

제15장

다국어 처리



1. 다국어 처리의 개요

❖ 다국어 처리

- 웹 브라우저를 사용하는 국가에 따라 다양한 언어 및 지역을 지원하는 서비스
- 다른 언어와 지역적 차이를 기술 변경 없이 소프트웨어에 바로 적용하는 것
- JSP 페이지에 JSTL의 `fmt` 태그를 이용하면 언어별로 페이지를 따로 만들 필요 없이 아주 간단하게 다국어를 지원할 수 있음
- 다국어는 다양한 언어와 지역에 적용될 수 있도록 하는 국제화(internationalization, i18n)와 언어별 구성 요소를 추가하여 특정 지역의 언어나 문화에 맞추는 지역화(localization, L10n)를 포함

1. 다국어 처리의 개요

❖ 1.1 지역화

- 사용 국가별 환경에서 특정 언어와 지역에 맞게 적합화하는 것
- L10n으로 표기
- 지역화에 주로 고려되는 사항
 - 숫자, 날짜, 시간의 형식
 - 화폐의 표시
 - 키보드의 지원
 - 문자열의 순서와 정렬
 - 심벌, 아이콘, 색상
 - 문화에 따라 오해의 소지가 있거나 의미가 없는 문자, 그림
 - 지역별 법률의 차이

1. 다국어 처리의 개요

❖ 1.2 국제화

- 여러 국가에서 사용할 수 있도록 다국어를 지원하는 것
- i18n으로 표기
- 국제화는 어느 국가에서나 사용할 수 있게 하는 지역화 기능을 포함
- 국제화는 주로 다음과 같은 처리를 포함하여 지원해야 함
 - 국제화 지원은 유니코드의 사용이나 기존의 인코딩을 적절히 처리하고 사용자 인터페이스에 표시할 문자열에는 문자 코드가 포함되지 않도록 설계 및 개발해야 함.
 - 국제화를 처리하는 정보에 언어 정보를 포함하거나, 세로쓰기/가로쓰기/우측에서 좌측으로의 가로쓰기 등 언어의 특성을 반영하는 처리 등을 지원해야 함.
 - 날짜와 시간 표시, 지역의 달력, 숫자 표시, 리스트의 정렬과 표시, 인명이나 주소의 처리 등 언어의 특성(언어적·지역적·문화적 특성 등)에 대한 사용자 설정을 지원해야 함
 - 국제화는 사용자의 요청이나 설정에 따라 필요시 사용되도록 지역화 정보를 코드와 분리해야 함

2. Locale 클래스를 이용한 다국어 처리

❖ Locale 클래스

- 특정 지리적·정치적·문화적 지역을 나타내는 클래스
- 사용자의 지역 환경에 따라 결정되는 지역적 문화(언어, 날짜, 시간 등)의 정보를 담고 있음
 - 예를 들어 웹 페이지에 보이는 메시지가 여러 가지 언어로 주어졌을 때 사용자가 어떤 언어로 출력할 것인지 결정할 수 있게 하는 수단이 바로 Locale 클래스
- Locale 클래스는 단순한 메시지뿐 아니라 숫자, 날짜, 시간 등을 표현하는 데 사용
- Locale 객체의 생성은 request 내장 객체를 이용하여 현재 웹 브라우저에 미리 정의된 언어나 국가 정보를 가져오는 방법

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

- Locale 클래스를 사용하려면 JSP 페이지에 page 디렉티브 태그의 import 속성으로 패키지 java.util.Locale을 설정

2. Locale 클래스를 이용한 다국어 처리

❖ 2.1 로케일 감지하기

- 웹 브라우저에 설정된 국가와 언어 이름을 알아내는 것을 로케일 감지라고 함

- 로케일 감지 메소드의 종류

메소드	반환 유형	설명
getDefault()	static Locale	디폴트 로케일의 현재 값을 가져옵니다.
getCountry()	String	현재 로케일의 국가/지역 코드(대문자)를 가져옵니다.
getDisplayCountry()	String	현재 로케일의 국가 이름을 가져옵니다.
getLanguage()	String	현재 로케일의 언어 코드(소문자)를 가져옵니다.
getDisplayLanguage()	String	현재 로케일의 언어 이름을 얻어옵니다.

