

Localization of websites: An overview

- Character set
 - Encoding
 - HTML
 - Content negotiation
 - Localization support in Java API
 - Servlets
 - JSP
 - Databases
 - php
-
-

Character Sets

- Mapping between character and byte string
- ASCII and EBCDIC - 7 bit tuple
- ISO-8859-1 - 8-bit tuple
- Lot of character sets e.g. JIS



Unicode

- Each character can be represented in 2 bytes
- 256 pages of 256 character each
- Higher byte represents page number and lower byte cell number
- First page is identical to Latin-1



Encoding

- While exchanging data with other applications
 - Types of encoding:
 - Raw Unicode data in network byte order (*big endian*)
 - *UTF-8: ASCII characters (with character code ≤ 127) are written as is, characters with higher codes are encoded with escape characters.*
 - *UTF-7: Low seven bits are used. Used for sending e-mail and news messages.*
-
-

HTML meta tag

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  <title>Akshar</title>
</head>
<body>
<b>#2309;#2331;#2352;#8203;</b>
<br>
#2392;#2350;#2381;#2346;#2381;#2351;#2370;#2335;#2352;
  #2349;#8204;#8204;#8204;#2404;#2352;#2340; #2392;#2387;
  #2346;#2352;#2367;#2346;#2387;#2392;#2381;#2358;#2381;#2399;
  #2350;#2387;
</body>
</html>
```

Content Negotiation

- Variant of content most appropriate for user
- Implementation of content negotiation:
 - Server-driven content negotiation
 - Agent-driven content negotiation
 - Transparent content negotiation



Server driven content negotiation

- HTTP 1.1 headers
 - Accept: text/plain;q=0.5,text/html
 - Accept-Charset: UTF-8, ISO-8859-1, UTF-7;q=0.9
 - Accept-Encoding: compress, gzip
 - Accept-Language: da, en-gb;q=0.8, en;q=0.7
- Disadvantages
 - Difficult to “guess” accurately
 - Inefficient and violation of user privacy

Agent driven negotiation

- Selection of best representation is performed by the user agent.
- Advantages:
 - No pain in determining user agent's capabilities
- Disadvantage:
 - Second request



Transparent negotiation

- Cache is supplied with list of available representation of responses.
- Cache performs server driven negotiation.
- Advantage:
 - Distributing negotiation load



Character Encoding in Java

- `java.lang.String` constructors
 - `String(byte[] bytes, int offset, int length, String encoding)`
 - `String(byte [] bytes, String encoding)`
 - `java.lang.String` Methods:
 - `getBytes(String encoding)`
 - `java.io.InputStreamReader` constructor
 - `OutputStreamWriter` constructor
-
-

Character encoding Support in Servlet API

- Reader and Writer object are constructed with character encoding supplied by the Content-Type header.
 - `setLocale()` for specifying character encoding and content language.
 - Use of Reader and Writer instead of `ServletInputStream` and `ServletOutputStream`.
 - Using `flushBuffer()` to flush buffers.
-
-

Localization support in JSP

- `java.util.ResourceBundle`
 - Key/value pairs
- Approaches
 - Internationalized pages with locale specific information from `ResourceBundle`
 - Seperate locale specific page with a servlet to dispatch page appropriately

JSP code to determine locale

```
<%-- Interpret user's locale choice --%>
<c:if test="{param['locale'] != null}">
  <fmt:setLocale value="{param['locale']}" scope="session" />
</c:if>

<%-- Offer locale choice to user --%>
<a href="locale-choice.jsp?locale=en-US">USA</a> -
<a href="locale-choice.jsp?locale=de-DE">Deutschland</a> -
<a href="locale-choice.jsp?locale=ja-JP">#26085;#26412;</a>
```

JSP: Formatting

- java.util.text
 - NumberFormat <fmt:formatNumber />
 - DateFormat <fmt:formatDate/>
 - MessageFormat <fmt:message/>



Example: Formatting date

```
<jsp:useBean id="now" class="java.util.Date" />
<fmt:formatDate value="{now}" timeStyle="long"
dateStyle="long" />
<p>
<fmt:bundle basename="Messages">
<fmt:formatDate value="{now}" timeStyle="long"
dateStyle="long" />
</fmt:bundle>
```

Javascript: Detecting foreign language support

- navigator.language (Netscape)
 - navigator.browserLanguage (IE-specific – Browser localized language)
 - navigator.systemLanguage (IE-specific – Windows OS – Localized language)
 - navigator.userLanguage (IE-specific – Windows OS – Regional settings)
-
-

Databases

- Oracle: Store Japanese data
 - Select * from v\$nls_parameters
 - Set NLS_LANG to JAPANESE_JAPAN.UTF8
 - Set character set parameter in java connection properties.
 - MySQL:
 - Configure using `--with-extra-charsets= list-of-charsets|complex|all|none`
 - Start server with `--default-character-set`
 - Change error message language with `--language`
-
-

Php: message catalog

```
<?php
    $messages=array(
        'en_US'=>array('biscuit'=>'biscuit'
            'candy'=>'candy'
        ),
        'en_GB'=>array('biscuit'=>'scone'
            'candy'=>'sweets'
        )
    );
    function msg($s) {
        global $LANG;
        global $messages;
        return $messages[$LANG][$s];
    }
?>

.....
<?php
    $LANG='en_GB';
    print msg('biscuit')."":\n";
?>
```

Questions

