



**SIMPLY RICH**

**ZK™**

# **クイックスタートガイド**

**Version 2.4.1**

July 2007

**Potix Corporation**

**Copyright © Potix Corporation. All rights reserved.**

The material in this document is for information only and is subject to change without notice. While reasonable efforts have been made to assure its accuracy, Potix Corporation assumes no liability resulting from errors or omissions in this document, or from the use of the information contained herein.

Potix Corporation may have patents, patent applications, copyright or other intellectual property rights covering the subject matter of this document. The furnishing of this document does not give you any license to these patents, copyrights or other intellectual property.

Potix Corporation reserves the right to make changes in the product design without reservation and without notification to its users.

The Potix logo and ZK are trademarks of Potix Corporation.

All other product names are trademarks, registered trademarks, or trade names of their respective owners.

# Table of Contents

はじめに.....	4
サーブレットコンテナ (Java Web Server)のご紹介.....	4
Java 言語のご紹介.....	4
統合開発環境 (IDE)のご紹介.....	4
<b>1. ダウンロード.....</b>	<b>5</b>
<b>2. デモファイルを実行してみよう.....</b>	<b>6</b>
<b>3. インストール.....</b>	<b>7</b>
方法 1: ZK をアプリケーションに添付する.....	7
方法 2: ZK ライブラリーを共有フォルダに置く.....	7
MySQL を使う.....	8
<b>4. ZK パッケージ.....</b>	<b>9</b>
/doc.....	9
/dist/lib.....	9
/dist/lib/zkforge.....	9
/dist/lib/ext.....	9
/dist/src.....	10
/dist/xsd.....	10
/dist/WEB-INF.....	10
<b>5. デモパッケージ.....</b>	<b>11</b>
/.....	11
/zkdemo.....	11
/MyApp.....	11
<b>6. 初めての ZK アプリ.....</b>	<b>12</b>
初めての Hello World.....	14

# はじめに

---

## サーブレットコンテナ (Java Web Server)のご紹介

Java でウェブアプリケーションを開発するにはサーブレットコンテナをインストールしなければなりません。Apache Tomcat は最も普及しているサーブレットコンテナの一つです。Tomcat をインストールする方法はとても簡単です。

Download	Installer for Windows: <a href="#">apache-tomcat-5.5.20.exe</a>
	A list of all available versions: <a href="http://tomcat.apache.org/download-55.cgi">http://tomcat.apache.org/download-55.cgi</a>
Documentation	<a href="http://tomcat.apache.org/tomcat-5.5-doc/index.html">http://tomcat.apache.org/tomcat-5.5-doc/index.html</a>

## Java 言語のご紹介

HTML に類似する ZUML という言語を使えば、Java が分からなくても ZK でリッチなユーザーインターフェースを実現することは可能です。但し、一つのウェブアプリケーションを完成するのに、Java の基本知識は必要です。Java のチュートリアルをいくつかリストアップしました。

Language Basic	<a href="http://java.sun.com/docs/books/tutorial/java/nutsandbolts/index.html">http://java.sun.com/docs/books/tutorial/java/nutsandbolts/index.html</a>
Class and Object	<a href="http://java.sun.com/docs/books/tutorial/java/concepts/index.html">http://java.sun.com/docs/books/tutorial/java/concepts/index.html</a> <a href="http://java.sun.com/docs/books/tutorial/java/javaOO/index.html">http://java.sun.com/docs/books/tutorial/java/javaOO/index.html</a> <a href="http://java.sun.com/docs/books/tutorial/java/IandI/index.html">http://java.sun.com/docs/books/tutorial/java/IandI/index.html</a>

## 統合開発環境 (IDE)のご紹介

Eclipse は最もよく知られている Java 統合開発環境の一つです。IDE の使用により、ウェブアプリケーションの開発・デバッグは簡単になります。さらに、ZK のソースコードをデバッグする作業を通して ZK への理解を深めることができます。

ステップ・バイ・ステップガイドは”small talks”より確認できます。

<http://www.zkoss.org/smalltalks/eclipse/ek.html>

## 1. ダウンロード

---

File	Description
zk-2.4.1.tar.gz zk-2.4.1.zip	ZK のバイナリパッケージです。ライブラリー・デモ用サンプルも含まれています。  このファイルがあれば ZK を使用することが可能です。
zk-demo-2.4.1.zip	独立した ZKdemo アプリケーションを WAR と EAR のフォーマットで提供します。デモアプリケーションのソースコードも含まれています。  デモ用サンプルを見るにはウェブサーバ等の設定は一切不要です。
zk-javadoc-2.4.1.zip	ZK フレームワークにて Java API を使用するマニュアルです。
zk-src-2.4.1.tar.gz	ZK フレームワークのソースで、サードパーティのライブラリーも含んでいます。

## 2. デモファイルを実行してみよう

---

Zk の能力を試すのであれば、下記の URL を参照しましょう。

<http://www.zkoss.org/zkdemo/userguide>.

