u550●JSPでCookieを使ったデータの保存

参考資料：
https://atmarkit.itmedia.co.jp/ait/articles/0109/19/news0002.html

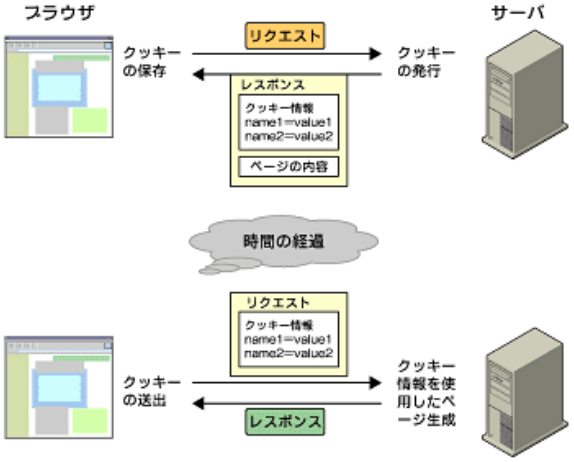
クッキーの扱いについて、JSPプログラムから値の格納と取得を行う方法について説明します。

クッキーとは

クッキーという概念がなかったころのWebシステムでは、サーバ側でアプリケーションを提供できても、ブラウザやユーザーが利用している端末にデータを書き込むことはできませんでした。クッキーは、その問題を解決するために考案された技術です。

クッキーがサポートされるブラウザに対して、サーバが小さなデータ（name=valueの組み合わせ：クッキー）をレスポンスのヘッダ部分に記述することで、その内容をブラウザ側に保存できます。ブラウザ側は、次回そのクッキーを発行したサイトに訪れた際に、持っているクッキーをサーバに提出します。サーバはその内容を確認することにより、前回自分がブラウザとやりとりしたデータを復元できます。

　クライアント側がクッキーを受け取った後は、ブラウザを閉じたり、コンピュータの電源を切ってもクッキーの情報を再度利用することができます。また、クッキーには有効期限を設定できるので、特定の期間を過ぎたクッキーを無効にすることができます。



今日のWebアプリケーションでは、このクッキーを利用して、次のようなことが行われています。

- 会員サイトのユーザーIDを保存して、次回の訪問でユーザーを認識する
- 最後に訪れた日時を保存して、ユーザーの訪問頻度を計る
- ユーザーがカスタマイズしたサイトの情報（ユーザーのし好）を保存しておき、次回もその設定を適用する
- 掲示板へ投稿するときに入力した名前とメールアドレスを保存しておき、次回の入力の手間を省く

クッキーを使用するプログラムの例

それでは、実際にクッキーを用いた例を見てみましょう。今回のプログラムは、**coo1.jsp**、**coo2.jsp**、**coo3.jsp**の3つのJSPページから構成され、それぞれ次のような処理を行います。

coo1.jsp	・このページにアクセスした時刻を文字列としてクッキーに設定する。
coo2.jsp	・coo1.jspで設定されたクッキーを取得して内容を表示する。 クッキーが取得できない場合はその旨のメッセージを表示する。
coo3.jsp	・クッキーを破棄する(有効期間がゼロ秒のクッキーを設定する)。

実際のプログラムの内容は次のようになります。

・coo1.jsp

192.168.56.10 - root@aso:/usr/local/src/apache-tomcat-9.0.0.M17/w VT

ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) 漢字コード(K) ヘルプ(H)

```
 1 <%@ page import="java.net.*, java.util.Date"
 2 contentType="text/html; charset=UTF-8" %>
 3 <%
 4 // 内容: クッキーを使用する(クッキーの設定)
 5 // 現在の時刻を取得
 6 Date now = new Date();
 7 // クッキーに格納する文字列を作成(URLエンコードをする)
 8 String value = URLEncoder.encode(now.toString());
 9 // 名前が"accesstime"、値が現在時刻であるクッキーを作成
10 Cookie cookie = new Cookie("accesstime",value);
11 // クッキーの設定
12 cookie.setMaxAge(7 * 24 * 60 * 60); //有効期間を1週間に設定
13 // クッキーを発行
14 response.addCookie(cookie);
15 %>
16 <!DOCTYPE HTML>
17 <html>
18 <head><title>クッキーを使用する</title></head>
19 <body>
20 <p>-- クッキーの設定 --</p>
21 <p>このページにアクセスした時刻をクッキーに設定しました。</p>
22 <p><a href="coo2.jsp">クッキーの内容を確認する</a></p>
23 </body>
24 </html>
25
26
"coo1.jsp" 26L, 901C
```

