



3G-Web サンプルプログラム インストールガイド

VeriTrans 3G

VeriTrans 3G-Web Sample Program for Java Installation Guide  
ver.1.4.1 (2020年07月～)

---

## 改訂履歴

版	改訂日	内容
1.0.0	2011/09	新規作成
1.1.0	2012/02	改訂
1.2.0	2012/09	社名変更
1.2.1	2013/03	「著作権に関するお問い合わせ先」の記載を削除 ベリトランス株式会社 テクニカルサポートのメールアドレスを変更
1.3.0	2013/08	ショッピングクレジット決済追加 2.6. サンプルプログラム・ライブラリ組込の表に WEB-INF/lib/commons-lang-2.6.jar を追加 2.7. サンプルプログラム設定の表に項目を追加 (DUMMY_PAYMENT_FLAG、FINISH_PAYMENT_ACCESS_URL、UNFINISH_PAYMENT_RETURN_URL、 ERROR_PAYMENT_RETURN_URL、LANG_ENABLE_FLAG、LANG)
1.3.1	2016/04	SHA-2/TLS1.1 以上対応 2.10. デプロイに説明追加
1.3.2	2016/05	2.9 ビルド に説明を追加
1.4.0	2017/03	銀行振込決済追加 2.6. サンプルプログラム・ライブラリ組込の表に項目を追加、変更 (追加 :jsp/common/finish_return.jsp、jsp/common/include_retrieve_info.jsp、css/finish_return_style.css 修正 :WEB-INF/src/mdk.bean.EncryptKeyBox.java→WEB-INF/src/mdk.bean.PostDataBox.java) 2.7. サンプルプログラム設定の表から項目を削除、修正
1.4.1	2020/07	開発ガイドの修正に合わせて文言を統一 TLS1.1 廃止に伴う修正および、古い Java7 向けの説明等を削除

## 目次

1. 導入の前に.....	4
1.1. 本ガイドの内容.....	4
1.2. 著作権、および問い合わせ先.....	4
1.2.1. 著作権.....	4
1.2.2. お問い合わせ先.....	4
2. 導入.....	5
2.1. 作業内容.....	5
2.2. Java 実行環境の確認・準備.....	5
2.3. ant の確認・準備.....	5
2.4. Web アプリケーションサーバー環境の確認・準備.....	6
2.5. サンプルプログラム展開.....	6
2.6. サンプルプログラム・ライブラリ組込.....	6
2.7. サンプルプログラム設定.....	9
2.8. ログ設定.....	10
2.9. ビルド.....	10
2.10. デプロイ.....	12
2.11. 動作確認.....	12

# 1. 導入の前に

## 1.1. 本ガイドの内容

本ガイドは、店舗様の EC サイトよりベリトランス(株)が提供する 3G-Web(以下、Web リンクサーバー)へ接続し、利用する際に参考となるサンプルプログラムのファイル構成、設定方法等について記載しています。

## 1.2. 著作権、および問い合わせ先

### 1.2.1. 著作権

本ドキュメントの著作権はベリトランス株式会社が保有しています。

Copyright © 2020 VeriTrans Inc., a Digital Garage company. All rights reserved.

### 1.2.2. お問い合わせ先

ベリトランス株式会社 テクニカルサポート

技術面に関するお問い合わせ先: [tech-support@veritrans.jp](mailto:tech-support@veritrans.jp)

## 2. 導入

### 2.1. 作業内容

Java 版サンプルプログラムの導入にあたり、以下の作業が必要となります。

- (1) java 実行環境の確認・準備
- (2) ant の確認・準備
- (3) Web アプリケーションサーバー環境の確認・準備
- (4) サンプルプログラム展開
- (5) サンプルプログラム・ライブラリ組込
- (6) サンプルプログラム設定
- (7) ログ設定
- (8) ビルド(build)
- (9) デプロイ(deploy)
- (10) 動作確認

