**程序国际化简介：**

随着网络的发展，大部分的Web站点面对的已经不是本地或者本国的浏览者，而是来 自全世界各国、各地区的浏览者。为此，国际化成为了Web站点不可或缺的一部分。

**Java国际化思想：**

1. Java国际化的思想是将程序中的信息存放在资源文件（\*.properties）中，程序根据 支持的国家及语言环境读取相应的资源文件。资源文件是key-value对，每一个资源 文件中的key是相同不变的，但value是根据不同国家语言进行定义的。
2. Java程序的国际化主要通过两个类完成。
3. java.util.Locale：用于提供本地信息，通常称它为语言环境。不同的语言、不 同的国家和地区采用不同嗯嗯Locale对象来表示
4. java.util.ResourceBundle：该类称为资源类，包含了特定语言环境的资源对象。 当程序需要一个特定的语言环境的资源时（例如字 符串资源），程序可以从适合当前用户语言环境的 资源包中加载它。采用这种方式可以编写独立于用 户语言环境的程序代码，而与特定语言环境相关的 信息则通过资源包来提供。
5. 为了实现Java程序的国际化，必须事先提供程序所需要的资源文件。资源文件的内 容由很多key-value对组成，其中key是value的唯一编号，而value是程序界面的显 示。资源文件的命名可以有3种形式：

·baseName.properties

·base\_language.properties

·baseName\_language\_country.properties

baseName是资源文件的基本名称，由用户自由定义，但是language和country必 须为Java所支持的语言和国家/地区的代码。例如：

·中国大陆：baseName\_zh\_CN.properties

·美国：baseName\_en\_US.properties

Java中的资源文件只支持ISO-8859-1编码格式字符，直接编写会出现乱码。但开发 者可以使用Java安装目录中的native2ascii.exe将中文进行转换然后在编写在资源 文件中。如果使用的是Eclipse编写资源文件，在保存资源时Eclipse会自动执行 native2ascii.exe命令，完成转码操作。

**Java支持的语言和国家：**

1）java.util.Locale类表示语言环境，该类的常用构造方法如下：

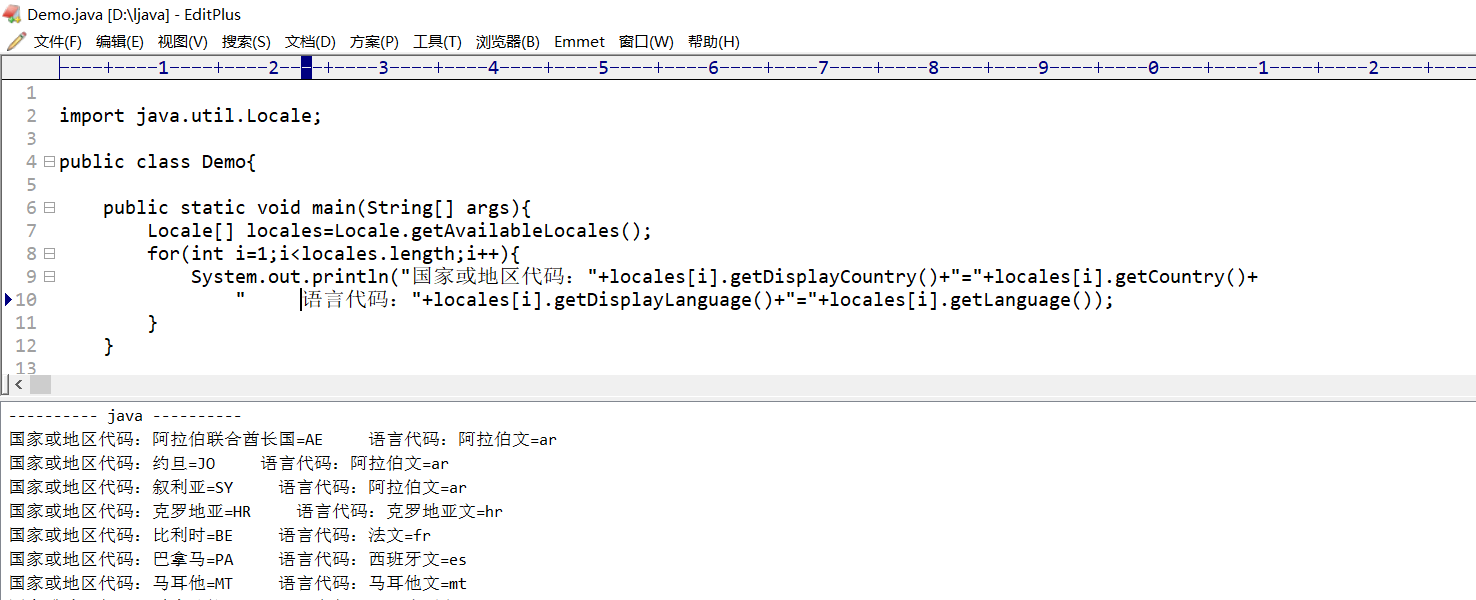
public Locale ( String language )

