

<http://d.hatena.ne.jp/littlebuddha/20100425/1272206006>

http://www.javaroad.jp/opensource/js_tomcat2.htm

CentOS 6.2 で Servlet を設定する

- httpd インストール
 - yum install httpd
 - yum install httpd-tools
- tomcat6 インストール
 - yum install tomcat6
 - yum install apache-tomcat-apis (DVD の場合はディスク 1)
 - yum install xalan-j2 (DVD の場合はディスク 1)
 - yum install jakarta-taglibs-standard (DVD の場合はディスク 2)
 - yum install tomcat6-webapps (DVD の場合はディスク 2)
- tomcat 自動起動
 - chkconfig tomcat6 on
- tomcat テスト
 - localhost:8080/ へアクセス確認
- apache と tomcat の連携
 - sudo vi /etc/httpd/conf.d/tomcat.conf

```
<Location /tomcat/>
# 下の設定は ajp プロトコルで Apache と Tomcat 間の通信を行う
ProxyPass ajp://localhost:8009/
# http プロトコルで Apache と Tomcat 間の通信を行いたい場合は
# 上記の設定をコメント行にして、下記をコメントアウトにする。
# ProxyPass http://localhost:8080/
</Location>
```

- localhost/tomcat/ へアクセス確認
- アプリの配置
 - Tomcat5.0 からコンテキストの設定方法が変わりました。コンテキストの設定は \$CATALINA_HOME/conf/[engine name]/[host name] ディレクトリ配下にコンテキスト名 .xml という XML を作成して行います。
 - sudo vi /etc/tomcat6/Catalina/localhost/testapp.xml

```
<Context path="/testapp"
docBase="/var/www/servlet_test"
reloadable="false"/>
```

- reloadable は開発以外は false にする。開発時は true。

ディレクトリ階層を深くする場合は設定ファイル名を「ディレクトリ # アプリ名 .xml」にする。

war ファイルの設置について

<http://www.alles.or.jp/~torutk/ojava/sabbath/deploytomcat.html>

Tomcat の場合、\$CATALINA_HOME/webapps の下にディレクトリを作成して、war を設置する。server.xml で

```
unpackWARs="true"
```

となっていれば、war が自動で解凍される。

また、<Context> 要素の docBase にディレクトリを指定した場合は、war ファイルを置いても認識されない。<Context> 要素で war ファイルを指定する場合は、docBase 属性に war ファイルを指定する。

この場合は、war ファイルが \$CATALINA_HOME/webapps に解凍される。

HelloWorld

ディレクトリ構成を作成する

```
aproot
|-WEB-INF
    |-classes
    |   |-HelloWorld.class
    |-lib
    |-web.xml
```

プログラムを作成

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorld extends HttpServlet {
    public void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException
    {
        PrintWriter out;

        response.setContentType("text/html; charset=Shift_JIS");
        out = response.getWriter();

        out.println("<html><body>");
        out.println("<h1>Hello World!</h1>");
        out.println("<p>Servlet のサンプル (HelloServlet.java) </p>");
        out.println("</body></html>");
    }
}
```

コンパイル

```
javac -cp /usr/share/java/apache-tomcat-apis/tomcat-servlet2.5-api.jar:. HelloWorld.java
```

web.xml の作成

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">

<web-app>
  <servlet>
    <servlet-name>hello</servlet-name>
    <servlet-class>HelloWorld</servlet-class>
  </servlet>

  <servlet-mapping>
    <servlet-name>
      hello
    </servlet-name>
    <url-pattern>
      /servlet/hello
    </url-pattern>
  </servlet-mapping>
</web-app>
```

```
</url-pattern>  
</servlet-mapping>  
</web-app>
```

実行

<http://localhost/tomcat/testapp/servlet/hello>

へアクセスして実行。