

## TERASOLUNA 2.0.5.1 移行ガイド(Web 版)

## 変更履歴

バージョン	日付	改訂箇所	改訂内容
2.0.5.1	2014/3/31	-	新規作成

## ■ 概要

本ドキュメントは、TERASOLUNA Server Framework for Java (Web 版) 2.0.5.0 で作成したアプリケーションを、2.0.5.1 へ移行する際の手順を示すドキュメントである。

### ◆ 2.0.5.1 の変更点概要

- 依存ライブラリのバージョンアップ
- build.xml のクラスパス設定変更
- commons-fileupload の脆弱性回避用のフィルタ設定追加
- リスナの登録順序の修正
- 「WB-05 コードリスト機能」国際化対応

## ■ 移行手順

2.0.5.0 から 2.0.5.1 への移行手順を説明する。

### 前提条件

TERASOLUNA Server Framework for Java (Web 版) 2.0.5.0 を使用したアプリケーションが正常に動作していること。

以後、本書ではこれをアプリケーションと呼称する。

### ① blank プロジェクトのダウンロード

TERASOLUNA Server Framework for Java (Web 版) 2.0.5.1 の blank プロジェクト (terasoluna-server4jweb-blank\_2.0.5.1.zip) を以下の URL よりダウンロードし、任意のフォルダに展開する。

ダウンロードサイトの URL :

➤ SourceForge.jp <http://sourceforge.jp/projects/terasoluna/releases/>

## ② 依存ライブラリの変更

TERASOLUNA フレームワークや Spring の依存ライブラリの差し替えを行う。  
差し替えるライブラリは以下の通りである。

2.0.5.0のライブラリ	2.0.5.1のライブラリ	種別
aspectjweaver-1.6.12.jar	aspectjweaver-1.7.4.jar	差し替え
commons-fileupload-1.2.jar	commons-fileupload-1.3.1.jar	差し替え
terasoluna-commons-2.0.5.0.jar	terasoluna-commons-2.0.5.1.jar	差し替え
terasoluna-dao-2.0.5.0	terasoluna-dao-2.0.5.1.jar	差し替え
terasoluna-ibatis-2.0.5.0	terasoluna-ibatis-2.0.5.1.jar	差し替え
terasoluna-thin-2.0.5.0	terasoluna-thin-2.0.5.1.jar	差し替え
-	terasoluna-ContentTypeLengthLimitFilter-1.0.0.jar	追加

「種別」に従って下記の修正を行う。

※「2.0.5.1 のライブラリ」は terasoluna-server4jweb-blank\_2.0.5.1.zip を展開したフォルダの lib フォルダ直下に格納されている。

### 差し替え：

アプリケーションの lib フォルダ直下から「2.0.5.0 のライブラリ」に記載した jar ファイルを削除し、「2.0.5.1 のライブラリ」に記載した jar ファイルを追加する。

### 追加：

「2.0.5.1 のライブラリ」に記載した jar ファイルをアプリケーションの lib フォルダ直下へ追加する。

同様に、Javadoc やソースコードが格納されている jar ファイルを差し替える。  
これらの jar ファイルは terasoluna-server4jweb-blank\_2.0.5.1.zip を展開したフォルダの terasoluna フォルダ配下に格納されている。

2.0.5.0のJavaDoc格納jarファイル	2.0.5.1のJavaDoc格納jarファイル	種別
terasoluna-commons-2.0.5.0-javadoc.jar	terasoluna-commons-2.0.5.1-javadoc.jar	差し替え
terasoluna-dao-2.0.5.0-javadoc.jar	terasoluna-dao-2.0.5.1-javadoc.jar	差し替え
terasoluna-ibatis-2.0.5.0-javadoc.jar	terasoluna-ibatis-2.0.5.1-javadoc.jar	差し替え
terasoluna-thin-2.0.5.0-javadoc.jar	terasoluna-thin-2.0.5.1-javadoc.jar	差し替え
-	terasoluna-ContentTypeLengthLimitFilter-1.0.0-javadoc.jar	追加

2.0.5.0のソースコード格納jarファイル	2.0.5.1のソースコード格納jarファイル	種別
terasoluna-commons-2.0.5.0-sources.jar	terasoluna-commons-2.0.5.1-sources.jar	差し替え
terasoluna-dao-2.0.5.0-sources.jar	terasoluna-dao-2.0.5.1-sources.jar	差し替え
terasoluna-ibatis-2.0.5.0-sources.jar	terasoluna-ibatis-2.0.5.1-sources.jar	差し替え
terasoluna-thin-2.0.5.0-sources.jar	terasoluna-thin-2.0.5.1-sources.jar	差し替え
-	terasoluna-ContentTypeLengthLimitFilter-1.0.0-sources.jar	追加

