

GKitTaglib

CheckFilter

(チェックフィルタ)

マニュアル

20080318 版

目次

CheckFilter 概要	3	ファイル保存	25
製品構成	3	XML 結合	26
インストール配布物.....	3	JAVA ファイル読み込み	26
動作環境	4	サーブレットコピー	26
セットアップ	5	パラメータコピー	27
インストール方法 (Windows)	5	パラメータ移動	27
インストール方法 (RedHat・Solaris)		サーブレット・パラメータ追加	27
.....	5	サーブレット・パラメータ削除	28
仕組み	5	サーブレット・パラメータ名前変更	
設定ファイルについて	6	28
CheckFilter.xml の設定	6	web.xml の記述について	29
config タグ	6	使用例	32
post タグ	6	jar ファイルの配置.....	32
parameter タグ.....	7	CheckFilter の設定.....	32
input タグ	9	web.xml の設定.....	32
permit タグ	9	サンプルの配置	32
output タグ	16	動作確認手順	33
substitution タグ	16	ライセンス	34
trim タグ.....	17	サポート	34
フィルター対象の JSP ファイルの設定		制約事項	35
.....	18	アトリビュート一覧	35
誤入力の扱いについて	19	config タグ	35
設定と異なる値が入力された場合.....	19	post タグ	35
ErrorManager	20	parameter タグ	35
ErrorInfo.....	20	input タグ	36
CheckFilterConfigTool	21	permit タグ	36
概要説明	21	output タグ	38
動作環境	21	substitution タグ	38
Eclipse Plugin 版インストール方法..	21	trim タグ	38
ツール画面説明.....	22		
機能説明	25		
新規作成	25		

CheckFilter 概要

CheckFilter は JSP/Servlet の WEB アプリケーションでサーバでの入力チェックを行います。CheckFilter を使用することにより、WEB アプリケーションでの入力チェックを単純なものとし、開発効率を飛躍的に高めることができます。

●CheckFilter にできること

本システムは以下のチェックを行います。

- ・ 文字の種類
- ・ 文字数
- ・ 必須入力
- ・ 入力許可文字・禁止文字
- ・ メールアドレス
- ・ 時間型フォーマット
- ・ 正規表現
- ・ 数字の最大値・最小値
- ・ 文字の置換

製品構成

インストールされる製品構成および動作環境を以下に説明します。

インストール配布物

[アプリケーションルート]

└[WEB-INF]

└[lib]

CheckFilter で使用するライブラリが格納されています。

下記の動作環境をサポートしています。

◆ サーバ

① オペレーティングシステム

- ・ Microsoft Windows Server 2003 日本語版
- ・ RedHat Linux Enterprise Edition version 3
- ・ Solaris 9 (SPARC)

② JSP コンテナ

- ・ Fujitsu Interstage Application Server V6.0/V7.0/V8.0/V9.0
- ・ BEA WebLogic Server 8.1J/9.2J/10.0J
- ・ IBM WebSphere Application Server V5.0/V6.1
- ・ Jakarta Tomcat 4.1/5.5/6.0
- ・ Servlet API2.2 JSP1.1 に準拠するコンテナ

③ JDK

- ・ 1.3.1_06 以上
- ・ J2SE SDK 1.3 以下の場合

J2SE SDK のバージョンが 1.3 以下の場合 SAX の API を使用可能にしてください。Java XML Pack に入っている JAXP (Java API for XML Processing) を使用します。Java XML Pack のダウンロードはこちらから行ってください。

<http://java.sun.com/xml/downloads/javaxmlpack.html>

ダウンロードしたファイルに含まれている jaxp-api.jar、dom.jar、sax.jar、rcesimple.jar を CLASSPATH に追加してください。

④ CPU

- ・ 500MHz 以上

⑤ メモリ

- ・ 512MB 以上

◆ クライアント

① オペレーティングシステム

- ・ Microsoft Windows 2000
- ・ Microsoft Windows XP
- ・ Microsoft Windows Vista

② ブラウザ

- ・ Microsoft Internet Explorer 5.5SP2/6.0SP1/6.0SP2/7.0

セットアップ

ここでは各OSでのセットアップ手順について、以下に説明します。

インストール方法 (Windows)

インストールする端末にCDを入れるとインストーラが自動的に起動します。インストーラ起動後は、インストーラの指示に従ってインストールを行ってください。

※自動的に起動しない場合はCD内の INSTALL.exe を実行してください。

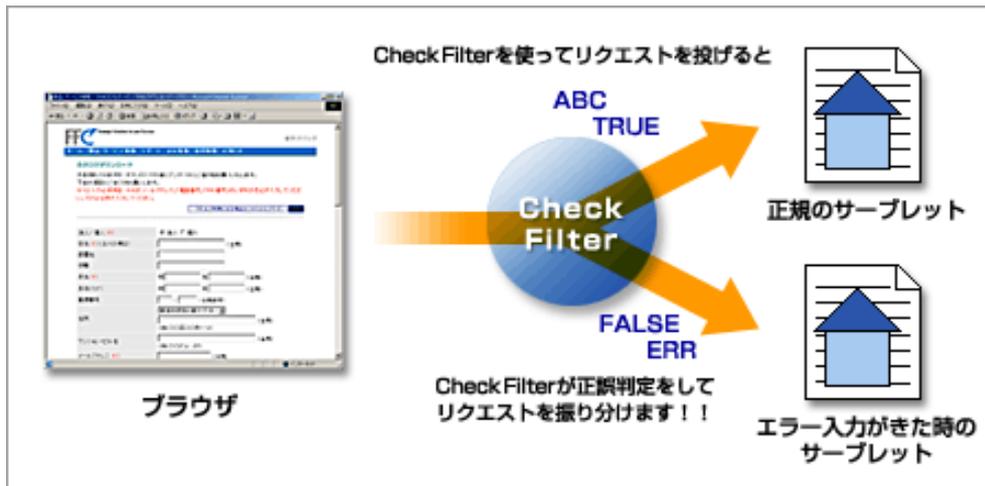
インストール方法 (RedHat・Solaris)

RedHat・Solaris でインストールを行うために CD の tar フォルダに TAR ファイルを用意してあります。tar コマンドでファイルを展開してください。