2. Locale 클래스를 이용한 다국어 처리

[로케일 감지 메소드 사용 예]

```
<% page contentType="text/html; charset=utf-8"%>
<% page import="java.util.Locale"%>
<%
    Locale locale = request.getLocale();
    String displayLanguage = locale.getDisplayLanguage();
    String language = locale.getLanguage();
    String displayCountry = locale.getDisplayCountry();
    String country = locale.getCountry();
%>
<html>
<head>
<title>Localization</title>
</head>
<body>
    <%
        out.println("로케일 언어 : " + displayLanguage + "<br />");
        out.println("로케일 언어 코드 : " + language + "<br />");
        out.println("로케일 국가 : " + displayCountry + "<br />");
        out.println("로케일 국가 코드 : " + country + "<br />");
    %>
</body>
</html>
```

로케일 언어	- 한국어
로케일 언어 코드	: ko
로케일 국가	: 대한민국
로케일 국가 코드	: KR

2. Locale 클래스를 이용한 다국어 처리

❖ 2.2 로케일 표현하기

■ 언어 설정

- 다양한 국가별 언어를 제대로 표현하기 위해 `response` 내장 객체의 `setHeader()` 메소드를 사용

[로케일의 언어 설정 예]

```
<% page contentType="text/html; charset=utf-8"%>
<% page import="java.util.Locale"%>
<%
    response.setContentType("text/html");
    response.setHeader("Content-Language", "es");
    String title = "Español";
%>
<html>
<head>
<title>
    <% out.print(title); %>
</title>
</head>
<body>
    <p>Idioma : Español</p>
    <p>&iexcl;Hola Mundo!</p>
</body>
</html>
```

Idioma : Español
¡Hola Mundo!

2. Locale 클래스를 이용한 다국어 처리

❖ 2.2 로케일 표현하기

■ 날짜와 시간 설정

- 사용자의 로케일에 따라 특정 날짜와 시간 형식을 표현하기 위해 **DateFormat** 클래스의 **getDateTimeInstance()** 메소드를 사용

[로케일의 날짜와 시간 설정 예]

```
<% page contentType="text/html; charset=utf-8"%>
<% page import="java.util.Locale,java.util.Date"%>
<% page import="java.text.DateFormat"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
    <%
        Locale locale = request.getLocale();
        String date = DateFormat.getDateTimeInstance(DateFormat.FULL,
            DateFormat.SHORT, locale).format(new Date());
    %>
    <p>로케일의 날짜 형식
    <p> <% out.print(date); %>
</body>
</html>
```

로케일의 날짜 형식

2016년 9월 5일 수요일 오후 4:32

2. Locale 클래스를 이용한 다국어 처리

❖ 2.2 로케일 표현하기

■ 통화와 숫자 설정

- 사용자의 로케일에 따라 특정 통화와 숫자를 표현하기 위해 **NumberFormat** 클래스의 **getCurrencyInstance()** 메소드를 사용

[로케일의 통화와 숫자 설정 예]

```
<% page contentType="text/html; charset=utf-8"%>
<% page import="java.util.Locale,java.util.Date"%>
<% page import="java.text.NumberFormat"%>
<html>
<head>

<title>Internationalization</title>
</head>
<body>
    <%
        Locale locale = request.getLocale();
        NumberFormat currency = NumberFormat.getCurrencyInstance(locale);
        NumberFormat percentage = NumberFormat.getPercentInstance(locale);
        String fcurrency = currency.format(1234567);
        String fpercentage = percentage.format(0.25);
    %>
    <p> 로케일의 통화 형식 : <% out.print(fcurrency); %>
    <p> 로케일의 비율 형식 : <% out.print(fpercentage); %></p>
</body>
</html>
```