まず、6行目で、クッキーに格納する値の準備をしています。**クッキーに格納できるオブジェクトは文字列（String）**だけです。また、HTTPプロトコルでは、cookieの値として使用できる文字の種類に制限がありますので、**値をURLエンコードしてから設定する**ようにします。具体的には、8行目の記述のように、java.net.URLEncoderクラスのencode()メソッドを使用します。

```
8: String value = URLEncoder.encode(now.toString());
```

> java.net.URLEncoderクラスを使用するために、1行目で java.net.*パッケージのインポートを宣言しています。encode()メソッドはスタティックなので、インスタンスを生成せずに使用できます。

10行目ではCookieのインスタンスを生成しています。Cookieのコンストラクタは、

```
Cookie(java.lang.String name, java.lang.String value)
```

のみです。クッキーを識別するための名前と、格納する文字列を引数に指定します。

12行目では、クッキーの有効期間を1週間に設定しています。setMaxAgeによる有効期間の指定は秒単位で行います。

> このほかにも、クッキーに対しては参照可能なパスの設定やコメント、バージョンなど各種の設定が可能ですが、ここでは簡単のために有効期間のみの設定を行っています。そのほかの項目を設定するメソッドは本稿の最後にまとめてあります。

クッキーの発行は、暗黙オブジェクトの１つである**response**オブジェクトの以下のメソッドを使用します。

```
addCookie(Cookie cookie)
```

14行目で、実際に作成したクッキーを引数として、このメソッドを実行しています。

```
14: response.addCookie(cookie);
```

実際には、サーバからブラウザ側に対してレスポンスが送り出されるときに、ヘッダ情報の一部としてクッキーの情報も送り出されます。

続いて、クッキーの内容を取得する**coo2.jsp**のプログラムの内容は次のようになります。

・coo2.jsp

192.168.56.10 - root@aso:/usr/local/src/apache-tomcat-9.0.0.M17/w VT

ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) 漢字コード(K) ヘルプ(H)

```
<%@ page import="java.net.*"
contentType="text/html; charset=UTF-8" %>
<%
// 内容: クッキーを使用する(クッキーの取得)
// クッキーの配列を取得
Cookie cookies[] = request.getCookies();
// 目的のクッキーを格納する変数
Cookie accesstimeCookie = null;
// それぞれのクッキーに対して名前を確認
if(cookies != null) {
for(int i = 0; i < cookies.length; i++) {
// 名前が "accesstime" であるかチェック
if(cookies[i].getName().equals("accesstime")) {
// 該当するクッキーを取得
accesstimeCookie = cookies[i];
}
}
}
// 表示する文字列
String accesstime;
// 該当するクッキーがみつからなかった場合
if(accesstimeCookie == null) {
accesstime = "記録がありません";
} else { // クッキーがみつかった場合は値を取得(URLデコードする)
accesstime = URLDecoder.decode(accesstimeCookie.getValue());
}
%>
<!DOCTYPE HTML>
<html>
<head><title>クッキーを使用する</title></head>
<body>
<p>-- クッキーの取得 --</p>
<p>cool.jsp に最後にアクセスした時刻<br>
<b>[ <%= accesstime %> ]</b>
<p><a href="cool.jsp">クッキーを設定する</a> |
<a href="coo3.jsp">クッキーを破棄する</a></p>
</body>
</html>
```

クッキーの取得は、クッキーの設定に比べて少々やっかいです。

暗黙オブジェクトの1つである**request**オブジェクトの、

```
getCookies()
```

メソッドを使用してクッキーを取得しますが、このメソッドでは個別のクッキーではなく、**複数のクッキーをひとまとまりの配列として取得**しま

す。そのため、この配列から目的のクッキーを探し出すための処理があらためて必要になります。

まず6行目でクッキーの配列を取得し、11～16行目のforループの中で目的のクッキーを探しています。クッキーの名前はCookieオブジェクトの**getName()**メソッドで取得できるため、次のような if 文で、名前の比較をして目的のクッキーであるかどうかを判定できます。

```
13: if(cookies[i].getName().equals("accesstime")) {
14: // 該当するクッキーを取得
15: accesstimeCookie = cookies[i];
16: }
```

さて、目的のクッキーを見つけることができた後は、**getValue()**メソッドで格納されている値（文字列）を取得します。ここで格納された文字列はURLエンコードされたものなので、URLデコードの処理が必要になることに注意しましょう。

