

第02期-理解模块加载机制

2020年7月15日 14:44

安装 Volo.Abp 包

Install-Package Volo.Abp.Core

创建第一个ABP模块

```
using Microsoft.Extensions.DependencyInjection;
using Volo.Abp.Modularity;

namespace AbpConsoleDemo
{
    public class AppModule : AbpModule
    {
    }
}
```

初始化应用程序

```
using System;
using Volo.Abp;

namespace AbpConsoleDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var application =
                AbpApplicationFactory.Create<AppModule>())
            {
                application.Initialize();

                Console.WriteLine("Press ENTER to stop
                application...");

                Console.ReadLine();
            }
        }
    }
}
```

AbpApplicationFactory 用于创建应用程序并加载 AppModule 作为启动模块，Initialize()方法启动应用程序。

第一个 Hello World 程序

```
using System;
using Volo.Abp.DependencyInjection;

namespace AbpConsoleDemo
{
    public class HelloWorldService : ITransientDependency
    {
        public void SayHello()
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

ITransientDependency 是 ABP 的一个特殊接口,它自动将服务注册为 Transient 生命周期。

现在,我们可以解析HelloWorldService并调用SayHello., 更改Program.cs, 如下所示

```
using System;
using Microsoft.Extensions.DependencyInjection;
using Volo.Abp;

namespace AbpConsoleDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var application =
                AbpApplicationFactory.Create<AppModule>())
            {
                application.Initialize();

                // 解析服务并使用它
                var helloWorldService =
                    application.ServiceProvider.GetService<Hello
WorldService>();

                helloWorldService.SayHello();

                Console.WriteLine("Press ENTER to stop
application...");

                Console.ReadLine();
            }
        }
    }
}
```

参考资料

[MyZony: 框架启动流程分析](#)

[MyZony: 模块系统的变化](#)

[使用 C# 代码实现拓扑排序](#)