※インストールされるものは Windows インストーラでインストールされるものと同一です。

仕組み

リクエストを行ったとき、入力データが設定どおりの場合は正規のリクエスト先へ、誤入力の場合はエラーリクエスト先へ振り分けられます。



設定ファイルについて

CheckFilter.xml でリクエストごとのフィルタ設定をします。

CheckFilter.xml の設定

CheckFilter.xml はどのパラメータにどのようなチェックをするかを記述する設定ファイルです。

config タグ

CheckFilter でチェックする設定内容を囲むタグです。この中にリクエストごとの設定を記述します。

【例】

```
<config>
  <post url="request" error="/error.jsp"/>
</config>
```

post タグ

対象となるリクエストの設定とエラーの際に遷移させるページの設定項目です。

◆アトリビュート

①url

リクエストの設定をします。記述する内容は SUBMIT する JSP に記述される下記の

```
<input type="hidden" name="gkit.checkfilter" value="request.jsp">
```

の value に設定した値です。

※入力チェックを行う為には" gkit.checkfilter" のキー名でチェックするリクエストを指定する必要があります。

※url アトリビュートは必ず記述してください。

【例】

JSP に下記のようにリクエスト gkit.checkfilter を設定した場合

```
<input type="hidden" name="gkit.checkfilter" value="request.jsp">
```

post の url を下記のように設定する必要があります。

```
<config>
  <post url="request.jsp" error="/error.jsp"/>/post>
</config>
```

同じにする

②error

エラーが発生した際にリクエストを送る JSP/Servlet を設定します。

【例】

エラー発生時にリクエストを送る JSP/Servlet ファイルを" error.jsp" と設定する場合

```
<config>
  <post url="request.jsp" error="/error.jsp"/>/post>
</config>
```

【省略時】

設定がない場合、リクエスト先にサブミットされます。

parameter タグ

対象となる JSP/Servlet のチェックするパラメータの設定項目です。

◆アトリビュート

①name

リクエストの name を設定します。

※name アトリビュートは必ず設定してください。

【例】

リクエストの name を“VALUE”に設定する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      </parameter>
    </post>
  </config>
```

input タグ

個々のリクエストパラメータのチェック項目を設定します。

※アトリビュートはありません。

【例】

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permitted="HALFWIDTH-NUMBER;"
        isable-type="" />
      </input>
    </parameter>
  </post>
</config>
```

permit タグ

何をチェックするか具体的なリクエストの制限項目を設定します。

◆アトリビュート

①type

入力を許可するタイプを設定します。設定されたタイプ以外の入力があった場合はエラーとなります。複数選択可能です。タイプの区切りに必ず“;”（セミコロン）を記述してください。

【例】

半角の数字とアルファベットの入力を許可する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permittedtype="HALFWIDTH-NUMBER;
        HALFWIDTH-ALPHABET"/>
      </input>
    </parameter>
  </post>
</config>
```

No	設定値	意味
1	ALLCHARS	文字列を制限しない
2	FULLWIDTH	全角すべて
3	FULLWIDTH-HIRAGANA	ひらがな
4	FULLWIDTH-KATAKANA	全角カタカナ
5	FULLWIDTH-ALPHABET	全角アルファベット
6	FULLWIDTH-ALPHABET-CAPITAL	全角アルファベット大文字
7	FULLWIDTH-ALPHABET-SMALL	全角アルファベット小文字
8	FULLWIDTH-NUMBER	全角数字
9	FULLWIDTH-KANJI	全角漢字
10	FULLWIDTH-OTHER	全角その他
11	HALFWIDTH	半角すべて
12	HALFWIDTH-ALPHABET-CAPITAL	半角アルファベット大文字
13	HALFWIDTH-ALPHABET-SMALL	半角アルファベット小文字
14	HALFWIDTH-NUMBER	半角カタカナ
15	HALFWIDTH-NUMBER	半角数字
16	HALFWIDTH-OTHER	半角その他
17	MAILADDRESS	メールアドレス
18	TIME-FORMAT	時間型フォーマット
19	FORMAT	フォーマット（正規表現）

②format

フォーマットを設定します。設定されフォーマット以外の入力があった場合はエラーとなります。
※フォーマットは正規表現で設定します。

【例】

時間型フォーマットを“YY/MM/DD hh:mm:ss”と設定する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permit format="YY/MM/DD hh:mm:ss:"/>
      </input>
    </parameter>
  </post>
</config>
```

③required

リクエストの入力が必須かどうかの設定をします。

“YES”：リクエストの中にパラメータが無い場合、空の場合にエラーとなる

“NO”：リクエストの中にパラメータが無い場合、空の場合はエラーにならない

【例】

リクエストの中に VALUE パラメータがない時にエラーとなるように設定する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permitted="HALFWIDTH-NUMBER:"
          required="YES"/>
      </input>
    </parameter>
  </post>
</config>
```

④maxlength

入力された文字の最大文字数を設定します。文字数が設定を超えた場合エラーとなります。
0 ~ 2147483647 の範囲で設定してください。

【例】

最大文字数を"10"に設定する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permittedtype="HALFWIDTH-NUMBER;"
          maxlength="10"/>
      </input>
    </parameter>
  </post>
</config>
```

⑤minlength

入力された文字の最小文字数を設定します。文字数が設定を下回った場合エラーとなります。
0 ~ 2147483647 の範囲で設定してください。

【例】

最小文字数を"3"に設定する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permittedtype="HALFWIDTH-NUMBER;" length="3"/>
      </input>
    </parameter>
  </post>
</config>
```

⑥maxvalue

入力された数値の最大値を設定します。設定した数値を超えた場合エラーとなります。
-2147483648 ~ 2147483647 の範囲で設定してください。

【例】

最大値を"10000"に設定する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permittedtype="HALFWIDTH-NUMBER;"
          maxvalue="10000"/>
      </input>
    </parameter>
  </post>
</config>
```

⑦minvalue

入力された数値の最小値を設定します。設定した数値を下回った場合エラーとなります。
-2147483648 ~ 2147483647 の範囲で設定してください。

【例】

最小値を"-10000"に設定する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permittedtype="HALFWIDTH-NUMBER;"
          minvalue="-10000"/>
      </input>
    </parameter>
  </post>
</config>
```

⑧disable-type

拒否するタイプを設定します。含まれていた場合エラーとなります。

【例】

半角の数字とアルファベットを入力するとエラーになる場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permittedisable-type="HALFWIDTH-NUMBER;
        HALFWIDTH-ALPHABET;"/>
      </input>
    </parameter>
  </post>
</config>
```

