16-JSON、AJAX、i18n

讲师：王振国

今日任务

1、什么是 JSON?

JSON（JavaScript Object Notation） 是一种轻量级的数据交换格式。易于人阅读和编写。同时也易于机器解析和生成。JSON 采用完全独立于语言的文本格式，而且很多语言都提供了对 json 的支持（包括 C, C++, C#, Java, JavaScript, Perl, Python 等）。 这样就使得 JSON 成为理想的数据交换格式。

json 是一种轻量级的数据交换格式。

轻量级指的是跟 xml 做比较。

数据交换指的是客户端和服务器之间业务数据的传递格式。

1.1、JSON 在 JavaScript 中的使用。

1.1.1、json 的定义

json 是由键值对组成，并且由花括号（大括号）包围。每个键由引号引起来，键和值之间使用冒号进行分隔，多组键值对之间进行逗号进行分隔。

json 定义示例：

```
var jsonObj = {
   "key1":12,
   "key2":"abc",
   "key3":true,
   "key4":[11,"arr",false],
   "key5":{
      "key5_1" : 551,
      "key5_2" : "key5_2_value"
   },
   "key6":[{
      "key6_1_1":6611,
      "key6_1_2":"key6_1_2_value"
```

```
  },{
     "key6_2_1":6621,
     "key6_2_2":"key6_2_2_value"
  }]
};
```

1.1.2、json 的访问

json 本身是一个对象。

json 中的 key 我们可以理解为是对象中的一个属性。

json 中的 key 访问就跟访问对象的属性一样： json 对象.key

json 访问示例：

```
alert(typeof(jsonObj));// object  json 就是一个对象
alert(jsonObj.key1); //12
alert(jsonObj.key2); // abc
alert(jsonObj.key3); // true
alert(jsonObj.key4);// 得到数组[11,"arr",false]
 // json 中 数组值的遍历
for(var i = 0; i < jsonObj.key4.length; i++) {
   alert(jsonObj.key4[i]);
}
alert(jsonObj.key5.key5_1);//551
alert(jsonObj.key5.key5_2);//key5_2_value
alert( jsonObj.key6 );// 得到 json 数组

// 取出来每一个元素都是 json 对象
var jsonItem = jsonObj.key6[0];
// alert( jsonItem.key6_1_1 ); //6611
alert( jsonItem.key6_1_2 ); //key6_1_2_value
```

1.1.3、json 的两个常用方法

json 的存在有两种形式。
一种是：对象的形式存在，我们叫它 json 对象。
一种是：字符串的形式存在，我们叫它 json 字符串。
一般我们要操作 json 中的数据的时候，需要 json 对象的格式。
一般我们要在客户端和服务器之间进行数据交换的时候，使用 json 字符串。

JSON.stringify() 把 json 对象转换成为 json 字符串

JSON.parse() 把 json 字符串转换成为 json 对象

示例代码：

```
// 把 json 对象转换成为 json 字符串
var jsonObjString = JSON.stringify(jsonObj); // 特别像 Java 中对象的 toString
alert(jsonObjString)
// 把 json 字符串。转换成为 json 对象
var jsonObj2 = JSON.parse(jsonObjString);
alert(jsonObj2.key1);// 12
alert(jsonObj2.key2);// abc
```

1.2、JSON 在 java 中的使用

1.2.1、javaBean 和 json 的互转

使用阿里的Jackson

```
@Test
public void test1(){
    Person person = new Person(1,"国哥好帅!");
    // 创建 Gson 对象实例
    Gson gson = new Gson();
    // toJson 方法可以把 java 对象转换成为 json 字符串
    String personJsonString = gson.toJson(person);
    System.out.println(personJsonString);
    // fromJson 把 json 字符串转换回 Java 对象
    // 第一个参数是 json 字符串
    // 第二个参数是转换回去的 Java 对象类型
    Person person1 = gson.fromJson(personJsonString, Person.class);
    System.out.println(person1);
}
```