③ build.xml のクラスパス設定変更

②依存ライブラリの変更に従って、build.xml のクラスパスの設定を修正する。  
アプリケーション直下の/ant/build.xml を②依存ライブラリの変更で使用した依存ライブラリ表を参照し、「種別」に従って下記の修正を行う。

## 差し替え：

「2.0.5.0 のライブラリ」に記載した jar ファイルから「2.0.5.1 のライブラリ」に記載した jar ファイルにクラスパス設定の jar ファイル名を変更する。

## 追加：

「2.0.5.1 のライブラリ」に記載した jar ファイルのクラスパス設定を追加する。

## ●修正例●

## 【2.0.5.0 の build.xml】

```
<!-- クラスパスの設定 -->
<property name="classpath.lib" value="
    * * * (中略) * * *
    ${lib.dir}/commons-fileupload-1.2.jar;
    * * * (中略) * * *
    ${lib.dir}/terasoluna-dao-2.0.5.0.jar;
    ${lib.dir}/terasoluna-ibatis-2.0.5.0.jar;
    * * * (中略) * * *
"/>
```

## 【2.0.5.1 の build.xml】

```
<!-- クラスパスの設定 -->
<property name="classpath.lib" value="
    * * * (中略) * * *
    ${lib.dir}/commons-fileupload-1.3.1.jar; ←クラスパス差し替えの例
    ${lib.dir}/terasoluna-ContentLengthLimitFilter-1.0.0.jar; ←クラスパスの追加の例
    * * * (中略) * * *
    ${lib.dir}/terasoluna-dao-2.0.5.1.jar;
    ${lib.dir}/terasoluna-ibatis-2.0.5.1.jar;
    * * * (中略) * * *
"/>
```

#### ④ commons-fileupload の脆弱性回避用のフィルタ設定追加

以下のように、commons-fileupload の脆弱性回避用のフィルタを web.xml に追加設定する。フィルタの中でも、最初に実行されるフィルタとして設定すること。

【web.xml】

```
*** (中略) ***
<!-- Content-Type サイズ制限(commons-fileupload 脆弱性回避用) -->
<filter>
  <filter-name>contentTypeLengthLimitFilter</filter-name>
  <filter-class>jp.terasoluna.fw.web.common.filter.ContentTypeLengthLimitFilter</filter-class>
  <!--
  <init-param>
    <param-name>limitBytes</param-name>
    <param-value>2048</param-value>
  </init-param>
  -->
</filter>
*** (中略) ***
<filter-mapping>
  <filter-name>contentTypeLengthLimitFilter</filter-name>
  <url-pattern>*.do</url-pattern>
</filter-mapping>
*** (中略) ***
```

### ⑤ リスナの登録順序の修正

リスナ(および関連するパラメータ設定)の登録(記述)順序を以下のように修正する。

(修正前) 「ContextLoaderListener → Log4jConfigListener」の順序

(修正後) 「Log4jConfigListener → ContextLoaderListener」の順序

#### 【web.xml】

※以下は、blank プロジェクトのデフォルト設定を例にした修正後の記述である。  
context-param の内容については、プロジェクトの要件によって下記と異なる値に変更している可能性がある。その際は、変更している設定値はそのまま、Log4jConfigListener と ContextLoaderListener の順序を入れ替えればよい。

\*\*\* (中略) \*\*\*

```
<listener>
  <listener-class>org.springframework.web.util.Log4jConfigListener</listener-class>
</listener>
<context-param>
  <param-name>log4jConfigLocation</param-name>
  <param-value>classpath:log4j.properties</param-value>
</context-param>
<context-param>
  <param-name>log4jExposeWebAppRoot</param-name>
  <param-value>false</param-value>
</context-param>
```

\*\* (中略) \*\*

```
<listener>
  <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>
    /WEB-INF/applicationContext.xml
  </param-value>
</context-param>
```

\*\*\* (中略) \*\*\*

## ⑥ 「WB-05 コードリスト機能」国際化対応に伴う変更

本設定の変更は「WB-05 コードリスト機能」にて、拡張ポイントを利用して、CodeListLoader インタフェースの実装クラスを作成していた場合のみ対応が必要となる。

「WB-05 コードリスト機能」の国際化対応に伴い、CodeListLoader インタフェースに追加された「getCodeBeans(Locale locale)」メソッドを実装する。

国際化対応が不要な場合は、引数なしの getCodeBean() を呼び出し、その結果を返却する処理を実装する。

### 【CodeListLoader 実装クラス】

```
*** (中略) ***  
    public CodeBean[] getCodeBeans(Locale locale) {  
        return getCodeBeans();  
    }  
*** (中略) ***
```

## ⑦ 動作確認

アプリケーションを起動させて、問題なく動作することを確認する。