No	設定値	意味
1	ALLCHARS	文字列をすべて制限する
2	FULLWIDTH	全角すべて
3	FULLWIDTH-HIRAGANA	ひらがな
4	FULLWIDTH-KATAKANA	全角カタカナ
5	FULLWIDTH-ALPHABET	全角アルファベット
6	FULLWIDTH-ALPHABET-CAPITAL	全角アルファベット大文字
7	FULLWIDTH-ALPHABET-SMALL	全角アルファベット小文字
8	FULLWIDTH-NUMBER	全角数字
9	FULLWIDTH-KANJI	全角漢字
10	FULLWIDTH-OTHER	全角その他
11	HALFWIDTH	半角すべて
12	HALFWIDTH-ALPHABET-CAPITAL	半角アルファベット大文字
13	HALFWIDTH-ALPHABET-SMALL	半角アルファベット小文字
14	HALFWIDTH-NUMBER	半角カタカナ
15	HALFWIDTH-NUMBER	半角数字
16	HALFWIDTH-OTHER	半角その他
17	MAILADDRESS	メールアドレス
18	TIME-FORMAT	時間型フォーマット
19	FORMAT	フォーマット（正規表現）

⑨disable-chars

拒否する文字列を設定します。設定した文字列が含まれていた場合はエラーとなります。複数設定する場合は“;”（セミコロン）で区切って記述します。

【例】

“あいう”を入力したときにエラーと設定する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permit type="ALLCHARS" disable-chars ="あいう"/>
      </input>
    </parameter>
  </post>
</config>
```

⑩enable-chars

許可する文字列を設定します。設定した文字列が含まれていた場合はエラーとはなりません。複数設定する場合は“;”（セミコロン）で区切って記述します。

【例】

“あいう”を入力したときにエラーにしない場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permit type="HALFWIDTH-NUMBER"
          enable-chars ="あいう"/>
      </input>
    </parameter>
  </post>
</config>
```

output タグ

リクエストのパラメータを変換する設定を記述します。

【例】

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <output>
        <substitution target="a" replacement="A">
      </output>
    </parameter>
  </post>
</config>
```

substitution タグ

変換する項目を設定します。output タグ内に記述します。

◆アトリビュート

①target

変換対象となる文字列を設定します。

②replacement

変換する文字列を設定します。

【例】

全角数字を半角数字に変換する場合

```
<config>
  <post url="request" error="/error.jsp">
    <parameter name="VALUE">
      <input>
        <permit type="ALLCHARS" />
      </input>
      <output>
        <substitution target="0" replacement="0"/>
        <substitution target="1" replacement="1"/>
      </output>
    </parameter>
  </post>
</config>
```

```

        <substitution target=" 2" replacement="2"/>
        <substitution target=" 3" replacement="3"/>
        <substitution target=" 4" replacement="4"/>
        <substitution target=" 5" replacement="5"/>
        <substitution target=" 6" replacement="6"/>
        <substitution target=" 7" replacement="7"/>
        <substitution target=" 8" replacement="8"/>
        <substitution target=" 9" replacement="9"/>
    </output>
</parameter>
</post>
</config>

```

trim タグ

トリムを設定し、空白を削除する設定を行います。“left”、“center”、“right”のいずれかひとつを設定します。output タグ内に記述します。

- ・ left : 左側の空白を削除する
- ・ right : 右側の空白を削除する
- ・ center : 左右の空白以外の空白を削除する

【例】

トリムを“center”に設定する場合

```

<config>
    <post url="request" error="/error.jsp">
        <parameter name="VALUE">
            <output>
                <trim type="center"/>
            </output>
        </parameter>
    </post>
</config>

```

フィルター対象の JSP ファイルの設定

フィルターをかける対象の JSP ファイルにあらかじめインプットタグを設定する必要があります。

設定ファイルのサンプル

request.jsp:

```
<%@ page language="java" pageEncoding="SHIFT_JIS" %>
<!DOCTYPE HTML PUBLIC "-//w3c//dtd html 4.0 transitional//en">
<html>
  <head>
    <title>requeset JSP</title>
  </head>
  <body bgcolor="#FFFFFF">
    <form action="/response.jsp" method="post">
      <input type="hidden" name="gkit.checkfilter" value="request.jsp">
      <input type="text" name="VALUE">
      <br>
      <input type="text" name="STRING">
      <br>
      <input type="submit" value="submit">
    </form>
  </body>
</html>
```

- ①form タグの中に上記サンプルのように input タグを作成します。
- ②アトリビュートの name を "gkit.checkfilter" とします。
- ③value を設定ファイル "CheckFilter.xml" の url 名と同じ名前で設定します。

この、設定ファイル CheckFilter.xml の url と同じ名前になったパラメータが CheckFilter でチェックされることとなります。

name と value の項目が記入されていないとフィルタが動作しないので必ず記入してください。

誤入力の扱いについて

設定と異なる値が入力された場合

設定と違った値が入力された場合、設定された error にリクエストがいきます。そのリクエスト先で以下のようにして、誤入力の内容を取得することができます。

エラー情報取得のサンプル(error.jsp)