로케일의 통화 형식 : ₩1,234,567

로케일의 비율 형식 : 25%

3. JSTL fmt 태그를 이용한 다국어 처리

❖ JSTL fmt 태그

- 다국어 문서 처리를 위한 국제화 및 지역화 태그
- 날짜와 숫자 등을 형식화하는 기능을 제공하는 JSTL 라이브러리인 JSTL fmt 태그는 특정 지역에 따라 다른 메시지를 출력할 때 사용
 - 한글 웹 브라우저는 한글 메시지를 출력할 때, 영문 웹 브라우저는 영어 메시지를 출력할 때 유용

❖ JSTL fmt 태그 라이브러리를 사용 방법

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

- JSTL 라이브러리인 jstl.jar 파일이 필요.
 - 배포 사이트: 아파치 홈페이지
 - 다운로드 파일: jstl-1.2.5.jar

3. JSTL fmt 태그를 이용한 다국어 처리

❖ JSTL fmt 태그의 종류

구분	태그 유형	설명
Locale 설정	setLocale	로케일을 설정합니다.
	requestEncoding	요청 파라미터의 문자 인코딩을 설정합니다.
메시지 처리	bundle	사용할 리소스بن들을 설정합니다.
	message	리소스بن들에서 로케일에 맞는 메시지를 가져와 출력합니다.
	setBundle	리소스بن들을 읽어와 특정 변수에 저장합니다.
구분	태그 유형	설명
날짜	formatDate	날짜 형식을 표현합니다.
	parseDate	문자열에서 원하는 패턴의 날짜 형식으로 변환합니다.
숫자	parseNumber	문자열에서 원하는 패턴의 숫자 형식으로 변환합니다.
	formatNumber	숫자 형식을 표현합니다.
시간	setTimeZone	특정 범위의 시간대를 설정합니다.
	timeZone	시간대를 설정합니다.

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.1 로케일 설정 태그의 기능과 사용법

■ setLocale 태그

- 국제화 태그가 사용할 로케일을 설정하는 태그

```
<fmt:setLocale  
  value="언어 코드[_국가 코드]"  
  [scope="{page|request|session|application}"]/> //기본 값은 page
```

[setLocale 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>  
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>  
<html>  
  <head>  
    <title>Internationalization</title>  
  </head>  
  <body>  
    <%=response.getLocale()%>  
  
    <fmt:setLocale value="ko" />  
    <p><%=response.getLocale()%>  
  
    <fmt:setLocale value="ja" />  
    <p><%=response.getLocale()%>  
  
    <fmt:setLocale value="en" />  
    <p><%=response.getLocale()%>  
  </body>  
</html>
```

ko_KR
ko
ja
en

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.1 로케일 설정 태그의 기능과 사용법

- requestEncoding 태그
 - 요청 파라미터의 문자 인코딩을 설정하는 태그

```
<fmt:requestEncoding value="문자 인코딩"/>
```

[requestEncoding 태그 사용 예]

```
<%@ page contentType="text/html; charset=euc-kr"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
  <fmt:requestEncoding value="euc-kr" />
  요청 파라미터 :
  <%
    out.print(request.getParameter("id"));
  %>
  <form method="post" action="#">
    <p> 아이디 <input type="text" name="id">
      <input type="submit" value="전송">
    </p>
  </form>
</body>
</html>
```

request객체의 setCharacterEncoding()과 동일한 기능을 한다.

요청 파라미터 : euc	
아이디	<input type="text" value="홍길동"/>
<input type="button" value="전송"/>	
요청 파라미터 : euc	
아이디	<input type="text" value="홍길동"/>
<input type="button" value="전송"/>	

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.2 메시지 처리 태그의 기능과 사용법

■ 리소스번들

- 메시지 처리 태그에서 사용하는 파일로 메시지 번들이라고도 함
- 리소스번들로 사용하는 파일은 보통 WEB-INF/classes/ 폴더에 있음
- 리소스번들은 `java.util.Properties` 클래스에 정의된 방법으로 메시지를 읽어오기 때문에 확장자가 `properties`인 파일이 반드시 있어야 함.
 - `java.util.Properties` 클래스는 알파벳이나 숫자, 라틴 문자 외의 언어를 유니코드 값으로 표현

■ 리소스번들로 사용하는 *.properties 파일의 종류

*.properties 파일	설명
파일 이름: properties	기본 메시지일 때 사용합니다.
파일 이름: ko.properties	한글 메시지일 때 사용합니다.
파일 이름: en.properties	영어 메시지일 때 사용합니다.

[한글, 영문 리소스번들 작성 예]