1.2.2、List 和 json 的互转

```
//    1.2.2、List 和 json 的互转
    @Test
    public void test2() {
        List<Person> personList = new ArrayList<>();

        personList.add(new Person(1, "国哥"));
        personList.add(new Person(2, "康师傅"));

        Gson gson = new Gson();

        // 把 List 转换为 json 字符串
        String personListJsonString = gson.toJson(personList);
        System.out.println(personListJsonString);
```

```
        List<Person> list = gson.fromJson(personListJsonString, new PersonListType().getType());
        System.out.println(list);
        Person person = list.get(0);
        System.out.println(person);
    }
```

1.2.3、map 和 json 的互转

```
//    1.2.3、map 和json 的互转
    @Test
    public void test3(){
        Map<Integer,Person> personMap = new HashMap<>();

        personMap.put(1, new Person(1, "国哥好帅"));
        personMap.put(2, new Person(2, "康师傅也好帅"));

        Gson gson = new Gson();
        // 把 map 集合转换成为 json 字符串
        String personMapJsonString = gson.toJson(personMap);
        System.out.println(personMapJsonString);

//        Map<Integer,Person> personMap2 = gson.fromJson(personMapJsonString, new
PersonMapType().getType());
        Map<Integer,Person> personMap2 = gson.fromJson(personMapJsonString, new
TypeToken<HashMap<Integer,Person>>(){}.getType());

        System.out.println(personMap2);
        Person p = personMap2.get(1);
        System.out.println(p);

    }
```

2、AJAX 请求

2.1、什么是 AJAX 请求

应用：
更新网页中某一部分的数据，其他数据不受影响。

AJAX 即“**A**synchronous **J**avascript **A**nd **X**ML”（异步 JavaScript 和 XML），是指一种创建交互式网页应用的网页开发技术。

ajax 是一种浏览器通过 js 异步发起请求，局部更新页面的技术。

Ajax 请求的局部更新，浏览器地址栏不会发生变化
局部更新不会舍弃原来页面的内容

异步请求，不会导致用户交互的阻塞，用户可以继续执行其他的操作

2.2、原生 AJAX 请求的示例：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="pragma" content="no-cache" />
    <meta http-equiv="cache-control" content="no-cache" />
    <meta http-equiv="Expires" content="0" />
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Insert title here</title>
    <script type="text/javascript">
      // 在这里使用 javaScript 语言发起 Ajax 请求，访问服务器 AjaxServlet 中 javaScriptAjax
      function ajaxRequest() {
//            1、我们首先要创建 XMLHttpRequest
        var xmlhttprequest = new XMLHttpRequest();
//            2、调用 open 方法设置请求参数

xmlhttprequest.open("GET","http://localhost:8080/16_json_ajax_i18n/ajaxServlet?action=javaScriptAj
ax",true)
//            4、在 send 方法前绑定 onreadystatechange 事件，处理请求完成后的操作。
        xmlhttprequest.onreadystatechange = function(){
          if (xmlhttprequest.readyState == 4 && xmlhttprequest.status == 200) {

            var jsonObj = JSON.parse(xmlhttprequest.responseText);

            // 把响应的数据显示在页面上
            document.getElementById("div01").innerHTML = "编号：" + jsonObj.id + "，姓名：" +
jsonObj.name;
          }
        }
//            3、调用 send 方法发送请求
        xmlhttprequest.send();
      }
    </script>
  </head>
  <body>
    <button onclick="ajaxRequest()">ajax request</button>
    <div id="div01">
    </div>
  </body>
</html>
```

2.3、jQuery 中的 AJAX 请求

$.ajax 方法

url　　表示请求的地址

type　　表示请求的类型 GET 或 POST 请求

data　　表示发送给服务器的数据

格式有两种：

一：name=value&name=value

二：{key:value}

success　　请求成功，响应的回调函数

dataType　　响应的数据类型

常用的数据类型有：

text 表示纯文本

xml 表示 xml 数据

json 表示 json 对象