```
// エラー情報を生成
ErrorManager em = (ErrorManager) request.getAttribute("error infor");
// エラー対象があるか
if(em != null) {
    for(int i=0; i <= em.getErrorCount(); i++) {
        // エラー対象の取得
        ErrorInfo error = em.getErrorItem(i);
        if(error!=null) {
            out.println("パラメータ名 : " + error.getParamName() + "<br>");
            out.println("変換前 : " + error.getOldValue() + "<br>");
            out.println("変換後 : " + error.getNewValue() + "<br>");
            out.println("エラーフラグ : " + error.getErrorFlag() + "<br>");
            out.println("エラー内容 : " + error.getErrorId() + "<br>");
            out.println("<br>");
        }
    }
}
```

ErrorManager

◆エラー情報管理クラス

ErrorManager クラスは request の情報を管理するクラスとなっています。このクラスからエラー情報を取得できるようになります。

メソッド名	パラメータ	戻り値	説明
getErrors	なし	Vector	全部の ErrorInfo 対象を取得 エラー対象がない場合は null が返る
getErrorCount	なし	int	ErrorInfo 対象の数を取得 エラー情報がない場合は 0 が返る
getErrorItem	int	ErrorInfo	インデックス対応の ErrorInfo 対象を取得
getErrorUrl	なし	URL	エラー処理ページの URL を取得

ErrorInfo

◆エラー情報クラス

ErrorInfo クラスはエラーとなっている情報を取得するためのクラスとなっています。こちらのクラスは ErrorManager クラスでエラー情報を生成してから使用してください。

メソッド名	パラメータ	戻り値	説明
getParamName	なし	String	パラメータ名を取得
getOldValue	なし	String	変換前パラメータ値を取得
getNewValue	なし	String	変換後パラメータ値を取得
getErrorFlag	なし	boolean	エラーフラグを取得 エラーの場合は true が返る
getErrorId	なし	int	エラー内容を取得

CheckFilterConfigTool

概要説明

CheckFilter Config Tool は、CheciFilter の設定ファイルを作成するツールです。
本ツールを使用することにより、CheckFilter の設定ファイルを効率的に作成できます。

動作環境

◆OS

- ・ Windows2000
- ・ RedHat9
- ・ Solaris(SPARC)

◆JDK

- ・ JDK 1.3.0 以降 ※JAXP が必要です

Eclipse Plugin 版インストール方法

統合開発環境 Eclipse2.1 でこの CheckFilter ConfigTool が使用できます。

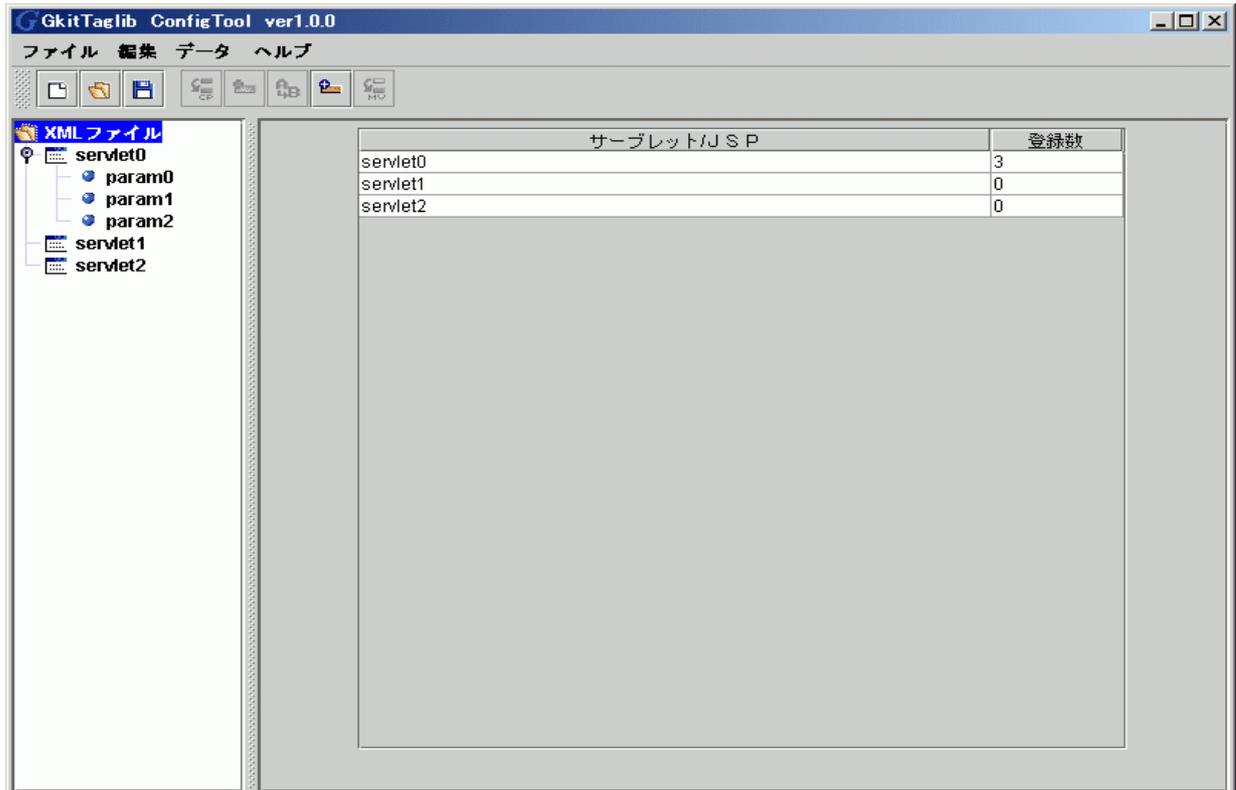
◆インストール

- 1.eclipse を停止します。
- 2.インストールフォルダにある jp.co.ffc.gkit.checkfilter.tool フォルダを eclipse の plugins フォルダに置いてください。
- 3.eclipse を起動します。
- 4.メニューバーの「ウィンドウ → ビューの表示 →その他」を選択します。