```
//message_ko.properties 파일인 경우  
NAME = 홍길순
```

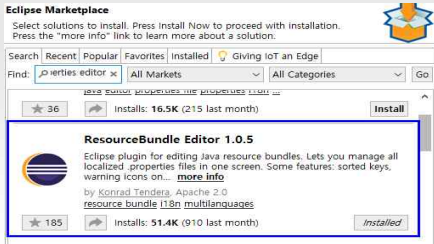
```
//message_en.properties 파일인 경우  
NAME = your name
```

Bundle? 영어로 묶음이라는 뜻이며 보통 상품을 구매했을 때 끼워주는 걸 말한다.
스마트폰을 구입했을 때 나누어 주는 이어폰이나 PC를 구입할 때 끼워주는 소프트웨어 등이다.

리소스 번들로 사용하는 *.properties파일은 보통 WEB-INF/classes/폴더 안에 위치한다.
보통 패키지를 만들어 관리를 하며, 로케일이 ko이면 xxxBundle_ko.properties를 읽어오며, 로케일이 롤리면 xxxBundle.properties처럼 언어코드가 붙어있지 않은 파일을 읽어온다.

3. JSTL fmt 태그를 이용한 다국어 처리

*.properties파일의 내용은 한글로 입력하면 안된다. 다시 말해 유니코드로 입력해야 된다.
그렇게 되면 한글자씩 입력할 때 마다, 유니코드를 찾아야 하는 불편함이 발생한다.
하지만 아래 **ResourceBundle Editor**가 그 불편함을 해소해주니 이클립스에서 플러그인으로 설치 하도록 하자.



The screenshot shows the Eclipse Marketplace interface. At the top, it says "Eclipse Marketplace" and "Select solutions to install. Press Install Now to proceed with installation. Press the 'more info' link to learn more about a solution." Below this is a search bar with "properties editor" entered. The search results show a list of plugins. The "ResourceBundle Editor 1.0.5" plugin is highlighted with a blue box. It has a star rating of 36, 16.5K installs, and an "Install" button. The description says "Eclipse plugin for editing Java resource bundles. Lets you manage all localized .properties files in one screen. Some features: sorted keys, warning icons on..." and includes a "more info" link. The author is listed as "by Konrad Tenders, Apache 2.0 resource bundle i18n multilanguages". At the bottom, it shows a star rating of 185, 51.4K installs, and an "Installed" button.

Eclipse Marketplace

Select solutions to install. Press Install Now to proceed with installation.
Press the "more info" link to learn more about a solution.

Search Recent Popular Favorites Installed Giving IoT an Edge

Find: properties editor x All Markets All Categories Go

java editor properties file properties editor ...

★ 36 Installs: 16.5K (215 last month) Install

ResourceBundle Editor 1.0.5

Eclipse plugin for editing Java resource bundles. Lets you manage all localized .properties files in one screen. Some features: sorted keys, warning icons on... [more info](#)

by Konrad Tenders, Apache 2.0
resource bundle i18n multilanguages

★ 185 Installs: 51.4K (910 last month) Installed

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.2 메시지 처리 태그의 기능과 사용법

■ bundle 태그

- 사용할 리소스변들을 설정하는 태그로 형식

```
<fmt:bundle basename="리소스변들" [prefix="key 이름"]>  
    //body 내용  
</fmt:bundle>
```

■ message 태그

- bundle 태그에 설정한 리소스변들에서 메시지를 읽어와 출력하는 태그

```
<fmt:message  
    key="메시지 key 이름"  
    [bundle="setBundle 태그의 변수 이름" ]  
    [var="메시지를 저장하는 변수 이름"]  
    [scope="{page|request|session|application}"] /> //기본 값은 page
```

3. JSTL fmt 태그를 이용한 다국어 처리

[bundle 태그와 message 태그 사용 예]

//message.properties 파일인 경우

name = admin

hello = How are You

//message_ko.properties 파일인 경우

name = \uAD00\uB9AC\uC790 //"관리자" 유니코드

hello = \uC548\uB155\uD558\uC138\uC694 //"안녕하세요" 유니코드

