JSPベスト・プラクティス: タイム・スタンプの力

JSPページにタイム・スタンプを追加し、Webサイトを独自化する

Brett D. McLaughlin, Sr.
著作家
O'Reilly Media, Inc.

2003年 7月 01日

Brett McLaughlin氏のJSPベスト・プラクティスの連載記事において、今回は、JSPページへタイム・スタンプを追加することによって、Webサイトの使いやすさを向上させる方法、および追加方法の様々なテクニックについて説明しています。

タイム・スタンプは、通常個人のサイトあるいはコンテンツ主導サイトのような単純なWebサイト上で使用され、ページがいつ更新されたかをユーザーに知らせる働きがあります。その結果、特にEコマースのようなサイトには、それらの本質的な実用性が見落とされがちです。今回は、JSPページへタイム・スタンプを追加することによって、サイトの独自化に門戸を開くこと、およびそれがユーザーにとっていかに安堵をもたらすかについて説明いたします。

タイム・スタンプの追加

JSPページに最終更新日時のタイム・スタンプを置くことは、他のタイプのページにタイム・スタンプを置くことと全く同じです。`java.io.File`クラスは、ちょうどこの目的のために`lastModified`メソッドを提供しています。唯一の一筋縄にはいかない部分は、Webコンテナーに展開されたJSPページへの`File`ハンドルの取り込みです。リスト1は、この難関を乗り越えるための短いコード(JSPページ内に置かれた場合はscriptlet)です。

`application`と`request`はどちらも、すべてのJSPページがアクセスすることができる組み込み変数であることに注意してください。`application`は、このJSPページで`javax.servlet.ServletContext`オブジェクトをインプリメントするクラスの例です。このオブジェクトはJSPページのservletパスを取得し、取得したパスは物理的なパスに変換されます。その後、このパスは`File`に変換されます。

リスト1. JSPへのfileハンドルの取り込み

```
String jspPath = application.getRealPath(request.getServletPath());
java.io.File jspFile = new java.io.File(jspPath);
```

この情報を理解するため実行に移すには、リスト2のようなコードを使用してください。

リスト2. タイム・スタンプの出力

```
<%
     String jspPath = application.getRealPath(request.getServletPath());
     java.io.File jspFile = new java.io.File(jspPath);
     out.println(new java.util.Date(jspFile.lastModified()));
%>
```

ファイルの最終更新日が決められれば、それは`Date`オブジェクトに変換され、画面に直接出力されます。これは結果を得るためのかなり基礎的なコードになっています。

再使用可能なタイム・スタンプ

タイム・スタンプを入れたいすべてのページに、上のようなコードを追加することは確かに苦痛(そして冗長でもある)となるでしょう。これに対するよりよいアプローチは、日付およびタイム・スタンプを表示する汎用的なJSPページを作成し、タイム・スタンプを入れたいすべての場所でこのJSPページをインクルードすることです。これについては、リスト2をtimestamp.jsp として保存し、タイム・スタンプを出力するために必要な場所で`jsp:include`メカニズムを使用すればよいです。リスト3は、JSPタイム・スタンプをインクルードしている単純なフッター・ページの例です。

リスト3. 他のJSPページでタイム・スタンプをインクルードする

```
<!-- Begin footer section -->
     <tr>
       <td width="91" align="left" valign="top" bgcolor="#330066"> </td>
       <td align="left" valign="top"> </td>
       <td class="footer" align="left" valign="top">
           <div align="center"><br>
           &copy; 2003 <a href="mailto:webmaster@newInstance.com">Brett McLaughlin</a><br>
       Last Modified: <jsp:include page="timestamp.jsp" flush="true" />
         </div></td>
           <td align="left" valign="top"> </td>
       <td width="141" align="right" valign="top" bgcolor="#330066"> </td>
     </tr>
</table>
<!-- End footer section -->
```

リスト3は親JSPページ(この場合、footer.jsp)に相当して、インクルードされるtime-stampファイルではないことに気づかれたことでしょう。つまり、最終更新日のタイム・スタンプは timestamp.jsp ではなくfooter.jsp から引き出されています。これは重要なことです。なぜなら time-stampファイルは静的ファイルであり、更新されないためです。したがって、最終更新日のクエリーは、footer.jsp のような動的に更新されるページにあるべきです。

