**配置ABP**

在预初始化事件中进行配置，示例：

public class SimpleTaskSystemModule : AbpModule

{

public override void PreInitialize()

{

//为你的应用添加语言

Configuration.Localization.Languages.Add(new LanguageInfo("en", "English", "famfamfam-flag-england", true));

Configuration.Localization.Languages.Add(new LanguageInfo("tr", "Türkçe", "famfamfam-flag-tr"));

//添加一个本地资源

Configuration.Localization.Sources.Add(

new XmlLocalizationSource(

"SimpleTaskSystem",

HttpContext.Current.Server.MapPath("~/Localization/SimpleTaskSystem")

)

);

//配置导航/菜单

Configuration.Navigation.Providers.Add<SimpleTaskSystemNavigationProvider>();

}

public override void Initialize()

{

IocManager.RegisterAssemblyByConvention(Assembly.GetExecutingAssembly());

}

}

**替换内置服务**

Configuration.ReplaceServices方法可以用来重写一个内置服务。例如，你可以用自定义的实现替换IAbpSession服务，如下所示：

**Configuration.ReplaceService**<IAbpSession,MySession>(DependencyLifeStyle.Transient);

**配置模块**

除了框架自身的启动配置，一个模块可扩展IAbpModuleConfigurations接口来提供配置切入点，例如：

...using Abp.Web.Configuration;

...public override void PreInitialize()

{

Configuration.Modules.AbpWebCommon().SendAllExceptionsToClients = true;

}

...

例子中，我们配置了用AbpWebCommon模块负责发送异常给客户端。

**为一个模块创建配置**

假设我们有一个名为MyModule的模块，它有一些配置属性。首先，我们为这些配置属性创建一个类：

public class MyModuleConfig

{

public bool SampleConfig1 { get; set; }

public string SampleConfig2 { get; set; }

}

然后，在依赖注入的预初始化事件里注册这个类：

IocManager.Register<MyModuleConfig>();

在本例中，它应该注册成单例，现在我们可以使用下面的代码，在我们模块的PreInitialize方法中配置MyModule：

Configuration.**Get<MyModuleConfig>()**.SampleConfig1 = false;