### 2.2. Java 実行環境の確認・準備

サンプルプログラム実行環境に Java Virtual Machine(JVM)が導入されており、またバージョンが 1.8 以降であることを確認して下さい。

### 2.3. ant の確認・準備

本書では、サンプルプログラムのビルドに ant を利用することを想定しています。

ant パッケージを導入する場合は、<http://ant.apache.org/> より最新のパッケージを取得して下さい。

導入手順は <http://ant.apache.org/manual/index.html> をご覧下さい。

## 2.4. Web アプリケーションサーバー環境の確認・準備

サンプルプログラムは Web アプリケーションサーバー上で実行されるプログラムです。サンプルプログラム実行環境上で Web アプリケーションサーバーが稼働している事、またそのサーバー上で Java アプリケーションが稼働することを確認して下さい。

## 2.5. サンプルプログラム展開

サンプルプログラムは圧縮された状態で提供されますので、解凍して展開します。

## 2.6. サンプルプログラム・ライブラリ組込

解凍後、サンプルプログラム稼働に必要なファイル及びライブラリが揃っている事を確認します。

ディレクトリ/ファイル名		説明	
conf	log4j.properties	サンプルプログラム用の log4j 設定が記載されています。	
	MessageResources.properties	エラーメッセージが定義されています。	
	VTWEBMDK.properties	マーチャントID 等が定義されています。	
dist		ビルド時に使用する work 領域です。	
WEB-INF	lib	commons-beanutils.jar	サンプルプログラムが使用している Struts および log4J のライブラリです。
		commons-digester.jar	
		commons-discovery-0.2.jar	
		commons-lang-2.6.jar	
		commons-logging-1.0.4.jar	
		commons-validator.jar	
		log4j-1.2.8.jar	
		struts.jar	
	src	設定ファイル、およびサンプルソースが置かれているディレクトリです。 ※ソースファイルは jp.co.veritrans.vtweb.mdk と jp.co.veritrans.vtweb.sample に収めているため、ディレクトリ構造を省きます。	
		mdk.bean.CommodityDetail.java	商品情報を保持する Bean です。
mdk.bean.EncryptionKey.java		暗号鍵を保持する Bean です。	
mdk.bean.PostDataBox.java		Web リンクサーバーからの受信情報を保持するクラスです。	
mdk.bean.ResultDataBean.java		決済結果を保持する Bean です。	
mdk.common.Constants.java		デフォルト値を保持するクラスです。	
mdk.common.DateUtil.java		日付系 Utility です。	
mdk.common.HashCodeCreator4Merchant.java		ハッシュコード作成クラスです。	
mdk.common.MDKUtil.java		Utility クラスです。	
mdk.common.ResultCheck.java	決済結果のチェックを行うクラスです。		

mdk.common.StringUtil.java	文字列系 Utility です。
mdk.conf.MerchantConf.java	VTWEBMDK.properties の値を保持するクラスです。
mdk.filter.SetCharacterEncodingFilter.java	文字コードフィルタリングを行うクラスです。
sample.server.DoPostActionVTW.java	Web リンクサーバーからの決済結果を受け取るサーバークラスです。
sample.server.action.CommodityForm.java	決済方法を指定しない場合の ActionForm クラスです。
sample.server.action.ConfirmAction.java	決済方法を指定しない場合の Action クラスです。
sample.server.action.DoPostActionBrowser.java	ブラウザからの決済結果を受け取り、比較する処理の Action クラスです。
sample.server.action.DoPostForm.java	ブラウザからの決済結果を受け取り、比較する処理の ActionForm クラスです。
sample.server.action.MerchantSampleAction.java	全 Action クラスの基底クラスです。
sample.server.action.bank.BankAction.java	銀行決済の Action クラスです。
sample.server.action.bank.BankForm.java	銀行決済の ActionForm クラスです。
sample.server.action.card.CardAction.java	カード決済の Action クラスです。
sample.server.action.card.CardForm.java	カード決済の ActionForm クラスです。
sample.server.action.cvs.CvsAction.java	コンビニ決済の Action クラスです。
sample.server.action.cvs.CvsForm.java	コンビニ決済の ActionForm クラスです。
sample.server.action.em.EmAction.java	電子マネー決済の Action クラスです。
sample.server.action.em.EmForm.java	電子マネー決済の ActionForm クラスです。
sample.server.action.oricosc.OricoscAction.java	ショッピングクレジット決済の Action クラスです。
sample.server.action.oricosc.OricoscForm.java	ショッピングクレジット決済の ActionForm クラスです。
sample.server.action.va.VaAction.java	銀行振込決済の Action クラスです。
sample.server.action.va.VaForm.java	銀行振込決済の ActionForm クラスです。
struts-bean.tld	struts 設定ファイルです。
struts-config.xml	
struts-html.tld	
struts-logic.tld	