さらに一歩先に進むには、様々なコンテンツ・ページからなりたっているサイトを考えてみてください。これらの各ページはそれぞれfooter.jsp ページをインクルードしており、footer.jsp ページはtimestamp.jsp をインクルードしています。最終更新日がトップページ(この場合コンテンツ・ページ)を照会すると、コンテンツが変更されてタイム・スタンプだけが変わっています。これがまさに望んでいた結果です。

タイム・スタンプのフォーマット

デフォルトのタイム・スタンプ出力-`Fri Mar 28 10:30:10 CST 2003`-はやや魅力に欠けています。幸運なことに、出力のフォーマットをコントロールするために`java.text.SimpleDateFormat`クラ

スを使用することができます。リスト4は、`java.text.SimpleDateFormat`を追加したtimestamp.jspです。

リスト4. タイム・スタンプのフォーマット

```
<%
     String jspPath = application.getRealPath(request.getServletPath());
     java.io.File jspFile = new java.io.File(jspPath);
     java.util.Date lastModified = new java.util.Date(jspFile.lastModified());
     java.text.SimpleDateFormat fmt =
       new java.text.SimpleDateFormat("MMM dd, yyyy, K:mm a (zz)");
     out.println(fmt.format(lastModified));
%>
```

出力は、`Mar 28, 2003, 10:30 AM (CST)`となり、ユーザーにとってはるかに分かりやすいタイムスタンプになっています。フォーマットの他のオプションについては、Javadocで`java.text`パッケージをお調べください。

WARの使用

WAR(Webアーカイブ)ファイルに、Webアプリケーションの一部としてJSPページをデプロイする時、わずかな制限に出くわすことになりますが、幸運にもかなり単純な回避策があります。サーブレット・コンテナーは、通常、WARファイルを一時ディレクトリへと展開し、その一時ディレクトリからコンテンツを供給することにより、WARファイルをデプロイしています。厄介なのは、JSPページに関連した`File`オブジェクトを取得できない状況のときです。それどころか、一時ファイルへのハンドルを作成することだけはでき、Webコンテナーが再起動するたびにハンドルは再作成されるかもしれません。もし放っておけば、最終更新日は再起動のたびに変更されるでしょう。

回避策はデプロイされることになっているサーバー上でWARファイルを手動で展開させることです。この手動での展開の不便さは比較的小さいものです(特にWARファイルを展開するための`zip`ツールを使用できるので)。また、一度行えば、タイム・スタンプはここで記述されたように機能するでしょう。

結論

サイト上のコンテンツが新鮮でタイムリーであることでユーザーを安心させることに加えて、タイム・スタンプを追加することがWebサイトの独自化に向けての最初のステップです。Webページ(あるいはJSPページ)にタイム・スタンプを加えたら、選択されたページが更新される際ユーザーにEメールするプログラムをスクリプトするのは簡単なことです。

次回は、カスタム・タグ・ライブラリーの使用を取り入れたタイム・スタンプについて紹介します。それまでは、ここで学習したテクニックを試みるていただくことをお勧めいたします。それでは、次回をお楽しみに。

著者について

Brett D. McLaughlin, Sr.

Brett McLaughlin氏は、Logo (小さな三角形を覚えていますか?) の時代からコンピューターの仕事をしています。現在の専門は、JavaおよびJava関連のテクノロジーを使ったアプリケーション・インフラストラクチャーの構築です。ここ数年は、Nextel Communications and Allegiance Telecom, Inc. でこれらのインフラストラクチャーの実装に携わっています。Brett氏は、Javaサーブレットを使ってWebアプリケーション開発のための再利用可能なコンポーネント・アーキテクチャーを構築するJava Apacheプロジェクトturbineの共同設立者の1人です。同氏はまた、オープン・ソースのEJBアプリケーション・サーバーであるEJBossプロジェクトと、オープン・ソースのXML Web公開エンジンであるCocoonにも貢献しています。