

Servlet 国际化

在我们开始之前，先来看看三个重要术语：

- **国际化 (i18n)**：这意味着一个网站提供了不同版本的翻译成访问者的语言或国籍的内容。
- **本地化 (l10n)**：这意味着向网站添加资源，以使其适应特定的地理或文化区域，例如网站翻译成印地文 (Hindi)。
- **区域设置 (locale)**：这是一个特殊的文化或地理区域。它通常指语言符号后跟一个下划线和一个国家符号。例如 "en_US" 表示针对 US 的英语区域设置。

当建立一个全球性的网站时有一些注意事项。本教程不会讲解这些注意事项的完整细节，但它会通过一个很好的实例向您演示如何通过差异化定位（即区域设置）来让网页以不同语言呈现。

Servlet 可以根据请求者的区域设置拾取相应版本的网站，并根据当地的语言、文化和需求提供相应的网站版本。以下是 request 对象中返回 Locale 对象的方法。

```
java.util.Locale request.getLocale()
```

检测区域设置

下面列出了重要的区域设置方法，您可以使用它们来检测请求者的地理位置、语言和区域设置。下面所有的方法都显示了请求者浏览器中设置的国家名称和语言名称。

| 序号 | 方法 & 描述 |
|----|---------------------------------------------------------------------------|
| 1 | String getCountry() 该方法以 2 个大写字母形式的 ISO 3166 格式返回该区域设置的国家/地区代码。 |
| 2 | String getDisplayCountry() 该方法返回适合向用户显示的区域设置的国家的名称。 |
| 3 | String getLanguage() 该方法以小写字母形式的 ISO 639 格式返回该区域设置的语言代码。 |
| 4 | String getDisplayLanguage() 该方法返回适合向用户显示的区域设置的语言的名称。 |
| 5 | String getISO3Country() 该方法返回该区域设置的国家的三个字母缩写。 |
| 6 | String getISO3Language() 该方法返回该区域设置的语言的三个字母的缩写。 |

实例

本实例演示了如何显示某个请求的语言和相关的国家：

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Locale;

public class GetLocale extends HttpServlet{

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException
    {
        // 获取客户端的区域设置
        Locale locale = request.getLocale();
        String language = locale.getLanguage();
        String country = locale.getCountry();

        // 设置响应内容类型
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        String title = "检测区域设置";
        String docType = "<!DOCTYPE html> \n";
        out.println(docType +
            "<html>\n" +
            "<head><title>" + title + "</title></head>\n" +
            "<body bgcolor=\"#f0f0f0\">\n" +
            "<h1 align=\"center\">" + language + "</h1>\n" +
            "<h2 align=\"center\">" + country + "</h2>\n" +
            "</body></html>");
    }
}
```

语言设置

Servlet 可以输出以西欧语言（如英语、西班牙语、德语、法语、意大利语、荷兰语等）编写的页面。在这里，为了能正确显示所有的字符，设置 Content-Language 头是非常重要的。

第二点是使用 HTML 实体显示所有的特殊字符，例如，"ñ" 表示 "ñ"，"¡" 表示 "¡"，如下所示：

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Locale;
```

```
public class DisplaySpanish extends HttpServlet{

    public void doGet(HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException
    {
        // 设置响应内容类型
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        // 设置西班牙语代码
        response.setHeader("Content-Language", "es");

        String title = "En Espa&ntilde;ol";
        String docType = "<!DOCTYPE html> \n";
        out.println(docType +
            "<html>\n" +
            "<head><title>" + title + "</title></head>\n" +
            "<body bgcolor=\"#f0f0f0\">\n" +
            "<h1>" + "En Espa&ntilde;ol:" + "</h1>\n" +
            "<h1>" + "&iexcl;Hola Mundo!" + "</h1>\n" +
            "</body></html>");
    }
}
```

特定于区域设置的日期

您可以使用 `java.text.DateFormat` 类及其静态方法 `getDateTimeInstance()` 来格式化特定于区域设置的日期和时间。下面的实例演示了如何格式化特定于某个给定的区域设置的日期：

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Locale;
import java.text.DateFormat;
import java.util.Date;

public class DateLocale extends HttpServlet{

    public void doGet(HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException
    {
        // 设置响应内容类型
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        // 获取客户端的区域设置
```

```
Locale locale = request.getLocale( );
String date = DateFormat.getDateInstance(
    DateFormat.FULL,
    DateFormat.SHORT,
    locale).format(new Date( ));

String title = "特定于区域设置的日期";
String docType = "<!DOCTYPE html> \n";
out.println(docType +
    "<html>\n" +
    "<head><title>" + title + "</title></head>\n" +
    "<body bgcolor=\"#f0f0f0\">\n" +
    "<h1 align=\"center\">" + date + "</h1>\n" +
    "</body></html>");
}
}
```

特定于区域设置的货币

您可以使用 `java.text.NumberFormat` 类及其静态方法 `getCurrencyInstance()` 来格式化数字（比如 `long` 类型或 `double` 类型）为特定于区域设置的货币。下面的实例演示了如何格式化特定于某个给定的区域设置的货币：

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Locale;
import java.text.NumberFormat;
import java.util.Date;

public class CurrencyLocale extends HttpServlet{

    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
    {
        // 设置响应内容类型
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        // 获取客户端的区域设置
        Locale locale = request.getLocale( );
        NumberFormat nft = NumberFormat.getCurrencyInstance(locale);
        String formattedCurr = nft.format(1000000);

        String title = "特定于区域设置的货币";
        String docType = "<!DOCTYPE html> \n";
        out.println(docType +
            "<html>\n" +
```

```
"<head><title>" + title + "</title></head>\n" +
"<body bgcolor=\"#f0f0f0\">\n" +
"<h1 align=\"center\">" + formattedCurr + "</h1>\n" +
"</body></html>");
}
}
```

特定于区域设置的百分比

您可以使用 `java.text.NumberFormat` 类及其静态方法 `getPercentInstance()` 来格式化特定于区域设置的百分比。下面的实例演示了如何格式化特定于某个给定的区域设置的百分比：

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.util.Locale;
import java.text.NumberFormat;
import java.util.Date;

public class PercentageLocale extends HttpServlet{

    public void doGet(HttpServletRequest request,
                       HttpServletResponse response)
        throws ServletException, IOException
    {
        // 设置响应内容类型
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();
        // 获取客户端的区域设置
        Locale locale = request.getLocale( );
        NumberFormat nft = NumberFormat.getPercentInstance(locale);
        String formattedPerc = nft.format(0.51);

        String title = "特定于区域设置的百分比";
        String docType = "<!DOCTYPE html> \n";
        out.println(docType +
            "<html>\n" +
            "<head><title>" + title + "</title></head>\n" +
            "<body bgcolor=\"#f0f0f0\">\n" +
            "<h1 align=\"center\">" + formattedPerc + "</h1>\n" +
            "</body></html>");
    }
}
```

 点我分享笔记