

搜索`symfony locale`, 查看Symfony的文档。我们往下看, 这里有段关于controller方法的示例代码, 在路径中添加了`_locale`占位符参数, 这个参数是个特殊的参数。当路径中使用这个参数时, Request对象会根据这个参数的值自动的设置语言环境。

回到项目, 打开PostController类。在`index()`方法的注解中, 我们可以使用这个占位符。使用大括号下划线`{_locale}`, 我们要限制`_locale`可以使用的字符串。在注解中我们添加额外的配置项回车, 我们添加`requirements`配置, `requirements`是个数组, 在`requirements`中我们设置`_locale`参数它只能为英文`en`或者中文`zh_CN`。

出现了下划线, 我们需要对注解的参数顺序调整一下, 现在就可以了。回到浏览器, 我们访问博客首页。这里我们输入`en`, 现在我们看到文章列表这里显示的英文。如果我们输入`zh_CN`, 它会翻译为中文。如果我们随便输入个地区, 那么将会报错, 提示没有找到这个路由。

回到项目, 我们还可以使用内联的方式来限制`_locale`参数, 可以使用的字符串, 在占位符大括号中添加尖括号, 这里输入`<en|zh_CN>`, 然后注释掉`requirements`配置。

回到浏览器, 我们输入`en`仍然可以访问, 输入`zh_CN`同样没有问题, 如果我们随便输一个字符串仍然会报错, 我们需要为controller中的所有方法都添加`_locale`前缀。

可以在类前添加注解, 我们复制一下Route注解, 粘贴修改一下, 删除`name`, 只保留`_locale`路径。修改`index()`方法的注解, 删除`_locale`参数。

回到浏览器, 我们访问`en`没有问题, 访问中文地区也没有问题。如果我们查看第一篇文章, 在文章详情页的地址前也对应的加上了地区码, 我们可能有很多controller类都需要限制`_locale`参数。

我们打开`config`目录, 打开`services.yaml`配置文件, 在`parameters`配置下我们添加一个配置, 叫做`support_locales`的。这里我们设置为`en|zh_CN`或者其他的地区。

```
#config/services.yaml
parameters:
    # ...
    support_locales: en|zh_CN|fr
```

然后回到controller, 我们可以使用双百分号的方式引入这个参数, 复制参数名称, 粘贴。回到浏览器, 刷新, 同样没有问题。我们访问`fr`也没有问题, 如果我们随便访问一个地区, 那就出错了。

```
#src/Controller/PostController.php
#[Route('/{_locale<%support_locales%>}')]
class PostController extends BaseController
{
    // ...
}
```

现在我们访问一下博客的首页, 虽然当前显示了一个页面, 但是错误码仍然是404的。我们需要添加一个额外的controller方法, 将首页的地址跳转到文章列表。

回到项目，我们新增一个controller，Controller名称叫做`HomeController`，在`index()`方法中，我们直接进行跳转，输入`$this->redirectToRoute()`。我们跳转到`post_index`路由，现在我们就需要为`post_index`路由传递参数了，我们需要为`post_index`路由传递`_locale`参数。这里我们直接输入`zh_CN`跳转到中文地区，然后调整一下路径。

```
#src/Controller/HomePageController.php

class HomeController extends AbstractController
{
    #[Route('/', name: 'home_page')]
    public function index(): Response
    {
        return $this->redirectToRoute('post_index', [
            '_locale' => 'zh_CN'
        ]);
    }
}
```

回到浏览器刷新首页，我们看到现在就自动跳转到了中文地区的页面，国际化的内容我们暂时讲到这里。在下节课，我们来学习另一个有趣的组件工作流。