参考文章：https://www.cnblogs.com/artech/p/inside-asp-net-core-1.html

文件提供器IFileProvider在 .net core 中比较常见吧，我们可以使用IFileProvider读取目录和文件

**简单示例**

static void Main (string[] args) {

    // 新建文件提供器，指定根目录

    IFileProvider PhysicalFileProvider = new PhysicalFileProvider ("E:\\ProjectTestDir");

    // 获取根目录下的内容

    foreach (var content in PhysicalFileProvider.GetDirectoryContents ("\\")) {

        // 输出路径

        Console.WriteLine (content.PhysicalPath);

        // 如果内容不是目录类型

        if (!content.IsDirectory) {

            // 入读文件

            Stream stream = content.CreateReadStream ();

            byte[] buf = new byte[stream.Length];

            stream.Read (buf, 0, buf.Length);

            Console.WriteLine (Encoding.ASCII.GetString (buf));

        }

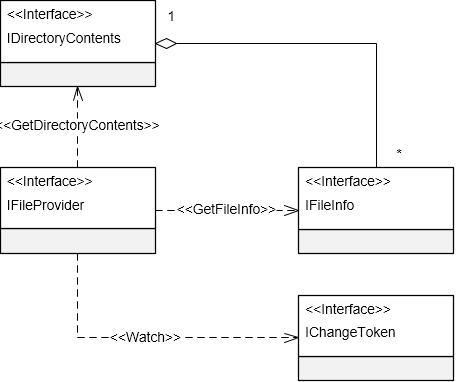
}

    Console.WriteLine ("Hello World!");

}

**IFileProvider设计图**

出自：https://www.cnblogs.com/artech/p/inside-asp-net-core-1.html



**类介绍**

public interface IFileProvider

{

    IFileInfo GetFileInfo(string subpath);

    IDirectoryContents GetDirectoryContents(string subpath);

    IChangeToken Watch(string filter);

}

IDirectoryContents：表示一个目录中的文件的集合，是IFileInfo的集合

IFileInfo: 表示一个文件或目录

public interface IFileInfo

{

    bool                Exists { get; }         // 是否存在

    bool                IsDirectory { get; }    // 是否是目录

    string              Name { get; }           // 名字

    string              PhysicalPath { get; }   // 完整路径名

    DateTimeOffset      LastModified { get; }   // 最后修改时间

    long                Length { get; }         // 如果是文件，则为文件的长度

    Stream              CreateReadStream();     // 如果是文件，返回文件的流

}

IChangeToken：

表示文件改变令牌，可以理解为，文本文件发生改变前后，所获取到的IChangeToken是不同的，利用这一功能，我们可以与ChangeToken配合监听文件改变

public class Program

{

    public static void Main(string[] args)

    {

        IFileProvider fileProvider = new PhysicalFileProvider(@"c:\test");

        // 监听文件改变

        ChangeToken.OnChange(

            () => fileProvider.Watch("data.txt"),   // 获取改变令牌函数

            () => LoadFileAsync(fileProvider));     // 当发生改变时的回调函数

    }

    public static async void LoadFileAsync(IFileProvider fileProvider)

    {

    }

}

**.net core 已提供的FileProvider**

PhysicalFileProvider：

包：Microsoft.Extensions.FileProviders.Physical

物理文件提供器，提供对硬盘数据的访问

EmbeddedFileProvider：

包：Microsoft.Extensions.FileProviders.Embedded

提供对内嵌程序集的文件的访问