

# Servlet/JSP (作成中)

2010年x月x日 作成

Firebird日本ユーザー会

Naoyuki Sano

<http://rururu.sakura.ne.jp/>

<http://rururublog.sblo.jp/>

## ■ Servlet/JSPを教わった人 . . . 2004/9/25 - 2005/5/21

秦 崇[はた たかし]さん(<http://秦崇.jp/>)

2009年3月

\* 前職で一緒に働いていた秦 崇氏が32歳の若さで亡くなりました。

ご冥福をお祈りします

<http://www.todo.ne.jp/activity.html>

前略、秦崇様。

<http://haniwafactory.blog28.fc2.com/blog-entry-3053.html>

<書籍>

3日で作るPHPアプリケーション

<http://www.amazon.co.jp/3日で作るPHPアプリケーション-秦-崇/dp/475615106X>

PHPでブログを超える(3日で作るPHPアプリケーションの紹介記事)

<http://ascii.jp/elem/000/000/114/114544/>

<セミナー>

- PHP Conference 2008

ライトニングトーク「PHPプログラマ・エンジニア育成のポイント」

<http://www.phppro.jp/events/1>

- OSC2008 Tokyo/Spring

45分でわかる Ajax丸かじり

<http://wiki.fdiary.net/tokushima/?osc2008-spring>

- PostgreSQL CE Silver 徹底解説セミナー (2006/3/3)

<http://www.linuxacademy.ne.jp/event/200603/postgresql-ce/>

1. 環境構築(Windows)
2. Servletのライフサイクル
3. Servletの作り方
4. HTMLフォームからの入力を取得(GET/POST)
5. 日本語の文字化け/クロスサイトスクリプティング(XSS)対策
6. Cookieの保存/取得
7. セッションの保存/取得
8. SevletContextオブジェクト **(作成中)**
9. RequestDispatcherオブジェクト **(作成中)**
10. JSPについて **(作成中)**
11. JSPタグ **(作成中)**
12. JSPのエラーについて **(作成中)**
13. JSPのメリット/デメリット **(作成中)**
14. MVCアーキテクチャー **(作成中)**
15. JSPの暗黙オブジェクト **(作成中)**
16. JSPでパッケージをインポート **(作成中)**
17. JSPでJavaBeansを使う **(作成中)**
18. カスタムタグ/アクションタグ **(作成中)**
19. JSPの例外処理 **(作成中)**
20. BMI値を取得するWebアプリケーションを作成 **(作成中)**
21. データベースを使用したWebアプリケーションを作成 **(作成中)**

## 1. 環境構築(Windows)

### 1、Java SE Development Kit (JDK) 6

以下のサイトからダウンロードしてインストール (インストール先の指定あり)

<http://java.sun.com/javase/ja/6/download.html>

### 2、環境変数の設定

環境変数PathにJavaをインストールしたディレクトリ¥binを入れる。

例) D:¥javaにインストールした場合

『;D:¥java¥bin』をPathの最後に挿入

### 3、コマンドプロンプトを立ち上げ、『java -version』と打ってバージョンが表示されたらOK!

### 4、Tomcat6

#### (1) ダウンロード

Windows Service Installer版をインストール

<http://tomcat.apache.org/download-60.cgi>

#### (2) インストール

ダウンロードしたファイルを実行し、ウィザードに従ってインストール (インストール先とパスワードの指定あり)

※Vistaの場合は、デフォルトのProgram Filesにインストールした場合、UACで書き込みが制限されて起動時にエラーになるので、デフォルト以外の場所にインストールする必要があります。

