## Module模块

1. abp通过icoManager注册容器
2. 实例通过继承AbpModule来发现服务

## Configuration

1. abp原生的注入AbpStartuponfiguration，Castle的依赖注入
2. 外部通过Dictionary对象和扩展方法巧妙的实现配置中心化

原生实现

第一行通过AbpCoreInstaller注册的是系统框架级的所有configuration.

接着通过容器生成IAbpStartupConfiguration的实例AbpStartupConfiguration对象。

最后调用AbpStartupConfiguration的Initialize方法，该方法最终调用容器生成他的小伙伴的各种实例（\*\*\*\*Configuration）

## 注入的配置

### AddAbp 和 UseAbp

AddAbp 用于注入服务（主要是替换原框架和配置abp框架使用的服务）

UseAbp 配置abp的使用 （管理中间过程和初始化一些实例）

### AddAbp主要做的操作有

### 启动核心类AbpBootstrapper

//1.创建AbpBootstrapper

var abpBootstrapper = AddAbpBootstrapper<TStartupModule>(services, optionsAction);

//2.配置

ConfigureAspNetCore(services, abpBootstrapper.IocManager);

//3.使用Castle替代自带的 Ioc 容器

return WindsorRegistrationHelper.CreateServiceProvider(abpBootstrapper.IocManager.IocContainer, services);