```
$("#ajaxBtn").click(function(){
   $.ajax({
      url:"http://localhost:8080/16_json_ajax_i18n/ajaxServlet",
      // data:"action=jQueryAjax",
      data:{action:"jQueryAjax"},
      type:"GET",
      success:function (data) {
         // alert("服务器返回的数据是：" + data);
         // var jsonObj = JSON.parse(data);
         $("#msg").html("编号：" + data.id + "，姓名：" + data.name);
      },
      dataType : "json"
   });
});
```

$.get 方法和$.post 方法

url　　请求的 url 地址

data　　发送的数据

callback　　成功的回调函数

type　　返回的数据类型

```
// ajax--get 请求
$("#getBtn").click(function(){
   $.get("http://localhost:8080/16_json_ajax_i18n/ajaxServlet","action=jQueryGet",function (data) {
      $("#msg").html(" get 编号：" + data.id + "，姓名：" + data.name);
   },"json");
});
// ajax--post 请求
$("#postBtn").click(function(){
   $.post("http://localhost:8080/16_json_ajax_i18n/ajaxServlet","action=jQueryPost",function (data)
{
      $("#msg").html(" post 编号：" + data.id + "，姓名：" + data.name);
```

```
    },"json");
});
```

$.getJSON 方法

url	请求的 url 地址
data	发送给服务器的数据
callback	成功的回调函数

```
// ajax--getJson 请求
$("#getJSONBtn").click(function(){
    $.getJSON("http://localhost:8080/16_json_ajax_i18n/ajaxServlet","action=jQueryGetJSON",function
(data) {
        $("#msg").html(" getJSON 编号：" + data.id + " , 姓名：" + data.name);
    });
});
```

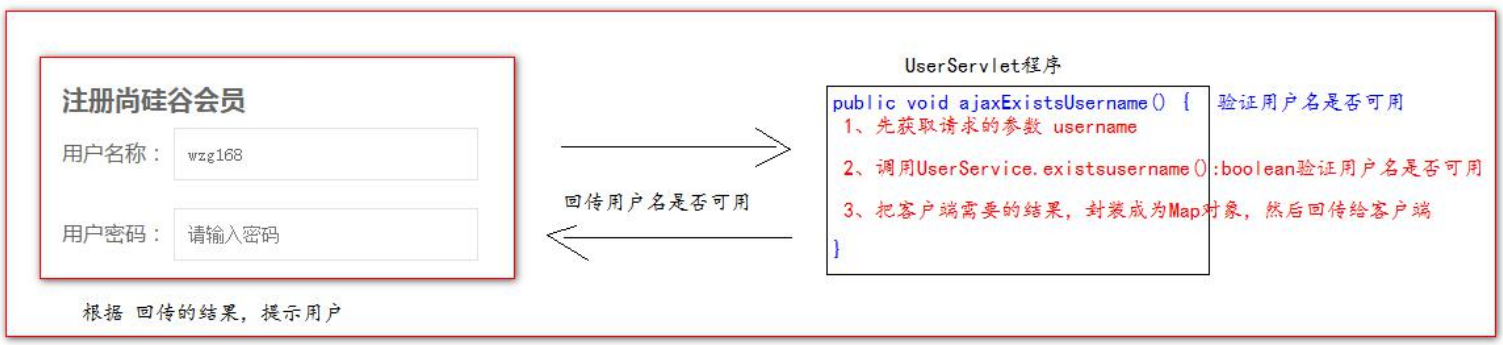
表单序列化 serialize()

serialize()可以把表单中所有表单项的内容都获取到，并以 name=value&name=value 的形式进行拼接。