> URLエンコードされた文字列は、java.net.URLDecodeクラスのdecode()メソッドで元の文字列に復元できます。エンコードの処理と同様、decode()メソッドはスタティックなので、インスタンスを生成せずに使用できます。

これらの、クッキーからの値の取得とURLデコードの処理は25行目で行っています。

```
25: accesstime = URLDecoder.decode(accesstimeCookie.getValue());
```

もし、目的のクッキーが見つからなかった場合は、

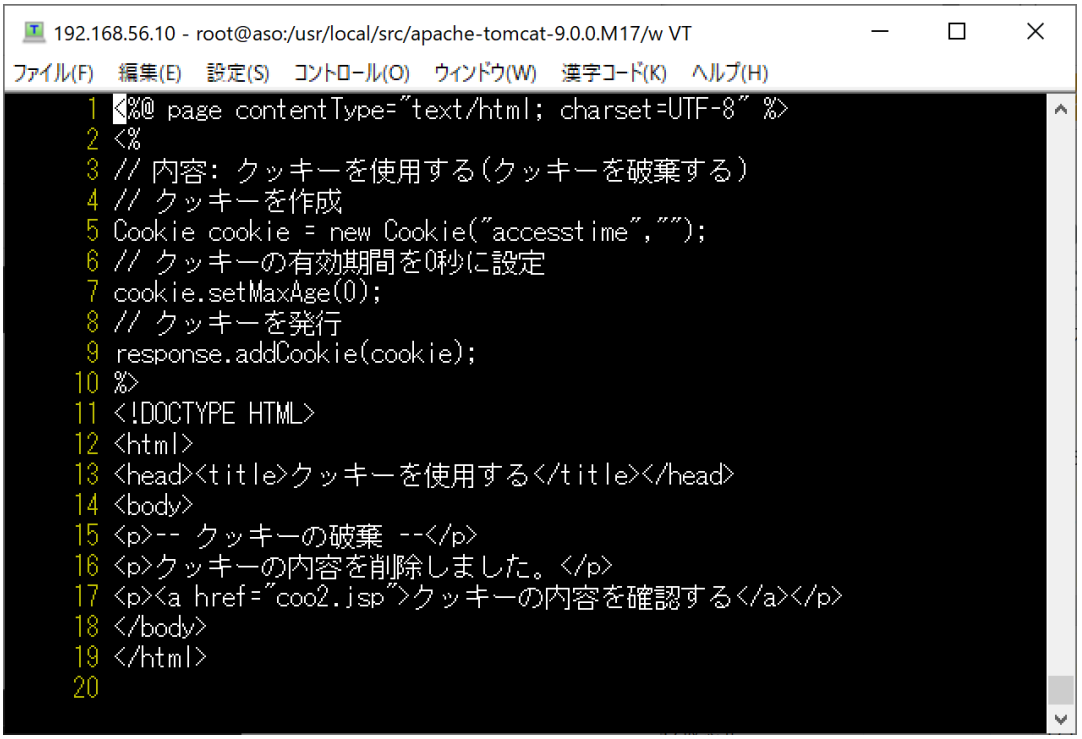
```
23: accesstime = "記録がありません";
```

とし、その結果を40行目でHTML文に出力しています。

```
34: <b>[ <%= accesstime %> ]</b>
```

続いて、**クッキーの破棄**を行う**coo3.jsp**のプログラムの内容は次のようになります。

・**coo3.jsp**



coo3.jspは、クッキーの破棄を行います。クッキーの破棄といっても、明示的にクッキーの内容を削除する手段はないため、既存のクッキーを**有効期間がゼロ秒のクッキー**に変更することで、ブラウザ側で破棄の処理を行うように促します。

5行目でクッキーを作成していますが、クッキーの名前は、削除したいクッキーの名前にします。次に、7行目で、クッキーの有効期間をゼロ秒に設定しています。

```
5: Cookie cookie = new Cookie("accesstime","");
7: cookie.setMaxAge(0);
```