	struts-nested.tld		
	struts-tiles.tld		
	tiles-defs.xml		
	web.xml	web 設定ファイルです。	
build.xml		ビルド用設定ファイルです。	
jsp	common	共通で使用する JSP ファイルを格納します。	
		finish_return.jsp	決済処理結果比較画面です。
		include_cart_table.jsp	画面の構成要素(商品)です。
		include_input_hidden.jsp	画面の構成要素(hidden)です。
		include_retrieve_info.jsp	画面の構成要素(session)です。
		include_paynowid.jsp	画面の構成要素(paynowid)です。
		include_retrieve_info.jsp	画面の構成要素(商品情報取得)です。
	iframe	インラインフレームサンプルで使用する JSP ファイルを格納します。	
		entry.jsp	決済方法を指定しない場合の画面です。
		entry_bank.jsp	銀行決済を選択した場合の画面です。
		entry_card.jsp	カード決済を選択した場合の画面です。
		entry_cvs.jsp	コンビニ決済を選択した場合の画面です。
		entry_em.jsp	電子マネー決済を選択した場合の画面です。
		entry_oricosc.jsp	ショッピングクレジット決済を選択した場合の画面です。
		entry_va.jsp	銀行振込決済を選択した場合の画面です。
		include_head.jsp	画面の構成要素(head)です。
		jump.jsp	Web リンクサーバーが提供する決済画面へ遷移する画面です。
		purchase.jsp	決済方法を選択する画面です。
	self	自画面遷移サンプルで使用する JSP ファイルを格納します。	
		entry.jsp	決済方法を指定しない場合の画面です。
entry_bank.jsp		銀行決済を選択した場合の画面です。	
entry_card.jsp		カード決済を選択した場合の画面です。	
entry_cvs.jsp		コンビニ決済を選択した場合の画面で	

			す。
		entry_em.jsp	電子マネー決済を選択した場合の画面です。
		entry_oricosc.jsp	ショッピングクレジット決済を選択した場合の画面です。
		entry_va.jsp	銀行振込決済を選択した場合の画面です。
		include_head.jsp	画面の構成要素(head)です。
		jump.jsp	Web リンクサーバーが提供する決済画面へ遷移する画面です。
		purchase.jsp	決済方法を選択する画面です。
css	finish_return_style.css		決済処理結果比較画面のスタイルシートです。
	style.css		PC ブラウザ用のスタイルシートです。
	style_m.css		モバイル画面用のスタイルシートです。
	iframe.css		インラインフレームサンプル用のスタイルシートです。
WEB-IMG	header.gif		インラインフレームサンプル用のヘッダー画像です。
	footer.gif		インラインフレームサンプル用のフッター画像です。
index.html			各画面へのリンク画面です。