```
// ajax 请求
$("#submit").click(function(){
    // 把参数序列化
    $.getJSON("http://localhost:8080/16_json_ajax_i18n/ajaxServlet","action=jQuerySerialize&" +
$("#form01").serialize(),function (data) {
        $("#msg").html(" Serialize 编号：" + data.id + " , 姓名：" + data.name);
    });
});
```

3、书城项目第九阶段

3.1、使用 AJAX 验证用户名是否可用



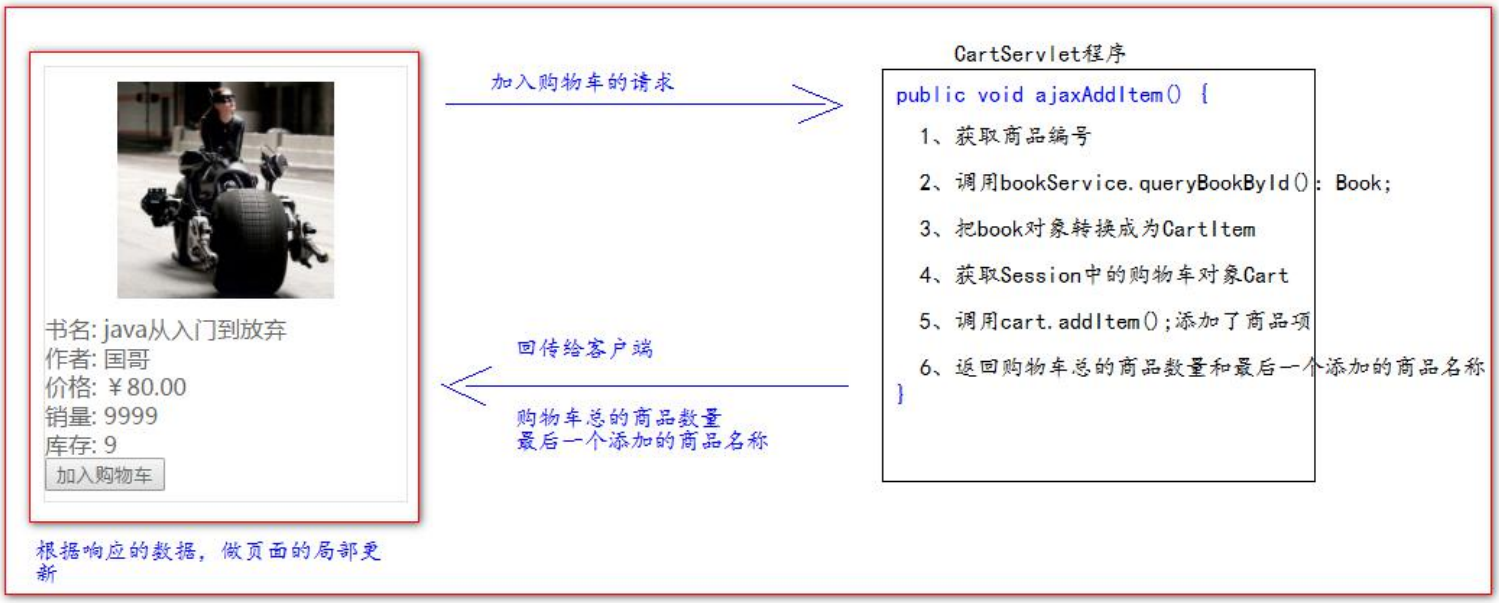
UserServlet 程序中 ajaxExistsUsername 方法：

```
protected void ajaxExistsUsername(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
    // 获取请求的参数 username
    String username = req.getParameter("username");
    // 调用 userService.existsUsername();
    boolean existsUsername = userService.existsUsername(username);
    // 把返回的结果封装成为 map 对象
    Map<String,Object> resultMap = new HashMap<>();
    resultMap.put("existsUsername",existsUsername);

    Gson gson = new Gson();
    String json = gson.toJson(resultMap);

    resp.getWriter().write(json);
}
```

regist.jsp 页面中的代码：

```
$("#username").blur(function () {
   //1 获取用户名
   var username = this.value;
   $.getJSON("http://localhost:8080/book/userServlet","action=ajaxExistsUsername&username=" +
username,function (data) {
      if (data.existsUsername) {
         $("span.errorMsg").text("用户名已存在！");
      } else {
         $("span.errorMsg").text("用户名可用！");
      }
   });
});
```

3.2、使用 AJAX 修改把商品添加到购物车



CartServlet 程序：

