# Locale类(本地化)

Java提供了一个本地化的对象,封装了当前语言、国家、环境等特征

Locale在java.util里面

|  |
| --- |
| //模拟中国语言等环境  // Locale locale = Locale.CHINA;  Locale locale = Locale.getDefault();//获取当前系统默认的语言环境  System.out.println(locale.getCountry()); //CN 国家简称  System.out.println(locale.getDisplayCountry()); //中国 国家名称  System.out.println(locale.getLanguage()); //zh 语言简称  System.out.println(locale.getDisplayLanguage());//中文 语言名称 |

规律:

getXXX() 返回的是简称

getDisplayXXX返回的是全称

# 国际化

## 2.1静态数据国际化

网站中显示的固定文本的国际化:”用户名””密码”

国际化软件:

1. 存储所有国家显示的文本字符串

a)文件:properties

b命名:基础名\_语言简称\_国家简称.properties(注意:国家简称可以省略)

例如:msg\_zh\_CN.properties

2) 程序中获取

ResourceBundle类,可以读取国际化的资源文件,上面的.properties就是国际化的资源文件

资源文件在src/resource包下面:

msg\_en\_US.properties

|  |
| --- |
| hello=Hello  username=UserName  pwd=Password |

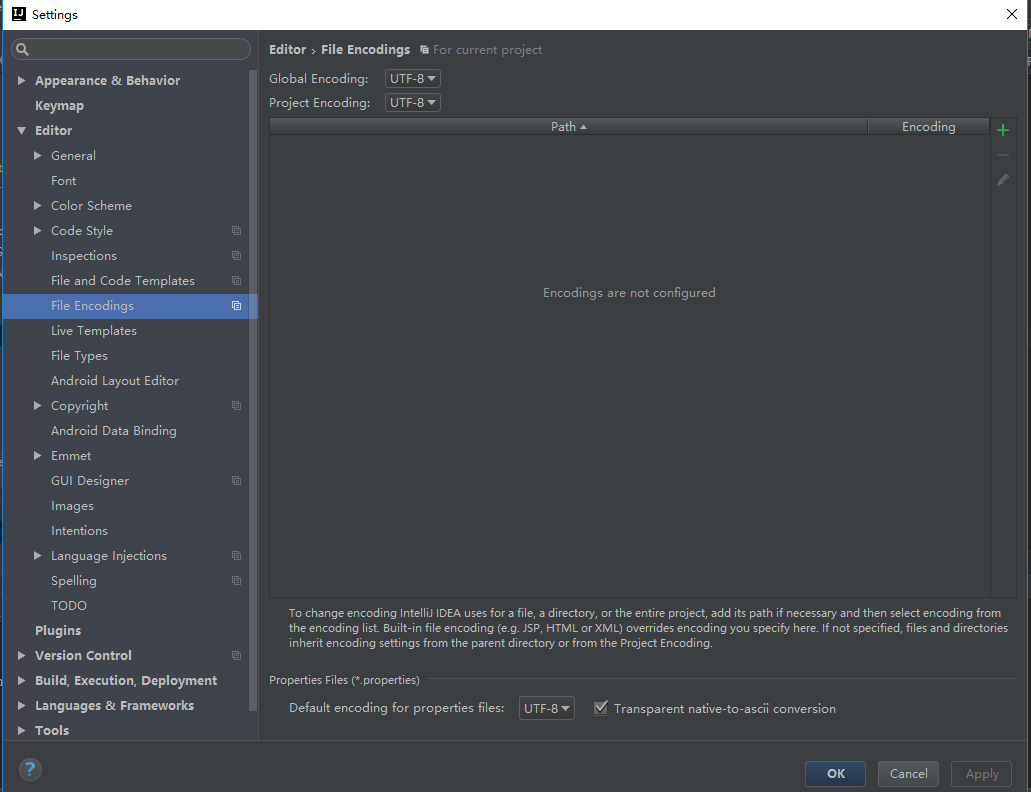
msg\_zh\_CN.properties

|  |
| --- |
| hello=你好  username=用户名  pwd=密码 |

代码:

|  |
| --- |
| //中文语言环境  Locale locale = Locale.CHINA;  // Locale locale = Locale.US;  //创建工具类对象ResourceBundle  ResourceBundle bundle = ResourceBundle.getBundle("resource.msg", locale);  //根据key获取配置文件中的值  String hello = bundle.getString("hello");  String username = bundle.getString("username");  String pwd = bundle.getString("pwd");  try {  //资源文件是gbk的编码,由于读取到的String是被编码成了iso8859-1的编码了,所以应该还原  System.out.println(new String(hello.getBytes("iso8859-1"), "gbk"));  System.out.println(new String(username.getBytes("iso8859-1"),"gbk"));  System.out.println(new String(pwd.getBytes("iso8859-1"),"gbk"));  String str = "中文";  System.out.println(new String(str.getBytes("unicode"),"unicode"));  }catch (Exception e){  } |

## idea配置解决中文乱码问题



注意:右下角的那个勾勾

## 2.3动态文本国际化

由于数值,货币,时间,日期等数据在程序中动态产生,无法像文字一样简单的

将他们从应用程序中分离出来,而是需要进行特殊处理,java提供了解决这些问题的api(java.util,java.text包中)