- 5.「ビューの表示」のツリーの「GkitTaglib」を開き、「CheckFilter ConfigTool 1.0.0」を選択し、「OK」を押下します。

XML ファイル、サーブレット、パラメータの3つの画面があります。

◆XML ファイル画面

初期画面ならびにツリーの XML ファイルを選択するとサーブレット名と各サーブレットのパラメータ登録数の一覧が Table で表示されます。XML ファイルはルートになっているので追加はできません。サーブレットの追加はツリーの XML ファイル名が選択状態のときに追加が行えます。



◆サーブレット画面

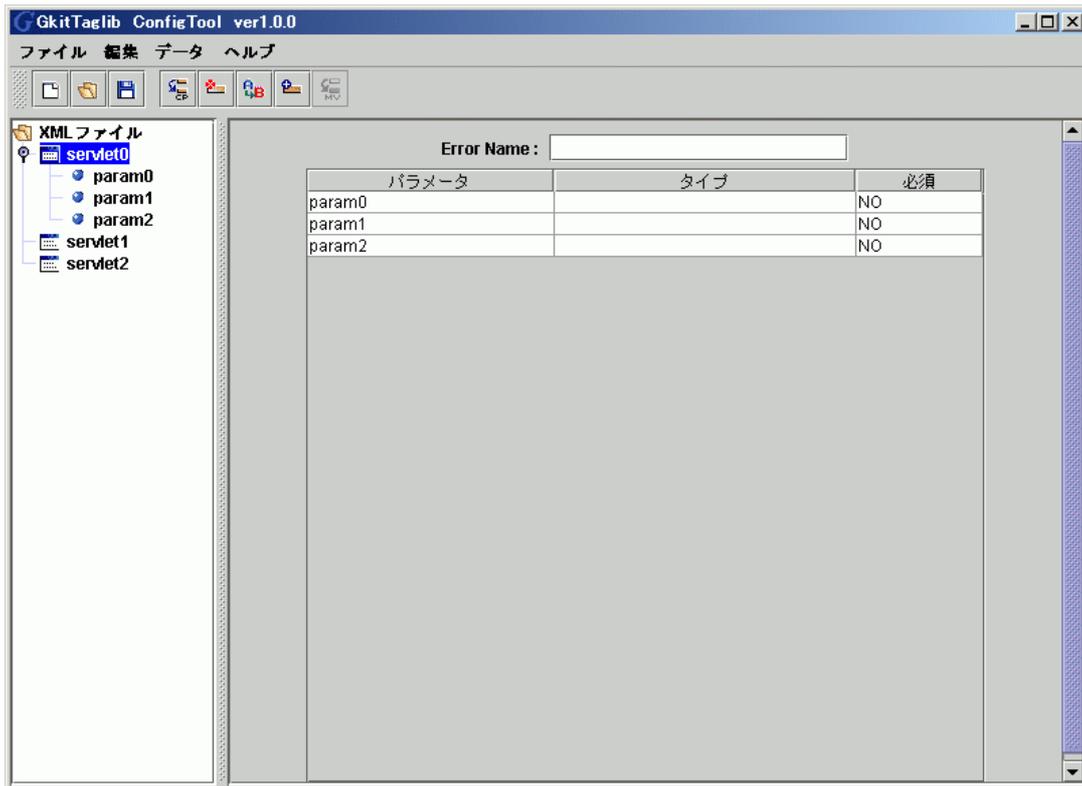
各サーブレット名が選択状態のとき、ツリーの右クリックでポップアップメニューやメニューバーの編集項目・ツールバーで、パラメータの追加や選択のサーブレット削除・コピー・名前変更が行えます。このサーブレット名と JSP ファイルでインプットタグ名 gkit.checkilter の value の値が一致していたら、サーブレット名の下階層に設定したパラメータをチェックすることになります。

◆エラーページ名の設定

エラーが発生したときにリクエストの転送先を記述します。エラーページは相対パスで指定してください。

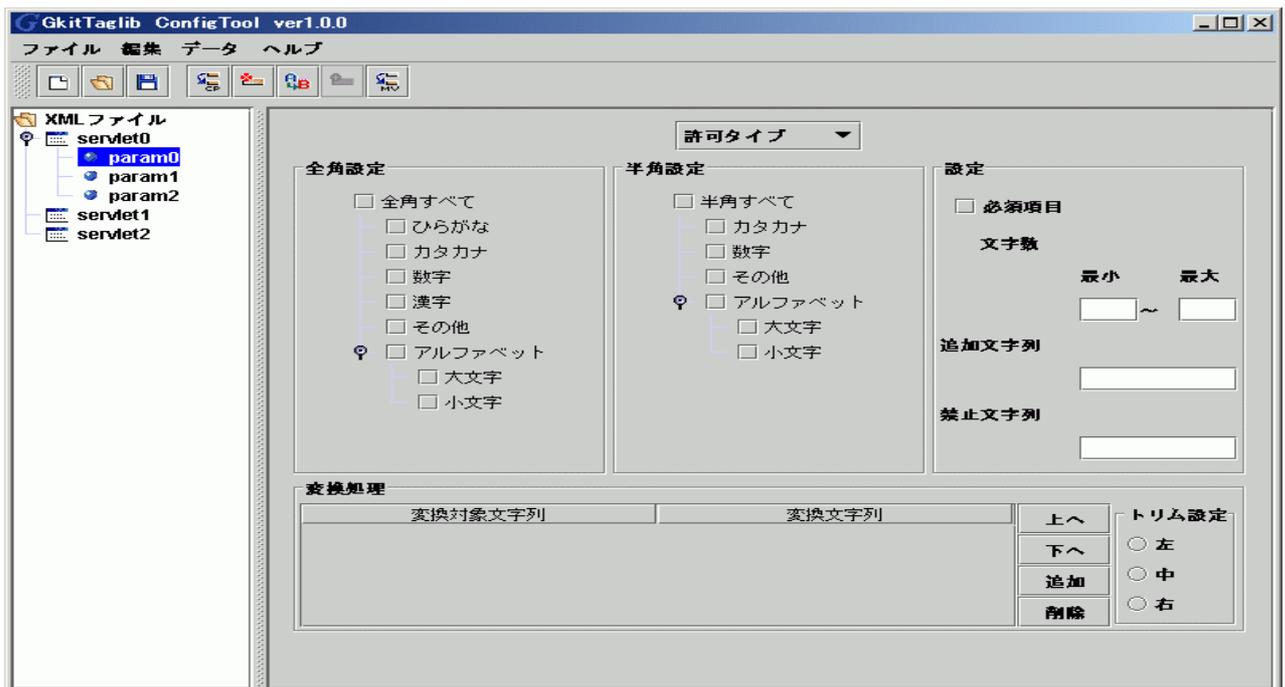
【例】

/error.jsp



◆パラメータ画面

ツリーの各パラメータ名を選択すると設定画面が表示されます。この設定画面で各パラメータのチェック項目を設定します。各パラメータ名が選択状態のとき、ツリーの右クリックでポップアップメニューやメニューバーの編集項目・ツールバーで、パラメータの削除・コピー・移動・名前変更が行えます。



①許可タイプ設定

ツリー型のチェックボックスに全角・半角設定をチェックするとそのチェックした項目が入力可能となります。チェックした項目以外を入力するとエラーとなります。その他、チェック項目以外で許可ならびに禁止設定する場合は、以下の項目を設定できます。

◆必須項目

必須項目のチェックボックスにチェックをつけるとそのパラメータは入力が必要となります。入力がない場合はエラーとなります。

◆文字数

入力文字数を最小値・最大値で設定します。最小値は0、最大値は2147483647まで設定できます。その設定した範囲外の場合はエラーとなります。

【例】

最小値3で最大値6だとしたら、2文字以内入力で送信したらエラーとなり、7文字以上入力でエラーとなります。

◆追加文字列

追加文字列を設定します。その文字列が入力されていないとエラーとなります。文字列を複数設定する場合は“;”で区切って入力してください。

◆禁止文字列

禁止文字列を設定します。その文字列が入力されるとエラーとなります。文字列を複数設定する場合は“;”で区切って入力してください。

◆変換処理

変換対象文字列と変換文字列を設定します。設定した対象文字列が入っている場合は変換されます。

【例】

変換対象文字列	変換文字列
<	&l t ;

◆トリム設定

左、中、右からトリム設定する場所を決めます。

②数値設定

数値を入力する場合に最小値、最大値を設定します。その範囲外のものはエラーとなります。最小値は“-2147483648”、最大値は“2147483647”まで設定できます。

※小数点は設定できません。

③メールアドレス設定

メールアドレス設定でチェックボックスにチェックを入れると設定したパラメータはメールアドレスのフォーマットでない場合にエラーとなります。

④フォーマット設定

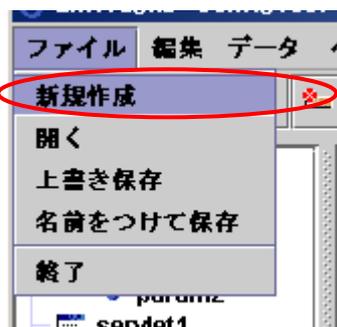
フォーマットの設定は時間型フォーマットと正規表現型フォーマットがあります。どちらかの設定ができ、フォーマットの設定がしてあるパラメータはその設定したフォーマット以外を入力するとエラーとなります。

機能説明

CheckFilter Config Tool の機能について説明します。

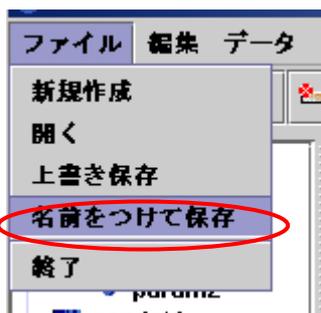
新規作成