```
protected void ajaxAddItem(HttpServletRequest req, HttpServletResponse resp) throws ServletException,
IOException {
    // 获取请求的参数 商品编号
    int id = WebUtils.parseInt(req.getParameter("id"), 0);
    // 调用 bookService.queryBookById(id):Book 得到图书的信息
    Book book = bookService.queryBookById(id);
    // 把图书信息，转换成为 CartItem 商品项
    CartItem cartItem = new CartItem(book.getId(),book.getName(),1,book.getPrice(),book.getPrice());
    // 调用 Cart.addItem(CartItem);添加商品项
    Cart cart = (Cart) req.getSession().getAttribute("cart");
    if (cart == null) {
        cart = new Cart();
        req.getSession().setAttribute("cart",cart);
    }
    cart.addItem(cartItem);

    System.out.println(cart);
    // 最后一个添加的商品名称
    req.getSession().setAttribute("lastName", cartItem.getName());

    //6、返回购物车总的商品数量和最后一个添加的商品名称
    Map<String,Object> resultMap = new HashMap<String,Object>();

    resultMap.put("totalCount", cart.getTotalCount());
    resultMap.put("lastName",cartItem.getName());

    Gson gson = new Gson();
    String resultMapJsonString = gson.toJson(resultMap);

    resp.getWriter().write(resultMapJsonString);
```

```
}
```

pages/client/index.jsp 页面

html 代码：

```
<div style="text-align: center">
   <c:if test="${empty sessionScope.cart.items}">
      <%--购物车为空的输出--%>
      <span id="cartTotalCount"> </span>
      <div>
         <span style="color: red" id="cartLastName">当前购物车为空</span>
      </div>
   </c:if>
   <c:if test="${not empty sessionScope.cart.items}">
      <%--购物车非空的输出--%>
      <span id="cartTotalCount">您的购物车中有 ${sessionScope.cart.totalCount} 件商品</span>
      <div>
