

# SpringMVC 国际化概述

---

## 简介：

<http://www.china-embassy.org/chn/>

假设：正在开发一个支持多国语言的Web应用程序，要求系统能够根据客户端的系统的语言类型返回对应的界面：英文的操作系统返回英文界面，而中文的操作系统则返回中文界面——这便是典型的i18n国际化问题。对于有国际化要求的应用系统，我们不能简单地采用硬编码的方式编写用户界面信息、报错信息等内容，而必须为这些需要国际化的信息进行特殊处理。

**简单来说，就是为每种语言提供一套相应的资源文件，并以规范化（国家语言代码对照表）命名的方式保存在特定的目录中，由系统自动根据客户端语言选择适合的资源文件。**

## 国家语言代码对照表：

国家/地区参数也由标准的ISO国家/地区代码表示，这些代码是由ISO-3166标准定义的，每个国家/地区由两个大写字母表示。用户可以从以下网址查看ISO-3166的标准代码：

开发中（尽量以下划线方式，例如：zh\_CN 中国 en\_US 美国

语言代码	国家/ 地区
"" (空字符串)	无变化的文化
af	公用荷兰语
af-ZA	公用荷兰语 - 南非
sq	阿尔巴尼亚
sq-AL	阿尔巴尼亚 -阿尔巴尼亚
ar	阿拉伯语
ar-DZ	阿拉伯语 -阿尔及利亚
ar-BH	阿拉伯语 -巴林
ar-EG	阿拉伯语 -埃及
ar-IQ	阿拉伯语 -伊拉克
ar-JO	阿拉伯语 -约旦
ar-KW	阿拉伯语 -科威特
ar-LB	阿拉伯语 -黎巴嫩
ar-LY	阿拉伯语 -利比亚
ar-MA	阿拉伯语 -摩洛哥
ar-OM	阿拉伯语 -阿曼
ar-QA	阿拉伯语 -卡塔尔
ar-SA	阿拉伯语 - 沙特阿拉伯
ar-SY	阿拉伯语 -叙利亚共和国
ar-TN	阿拉伯语 -北非的共和国
ar-AE	阿拉伯语 - 阿拉伯联合酋长国
ar-YE	阿拉伯语 -也门
hy	亚美尼亚
hy-AM	亚美尼亚的 -亚美尼亚
az	Azeri
az-AZ-Cyrl	Azeri-(西里尔字母的) 阿塞拜疆
az-AZ-Latn	Azeri(拉丁文)- 阿塞拜疆
eu	巴斯克

语言代码	国家/ 地区
eu-ES	巴斯克 -巴斯克
be	Belarusian
be-BY	Belarusian-白俄罗斯
bg	保加利亚
bg-BG	保加利亚 -保加利亚
ca	嘉泰罗尼亚
ca-ES	嘉泰罗尼亚 -嘉泰罗尼亚
zh-HK	华 - 香港的 SAR
zh-MO	华 - 澳门的 SAR
zh-CN	华 -中国
zh-CHS	华 (单一化)
zh-SG	华 -新加坡
zh_TW	华 -台湾
zh-CHT	华 (传统的)
hr	克罗地亚
hr-HR	克罗地亚 -克罗地亚
cs	捷克
cs-CZ	捷克 - 捷克
da	丹麦文
da-DK	丹麦文 -丹麦
div	Dhivehi
div-MV	Dhivehi-马尔代夫
nl	荷兰
nl-BE	荷兰 -比利时
nl-NL	荷兰 - 荷兰
en	英国
en-AU	英国 -澳洲
en-BZ	英国 -伯利兹

语言代码	国家/ 地区
en-CA	英国 - 加拿大
en-CB	英国 - 加勒比海
en-IE	英国 - 爱尔兰
en-JM	英国 - 牙买加
en-NZ	英国 - 新西兰
en-PH	英国 - 菲律宾共和国
en-ZA	英国 - 南非
en-TT	英国 - 千里达托贝哥共和国
en-GB	英国 - 英国
en_US	英国 - 美国
en-ZW	英国 - 津巴布韦
et	爱沙尼亚
et-EE	爱沙尼亚的 - 爱沙尼亚
fo	Faroese
fo-FO	Faroese- 法罗群岛
fa	波斯语
fa-IR	波斯语 - 伊朗王国
fi	芬兰语
fi-FI	芬兰语 - 芬兰
fr	法国
fr-BE	法国 - 比利时
fr-CA	法国 - 加拿大
fr-FR	法国 - 法国
fr-LU	法国 - 卢森堡
fr-MC	法国 - 摩纳哥
fr-CH	法国 - 瑞士
gl	加利西亚
gl-ES	加利西亚 - 加利西亚