また、ローカルサーバでデモを実行する方法は次の通りです。

1. (sourceforge.net) から zk-demo-2.4.1.zip をダウンロードします。
2. ファイルを解凍し、zkdemo-all.war か zkdemo-all.ear をウェブサーバまたはアプリケーションサーバにデプロイします。

ほとんどのサーバは管理用のコンソールを持ち、それを使えば簡単にアプリケーションをデプロイすることができます。手順は関連するマニュアルを参照するか、下記 ZK コミュニティーの URL をご確認ください:

[http://en.wikibooks.org/wiki/ZK/How-Tos#Installation\\_and\\_Configuration](http://en.wikibooks.org/wiki/ZK/How-Tos#Installation_and_Configuration),

Tomcat サーバの場合は、アプリケーションを \$TOMCAT\_DIR/webapps フォルダにコピーすれば Tomcat は自動的にデプロイしてくれます。

3. zkdemo-all.war のデプロイが完了しましたら、<http://localhost:8080/zkdemo-all/userguide> へアクセスしてみてください。ポート番号はウェブサーバの設定によって、必ずしも 8080 ではありません。

## 3. インストール

---

ZK はライブラリー集(the JAR files)を持っています。これらのライブラリーを Tomcat サーバへインストールする方法は二つあります。一つ目はこれらのファイルをウェブアプリケーションに添付(bundle)します。つまり、アプリケーションの WEB-INF/lib フォルダにコピーします。この方法のメリットは今後アプリケーションを他のウェブサーバにデプロイする際、サーバでの設定は不要になります。

二つ目はライブラリーを使用するウェブサーバまたはアプリケーションサーバにインストールします。つまり、共有フォルダにコピーします。(Tomcat の場合は shared/lib)。この方法のメリットは共有ディレクトリにアクセスできるすべてのアプリケーションは ZK を使用することが可能になります。

### 方法 1: ZK をアプリケーションに添付する

ZK のライブラリーをウェブアプリケーションの適当なフォルダにコピーすることで、アプリケーションをサーバへデプロイする際、サーバ側の設定は一切不要になります。

#### ZK ライブラリーを WAR ファイルに添付する(ウェブアプリ)

1. zk-2.4.1.zip または zk-2.4.1.tar.gz を解凍します。
2. dist/lib/\*.jar, dist/zkforge/\*.jar, dist/lib/ext/\*.jar をウェブアプリケーションの /WEB-INF/lib ディレクトリにコピーします。

dist/zkforge の下にある全部のライブラリーと libraries in dist/lib/ext の下にある一部のライブラリーはオプションライブラリーです。オプションファイルについては次の章をご覧ください。

#### ZK ライブラリーを EAR ファイルに添付する(the Java EE application)

1. zk-2.4.1.zip か zk-2.4.1.tar.gz を解凍します。
2. dist/lib/\*.jar, dist/zkforge/\*.jar, dist/lib/ext/\*.jar を Java Ee アプリケーションのホームディレクトリにコピーします。

dist/zkforge の下にある全部のライブラリーと libraries in dist/lib/ext の下にある一部のライブラリーはオプションです。オプションファイルについては次の章をご覧ください。

### 方法 2: ZK ライブラリーを共有フォルダに置く

この方法は ZK ライブラリーを共有フォルダに置きます。やり方はウェブサーバによって違います。

#### ZK を Tomcat にインストール

1. <http://tomcat.apache.org> から Tomcat をダウンロードし、インストールします。
2. Tomcat を停止します。
3. zk-2.4.1.zip もしくは zk-2.4.1.tar.gz を解凍します。

4. `dist/lib/*.jar` と `dist/zkforge/*.jar` と `dist/lib/ext/*.jar` を `$TOMCAT_HOME1/shared/lib` にコピーします。  
`dist/zkforge` の全てのライブラリーと `libraries in dist/lib/ext` にある一部のライブラリーはオプションです。オプションファイルについては次の章をご覧ください。
5. Tomcat を再起動します。