         您刚刚将<span style="color: red" id="cartLastName">${sessionScope.lastName}</span>加入到了购
物车中
      </div>
   </c:if>
</div>
```

javaScript 代码：

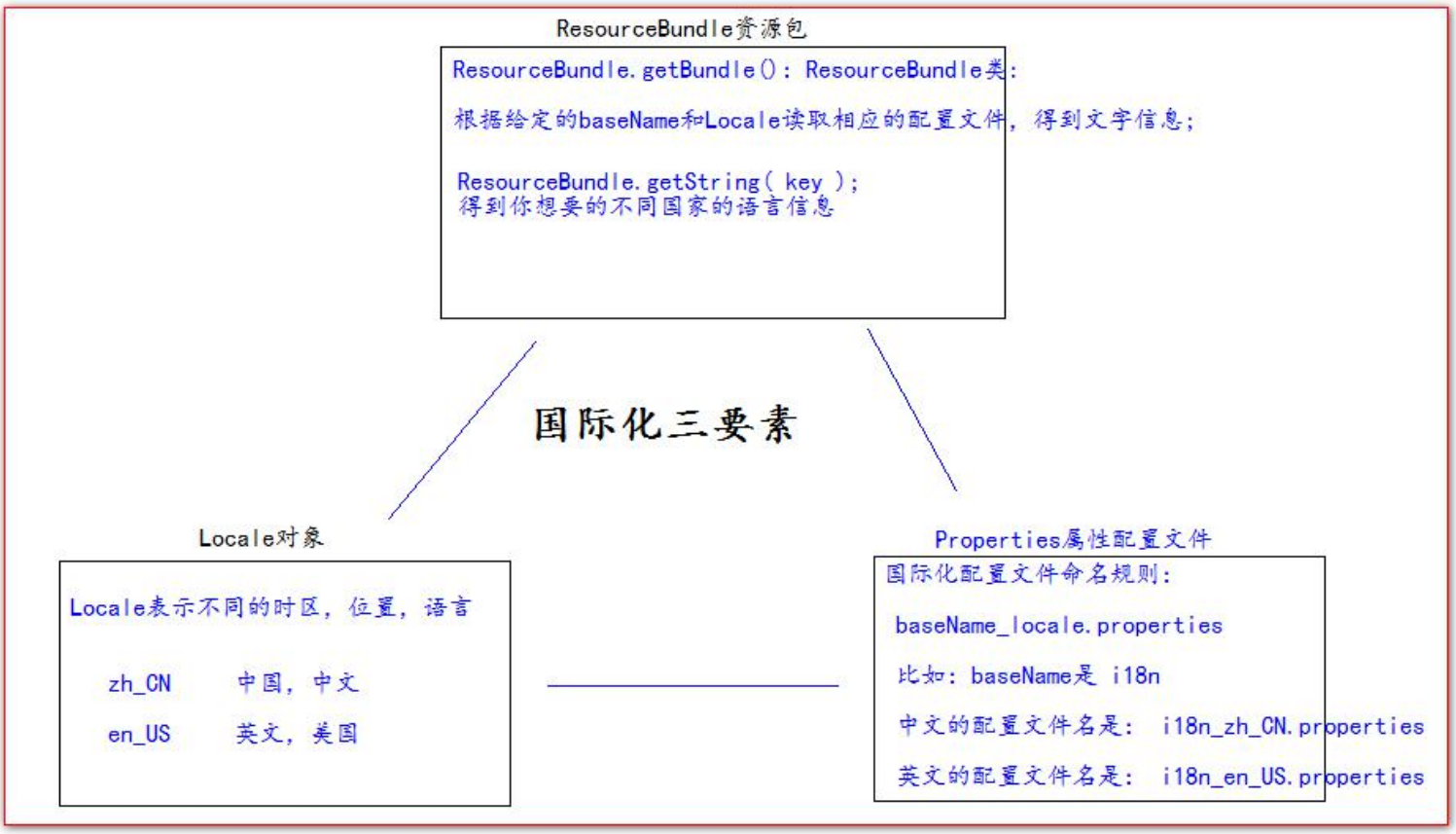
```
<Script type="text/javascript">
   $(function () {
      // 给加入购物车按钮绑定单击事件
      $("button.addToCart").click(function () {
         /**
          * 在事件响应的function 函数 中，有一个this 对象，这个this 对象，是当前正在响应事件的dom 对象
          * @type {jQuery}
          */
         var bookId = $(this).attr("bookId");
         // location.href = "http://localhost:8080/book/cartServlet?action=addItem&id=" + bookId;

         // 发ajax 请求，添加商品到购物车
         $.getJSON("http://localhost:8080/book/cartServlet","action=ajaxAddItem&id=" +
bookId,function (data) {
            $("#cartTotalCount").text("您的购物车中有 " + data.totalCount + " 件商品");
            $("#cartLastName").text(data.lastName);
         })
      });
   });
</Script>
```

4、i18n 国际化（了解内容）

4.1、什么是 i18n 国际化？

- 国际化（Internationalization）指的是同一个网站可以支持多种不同的语言，以方便不同国家，不同语种的用户访问。
- 关于国际化我们想到的最简单的方案就是为不同的国家创建不同的网站，比如苹果公司，他的英文官网是：http://www.apple.com 而中国官网是 http://www.apple.com/cn
- 苹果公司这种方案并不适合全部公司，而我们希望相同的一个网站，而不同人访问的时候可以根据用户所在的区域显示不同的语言文字，而网站的布局样式等不发生改变。
- 于是就有了我们说的国际化，国际化总的来说就是同一个网站不同国家的人来访问可以显示出不同的语言。但实际上这种需求并不强烈，一般真的有国际化需求的公司，主流采用的依然是苹果公司的那种方案，为不同的国家创建不同的页面。所以国际化的内容我们了解一下即可。
- 国际化的英文 Internationalization，但是由于拼写过长，老外想了一个简单的写法叫做 I18N，代表的是 Internationalization 这个单词，以 I 开头，以 N 结尾，而中间是 18 个字母，所以简写为 I18N。以后我们说 I18N 和国际化是一个意思。

4.2、国际化相关要素介绍



时区对象Locale;
配置文件properties;
ResouceBundle对象管理配置文件

注意配置文件的命名规则

4.3、国际化资源 properties 测试

配置两个语言的配置文件：

i18n_en_US.properties 英文