### 5、フォルダ作成

C:¥Tomcat 6.0 (Tomcatのインストールパスを指定)のwebappsフォルダに以下の階層のフォルダを作成

¥study

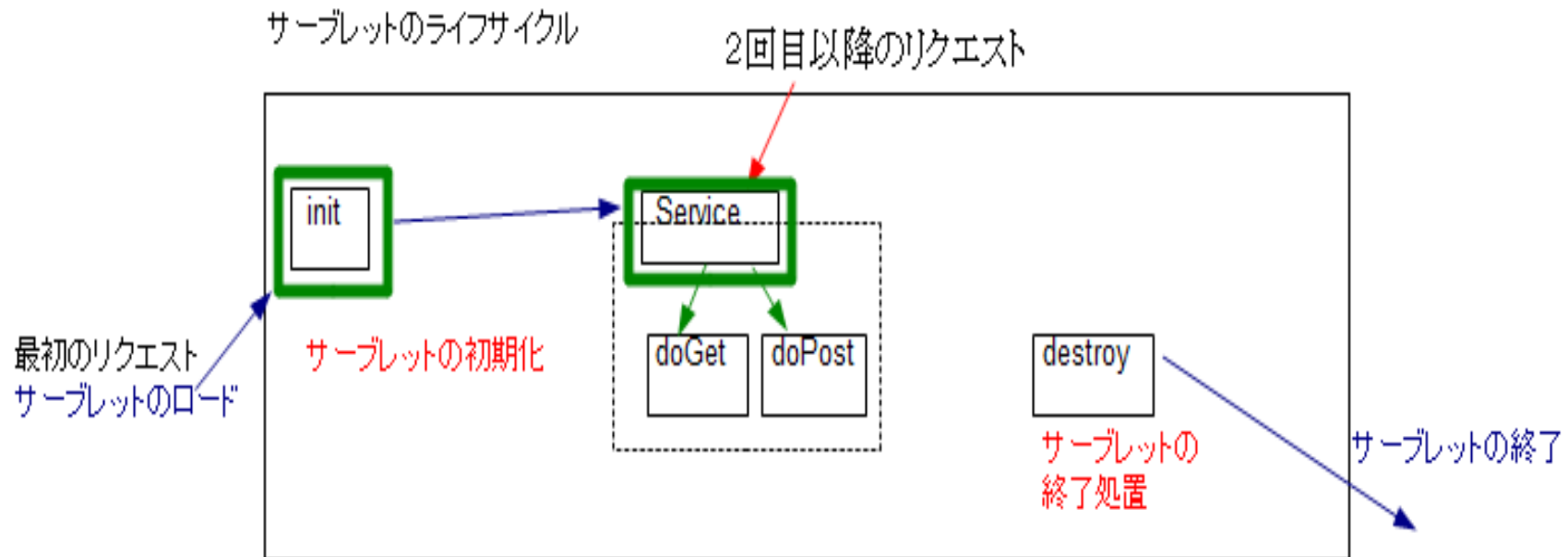
¥WEB-INF

¥classes

### 6、環境変数

- JAVA\_HOME : javaのインストールパスを指定
- CATALINA\_HOME : C:¥Tomcat 6.0 (Tomcatのインストールパスを指定)
- CLASSPATH : .;%CATALINA\_HOME%lib¥servlet-api.jar;%CATALINA\_HOME%lib¥jsp-api.jar;  
%CATALINA\_HOME%webapps¥study¥WEB-INF¥classes

## 2. Servletのライフサイクル



### 3. Servletの作り方

(1) 必要なパッケージをimport

```
java.io.IOException;  
java.io.PrintWriter;  
javax.servlet.ServletException;  
javax.servlet.http.;;  
javax.servlet.http.HttpServletRequest;  
javax.servlet.http.HttpServletResponse;
```

(2) HttpServletクラスを継承

(3) 必要なメソッド(doGet, doPost)をオーバーライド

(4) ContentType(出力形式)の決定

(5) Writerオブジェクトの取得

■ HelloWorld.java (動かし方については次のページで説明)

```
package test01;  
import java.io.IOException;  
import java.io.PrintWriter;  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
public class HelloWorldServlet extends HttpServlet {  
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {  
        response.setContentType("text/html;charset=Windows-31J");  
        PrintWriter out = response.getWriter();  
        out.println("<html>");  
        out.println("<head> <title>Hello World</title></head>");  
        out.println("<body>Hello World!</body>");  
        out.println("</html>");  
    }  
}
```

- (1) ソースコード(HelloWorld.java)を任意の場所(C:¥javastudy)に保存してコンパイル  
C:¥javastudy> javac HelloWorld.java
- (2) %CATALINA\_HOME%¥webapps¥study¥WEB-INF¥classesに「test01」フォルダを作成
- (3) コンパイルしてエラーが出なかったらクラスファイル(HelloWorldServlet.class)を  
%CATALINA\_HOME%¥webapps¥study¥WEB-INF¥classes¥test01にコピー
- (4) Web.xmlの作成 (ファイルの場所は%CATALINA\_HOME%¥webapps¥study¥WEB-INF)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
version="2.5">