## 2.7. サンプルプログラム設定

設定ファイルの値を環境に合わせて変更してください。

パラメータ	値
MERCHANT_ID	マーチャント ID ペリトランス指定の値を設定してください。
SEED	Web リンクサーバーへ送信するデータの検証用ハッシュシード ペリトランス指定の値を設定してください。
DUMMY_PAYMENT_FLAG	ダミー取引フラグ 0:本番取引、1:ダミー取引
SETTLEMENT_TYPE	決済方法 00:決済方法指定なし、01:カード決済、 02:コンビニ決済、03:電子マネー決済、 04:銀行決済、05:ショッピングクレジット決済、 06:銀行振込決済

VTWEB_REGIST_URL	Web リンクサーバーの暗号鍵取得 URL 弊社から依頼がない場合は変更しないでください。
VTWEB_SETTLEMENT_URL	Web リンクサーバーの決済 URL 弊社から依頼がない場合は変更しないでください。
FINISH_PAYMENT_ACCESS_URL	決済結果通知先 URL EC サイトの決済完了報告時の URL に変更してください。
FINISH_PAYMENT_RETURN_URL	決済完了後戻り URL お客様の決済戻り URL に変更してください。
UNFINISH_PAYMENT_RETURN_URL	未決済時戻り URL EC サイトの未決済時の戻り URL に変更してください。
ERROR_PAYMENT_RETURN_URL	決済エラー時戻り URL EC サイトの決済エラー時の戻り URL に変更してください。
LANG_ENABLE_FLAG	言語選択可否フラグ
LANG	使用言語
CARD_CAPTURE_FLAG	売り上げフラグ 0: 与信のみ、1: 与信・売上。指定が無い場合は、0
SHOP_NAME	モバイル Edy 用ショップ名
SCREEN_TITLE	Suica 用商品名
CONTENTS	請求内容
CONTENTS_KANA	請求内容カナ
DUMMY_COMMODITY_ID	商品情報の商品 ID 未入力時に設定するダミー値
DUMMY_COMMODITY_JANCODE	商品情報の JAN.CODE 未入力時に設定するダミー値

## 2.8. ログ設定

必要に応じてログファイル及びログレベルの設定を行って下さい、

設定内容に関しては <http://logging.apache.org/log4j/> を参照して下さい。

## 2.9. ビルド

(1) サンプルプログラム解凍先に移動します。

```
$cd /tmp/vtweb/mdk <<お客様の展開先に読み替えて下さい。
```

(2) build.xml の内容を修正してください。

```
6 行目:<property name="webapp.home" value="Web アプリケーションサーバー" />
```

例(UNIX 系で tomcat をご利用の場合): /usr/local/tomcat

例(Windows で tomcat をご利用の場合): C:/apache-tomcat

Web アプリケーションサーバーのディレクトリ構成に合わせて、次の箇所も修正してください。

18 行目: <fileset dir="\${webapp.home}/common/lib" includes="\*.jar" />

例(Tomcat6 以降をご利用の場合): <fileset dir="\${webapp.home}/lib" includes="\*.jar" />

(3) ビルドします。

```
$ant -f build.xml
```

以下の警告メッセージが出力されますが、動作には支障ございませんのでご了承ください。

[javac] 注:入力ファイルの操作のうち、未チェックまたは安全ではないものがあります。

[javac] 注:詳細については、`-Xlint:unchecked` オプションを指定して再コンパイルしてください。

(4) サンプルプログラム解凍先に War ファイル(VTWebJava.war)が作成されたことを確認します。

## 2.10. デプロイ

作成された war ファイルを Web アプリケーションサーバーにデプロイします。

例(UNIX 系で tomcat をご利用の場合): `cp VTWebJava.war $TOMCAT_HOME/webapps`

例(Windows で tomcat をご利用の場合): `cp VTWebJava.war %TOMCAT_HOME%/webapps`

## 2.11. 動作確認

Web ブラウザで以下 URL にアクセスし、稼働することを確認して下さい。

`http://(導入サーバーベース URL)/ VTWebJava /index.html`