```
username=username
password=password
sex=sex
age=age
regist=regist
boy=boy
email=email
girl=girl
reset=reset
submit=submit
```

i18n_zh_CN.properties 中文

```
username=用户名
password=密码
sex=性别
age=年龄
regist=注册
boy=男
girl=女
email=邮箱
reset=重置
submit=提交
```

国际化测试代码：

```
public class I18nTest {
    @Test
    public void testLocale(){
        // 获取你系统默认的语言。国家信息
//        Locale locale = Locale.getDefault();
//        System.out.println(locale);

//        for (Locale availableLocale : Locale.getAvailableLocales()) {
//            System.out.println(availableLocale);
//        }

        // 获取中文，中文的常量的 Locale 对象
        System.out.println(Locale.CHINA);
        // 获取英文，美国的常量的 Locale 对象
        System.out.println(Locale.US);
    }
```

```
    @Test
    public void testI18n(){
        // 得到我们需要的Locale 对象
        Locale locale = Locale.CHINA;
        // 通过指定的basename 和Locale 对象，读取 相应的配置文件
        ResourceBundle bundle = ResourceBundle.getBundle("i18n", locale);

        System.out.println("username: " + bundle.getString("username"));
        System.out.println("password: " + bundle.getString("password"));
        System.out.println("Sex: " + bundle.getString("sex"));
        System.out.println("age: " + bundle.getString("age"));
    }
}
```

1. 定义配置文件，按照命名规则命名，同时将信息写入其中；
2. 通过ResourceBundle对象来管理配置文件；
3. 获取配置文件中的信息

只要请求头向服务器请求了某种时区的语言，则服务器就会返回该种语言的页面

4.4、通过请求头国际化页面

```
<%@ page import="java.util.Locale" %>
<%@ page import="java.util.ResourceBundle" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="cache-control" content="no-cache" />
<meta http-equiv="Expires" content="0" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
   <%
      // 从请求头中获取Locale 信息（语言）
      Locale locale = request.getLocale();
      System.out.println(locale);
      // 获取读取包（根据 指定的baseName 和Locale 读取 语言信息）
      ResourceBundle i18n = ResourceBundle.getBundle("i18n", locale);

   %>
   <a href="">中文</a>|
   <a href="">english</a>
   <center>
      <h1><%=i18n.getString("regist")%></h1>
      <table>
      <form>
         <tr>
            <td><%=i18n.getString("username")%></td>
```

```
            <td><input name="username" type="text" /></td>
        </tr>
        <tr>
            <td><%=i18n.getString("password")%></td>
            <td><input type="password" /></td>
        </tr>
        <tr>
            <td><%=i18n.getString("sex")%></td>
            <td>
                <input type="radio" /><%=i18n.getString("boy")%>
                <input type="radio" /><%=i18n.getString("girl")%>
            </td>
        </tr>
        <tr>
            <td><%=i18n.getString("email")%></td>
            <td><input type="text" /></td>
        </tr>
        <tr>
            <td colspan="2" align="center">
            <input type="reset" value="<%=i18n.getString("reset")%>" />  
            <input type="submit" value="<%=i18n.getString("submit")%>" /></td>
        </tr>
        </form>
    </table>
    <br /> <br /> <br /> <br />
  </center>
  国际化测试：
  <br /> 1、访问页面，通过浏览器设置，请求头信息确定国际化语言。
  <br /> 2、通过左上角，手动切换语言
</body>
</html>
```

4.5、通过显示的选择语言类型进行国际化