public Locale ( String language , String country )

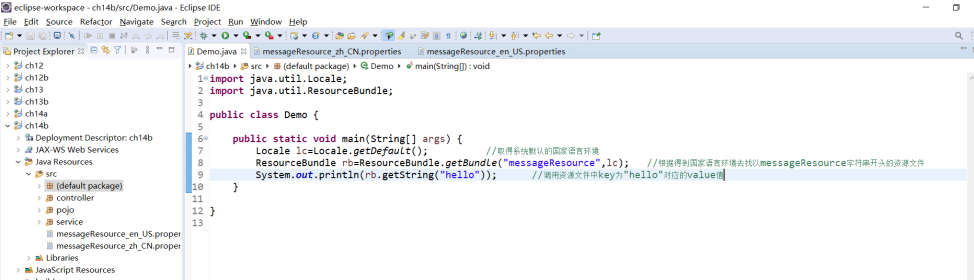
其中，language表示语言，它的取值是由小写的两个字母组成的语言代码；

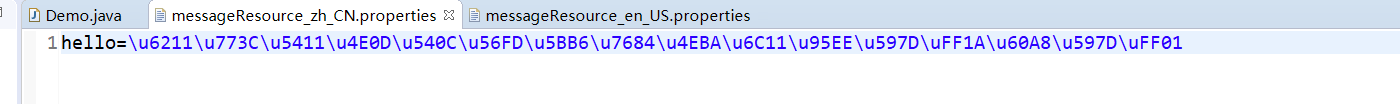
country表示国家或地区，它的取值是由大写组成的国家或地区的代码。

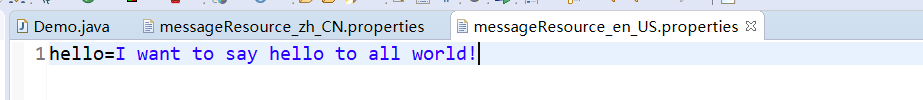
1. Java并不能支持所有国家和语言，如果需要获取Java所支持的语言和国家，开发者 可以通过调用Locale类的getAttributeLocales( )方法获取，结果返回一个Locale数组， 该数组中包含了Java所支持的语言和国家。



**实现Java程序的国际化：**







需要注意的是程序查找资源文件时具有优先级的文件，如下由高到低：

messageResource\_zh\_CN.properties

messageResource\_zh.properties

messageResource.properties

如果都没有找到，则出现异常。

**带占位符的国际化信息：**

1）在资源文件中可以为value值添加占位符，以用于动态的数据替换。例如：

hello={0} , 欢迎学习Spring MVC

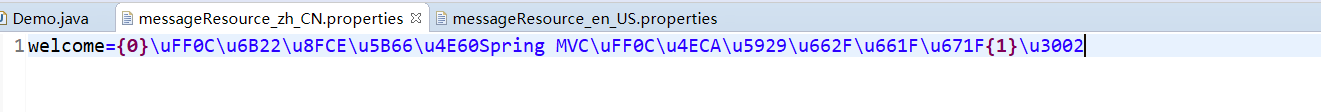
在上述中，花括号中的数字代表一个占位符，可以动态的向其填充数据。但只能使用 0~9的数字，所以说动态文件只能替换10个。例如：

