-contenitore e collegarsi agli argomenti correlati. Poiché la Documentazione per unità-contenitore è nuova, potrebbe essere necessario creare versioni iniziali di tali argomenti correlati.

Versioni

Versione	Note di rilascio	Data di rilascio
2.0.0	Unità 2	2011-05-05
2.1.0	Unità 2.1	2011-05-11
3.0.0	Unità 3	2013/04/26
3.5.0	Unità 3.5	2015/05/15
4.0.0	Unità consegnata alla comunità OSS	2015/10/06

Examples

Installazione

Per iniziare, devi solo installare il pacchetto **Unity** nuget. Esegui il seguente comando dalla console del gestore pacchetti:

```
PM> Install-Package Unity
```

In alternativa, è possibile utilizzare Visual Studio per installare Unity su un particolare progetto utilizzando l' *opzione Gestisci pacchetti NuGet per soluzione* in Strumenti -> NuGet Package Manager.

Ciao mondo

```
interface IGreeter
{
```

```

    void Greet();
}

class Greeter : IGreeter
{
    public void Greet()
    {
        Console.WriteLine("Hello World");
    }
}

class SpanishGreeter : IGreeter
{
    public void Greet()
    {
        Console.WriteLine("Hola Mundo");
    }
}

class FrenchGreeter : IGreeter
{
    public void Greet()
    {
        Console.WriteLine("Bonjour le Monde");
    }
}

class Program
{
    static void Main(string[] args)
    {
        var container = new UnityContainer()
            .RegisterType<IGreeter, SpanishGreeter>("spanish")
            .RegisterType<IGreeter, FrenchGreeter>("french")
            .RegisterType<IGreeter, Greeter>();

        //Get default registration. Outputs "Hello World"
        var greeter = container.Resolve<IGreeter>();
        greeter.Greet();

        //Get specific named registration. Outputs "Hola Mundo"
        greeter = container.Resolve<IGreeter>("spanish");
        greeter.Greet();

        //Get all named registrations (excludes the default one)
        //Outputs "Hola Mundo" and "Bonjour le Monde"
        foreach (var g in container.ResolveAll<IGreeter>())
        {
            g.Greet();
        }

        Console.ReadLine();
    }
}

```

Costruttore di iniezione

```

interface IService
{

```

```

        void ProcessRequest();
    }

    interface IRepository
    {
        IEnumerable<string> GetData();
    }

    class HelloWorldRepository : IRepository
    {
        public IEnumerable<string> GetData()
        {
            return new[] { "Hello", "World" };
        }
    }

    class HelloWorldService : IService
    {
        private readonly IRepository repo;
        public HelloWorldService(IRepository repo)
        {
            this.repo = repo;
        }
        public void ProcessRequest()
        {
            Console.WriteLine(String.Join(" ", this.repo.GetData()));
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            var container = new UnityContainer()
                .RegisterType<IRepository, HelloWorldRepository>()
                .RegisterType<IService, HelloWorldService>();

            //Unity automatically resolves constructor parameters that knows about.
            //It will return a HelloWorldService with a HelloWorldRepository
            var greeter = container.Resolve<IService>();
            //Outputs "Hello World"
            greeter.ProcessRequest();

            Console.ReadLine();
        }
    }
}

```

Leggi Iniziare con unità-contenitore online: <https://riptutorial.com/it/unity-container/topic/5292/iniziare-con-unita-contenitore>

Capitolo 2: Unity WebAPI

Examples

Impostazione di Unity con Web API.

1. Aggiungi Unity al tuo progetto.

Se usi [NuGet](#) puoi usare il [pacchetto Unity](#) . Esegui `Install-Package Unity` nella console di Gestione pacchetti. Questo aggiungerà la libreria Unity (e le sue dipendenze) al tuo progetto.

2. Creare un'implementazione di [IDependencyResolver](#) .

Per esempio:

```
public class UnityResolver : IDependencyResolver
{
    protected IUnityContainer Container;

    public UnityResolver(IUnityContainer container)
    {
        if (container == null)
        {
            throw new ArgumentNullException("container");
        }
        this.Container = container;
    }

    public object GetService(Type serviceType)
    {
        try
        {
            return Container.Resolve(serviceType);
        }
        catch (ResolutionFailedException)
        {
            return null;
        }
    }

    public IEnumerable<object> GetServices(Type serviceType)
    {
        try
        {
            return Container.ResolveAll(serviceType);
        }
        catch (ResolutionFailedException)
        {
            return new List<object>();
        }
    }

    public IDependencyScope BeginScope()
    {
        var child = Container.CreateChildContainer();
    }
}
```

```

        return new UnityResolver(child);
    }

    public void Dispose()
    {
        Container.Dispose();
    }
}

```

3. Registrare `IDependencyResolver` nel proprio `WebApiConfig` .

```

public static class WebApiConfig
{
    public static void Register(HttpConfiguration config)
    {
        // Routes goes here..

        // Create your container.
        var container = new UnityContainer();

        // Do registrations here...

        // Assign your container.
        config.DependencyResolver = new UnityResolver(container);
    }
}

```

Leggi Unity WebAPI online: <https://riptutorial.com/it/unity-container/topic/6396/unity-webapi>

Titoli di coda

S. No	Capitoli	Contributors
1	Iniziare con unità-contenitore	Clemens Tolboom , Community , Daniel J.G.
2	Unity WebAPI	smoksnes