语言代码	国家/ 地区
ka	格鲁吉亚州
ka-GE	格鲁吉亚州 -格鲁吉亚州
de	德国
de-AT	德国 -奥地利
de-DE	德国 -德国
de-LI	德国 -列支敦士登
de-LU	德国 -卢森堡
de-CH	德国 -瑞士
el	希腊
el-GR	希腊 -希腊
gu	Gujarati
gu-IN	Gujarati-印度
he	希伯来
he-IL	希伯来 -以色列
hi	北印度语
hi-IN	北印度的 -印度
hu	匈牙利
hu-HU	匈牙利的 -匈牙利
is	冰岛语
is-IS	冰岛的 -冰岛
id	印尼
id-ID	印尼 -印尼
it	意大利
it-IT	意大利 -意大利
it-CH	意大利 -瑞士
ja	日本
ja-JP	日本 -日本
kn	卡纳达语

语言代码	国家/ 地区
kn-IN	卡纳达语 -印度
kk	Kazakh
kk-KZ	Kazakh-哈萨克
kok	Konkani
kok-IN	Konkani-印度
ko	韩国
ko-KR	韩国 -韩国
ky	Kyrgyz
ky-KZ	Kyrgyz-哈萨克
lv	拉脱维亚
lv-LV	拉脱维亚的 -拉脱维亚
lt	立陶宛
lt-LT	立陶宛 -立陶宛
mk	马其顿
mk-MK	马其顿 -FYROM
ms	马来
ms-BN	马来 -汶莱
ms-MY	马来 -马来西亚
mr	马拉地语
mr-IN	马拉地语 -印度
mn	蒙古
mn-MN	蒙古 -蒙古
no	挪威
nb-NO	挪威 (Bokm?l) - 挪威
nn-NO	挪威 (Nynorsk)- 挪威
pl	波兰
pl-PL	波兰 -波兰
pt	葡萄牙

语言代码	国家/ 地区
pt-BR	葡萄牙 -巴西
pt-PT	葡萄牙 -葡萄牙
pa	Punjab 语
pa-IN	Punjab 语 -印度
ro	罗马尼亚语
ro-RO	罗马尼亚语 -罗马尼亚
ru	俄国
ru-RU	俄国 -俄国
sa	梵文
sa-IN	梵文 -印度
sr-SP-Cyrl	塞尔维亚 -(西里尔字母的) 塞尔
sr-SP-Latn	塞尔维亚 (拉丁文)- 塞尔维亚共
sk	斯洛伐克
sk-SK	斯洛伐克 -斯洛伐克
sl	斯洛文尼亚
sl-SI	斯洛文尼亚 -斯洛文尼亚
es	西班牙
es-AR	西班牙 -阿根廷
es-BO	西班牙 -玻利维亚
es-CL	西班牙 -智利
es-CO	西班牙 -哥伦比亚
es-CR	西班牙 - 哥斯达黎加
es-DO	西班牙 - 多米尼加共和国
es-EC	西班牙 -厄瓜多尔
es-SV	西班牙 - 萨尔瓦多
es-GT	西班牙 -危地马拉
es-HN	西班牙 -洪都拉斯
es-MX	西班牙 -墨西哥

语言代码	国家/ 地区
es-NI	西班牙 -尼加拉瓜
es-PA	西班牙 -巴拿马
es-PY	西班牙 -巴拉圭
es-PE	西班牙 -秘鲁
es-PR	西班牙 - 波多黎各
es-ES	西班牙 -西班牙
es-UY	西班牙 -乌拉圭
es-VE	西班牙 -委内瑞拉
sw	Swahili
sw-KE	Swahili-肯尼亚
sv	瑞典
sv-FI	瑞典 -芬兰
sv-SE	瑞典 -瑞典
syr	Syriac
syr-SY	Syriac-叙利亚共和国
ta	坦米尔
ta-IN	坦米尔 -印度
tt	Tatar
tt-RU	Tatar-俄国
te	Telugu
te-IN	Telugu-印度
th	泰国
th-TH	泰国 -泰国
tr	土耳其语
tr-TR	土耳其语 -土耳其
uk	乌克兰
uk-UA	乌克兰 -乌克兰
ur	Urdu



语言代码	国家/ 地区
ur-PK	Urdu-巴基斯坦
uz	Uzbek
uz-UZ-Cyrl	Uzbek-(西里尔字母的) 乌兹别克
uz-UZ-Latn	Uzbek(拉丁文)- 乌兹别克斯坦
vi	越南
vi-VN	越南 -越南

## 功能实现:

### 1 基于浏览器请求的实现

### 2 根据用户请求定义的语言不同实现

### 1 基于浏览器请求的实现

- 1、默认情况下，SpringMVC 根据 Accept-Language 参数 判断客户端的本地化类型。
- 2、当接受到请求时，SpringMVC 会在上下文中查找一个本地化解析器（LocalResolver）Locale，找到后使用它获取请求 所对应的本地化类型信息。
- 3、SpringMVC 还允许装配一个动态更改本地化类型的拦截 器，这样通过指定一个请求参数就可以控制单个请求的本地化类型。