初期状態、新規作成によって新規に XML ファイルを作成することができます。



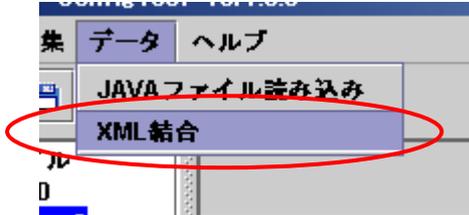
ファイル保存

名前をつけて保存、上書き保存など XML ファイルを作成することができます。encoding で指定された文字コードで保存を行います。XML 文字コードは UTF-8、SHIFT_JIS、EUC-JP となります。



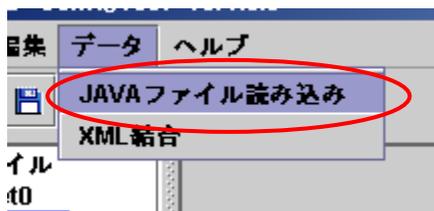
XML 結合

それぞれの環境で作成したXML ファイルを結合し、1つのXML ファイルとすることができます。文字コードがそれぞれのファイルで異なっても問題なく連結することができます。



JAVA ファイル読み込み

作成した Java ファイル、JSP ファイルを読み込んでツールに反映できます。取得するパラメータのメソッド (getParameter メソッド) を取得し、ツールにパラメータとして反映することができます。



サーブレットコピー

サーブレットの設定をコピーし、別な名前のサーブレット設定を作成します。操作はツリーのサーブレット名が選択状態の時に行うことができ、メニューバー・ツールバー・ツリー部の右クリックメニューで表示されるポップアップメニューから操作できます。コピー対象のサーブレットをコピーした場合にサーブレット階層の一番下にコピーしたサーブレットが追加されます。



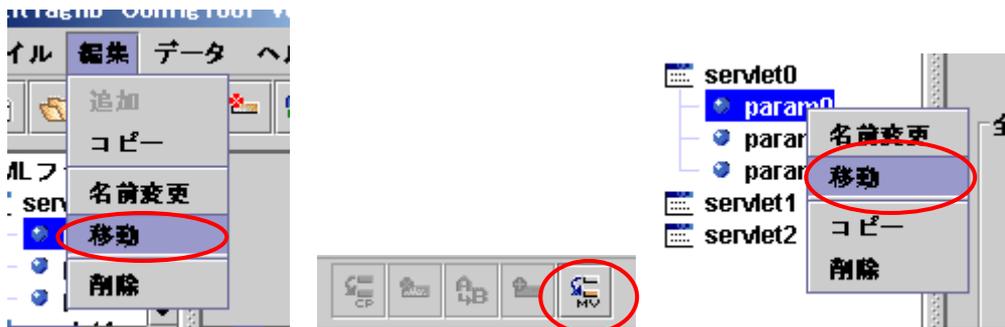
パラメータコピー

パラメータの設定をコピーし、別な名前パラメータ設定を作成します。操作はツリーのパラメータ名が選択状態の時に行うことができ、メニューバー・ツールバー・ツリー部の右クリックメニューで表示されるポップアップメニューから操作できます。コピー対象のパラメータをコピーした場合に、コピー元となるパラメータの親サブレット内のパラメータ階層の一番下にコピーしたパラメータが追加されます。



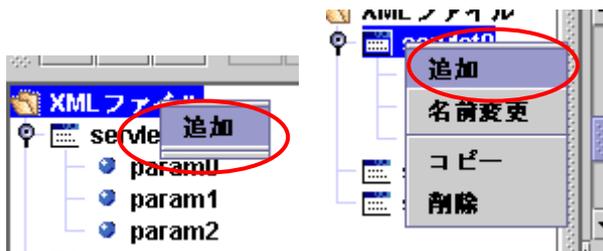
パラメータ移動

設定してあるパラメータを他のサブレットへ移動することができます。操作はツリーのパラメータ名が選択状態の時に行うことができ、メニューバー・ツールバー・ツリー部の右クリックメニューで表示されるポップアップメニューから操作できます。移動対象のパラメータ名を移動したときに、リスト表示のダイアログが表示され、移動先のサブレットを選択して実行をすれば、移動先にパラメータが移動します。



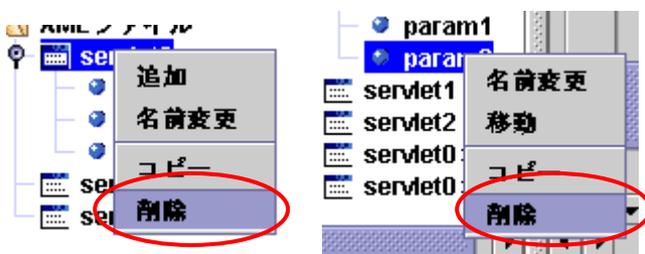
サブレット・パラメータ追加

サブレット・パラメータを追加することができます。ツリーのXMLファイルのところを選択されていたらサブレットが追加され、サブレットが選択されていたらパラメータが追加することができます。操作は各サブレット・パラメータ名が選択状態の時に行うことができ、メニューバー・ツールバー・ツリー部の右クリックメニューで表示されるポップアップメニューから操作できます。



サーブレット・パラメータ削除

ツリーの選択されているサーブレットかパラメータを削除することができます。操作は各サーブレット・パラメータ名が選択状態の時に行うことができ、メニューバー・ツールバー・ツリー部の右クリックメニューで表示されるポップアップメニューから操作できます。



サーブレット・パラメータ名前変更

ツリーの選択されているサーブレットがパラメータの名前を変更することができます。操作は各サーブレット・パラメータ名が選択状態の時に行うことができ、メニューバー・ツールバー・ツリー部の右クリックメニューで表示されるポップアップメニューから操作できます。



web.xml の記述について

web.xml には CheckFilter を使用するための設定を行います。

リクエストが `jp.co.ffc.gkit.checkfilter.CheckFilter` を通るようにフィルタ設定を行います。

◆マッピング

`jp.co.ffc.gkit.checkfilter.CheckFilter` にフィルタマッピングを行います。