9行目で、このクッキーの発行を行っていますが、これは**coo1.jsp**の方法とまったく同じです。

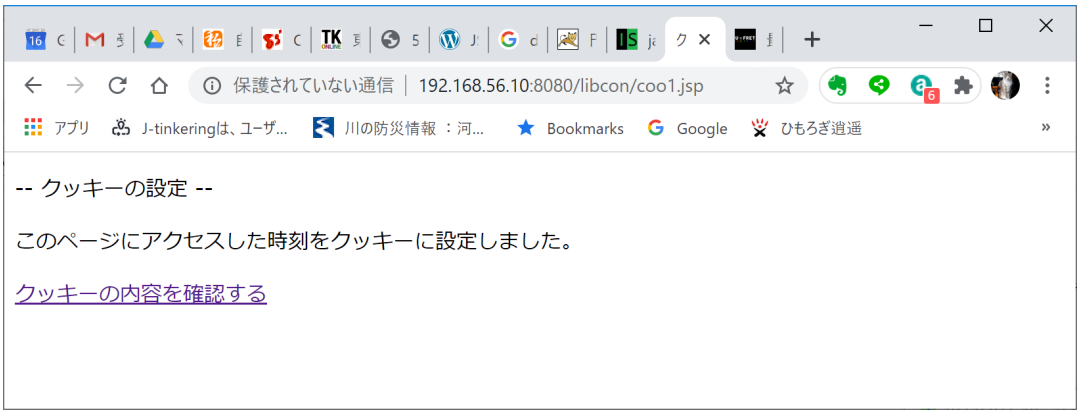
さて、再度**coo2.jsp**へアクセスすることで、この方法で本当にクッキーが削除されたのかどうかを確認できます。17行目で、**coo2.jsp**へのリンクを表示しています。

プログラムの実行結果

実際に上記のプログラムを実行してみましょう。一連の処理の流れは次のようになります（上から下の順に遷移します）。

coo1.jspの実行画面

クッキーの設定 アクセスした時刻を表す文字列がクッキーに設定されました

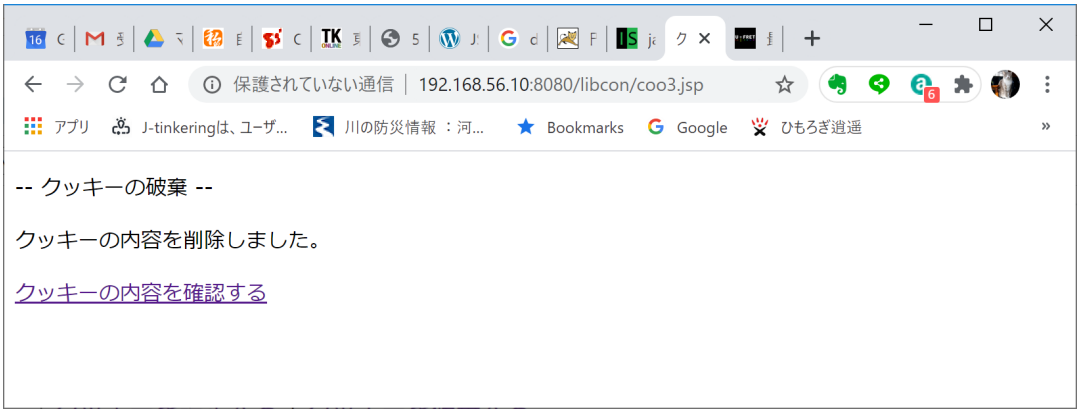


coo2.jspの実行画面

クッキーの取得 coo1.jsp で設定されたクッキーを取得しました。確かに時刻がクッキーに格納されていました

coo3.jspの実行画面

クッキーの破棄 クッキーの有効期限をゼロ秒にして、再度クッキーを設定します



coo2.jspの実行画面

クッキーの取得 クッキーは取得できていません。coo3.jsp の処理で破棄されたことが確認できました。

■Cookieクラスのメソッド

簡単なサンプルプログラムをもとに、クッキーの作成とその格納、および取得の様子を見てきました。ここで述べた以外にも、Cookieクラスには多くのメソッドがあり、クッキーの振る舞いを細かく設定できるようになっています。Cookieクラスの振る舞いを設定するそのほかのメソッドには次のようなものがあります。

	メソッド	説明
	setComment(java.lang.String purpose)	クッキーの目的を説明するためのコメント文を設定します（クッキーのバージョンが1以上である場合に使用されます）
	setDomain(java.lang.String	クッキーにアクセス可

	pattern)	能なサーバまたはドメインを設定します
	setPath(java.lang.String uri)	クッキーにアクセス可能なURLパスを設定します
	setSecure(boolean flag)	HTTPSなどの暗号化されたセキュアなプロトコルで送られるかどうかを指定します
	setVersion(int v)	クッキーのバージョンを指定します

完成したjspの実行画面のスクショを提出してください！