

第15期-虚拟文件系统

2020年7月31日 9:39

虚拟文件系统是对文件系统的抽象，不用关心文件的存储位置和方式，抽象了一种树形文件管理机制，基于微软 .NET Core 文件系统扩展。

安装虚拟文件库

```
abp add-package Volo.Abp.VirtualFileSystem
```

```
Install-Package Volo.Abp.VirtualFileSystem
```

将文件嵌入到程序集

```
<Project Sdk="Microsoft.NET.Sdk.Web">

  <PropertyGroup>
    <TargetFramework>netcoreapp3.1</TargetFramework>
  </PropertyGroup>

  <ItemGroup>
    <EmbeddedResource Include="Resource\*" />
  </ItemGroup>

</Project>
```

此配置以递归方式添加项目的 **MyResources** 文件夹下的所有文件(包括将来新添加的文件)

将嵌入在程序集中的文件注册到虚拟文件系统

```
Configure<AbpVirtualFileSystemOptions>(options =>
{
    options.FileSets.AddEmbedded<MyModule>();
});

Configure<AbpVirtualFileSystemOptions>(options =>
{
    options.FileSets.AddEmbedded<MyModule>(
        baseNamespace: "BookStore.MyFiles",
        baseFolder: "/MyFiles"
    );
});
```

在开发过程中处理嵌入式文件

```
[DependsOn(typeof(MyModule))]  
public class MyWebAppModule : AbpModule  
{  
    public override void ConfigureServices(ServiceConfigurationContext  
context)  
    {  
        var hostingEnvironment = context.Services.GetHostingEnvironment();  
  
        if (hostingEnvironment.IsDevelopment()) //only for development  
time  
        {  
            Configure<AbpVirtualFileSystemOptions>(options =>  
            {  
                options.FileSets.ReplaceEmbeddedByPhysical<MyModul  
e>(  
                    Path.Combine(  
                        hostingEnvironment.ContentRootPath  
,  
                        string.Format(  
                            ".. {0} MyModuleProject",  
                            Path.DirectorySeparatorCha  
r  
                            )  
                    )  
                );  
            });  
        }  
    }  
}
```

使用虚拟文件系统

```
public class MyService  
{  
    private readonly IVirtualFileProvider _virtualFileProvider;  
  
    public MyService(IVirtualFileProvider virtualFileProvider)  
    {  
        _virtualFileProvider = virtualFileProvider;  
    }  
  
    public void Foo()  
    {  
        //Getting a single file  
        var file =  
_virtualFileProvider.GetFileInfo("/MyResources/js/test.js");  
  
        var fileContent = file.ReadAsString();  
    }  
}
```

```
        //Getting all files/directories under a directory
        var directoryContents =
        _virtualFileProvider .GetDirectoryContents("/MyResources/js");
    }
}
```

与 ASP.NET Core 集成

```
Configure<AbpAspNetCoreContentOptions>(options =>
{
    options.AllowedExtraWebContentFileExtensions.Add(".html");
    options.AllowedExtraWebContentFolders.Add("/ClassLibrary1");
});
```

```
app.UseVirtualFiles();
```

默认情况下，ASP.NET Core 仅允许 wwwroot 文件夹包含客户端使用的静态文件，当你使用 UseVirtualFiles 中间件时以下文件夹也可以包含静态文件：Pages、Views 和 Themes

这允许你可以在 .cshtml 文件所在目录添加 .js, .css... 文件，更易于开发和维护你的项目。

参考资料

[ASP.NET Core 中的文件提供程序](#)