```
<filter>
  <filter-name>CheckFilter</filter-name>
  <filter-class>jp.co.ffc.gkit.checkfilter.CheckFilter</filter-class>
</filter>

<filter-mapping>
  <filter-name>CheckFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

◆設定ファイルの指定

初期化時に CheckFilter が使用する設定ファイルを指定できます。

初期パラメータ `XMLPath` で設定ファイルを指定してください。

```
<init-param>
  <param-name>XMLPath</param-name>
  <param-value>/WEB-INF/CheckFilter.xml</param-value>
</init-param>
```

◆再読み込みの設定

初期パラメータ `Reload` で設定ファイルが変更されたときに、更新するかどうか設定します。

- true : 更新する
- false : 更新しない

```
<init-param>
  <param-name>Reload</param-name>
  <param-value>>false</param-value>
</init-param>
```

◆文字コードの設定

初期パラメータ Encoding でリクエストされるエンコードの文字コード設定を行います。

```
<init-param>  
  <param-name>Encoding</param-name>  
  <param-value>Shift_JIS</param-value>  
</init-param>
```

◆エンコードの設定

初期化パラメータ EncordSet でリクエストに対しエンコードするかどうかを設定します。

- ・ true : エンコード設定を行う (デフォルト)
- ・ false : エンコード設定を行わない

```
<init-param>  
  <param-name>EncordSet</param-name>  
  <param-value>>false</param-value>  
</init-param>
```

【例】

web.xml の設定例です。

```
<?xml version="1.0" encoding="Shift_JIS"?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
  "http://java.sun.com/j2ee/dtds/web-app_2_3.dtd">

<web-app>
  <filter>
    <filter-name>CheckFilter</filter-name>
    <filter-class>jp.co.ffc.gkit.checkfilter.CheckFilter</filter-class>
    <init-param>
      <param-name>XMLPath</param-name>
      <param-value>/WEB-INF/CheckFilter.xml</param-value>
    </init-param>
    <init-param>
      <param-name>Reload</param-name>
      <param-value>>false</param-value>
    </init-param>
    <init-param>
      <param-name>Encoding</param-name>
      <param-value>Shift_JIS</param-value>
    </init-param>
    <init-param>
      <param-name>EncordSet</param-name>
      <param-value>>false</param-value>
    </init-param>
  </filter>

  <filter-mapping>
    <filter-name>CheckFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>
</web-app>
```

使用例

CheckFilter を動かすための環境について、以下に説明します。

jar ファイルの配置

[アプリケーションルート]

└[WEB-INF]

└[class]

└[lib]

[アプリケーションルート]/WEB-INF/lib に、以下の jar ファイルを設置します。

- gkit-checkfilter-1.0.0.jar
- gkit-checkfilter-1.0.0Lic.jar
- jakarta-oro-2.0.7.jar

CheckFilter の設定

CheckFilter の設定については 5、6、7 章を参照してください。設定ファイルは[アプリケーションルート]/WEB-INF に配置します。

web.xml の設定

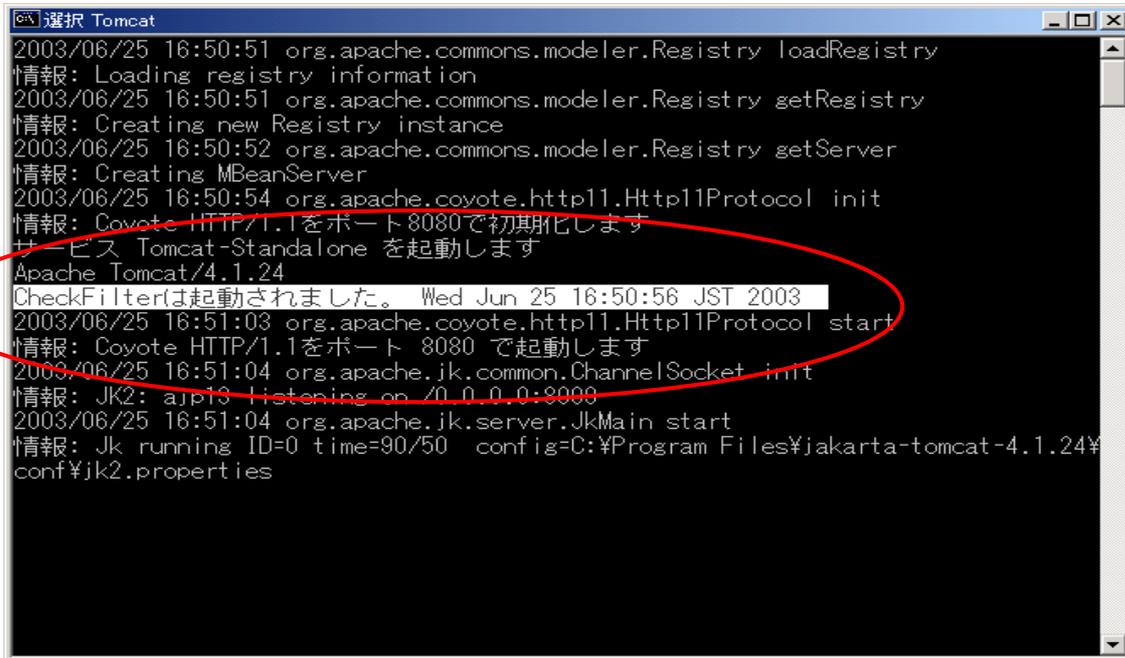
web.xml の設定については、8 章を参照してください。設定ファイルは[アプリケーションルート]/WEB-INF に配置します。

サンプルの配置

サンプルファイルの「request.jsp」「response.jsp」「error.jsp」を[アプリケーションルート]/に配置します。

動作確認手順

①Tomcat を起動します。“CheckFilter は起動されました。”という文字が表示されるのを確認してください。



```
2003/06/25 16:50:51 org.apache.commons.modeler.Registry loadRegistry
情報: Loading registry information
2003/06/25 16:50:51 org.apache.commons.modeler.Registry getRegistry
情報: Creating new Registry instance
2003/06/25 16:50:52 org.apache.commons.modeler.Registry getServer
情報: Creating MBeanServer
2003/06/25 16:50:54 org.apache.coyote.http11.Http11Protocol init
情報: Coyote HTTP/1.1をポート 8080で初期化します
サービス Tomcat-Standalone を起動します
Apache Tomcat/4.1.24
CheckFilterは起動されました。 Wed Jun 25 16:50:56 JST 2003
2003/06/25 16:51:03 org.apache.coyote.http11.Http11Protocol start
情報: Coyote HTTP/1.1をポート 8080 で起動します
2003/06/25 16:51:04 org.apache.jk.common.ChannelSocket init
情報: JK2: ajp13 listening on /0.0.0.0:8000
2003/06/25 16:51:04 org.apache.jk.server.JkMain start
情報: Jk running ID=0 time=90/50 config=C:\Program Files\jakarta-tomcat-4.1.24\
conf\jk2.properties
```