  <servlet>
    <servlet-name>HelloWorldServlet</servlet-name>
    <servlet-class>test01.HelloWorldServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>HelloWorldServlet</servlet-name>
    <url-pattern>/test01/HelloWorldServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

- (5) アプリケーションを再起動  
ブラウザを起動して「<http://localhost:8080/manager/html>」にアクセスして、  
「study」を探して「再ロード」のリンクをクリック
- (6) ブラウザを起動して「<http://localhost:8080/study/test01/HelloWorldServlet>」  
にアクセスして画面に「HelloWorld」と表示されれば成功です。

#### 4. HTMLフォームからの入力を取得(GET/POST) (1 / 4)

(1) ソースコード(getpostsample.java)を任意の場所(C:¥javastudy)に保存 (ソースコードは次のページに続きがあります)

```
package test02;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class getpostsample extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=Windows-31J");
        PrintWriter out = response.getWriter();
        String name = request.getParameter("name");
        int age = Integer.parseInt(request.getParameter("age"));
        out.println("<html>");
        out.println("<head><title>Getの情報</title></head>");
        out.println("<body><b>Getの情報</b>");
        out.println("あなたが入力した名前と年齢に間違いはないですか? <br>");
        out.println("名前 : " + name);
        out.println("年齢 : " + age);
        out.println("</body>");
        out.println("</html>");
    }
}
```



#### 4. HTMLフォームからの入力を取得(GET/POST) (2 / 4)

```
public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=Windows-31J");
    PrintWriter out = response.getWriter();
    String name = null,value = null;
    int age = 0;
    Enumeration names = request.getParameterNames();
    while (names.hasMoreElements()) {
        value = (String)names.nextElement();
        if (value.equals("name")) {
            name = request.getParameter(value);
        } else if(value.equals("age")) {
            age = Integer.parseInt(request.getParameter(value));
        }
    }
    out.println("<html>");
    out.println("<head><title>Postの情報</title></head>");
    out.println("<body><b>Postの情報</b>");
    out.println("あなたが入力した名前と年齢に間違いはないですか? <br>");
    out.println("名前 : " + name);
    out.println("年齢 : " + age);
    out.println("</body>");
    out.println("</html>");
}
}
```

(2) コンパイル

```
C:¥javastudy> javac getpostsample.java
```

## 4. HTMLフォームからの入力を取得(GET/POST) (3 / 4)

### (3) GETのパラメータ入力画面作成(getsample.html)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS" />
<title>GET</title>
</head>
<body>
<form id="frm02" action="/study/test02/getpostsample" method="get">
あなたの名前は<input id="name" name="name" type="text" maxlength="10" value="" /><br />
あなたの年齢は<input id="age" name="age" type="text" maxlength="3" value="" /><br />
<input id="ok" type="submit" value="OK" />
</form>
</body>
</html>
```

### (4) POSTのパラメータ入力画面作成(postsample.html)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS" />
<title>POST</title>
</head>
<body>
<form id="frm03" action="/study/test02/getpostsample" method="post">
あなたの名前は<input id="name" name="name" type="text" maxlength="10" value="" /><br />
あなたの年齢は<input id="age" name="age" type="text" maxlength="3" value="" /><br />
<input id="ok" type="submit" value="OK" />
</form>
</body>
</html>
```

#### 4. HTMLフォームからの入力を取得(GET/POST) (4 / 4)

- (5) %CATALINA\_HOME%\webapps\study\に「test02」フォルダを作成
- (6) getsample.htmlとpostsample.htmlを%CATALINA\_HOME%\webapps\study\に「test02」フォルダにコピー
- (7) %CATALINA\_HOME%\webapps\study\WEB-INF\classesに「test02」フォルダを作成
- (8) コンパイルしてエラーが出なかったらクラスファイル(getpostsample.class)を  
%CATALINA\_HOME%\webapps\study\WEB-INF\classes\test02にコピー
- (9) Web.xmlに以下の内容を追加 (ファイルの場所は%CATALINA\_HOME%\webapps\study\WEB-INF)

```
<servlet>
  <servlet-name>getpostsample</servlet-name>
  <servlet-class>test02.getpostsample</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>getpostsample</servlet-name>
  <url-pattern>/test02/getpostsample</url-pattern>
</servlet-mapping>
```

