



eBook Gratuit

APPRENEZ

unity-container

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#unity-

container

Table des matières

À propos	1
Chapitre 1: Démarrer avec unity-container	2
Remarques.....	2
Versions.....	2
Exemples.....	2
Installation.....	2
Bonjour le monde.....	2
Constructeur Injection.....	3
Chapitre 2: Unity WebAPI	5
Exemples.....	5
Configurer Unity avec Web API.....	5
Crédits	7

À propos

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

Chapitre 1: Démarrer avec unity-container

Remarques

Unity Container (Unity) est un conteneur d'injection de dépendance léger et extensible. Il facilite la création d'applications faiblement couplées et offre aux développeurs les avantages suivants: Création d'objet simplifiée, en particulier pour les structures et dépendances d'objets hiérarchiques. [<https://msdn.microsoft.com/en-us/library/ff647202.aspx>]

Il devrait également mentionner tous les grands sujets dans l'unité-conteneur et établir un lien avec les sujets connexes. Comme la documentation de unity-container est nouvelle, vous devrez peut-être créer des versions initiales de ces rubriques connexes.

Versions

Version	Notes de version	Date de sortie
2.0.0	Unité 2	2011-05-05
2.1.0	Unité 2.1	2011-05-11
3.0.0	Unité 3	2013-04-26
3.5.0	Unité 3.5	2015-05-15
4.0.0	Unité transférée à la communauté OSS	2015-10-06

Exemples

Installation

Pour commencer, il vous suffit d'installer le package nuget **Unity** . Exécutez la commande suivante à partir de la console du gestionnaire de packages:

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

Vous pouvez également utiliser Visual Studio pour installer Unity sur un projet particulier à l'aide de l' *option Manage NuGet Packages for Solution* sous Tools -> NuGet Package Manager.

Bonjour le monde

```
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();
    }
}

```

Constructeur Injection

```

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();
    }
}

```

Lire Démarrer avec unity-container en ligne: <https://riptutorial.com/fr/unity-container/topic/5292/demarrer-avec-unity-container>

Chapitre 2: Unity WebAPI

Exemples

Configurer Unity avec Web API.

1. Ajoutez Unity à votre projet.

Si vous utilisez [NuGet](#), vous pouvez utiliser le [package Unity](#) . Exécutez `Install-Package Unity` dans la console du gestionnaire de packages. Cela ajoutera la bibliothèque Unity (et ses dépendances) à votre projet.

2. Créez une implémentation de [IDependencyResolver](#) .

Par exemple:

```
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. Enregistrez votre `IDependencyResolver` dans votre `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);
    }
}
```

Lire Unity WebAPI en ligne: <https://riptutorial.com/fr/unity-container/topic/6396/unity-webapi>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec unity-container	Clemens Tolboom , Community , Daniel J.G.
2	Unity WebAPI	smoksnes