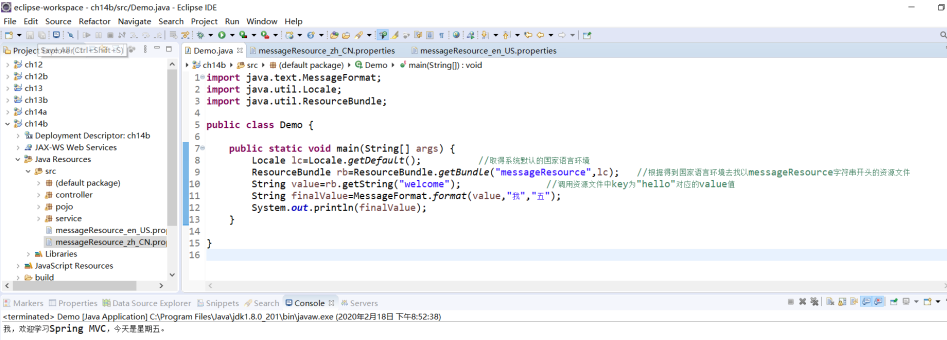
hello={0} , 欢迎学习 Spring MVC，今天星期{1}

2）要想将占位符替换为消息文本，则需要使用java.text.MessageFormat类，该类提供 了一个静态方法format( )，用来格式化带参数的文本。format( )方法的定义如下：

public staitc String format ( String pattern , Object ...arguments )

其中，pattern字符串就是一个带占位符的字符串（也就是带占位符的value值）， 消息文本中的数字占位符将按照方法参数的顺序被替换。如下：





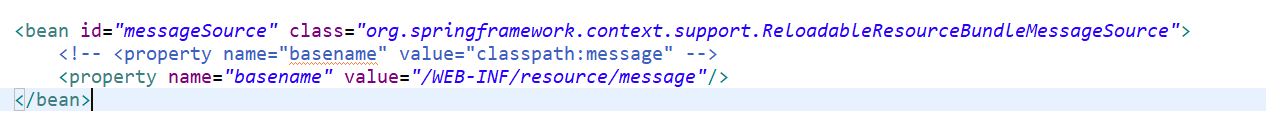
在格式化占位符时，如果参数不够，则未被格式化的占位符将会被当做字符串显示； 如果参数多于需要格式化的占位符，则会忽略。

**Spring MVC国际化简介：**

1. Spring MVC国际化是建立在Java国际化的基础之上的，只是将Java国际化的功能进 行封装和简化，式开发者使用时更简单、方便。
2. 将应用国际化时需要具备两个条件：
3. 将文本信息放到资源文件中
4. 选择和读取正确位置的资源属性文件

**Spring MVC加载国际化资源文件：**

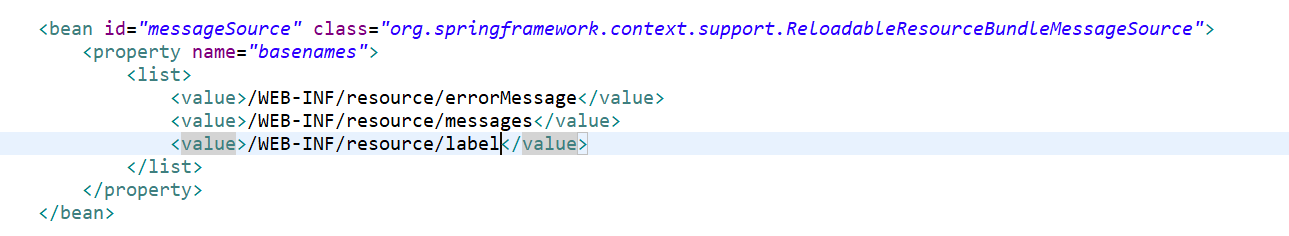
在Spring MVC中不能直接使用ResourceBundle加载资源属性文件，而是利用bean （messageSource）告知Spring MVC框架资源文件放在了哪里。在Spring MVC配置文件 中配置代码如下：