- (10) アプリケーションを再起動  
ブラウザを起動して「<http://localhost:8080/manager/html>」にアクセスして、「study」を探して「再ロード」のリンクをクリック
- (11) ブラウザを起動して「<http://localhost:8080/study/test02/getsample.html>」 または  
「<http://localhost:8080/study/test02/postsample.html>」にアクセス
- (12) 名前に英語(日本語を入力すると文字化けしますのでやめてください)、年齢に数字(数字以外を入力するとエラーになりますのでやめてください)を入力して「OK」ボタンをクリックしてください。以下の情報がブラウザに表示されれば成功です。

【GETまたはPOST】の情報 あなたが入力した名前と年齢に間違いありませんか？

名前：【入力した名前】 年齢：【入力した年齢】

## 5. 日本語の文字化け／クロスサイトスクリプティング(XSS)対策 ( 1 / 3 )

ここでは、クロスサイトスクリプティング(XSS)対策と日本語の文字化けを防ぐ共通のクラスを作成後に、

**「4. HTMLフォームからの入力を取得(GET/POST)」**のプログラムを使用する形に修正します。

(1) クロスサイトスクリプティング(XSS)対策と日本語の文字化けを防ぐ共通のクラス作成  
(ファイル名は「ServletUtil.java」、ファイルの場所は「C:¥javastudy¥util」)

```
package util;
public class ServletUtil {
    public static String strEscape(String strEsc) {
        StringBuffer strResult = new StringBuffer();
        for ( int i = 0; i < strEsc.length(); i++) {
            switch (strEsc.charAt(i)) {
                case '&':
                    strResult.append("&amp;");
                    break;
                case '<':
                    strResult.append("&lt;");
                    break;
                case '>':
                    strResult.append("&gt;");
                    break;
                case '¥r':
                    break;
                case '¥n':
                    strResult.append("<br>¥r¥n");
                    break;
                default:
                    strResult.append(strEsc.charAt(i));
                    break;
            }
        }
        return strResult.toString();
    }

    public static String strEncode(String strEnc) {
        try {
            return new String(strEnc.getBytes("ISO-8859-1"), "JISAutoDetect");
        } catch ( Exception e ) {
            return "Encode error";
        }
    }
}
```

## 5. 日本語の文字化け／クロスサイトスクリプティング(XSS)対策 (2 / 3)

### (2) コンパイル

C:¥javastrudy> javac util¥ServletUtil.java

### (3) getpostsample.javaの修正 (C:¥javastudy¥test02に移動してから作業をしてください)

```
package test02;
import util.ServletUtil;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class getpostsample extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=Windows-31J");
        PrintWriter out = response.getWriter();
        String name = ServletUtil.strEscape(ServletUtil.strEncode(request.getParameter("name")));
        int age = Integer.parseInt(ServletUtil.strEscape(ServletUtil.strEncode(request.getParameter("age"))));
        out.println("<html>");
        out.println("<head> <title>Get / Postの情報</title> </head>");
        out.println("<body> <b>Get / Postの情報</b>");
        out.println("あなたが入力した名前と年齢に間違いはないですか？ <br>");
        out.println("名前 : " + name);
        out.println("年齢 : " + age);
        out.println("</body>");
        out.println("</html>");
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }
}
```

## 5. 日本語の文字化け／クロスサイトスクリプティング(XSS)対策 (3 / 3)

(4) コンパイル

```
C:¥javastudy> javac test02¥getpostsample.java
```

(5) クラスファイル(getpostsample.class)を%CATALINA\_HOME%¥webapps¥study¥WEB-INF¥classes¥test02にコピー

(6) %CATALINA\_HOME%¥webapps¥study¥に「util」フォルダを作成

(7) クラスファイル(ServletUtil.class)を%CATALINA\_HOME%¥webapps¥study¥WEB-INF¥classes¥utilにコピー

(8) アプリケーションを再起動

ブラウザを起動して「<http://localhost:8080/manager/html>」にアクセスして、「study」を探して「再ロード」のリンクをクリック

(9) ブラウザを起動して「<http://localhost:8080/study/test02/getsample.html>」 または

「<http://localhost:8080/study/test02/postsample.html>」にアクセス

(10) 名前に英語または日本語、年齢に数字(数字以外を入力するとエラーになりますのでやめてください)を入力して「OK」ボタンをクリックしてください。以下の情報がブラウザに表示されれば成功です。