上記手順に従ってライブラリーのコピーが完了したら、ウェブアプリケーション( `demo/bin/zkdemo.war` ) を Tomcat にデプロイしましょう。Tomcat マネジャーを利用してデプロイするか、ファイルを `$TOMCAT_HOME/webapps` にコピーします。ファイルをコピーする場合、Tomcat を一旦停止しなければなりません。デプロイが完了したら `http://localhost:8080/zkdemo/userguide` より結果を確認できます。なお、上記のパスはウェブサーバの設定により異なります。

## ZK を Jetty にインストール

1. `http://www.mortbay.org/jetty/index.html` から Jetty をダウンロードし、インストールします。
2. Jetty を停止します。
3. `zk-2.4.1.zip` もしくは `zk-2.4.1.tar.gz` を解凍します。
4. `dist/lib/*.jar` と `dist/zkforge/*.jar` と `dist/lib/ext/*.jar` を `$JETTY_HOME/ext` にコピーします。`dist/zkforge` にある全てのライブラリーと `libraries in dist/lib/ext` にある一部のライブラリーはオプションライブラリーです。オプションファイルについては次の章をご覧ください。
5. Jetty を再起動します。

上記手順に従ってライブラリーのコピーが完了したら、ウェブアプリケーション( `demo/bin/zkdemo.war` ) を `$JETTY_HOME/webapps` にコピーしましょう。デプロイ成功したら `http://localhost:8080/zkdemo/userguide` より結果を確認できます。なお、上記のパスはウェブサーバの設定により異なります。

## MySQL<sup>2</sup>を使う

`zscript` にてコネクションを張るには MySQL JDBC ドライバ( `mysql-connector-*.jar` ) を `$TOMCAT_DIR/common/lib` に置かなければなりません。

---

<sup>1</sup> `$TOMCAT_HOME` は Tomcat をインストールしたホームディレクトリーです。

<sup>2</sup> `http://www.mysql.com`



## 4. ZK パッケージ

---

この章は zk-2.4.1.zip について説明します。

### /doc

コピーライト・リリースノート等ドキュメントはこのフォルダに入っています。

### /dist/lib

ZK を実行するのに必要なライブラリーはこのフォルダにあります。

### /dist/lib/zkforge

FCKeditor や Dojo 等 ZK Forge のコンポーネントはこのフォルダに入っています。これらのコンポーネントを使用する場合のみ、下記のファイルが必要です。

File	Description
dojoz.jar	ZK DOJO コンポーネントを使う場合使用
fckez.jar	ZK FCKeditor コンポーネントを使う場合使用
gmapsz.jar	ZK Google Maps コンポーネントを使う場合使用
timelinez.jar json_simple.jar	ZK Timeline コンポーネントを使う場合使用

### /dist/lib/ext

ZK を実行するのに必要な外部ライブラリーはこちらになります。一般的なライブラリーですので、既にインストールされている可能性があります。

オプションの jar ファイルは以下です。デザインにより、必要なものだけコピーしましょう。

File	Description
commons-fileupload.jar commons-io.jar	これらのファイルでアップロード機能を実現する場合に使う Version: Commons Fileupload 1.2 and Commons IO 1.3.1
commons-el.jar	EL を評価する場合に使う Version: Commons EL 1.0
jcommon.jar jfreechar.jar	ZUL のチャートコンポーネントを使用する場合に使う。 Version: JFreeChart 1.0.5 and JCommon 1.0.9
bsh.jar	Java interpreter (BeanShell)でスクリプトを作成する場合に使う Version: BeanShell 2.0b4
js.jar	JavaScript (Rhino)でスクリプトを作成する場合に使う

File	Description
	Version: Rhino 1.6R5
groovy.jar	Groovy でスクリプトを作成する場合に使う  Version: Groovy 1.0 (groovy-all)
jruby.jar	Ruby (JRuby)でスクリプトを作成する場合に使う  Version: JRuby 1.0 (jruby-complete)
Filters.jar	captcha コンポーネントを使用する場合に使う  Version: JH Labs Java Image Filters

## /dist/src

Eclipse 等 IDE をデバッグするに使うソースコードは JAR ファイルの形で保存されています。これらのファイルからバイナリライブラリーを編集することはできません。編集する場合は zk-src-2.4.1.tar.gz をダウンロードし、解凍してください。