```
<%@ page import="java.util.Locale" %>
<%@ page import="java.util.ResourceBundle" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
      pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="cache-control" content="no-cache" />
<meta http-equiv="Expires" content="0" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
```

```
<body>
  <%
     // 从请求头中获取Locale 信息（语言）
     Locale locale = null;

     String country = request.getParameter("country");
     if ("cn".equals(country)) {
        locale = Locale.CHINA;
     } else if ("usa".equals(country)) {
        locale = Locale.US;
     } else {
        locale = request.getLocale();
     }

     System.out.println(locale);
     // 获取读取包（根据 指定的baseName 和Locale 读取 语言信息）
     ResourceBundle i18n = ResourceBundle.getBundle("i18n", locale);

  %>
  <a href="i18n.jsp?country=cn">中文</a>|
  <a href="i18n.jsp?country=usa">english</a>
  <center>
     <h1><%=i18n.getString("regist")%></h1>
     <table>
     <form>
        <tr>
           <td><%=i18n.getString("username")%></td>
           <td><input name="username" type="text" /></td>
        </tr>
        <tr>
           <td><%=i18n.getString("password")%></td>
           <td><input type="password" /></td>
        </tr>
        <tr>
           <td><%=i18n.getString("sex")%></td>
           <td>
              <input type="radio" /><%=i18n.getString("boy")%>
              <input type="radio" /><%=i18n.getString("girl")%>
           </td>
        </tr>
        <tr>
           <td><%=i18n.getString("email")%></td>
           <td><input type="text" /></td>
        </tr>
        <tr>
           <td colspan="2" align="center">
           <input type="reset" value="<%=i18n.getString("reset")%>" />  
           <input type="submit" value="<%=i18n.getString("submit")%>" /></td>
        </tr>
```

```
        </form>
      </table>
      <br /> <br /> <br /> <br />
   </center>
   国际化测试：
   <br /> 1、访问页面，通过浏览器设置，请求头信息确定国际化语言。
   <br /> 2、通过左上角，手动切换语言
</body>
</html>
```

4.6、JSTL 标签库实现国际化

```
    <%--1 使用标签设置 Locale 信息--%>
   <fmt:setLocale value="" />
    <%--2 使用标签设置 baseName--%>
    <fmt:setBundle basename=""/>
    <%--3 输出指定 key 的国际化信息--%>
    <fmt:message key="" />
```

```
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="pragma" content="no-cache" />
<meta http-equiv="cache-control" content="no-cache" />
<meta http-equiv="Expires" content="0" />
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
   <%--1 使用标签设置 Locale 信息--%>
   <fmt:setLocale value="${param.locale}" />
   <%--2 使用标签设置 baseName--%>
   <fmt:setBundle basename="i18n"/>


   <a href="i18n_fmt.jsp?locale=zh_CN">中文</a>|
   <a href="i18n_fmt.jsp?locale=en_US">english</a>
   <center>
      <h1><fmt:message key="regist" /></h1>
      <table>
      <form>
         <tr>
```

```
            <td><fmt:message key="username" /></td>
            <td><input name="username" type="text" /></td>
        </tr>
        <tr>
            <td><fmt:message key="password" /></td>
            <td><input type="password" /></td>
        </tr>
        <tr>
            <td><fmt:message key="sex" /></td>
            <td>
                <input type="radio" /><fmt:message key="boy" />
                <input type="radio" /><fmt:message key="girl" />
            </td>
        </tr>
        <tr>
            <td><fmt:message key="email" /></td>
            <td><input type="text" /></td>
        </tr>
        <tr>
            <td colspan="2" align="center">
            <input type="reset" value="<fmt:message key="reset" />" />  
            <input type="submit" value="<fmt:message key="submit" />" /></td>
        </tr>
        </form>
    </table>
    <br /> <br /> <br /> <br />
  </center>
</body>
</html>
```