

# Linux环境.NET Core zh-CN中文语言问题

Published on Nov 15, 2021 UTC by Wang Ye <<https://wangye.org>>

.NET Core多语言本地化支持可以让所开发的应用更加国际化，但是不同操作系统平台之间可能会遇到一些特有的部署问题，本文将介绍我在Linux环境下部署国际化应用过程中遇到的zh-CN中文资源丢失问题的解决方案，当然可能有其他更好的方案，这里不做推荐，本文操作使用的软件环境是Debian 11、ASP.NET 5.0、nginx，仅供参考。

## 1 多语言国际化应用的建立

.NET Core本地化方式参考[《Globalization and localization in ASP.NET Core》](#)，通常情况下，在Startup.cs的ConfigureServices部分会包含如下代码：

```
services.AddLocalization(options => options.ResourcesPath =
"Resources");
services.AddMvc()

.AddViewLocalization(Microsoft.AspNetCore.Mvc.Razor.LanguageViewLocati
onExpanderFormat.Suffix)
.AddDataAnnotationsLocalization();
```

随后在中间件（Middleware）中会插入类似下面的代码：

```
var cultures = new List<CultureInfo> {
    new CultureInfo("en"),
    new CultureInfo("zh")
};
app.UseRequestLocalization(options => {
    options.DefaultRequestCulture = new
Microsoft.AspNetCore.Localization.RequestCulture("en");
    options.SupportedCultures = cultures;
    options.SupportedUICultures = cultures;
```

```
});
```

具体代码的含义我这里不再累述，大家可以参考官方文档，通过上述配置我们可以在项目根目录下建立Resources文件夹，文件夹里对应于根路径建立相应语言的本地化语言支持文件，比如Pages文件夹包含名为Default.cshtml的Razor模板文件，那么Resources文件夹也要对应的有Pages文件夹和Default.<LANGUAGE>.resx语言文件，比如简体中文是Default.zh-CN.resx，建立完成后我们就可以配置语言文件并建立多语言国际化应用。

## 2 Linux环境下zh-CN语言问题

按照步骤1的配置，在Windows环境下一切正常，但是通过Docker部署在Linux (Debian)环境下发现zh-CN语言功能失效了，究其原因发现在Windows环境下编译会自动生成zh-CN文件夹，文件夹中包含\*.resources.dll的编译的语言包文件，但是在Linux环境下这个zh-CN的语言包却无法生成，导致简体中文无法使用。

搜索网络发现这个问题已经有人发现，其中有人提出了两种解决方案，第一种是安装zh-CN语言包支持，第二种是将zh-CN修改为zh-Hans，比如比如简体中文Default.zh-CN.resx应该修改成Default.zh-Hans.resx，综合考虑，第一种方案需要修改系统配置，如果是在Docker里还好，否则有可能会影响其他程序的运行，本着最小化系统影响的原则，我决定采取第二种方案。

## 3 解决方案

### 3.1 修改项目语言文件类型

将所有的语言文件\*.zh-CN.resx改成\*.zh-Hans.resx，重新编译项目，发现Linux系统下正常生成zh-Hans文件夹，其中包含了我们所需要的语言文件，Deploy部署项目，浏览器测试正常，问题成功解决！

问题真的解决了吗？其实不然，在测试其他浏览器的时候发现本地化语言切换不了，也就是说本来是中文浏览器的用户访问网站，系统依然使用的是英文界面方案，因为英文en是我们默认的方案，所以可以判定这个解决办法有瑕疵，具体问题出在哪儿呢？

### 3.2 修改HTTP请求Accept-Language项

查阅官方文档发现这个本地化判断的几种方式，有个选项就是判断HTTP请求的Accept-Language项指示内容，比如Accept-Language: zh-CN, zh;q=0.8, en-US;q=0.6, en;q=0.4，表示根据浏览器用户接受的语言权重依次确定本地化语言，很显然，如果用户的浏览器Accept-Language仅仅发出zh-CN选项，那么我们的应用将因为找不到这个语言资源而发生fallback，进而调用了默认en语言，实际上我测试下来一部分浏览器可以发出zh-Hans，另外一部分浏览器只有zh-CN，如何解决这个问题呢？

那就是通过反向代理服务器修改Accept-Language，将zh-CN替代为zh-Hans。这里我使用nginx作为反向代理服务器，ngx\_headers\_more模块作为修改利器，使用下面的配置即可：

```
map $http_accept_language $my_accept_language {
    default $http_accept_language;
    "~*.*zh-Hans.*" $http_accept_language;
    "~*(.*) (zh-CN) (.*)" "$1$2, zh-Hans, zh-Hans-CN$3";
}
more_set_input_headers -r "Accept-Language: $my_accept_language";
```

通过上述配置，这个问题算是较好的解决了，当然也期待有更为优雅的方法。

## External References (Links)

《Globalization and localization in ASP.NET Core》

<https://docs.microsoft.com/en-us/aspnet/core/fundamentals/localization?>

<https://wangye.org/posts/2021/11/zh-cn-language-missing-in-net-core-under-linux.html>  
[view=aspnetcore-5.0](#)