中文:1987-09-09 ￥1000

英文:Sep/09 1987 $1000

|  |
| --- |
| public class Main {  public static void main(String args[])throws Exception{  // test1();  // test2();  // test3();  // test4();  // test5();  // test6();  test7();  }  //国家化货币-正  public static void test2(){  //模拟语言环境  Locale locale = Locale.CHINA;  //Locale locale = Locale.US;  double d = 100;  NumberFormat currencyInstance = NumberFormat.getCurrencyInstance(locale);  //format和parse方法相反  String format = currencyInstance.format(d);  System.out.println(format); //$100.0  }  //国家化货币-反  public static void test3()throws Exception{  String str = "$100";  int num = 10;  //分析str值是哪个国家的货币  Locale us = Locale.US;  NumberFormat currencyInstance = NumberFormat.getCurrencyInstance(us);  //解析货币  Number n = currencyInstance.parse(str);  System.out.println(n.intValue()\*num);  }  //国际化数字-正  public static void test4()throws Exception{  // Locale locale = Locale.CHINA;  Locale locale = Locale.US;  NumberFormat numberInstance = NumberFormat.getNumberInstance();  String str = numberInstance.format(100000);  System.out.println(str);//100,000  }  //国际化数字-反  public static void test5()throws Exception{  String str = "100,000";  Locale locale = Locale.CHINA;  NumberFormat numberInstance = NumberFormat.getNumberInstance();  Number parse = numberInstance.parse(str);  System.out.println(parse.intValue());//100000  }  //国际化日期-正  /\*\*  \* 日期  \* FULL 2017年12月26日 星期二  \* LONG 2017年12月26日  \* MEDIUM 2017-12-26  \* SHORT 17-12-26  \*  \*  \* 时间  \* FULL 下午04时47分32秒 CST  \* LONG 下午04时47分50秒  \* MEDIUM 16:48:16  \* SHORT 下午4:48  \*/  public static void test6()throws Exception{  //日期格式  int dateStyle = DateFormat.SHORT;  //时间格式  int timeStyle = DateFormat.SHORT;  Locale locale = Locale.CHINA;  // Locale locale = Locale.US;  DateFormat dateTimeInstance = DateFormat.getDateTimeInstance(dateStyle,timeStyle,locale);  String format = dateTimeInstance.format(new Date());  System.out.println(format);  }  //国际化日期-反  public static void test7() throws Exception{  String str ="09-11-28 上午10时25分39秒 CST";  Locale locale = Locale.CHINA;  DateFormat dateTimeInstance = DateFormat.getDateTimeInstance(DateFormat.SHORT, DateFormat.FULL);  Date date = dateTimeInstance.parse(str);  System.out.println(date);  }  private static void test1() {  //国际化货币  NumberFormat currencyInstance = NumberFormat.getCurrencyInstance();  //国际化数字  NumberFormat numberInstance = NumberFormat.getNumberInstance();  //国际化百分比  NumberFormat percentInstance = NumberFormat.getPercentInstance();  //国际化日期  // DateFormat.getDateTimeInstance()  }  } |

# JSP页面国际化

## 3.1jsp语法页面国际化(不用)

资源文件在src/resource包下面:

msg\_en\_US.properties

|  |
| --- |
| hello=Hello  username=UserName  pwd=Password |

msg\_zh\_CN.properties

|  |
| --- |
| hello=你好  username=用户名  pwd=密码 |

login.jsp中:

|  |
| --- |
| <%@ page import="java.util.ResourceBundle" %>  <%@ page contentType="text/html;charset=UTF-8" language="java" %>  <html>  <head>  <%  ResourceBundle bundle = ResourceBundle.getBundle("resource.msg", request.getLocale());  %>  <title><%=bundle.getString("title")%></title>  </head>  <body>  <form action="">  <table align="center" border="1">  <tr>  <td><%=new String(bundle.getString("username").getBytes("iso8859-1"),"gbk")%></td>  <td>  <input type="text" name="userName">  </td>  </tr>  <tr>  <td><%=new String(bundle.getString("pwd").getBytes("iso8859-1"),"gbk")%></td>  <td>  <input type="password" name="pwd">  </td>  </tr>  <tr>  <td>  <input type="submit" value="<%=new String(bundle.getString("submit").getBytes("iso8859-1"),"gbk")%>">  </td>  </tr>  </table>  </form>  </body>  </html> |

## 3.2jstl国际化

<fmt:setLocale value=” ${pageContext.request.locale}”></fmt:setLocale>设置本地化对象

<fmt:setBundle basename="resource.msg" var="bundle"></fmt:setBundle> 设置工具类

注意:var属性是设置一个变量名称,就像在java代码里面: ResourceBundle bundle = ResourceBundle.getBundle("resource.msg", Locale.getDefault()); 这个变量”bundle”

<fmt:message bundle="${bundle}" key="title"></fmt:message> 显示国际化文本

注意:key的值为资源文件的key

使用jstl必须使用两个库:jstl.jar和standard.jar

|  |
| --- |
| <%@ page contentType="text/html;charset=UTF-8" language="java" %>  <%--引入jstl国际化与格式化标签库--%>  <%@taglib uri="http://java.sun.com/jstl/fmt\_rt" prefix="fmt" %>  <html>  <head>  <fmt:setLocale value="${pageContext.request.locale}"></fmt:setLocale>  <fmt:setBundle basename="resource.msg" var="bundle"></fmt:setBundle>  <title>  <fmt:message bundle="${bundle}" key="title"></fmt:message>  </title>  </head>  <body>  <form action="">  <table align="center" border="1">  <tr>  <td>  <fmt:message bundle="${bundle}" key="username"></fmt:message>  </td>  <td>  <input type="text" name="userName">  </td>  </tr>  <tr>  <td>  <fmt:message bundle="${bundle}" key="pwd"></fmt:message>  </td>  <td>  <input type="password" name="pwd">  </td>  </tr>  <tr>  <td>  <input type="submit" value="<fmt:message key="submit" bundle="${bundle}"></fmt:message>">  </td>  </tr>  </table>  </form>  </body>  </html> |

## 3.3jstl格式化

格式化日期:

<fmt:formatDate value="<%=new Date()%>" pattern="yyyyMMdd"></fmt:formatDate>