```
<% page contentType="text/html; charset=utf-8"%>
<% taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
    <fmt:bundle basename="resourceBundle.message">
        <p> <fmt:message key="name" />
            <fmt:message key="hello" var="msg" />
            <p>${msg}
        </fmt:bundle>
    </body>
</html>
```

관리자

안녕하세요

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.2 메시지 처리 태그의 기능과 사용법

■ **setBundle** 태그

- 리소스변들을 가져와 변수로 저장한 후 JSP 페이지 어디에서나 사용할 수 있는 태그
- bundle 태그를 대체하여 사용할 수 있음

```
<fmt:setBundle  
    basename="리소스변들"  
    [var="리소스변들을 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/> //기본 값은 page
```

[setBundle 태그와 message 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>  
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>  
<html>  
  <head>  
    <title>Internationalization</title>  
  </head>  
  <body>  
    <fmt:setBundle var="message" basename="resourceBundle.message" />  
    <p> <fmt:message bundle="${message}" key="name" />  
      <fmt:message bundle="${message}" key="hello" var="msg" />  
    <p>${msg}  
  </body>  
</html>
```

관리자
안녕하세요

3. JSTL fmt 태그를 이용한 다국어 처리

- JSTL fmt 태그를 사용하기 위해 jstl-1.2.jar 파일을 다운로드하여 /WebContent/WEB-INF/lib/ 폴더에 추가
- /src/ 폴더에 다음과 같이 2개의 리소스번들을 작성

```
01 title=자바 서버 페이지  
02 username=관리자  
03 password=1234
```

```
01 title=Java Server Pages  
02 username=admin  
03 password=1234
```

3. JSTL fmt 태그를 이용한 다국어 처리

```
01 <% page contentType="text/html; charset=utf-8"%>
02 <% taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
03 <html>
04 <head>
05 <title>Internationalization</title>
06 </head>
07 <body>
08 <p> -----기본 로케일-----
09 <fmt:setBundle basename="ch09.com.bundle.
    myBundle" var="resourceBundle" />
10 <p> 제목 : <fmt:message key="title" bundle=
    "${resourceBundle}" />
11 <p> <fmt:message key="username" var="userMsg" bundle="${resourceBundle}" />
12 <p> 이름 : ${userMsg}
13
14 <p> -----영문 로케일 -----
15 <fmt:setLocale value="en" />
16 <fmt:setBundle basename="ch09.com.bundle.myBundle" var=
    "resourceBundle" />
17 <p> 제목 : <fmt:message key="title" bundle="${resourceBundle}" />
18 <p> 이름 : <fmt:message key="username" bundle="${resourceBundle}" />
19 </body>
20 </html>
```



3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.3 숫자 태그의 기능과 사용법

- formatNumber 태그
 - 숫자를 형식에 맞춰 출력하는 태그

```
<fmt:formatNumber value="형식화할 숫자"  
    [type="{number|currency|percent}" ] //기본 값은 number  
    [pattern="사용자 정의 패턴"]  
    [currencyCode="통화 코드"] //type="currency"일 때만 적용  
    [currencySymbol="통화 기호"] //type="currency"일 때만 적용  
    [groupingUsed="천 단위마다 구분 기호{true|false}"] //기본 값은 true  
    [maxIntegerDigits="최대 자릿수"]  
    [minIntegerDigits="최소 자릿수"]  
    [maxFractionDigits="소수점 이하 최대 자릿수"]  
    [minFractionDigits="소수점 이하 최소 자릿수"]  
    [var="형식화된 결과를 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/> //기본 값은 page
```

3. JSTL fmt 태그를 이용한 다국어 처리

[formatNumber 태그 사용 예]

```
<% page contentType="text/html; charset=utf-8"%>
<% taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
  <p>숫자 : <fmt:formatNumber value="1234.567" type="number" />
  <p>통화 : <fmt:formatNumber value="1234.567" type="currency"
    currencySymbol="원" />
  <p>퍼센트 : <fmt:formatNumber value="1234.567" type="percent" />
  <p>패턴(.0000): <fmt:formatNumber value="1234.567" pattern=".0000" />
</body>
</html>
```