### 1 在Spring配置文件中注册国际化bean

spring\_mvc.xml 中进行配置

```
<!-- spring mvc国际化 资源文件
    bean id 必须存在, 必须要叫 messageSource
-->
<bean id="messageSource"
    class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
    <!-- 到类路径下找: i18n_国际化语言编码.properties 配置文件 -->
    <property name="basename" value="classpath:i18n" />
    <property name="defaultEncoding" value="UTF-8"/>
</bean>
```

### 2 编写相应国际化资源文件

```
i18n_en_US.properties
i18n_zh_CN.properties
```

i18n\_en-US.properties

```
# commons
insert=insert
delete=delete
view=view
update=update

# user \u56FD\u9645\u5316
user.id=id
user.name=name
user.age=age
user.remark=remark
user.states.0=invalid
user.states.1=valid
```

### 3 页面中通过数据格式化完成（需要通过Servlet跳转）

jstl标签实现：

```
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<td><fmt:message key="user.name"/></td>
```

spring标签实现：

```
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>
<td><spring:message code="user.id"/></td>
```

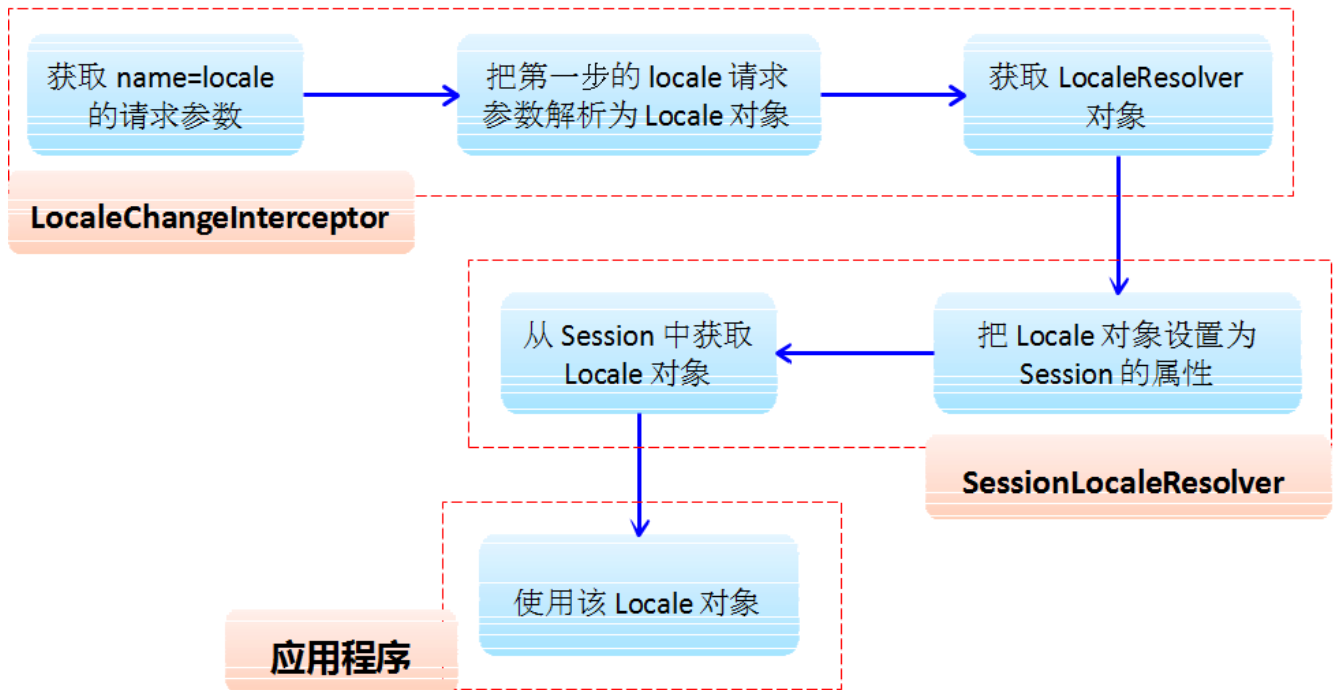
可通过浏览器的方式改变语言，进行效果查看。

## 2 根据用户请求定义的语言不同实现

指定一个请求参数就可以控制单个请求的本地化语言

SessionLocaleResolver & LocaleChangeInterceptor工作原理

中文 英语 法语



## 1 在Spring配置文件中注册国际化bean

```
<!-- 配置LocaleChangeInterceptor 主要用于获取请求中的locale信息，将期转为Locale对象，获取
LocaleResolver对象-->
<mvc:interceptors>
    <bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor"></bean>
</mvc:interceptors>

<!-- 配置SessionLocaleResolver用于将Locale对象存储于Session中供后续使用
    需要注意bean id的配置，值必须为localeResolver
-->
<bean id="localeResolver"
class="org.springframework.web.servlet.i18n.SessionLocaleResolver"></bean>
```

## 2 编写相应的语言配置

```
<a href="i18n?locale=zh_CN">中文</a>
<a href="i18n?locale=en_US">英文</a>
```

```
@RequestMapping("i18n")
public String i18n(){
    return "/index.jsp";
}
```