**Get/Postの情報** あなたが入力した名前と年齢に間違いはないですか？

名前：【入力した名前】 年齢：【入力した年齢】

## **6. Cookieの保存／取得 ( 1 / 4 )**

### (1) Cookie保存する値の入力画面作成(cookie.html)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=Shift_JIS" />
<title>Cookie</title>
</head>
<body>
<form id="frm03" action="/study/test03/savecookie" method="get">
名前を入力してください。<br />
<input id="name" name="name" type="text" maxlength="10" value="" /><br />
<input id="save" type="submit" value="Cookie保存" />
</form>
</body>
</html>
```

(2) %CATALINA\_HOME%\webapps\study\に「test03」フォルダを作成

(3) cookie.htmlを%CATALINA\_HOME%\webapps\study\に「test03」フォルダにコピー

(4) %CATALINA\_HOME%\webapps\study\WEB-INF\classesに「test03」フォルダを作成

## 6. Cookieの保存／取得 (2 / 4)

(5) ソースコード保存先(C:¥javastudy)に「test03」フォルダ作成

(6) 「C:¥javastudy¥test3」フォルダに以下の「savecookie.java」を作成

```
package test03;
import util.ServletUtil;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Cookie;
import java.net.URLEncoder;

public class savecookie extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String name = ServletUtil.strEscape(ServletUtil.strEncode(request.getParameter("name")));
        name = java.net.URLEncoder.encode(name,"Windows-31J");
        Cookie ck = new Cookie("username", name);
        ck.setMaxAge(60*60*60*24*30);
        response.addCookie(ck);
        response.setContentType("text/html;charset=Windows-31J");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Cookie保存</title></head>");
        out.println("<body><b>Cookieに情報を保存しました。</b>");
        out.println("<a href=¥\"/study/test03/readcookie¥\">Cookie情報の確認</a>");
        out.println("</body>");
        out.println("</html>");
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }
}
```



## 6. Cookieの保存／取得 (3 / 4)

### (7) コンパイル

```
C:¥javastudy> javac test03¥savecookie.java
```

### (8) 「C:¥javastudy¥test3」フォルダに以下の「readcookie.java」を作成

```
package test03;
import util.ServletUtil;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.Cookie;
import java.net.URLDecoder;

public class readcookie extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        Cookie[] cks = request.getCookies();
        String name = "";
        if ( cks != null ) {
            for ( int i = 0; i < cks.length; i++ ) {
                if ( cks[i].getName().equals("username") ) {
                    name = java.net.URLDecoder.decode(cks[i].getValue(),"Windows-31J");
                    name = ServletUtil.strEscape(name);
                }
            }
        }
        response.setContentType("text/html;charset=Windows-31J");
        PrintWriter out = response.getWriter();
        out.println("<html>");
        out.println("<head><title>Cookie情報の表示</title></head>");
        out.println("<body><b>Cookieから情報を取得しました。</b>");
        out.println("名前 : " + name);
        out.println("</body>");
        out.println("</html>");
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }
}
```

## 6. Cookieの保存／取得 (4 / 4)

### (9) コンパイル

C:¥javastudy> javac test03¥readcookie.java

(10) クラスファイル(savecookie.class, readcookie.class)を

%CATALINA\_HOME%¥webapps¥study¥WEB-INF¥classes¥test03にコピー

(11) Web.xmlに以下の内容を追加 (ファイルの場所は%CATALINA\_HOME%¥webapps¥study¥WEB-INF)

```
<servlet>
  <servlet-name>savecookie</servlet-name>
  <servlet-class>test03.savecookie</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>savecookie</servlet-name>
  <url-pattern>/test03/savecookie</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>readcookie</servlet-name>
  <servlet-class>test03.readcookie</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>readcookie</servlet-name>
  <url-pattern>/test03/readcookie</url-pattern>
</servlet-mapping>
```

(12) アプリケーションを再起動

ブラウザを起動して「<http://localhost:8080/manager/html>」にアクセスして、「study」を探して「再ロード」のリンクをクリック

(13) ブラウザを起動して「<http://localhost:8080/study/test03/cookie.html>」にアクセス

(14) 名前を入力して「Cookie保存」ボタンをクリックしてください。

(15) 次の画面で「Cookie情報の確認」をクリックしてください。

(16) 以下の情報がブラウザに表示されれば成功です。

Cookieから情報を取得しました。 名前：【入力した名前】

## 7. セッションの保存／取得 (1 / 5)

- (1) %CATALINA\_HOME%\webapps\study\に「test04」フォルダを作成
- (2) %CATALINA\_HOME%\webapps\study\WEB-INF\classesに「test04」フォルダを作成
- (3) 「C:\javastudy\test4」フォルダに以下の「sessioninput.java」を作成