## /dist/xsd

ZK アプリケーションを開発する際便利な XSD ファイルはこちらです。

## /dist/WEB-INF

ZK アプリケーションを開発する際便利な TLD ファイルはこのフォルダに入っています。

## 5. デモパッケージ

---

この章は zk-demo-2.4.1.zip について説明します。

/

zkdemo.war, zkdemo-all.war\*zkdemo.ear\*zkdemo-all.ear 計 3 つの実行ファイルがこのフォルダに入っています。  
詳細は **3.インストール** を参考にしてください。

### /zkdemo

ライブデモのソースコードはこちらに入っています。

### /MyApp

このディレクトリーの下でウェブアプリケーションを新規作成しましょう。

## 6. 初めての ZK アプリ

### WEB-INF/web.xml を用意しましょう

まずは以下の内容を WEB-INF の下にある web.xml にコピーしましょう。新しいアプリケーションを開発するたびに必ずこれを実行しなければなりません。次に ZUL・ZHTML ファイルを適当なディレクトリにコピーします。

```
<web-app version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">

    <!-- //// -->
    <!-- ZK -->
    <listener>
        <description>Used to cleanup when a session is destroyed</description>
        <display-name>ZK Session Cleaner</display-name>
        <listener-class>org.zkoss.zk.ui.http.HttpSessionListener</listener-class>
    </listener>
    <servlet>
        <description>ZK loader for evaluating ZK pages</description>
        <servlet-name>zkLoader</servlet-name>
        <servlet-class>org.zkoss.zk.ui.http.DHtmlLayoutServlet</servlet-class>

        <!-- Must. Specifies URI of the update engine
        (DHtmlUpdateServlet). -->
        <init-param>
            <param-name>update-uri</param-name>
            <param-value>/zkau</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup><!-- MUST -->
    </servlet>
    <servlet-mapping>
        <servlet-name>zkLoader</servlet-name>
        <url-pattern>*.zul</url-pattern>
    </servlet-mapping>
    <servlet-mapping>
        <servlet-name>zkLoader</servlet-name>
        <url-pattern>*.zhtml</url-pattern>
    </servlet-mapping>
    <servlet>
        <description>The asynchronous update engine for ZK</description>
        <servlet-name>auEngine</servlet-name>
        <servlet-class>org.zkoss.zk.au.http.DHtmlUpdateServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>auEngine</servlet-name>
        <url-pattern>/zkau/*</url-pattern>
    </servlet-mapping>
    <!-- //// -->
```

```

<!-- MIME mapping -->
<mime-mapping>
    <extension>gif</extension>
    <mime-type>image/gif</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>html</extension>
    <mime-type>text/html</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>htm</extension>
    <mime-type>text/html</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>jad</extension>
    <mime-type>text/vnd.sun.j2me.app-descriptor</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>jpeg</extension>
    <mime-type>image/jpeg</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>jpg</extension>
    <mime-type>image/jpeg</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>js</extension>
    <mime-type>application/x-javascript</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>png</extension>
    <mime-type>image/png</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>txt</extension>
    <mime-type>text/plain</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>xml</extension>
    <mime-type>text/xml</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>zhtml</extension>
    <mime-type>text/html</mime-type>
</mime-mapping>
<mime-mapping>
    <extension>zul</extension>
    <mime-type>text/html</mime-type>
</mime-mapping>

<welcome-file-list>

```

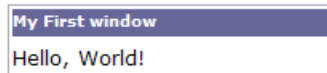
```
<welcome-file>index.zul</welcome-file>
<welcome-file>index.zhtml</welcome-file>
<welcome-file>index.html</welcome-file>
<welcome-file>index.htm</welcome-file>
</welcome-file-list>
</web-app>
```

## 初めての Hello World

以下のプログラムと同じ内容の hello.zul ファイルを作りましょう。そうすればブラウザ( <http://localhost:8080/zkdemo/hello.zul> )から結果が確認できます。パスはウェブサーバの設定により異なります。

```
<window title="My First window" border="normal" width="200px">
    Hello, World!
</window>
```

結果は以下の通りです：



ここで注意しなければならないのは、hello.zul は XUL と類似していますが、それは XUL ではなく ZUML です。ZL ロードーは ZUML を IE 等のブラウザが読める HTML ページに変換します。詳しくは Developer's Guide をご覧ください。