1）在上述代码中，classpath:message指的是classpath路径下的国际化资源文件；如果 国际化资源文件不是放在classpath下的，则也可以指定具体的位置。

2）另外，”messageSource”bean是ReloadableResourceBundleMessageSource类实现的， 它是不能重新加载的，如果修改了国际化资源文件，需要重启JVM。

3）如果有一组属性文件，则用basenames替换basename，如下：



配置时可以忽略后缀properties，且需要注意的是，这里不仅仅只是配置国际化资 源文件的，例如在数据验证时，如果编写了资源文件，也是放在这里的。

**语言区域的选择：**

在Spring MVC中可以使用语言区域解析器bean选择语言区域，该bean有3个常见实 现，即AcceptHeaderLocaleResolver、SessionLocaleResolver和CookieLocalResolver

1. AcceptHeanderLocaleResolver

根据浏览器Http Head（Http头信息）中的accept-language域设定，在accept-language 域中一般包含了当前操作系统的语言设定，可通过HttpServlet.getLocale( )方法获得 此域的内容。这种Locale是无法改变的，只能根据当前操作系统的语言而定。

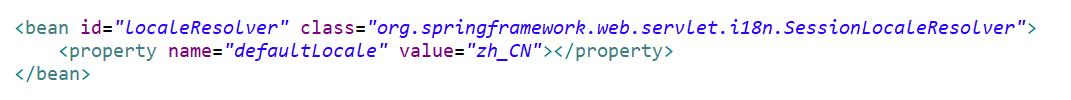
1. SessionLocaleResolver

根据用户本次会话过程中的语言设定决定语言区域（例如用户进入首页时选择语言 种类，则此次会话周期内统一使用该语言设定）。

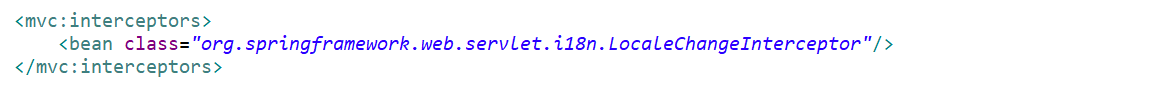
1. CookieLocaleResolver

GenuineCookie判定用户的语言设定（Cookie保存着用户上一次的语言设定参数）。

使用SessionLocaleResolver实现比较方便用户选择喜欢的语言种类。



如果采用的是基于SessionLocaleResolver和CookieLocaleResolver选择语言区域，必须 配置LocaleChangeInterceptor拦截器，如下：



**使用message标签显示国际化信息：**

1）在Spring MVC框架中可以使用Spring的message标签在JSP页面中显示国际化消息。 在使用message标签时需要在JSP使用<taglib>标签引入spring标签，代码如下：

**<%@tablib prefix=”spring” uri=”http://ww.springframework.org./tags”%>**

2）message标签有以下常用属性：

·code：获得国际化信息的key

·arguments：表示需要给占位符格式化的信息，可以由多个值。例如：

**<spring:message code=”third” arguments=”888,999”/>**

表示为key为third对应的value的占位符进行格式化，并显示最终结果。

·argumentSeparator：用来分割该标签参数的字符，默认为逗号

·text：如果code指定的key不存在或code属性不存在无法获取对应的value时所 显示的默认文本信息。