```
package test04;
import util.ServletUtil;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class sessioninput extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=Windows-31J");
        PrintWriter out = response.getWriter();
        HttpSession ses = request.getSession(true);
        out.println("<html>");
        out.println("<head> <title>セッションに保存する情報入力</title></head>");
        out.println("<form id=\"frm04\" action=\"/study/test04/savesession\" method=\"post\">");
        out.println("<body> <b>名前を入力して送信ボタンをクリックしてください。</b><br />");
        out.println("名前 : <input id=\"name\" name=\"name\" type=\"text\" value=\"\" />");
        out.println("<input id=\"input\" type=\"submit\" value=\"送信\" />");
        out.println("</form>");
        out.println("</body>");
        out.println("</html>");
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }
}
```

## 7. セッションの保存／取得 (2 / 5)

### (4) コンパイル

C:¥javastudy> javac test04¥sessioninput.java

(5) 「C:¥javastudy¥test4」フォルダに以下の「savesession.java」を作成

```
package test04;
import util.ServletUtil;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class savesession extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=Windows-31J");
        PrintWriter out = response.getWriter();
        HttpSession ses = request.getSession(true);
        String name = ServletUtil.strEscape(ServletUtil.strEncode(request.getParameter("name")));
        ses.setAttribute("name", name);
        out.println("<html>");
        out.println("<head> <title>Session情報保存</title> </head>");
        out.println("<body> <b>Session変数に情報を保存しました。 </b>");
        out.println("<a href=¥\"/study/test04/readsession¥\">Session情報の確認</a>");
        out.println("</body>");
        out.println("</html>");
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }
}
```

## 7. セッションの保存／取得 (3 / 5)

### (6) コンパイル

```
C:¥javastudy> javac test04¥savesession.java
```

### (7) 「C:¥javastudy¥test4」フォルダに以下の「readsession.java」を作成

```
package test04;
import util.ServletUtil;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Enumeration;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class readsession extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html;charset=Windows-31J");
        PrintWriter out = response.getWriter();
        HttpSession ses = request.getSession(true);
        String name = (String)ses.getAttribute("name");
        name = ServletUtil.strEscape(name);
        out.println("<html>");
        out.println("<head><title>Session情報の表示</title></head>");
        out.println("<body><b>Sessionに保存した情報を取得しました。</b><br>");
        out.println("名前 : " + name);
        out.println("</body>");
        out.println("</html>");
        ses.invalidate();
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doGet(request, response);
    }
}
```

## 7. セッションの保存/取得 (4 / 5)

(8) コンパイル

```
C:¥javastudy> javac test04¥readession.java
```

(9) クラスファイル(sessioninput.class, savesession.class, readsession.class)を  
%CATALINA\_HOME%¥webapps¥study¥WEB-INF¥classes¥test04にコピー

(10) Web.xmlに以下の内容を追加 (ファイルの場所は%CATALINA\_HOME%¥webapps¥study¥WEB-INF)

```
<servlet>
  <servlet-name>sessioninput</servlet-name>
  <servlet-class>test04.sessioninput</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>sessioninput</servlet-name>
  <url-pattern>/test04/sessioninput</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>savesession</servlet-name>
  <servlet-class>test04.savesession</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>savesession</servlet-name>
  <url-pattern>/test04/savesession</url-pattern>
</servlet-mapping>
<servlet>
  <servlet-name>readsession</servlet-name>
  <servlet-class>test04.readsession</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>readsession</servlet-name>
  <url-pattern>/test04/readsession</url-pattern>
</servlet-mapping>
```

## 7. セッションの保存／取得 (5 / 5)

(11) アプリケーションを再起動

ブラウザを起動して「<http://localhost:8080/manager/html>」にアクセスして、  
「study」を探して「再ロード」のリンクをクリック

(12) ブラウザを起動して「<http://localhost:8080/study/test04/sessioninput>」にアクセス

(13) 名前を入力して「送信」ボタンをクリックしてください。

(14) 次の画面で「Session情報の確認」をクリックしてください。

(15) 以下の情報がブラウザに表示されれば成功です。

Sessionに保存した情報を取得しました。

名前：【入力した名前】

x. xxxxx (x/ x)