숫자 : 1,234.567
통화 : 원 1,235
퍼센트 : 123,457%
패턴(.0000): 1234.5670

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.4 날짜 태그의 기능과 사용법

- formatDate 태그

- 날짜 정보를 담고 있는 객체를 형식화하여 출력하는 태그

```
<fmt:formatDate value="형식화할 날짜"  
    [type="{time|date|both}"]  
    [dateStyle="{default|short|medium|long|full}"]  
    [timeStyle="{default|short|medium|long|full}"]  
    [pattern="사용자 정의 패턴"]  
    [timeZone="타임존"]  
    [var="형식화된 결과를 저장할 변수 이름"]  
    [scope="{page|request|session|application}"]/>
```


3. JSTL fmt 태그를 이용한 다국어 처리

[formatDate 태그 사용 예]

```
<% page contentType="text/html; charset=utf-8"%>
<% taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization</title>
</head>
<body>
  <jsp:useBean id="now" class="java.util.Date" />
  <p> date 형식 : <fmt:formatDate value="${now}" type="date" />
  <p> time 형식 : <fmt:formatDate value="${now}" type="time" />
  <p> both 형식 : <fmt:formatDate value="${now}" type="both" />

</body>
</html>
```

date 형식 : 2018. 9. 5.

time 형식 : 오후 5:02:59

both 형식 : 2018. 9. 5. 오후 5:02:59

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.5 시간 태그의 기능과 사용법

- timeZone 태그
 - 시간대별로 시간을 처리하는 태그

```
<fmt:timeZone value="타임존">
    //
</fmt:timeZone>
```

[timeZone 태그 사용 예]

```
<%@ page contentType="text/html; charset=utf-8"%>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt"%>
<html>
<head>
<title>Internationalization
</title>
</head>
<body>
    ...
    </fmt:timeZone>
    <%><fmt:timeZone value="GMT">
        런던 : <fmt:formatDate value="${ now }" type="both" dateStyle="full"
        timeStyle="full" />
    </fmt:timeZone>
</body>
</html>
```

한국 : 2018년 9월 5일 수요일 오후 3시 4분 23초 대한민국 표준시
뉴욕 : 2018년 9월 5일 수요일 오전 12시 4분 23초 GMT-08:00
런던 : 2018년 9월 5일 수요일 오전 3시 4분 23초 그리니치 표준시

3. JSTL fmt 태그를 이용한 다국어 처리

❖ 3.5 시간 태그의 기능과 사용법

■ setTimeZone 태그

- 특정 영역 범위의 시간대별로 시간을 처리하는 태그

```
<fmt:setTimeZone value="타임존"  
    [var="시간대 결과를 저장할 변수 이름"]  
    [scope="{page|request|session|application}"] />
```

[setTimeZone 태그 사용 예]

```
<? page contentType="text/html; charset=utf-8" %>  
<? taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>  
<html>  
<head>  
<title>Internationalization</title>  
</head>  
<body>  
    <jsp:useBean id="now" class="java.util.Date" />  
    <p> 한국 : <fmt:formatDate value="${now}" type="both" dateStyle="full"  
        timeStyle="full" />  
    <p><fmt:setTimeZone value="GMT-8" />  
        뉴욕 : <fmt:formatDate value="${ now }" type="both" dateStyle="full"  
            timeStyle="full" />  
    <p><fmt:setTimeZone value="GMT/" />  
        런던 : <fmt:formatDate value="${ now }" type="both" dateStyle="full"  
            timeStyle="full" />  
</body>  
</html>
```

한국 : 2018년 9월 3일 수요일 오후 3시 3분 10초 대한민국 표준시
뉴욕 : 2018년 9월 3일 수요일 오전 12시 3분 10초 GMT-08:00
런던 : 2018년 9월 3일 수요일 오전 8시 3분 10초 그리니치 표준시

감사합니다.