②ブラウザで下記の URL にアクセスします。

[http://localhost:8080/\[アプリケーションルート\]/request.jsp](http://localhost:8080/[アプリケーションルート]/request.jsp)

③表示されるテキストフィールドに文字を入力して「送信」ボタンを押下してください。

入力にエラーがある場合、エラー画面が表示されます。エラーが無い場合は正常の画面が表示されます。

ライセンス

CheckFilter は CPU ライセンスとなっております。運用されるサーバの CPU 1 個に対し、1 つのライセンスが必要になります。

製品名	価格	型番
GkitTaglib InputFieldV2	200,000 円(税込み210,000 円)	A2091393-71
1 ライセンス	160,000 円(税込み168,000 円)	A2091393-72
5 ライセンス	770,000 円(税込み808,500 円)	A2091393-73
20 ライセンス	3,000,000 円(税込み 3,150,000 円)	A2091393-74

※開発ライセンスはフリーです。 ※CheckFilter は InputFieldV2 に含まれています。

サポート

製品の問い合わせは基本的に無償となっております。サポートのやり取りはすべて電子メールまたは FAX にて対応させていただいております。ご質問内容や状況によっては早急に回答できないことがあります。予めご了承ください。

電子メール : gkit@ml.ffc.co.jp

FAX 番号 : 03-5324-1667 (GKitTaglib CheckFilter と明記してください)

お問合せの際は以下の内容にてお問合せいただきますようお願いいたします。

- ・タイトル : お問い合わせの内容が判断できるような簡潔なものとしてください。
- ・連絡先 : お客様の会社名、氏名、メールアドレス、電話番号、FAX 番号
- ・製品名 : お問い合わせの製品名を記載してください。
- ・開発環境 : OS、AS、JDK の製品名およびバージョン
- ・運用環境 : OS、AS、JDK の製品名およびバージョン
- ・内容 : お問い合わせの内容を詳細に記載してください。

障害と思われる現象についてはどのような操作をしたときに発生するのか、また発生頻度（常に発生するのか、特定の条件でのみ発生するのかなど）も必ず記載してください。

可能であれば、スタンドアロンでの現象が確認できる再現プログラムをお送りください。再現確認の時間が短縮されることによって、より早い解決に繋がります。

制約事項

- ・複数の Input に同じ name 属性が設定されている場合、正常にチェックを行うことができません。
- ・エラー情報は入力が正常な場合は送られません。
- ・ CheckFilter は Servlet の Filter 技術により作られています。他の Filter と併用して使用する場合は web.xml に設定してご利用ください。

アトリビュート一覧

CheckFilter のアトリビュート一覧です。

config タグ

設定内容を囲むタグです。アトリビュートはありません。

post タグ

チェック対象の JSP/Servlet の設定項目です。

No	アトリビュート名	省略	意味	省略時の動作
1	url	不可	チェック対象の JSP/Servlet を設定	省略不可
2	error	可	エラーが発生した際にリクエストを送る JSP/Servlet を設定	url で設定した JSP/Servlet に送る

parameter タグ

個々のパラメータの設定項目です。

No	アトリビュート名	省略	意味	省略時の動作
1	name	不可	リクエストの name を設定	省略不可

input タグ

個々のリクエストの設定項目です。アトリビュートはありません。

permit タグ

具体的なリクエストの制限項目を設定します。

No	アトリビュート名	省略	意味	設定値
1	type	可	入力を許可するタイプを設定。設定されたタイプ以外の入力があった場合はエラーとする。複数選択可能。	ALLCHARS：文字列を制限しない
				FULLWIDTH：全角すべて
				FULLWIDTH-HIRAGANA：ひらがな
				FULLWIDTH-KATAKANA：全角カタカナ
				FULLWIDTH-ALPHABET：全角アルファベット
				FULLWIDTH-ALPHABET-CAPITAL：全角アルファベット大文字
				FULLWIDTH-ALPHABET-SMALL：全角アルファベット小文字
				FULLWIDTH-NUMBER：全角数字
				FULLWIDTH-KANJI：全角漢字
				FULLWIDTH-OTHER：全角その他
				HALFWIDTH：半角すべて
				HALFWIDTH-ALPHABET：半角アルファベット
				HALFWIDTH-ALPHABET-CAPITAL：半角アルファベット大文字
				HALFWIDTH-ALPHABET-SMALL：半角アルファベット小文字
				HALFWIDTH-NUMBER：半角数字
				HALFWIDTH-KATAKANA：半角カタカナ
HALFWIDTH-OTHER：半角その他				
MAILADDRESS：メールアドレス				
TIME-FORMAT：時間型フォーマット				
FORMAT：フォーマット（正規表現）				
2	format	可	フォーマット設定	例) 時間型フォーマット YY/MM/DD

			設定されたフォーマット以外の入力があった場合はエラーとする。	hh:mm:ss フォーマットは正規表現で指定する。
3	required	可	リクエストの入力が必須かどうかの設定。YES の場合はリクエストが存在しない場合はエラーとする。	NO (必須でない。デフォルト) YES (必須。)
4	maxlength	可	入力された文字の最大文字数を設定。文字数が設定を超えた場合エラーとする。	0 ~ 2147483647
5	minlength	可	入力された文字の最小文字数を設定。文字数が設定を下回った場合エラーとする。	0 ~ 2147483647
6	maxvalue		入力された数値の最大値を設定。設定した数値を超えた場合エラーとする。	-2147483648 ~ 2147483647
7	minvalue		入力された数値の最小値を設定。設定した数値を下回った場合エラーとする。	-2147483648 ~ 2147483647
8	disable-type		拒否するタイプを設定。	type と同じ
9	disable-chars		拒否する文字列を設定。設定した文字列が含まれていた場合はエラーとする。	文字列
10	enable-chars		許可する文字列を設定。設定した文	文字列

			字列が含まれていない場合はエラーとはならない。	
--	--	--	-------------------------	--

output タグ

Filter から JSP/Servlet にリクエストする前に変換項目を変換する設定を記述します。アトリビュートはありません。

substitution タグ

変換する項目を設定します。

No	アトリビュート名	省略	意味	設定値
1	target	可	変換対象となる文字列を設定する。	文字列
2	replacement	可	変換する文字列を設定する。	文字列

trim タグ

空白を削除する設定をします。

No	アトリビュート名	省略	意味	設定値
1	type	可	空白の削除の設定	left,center,right のいずれかひとつ。



Strategic Solutions for your Success