1. 第一步，SpringMVC配置文件

配置一个bean，ResourceBundleMessageSource，

有一个属性

<bean id=*"messageSource"*

class=*"org.springframework.context.support.ResourceBundleMessageSource"*>

<property name=*"basename"* value=*"i18n"*></property>

</bean>

1. 写国际化配置文件

在src下创建一个i18n.properties文件

属性文件的键为需要国际化的属性名，值为国际化后的值

i18n.user=User

i18n.password=Password

在创建带语言和国家后缀的配置文件

在src下创建一个i18n\_zh\_CN.properties文件

i18n.user=\u7528\u6237\u540D

i18n.password=\u5BC6\u7801

1. 导入fmt标签库

<%@taglib uri=*"http://java.sun.com/jsp/jstl/fmt"* prefix=*"fmt"* %>

使用fmt标签

<fmt:message key=*"i18n.user"*></fmt:message>

Key为配置文件中的键，这是文本，数值和日期也有对应的标签

1. 在handler方法中获取配置文件信息
2. 先获取解析国际化的对象

@Autowired

**private** ResourceBundleMessageSource messageSource;

1. 写handler方法

获取国际化配置文件中的属性值，需要在handler方法中填入参数locale

@RequestMapping("/i18n")

**public** String testI18n(Locale locale) {

//获取val值，需要三个参数，第一个为需要获取的键，第二个为获取键需要的参数，第三个为区域（中国），headler方法参数中可以自动获取这个locale

String val=messageSource.getMessage("i18n.user", **null**, locale);

System.***out***.println(val);

**return** "i18n2";

}

1. 通过超链接切换locale
2. 配置Bean

配置SessionLocaleResolver

这个对象能帮我们获取locale

<bean id=*"localeResolver"* class=*"org.springframework*

*.web.servlet.i18n.SessionLocaleResolver"*></bean>

配置LocaleChangeInterceptor

这个拦截器能帮我们改变locale

<mvc:interceptors>

<bean class=*"org.springframework.web.servlet.*

*i18n.LocaleChangeInterceptor"*></bean>

</mvc:interceptors>

<mvc:interceptors>是配置拦截器的标签

1. 通过请求参数的方式发出需要的语言

<a href=*"i18n?locale=zh\_CH"*>中文</a><br><br>

<a href=*"i18n?locale=en\_US"*>English</a>

配置的两个bean根据获取到语言然后自动设置