

# SMBC マルチペイメントサービス

モジュール・タイプ

(Java 版 インストールガイド)

2020年4月14日 1.08版

- ☑ 本書の著作権は「SMBC GMO PAYMENT 株式会社(以下、弊社という)」に帰属します。
- ☑ 本書のいかなる部分においても、弊社に事前の書面による許可なく、電子的、機械的を含むいかなる手段や形式によってもその複製、改変、頒布、ならびにそれらに類似する行為を禁止します。
- ☑ 本仕様書は加盟店様と弊社との間で締結した機密保持契約において機密情報として規定される情報です。本仕様書の取扱いは機密保持契約の規定に従ってください。
- ☑ 本書の内容は、予告なく随時更新されます。
- ☑ 本書の内容について万全を期しておりますが、万一記載もれ等お気づきの点がございましたら、当社までご連絡下さい。

# 目次

1. 動作環境について.....	1
1.1. 必要条件 .....	1
2. インストール手順.....	2
2.1. Java クラスライブラリの配置.....	2
2.2. 設定ファイルおよびテンプレートファイルの配置 .....	2
3. 設定ファイルについて.....	3
3.1. 接続情報定義(connector.properties).....	3
3.2. ロギング定義(logging.properties) .....	4
3.3. 本人認証用リダイレクトページのテンプレート .....	5
3.4. 支払い手続き開始 IF 呼出し用リダイレクトページのテンプレート .....	6
3.4.1. PayPal 決済 .....	6
3.4.2. WebMoney 決済.....	7
3.5. 接続コネクション定義(gmoconnection.properties).....	8
4. 呼出しサンプル.....	9

## 変更履歴

---

### 2008年9月1日 - 1.00版

---

- ・ 新規作成

### 2009年9月15日 - 1.01版

---

- ・ 対象動作条件に Java1.6.0 を追加

### 2010年2月22日 - 1.02版

---

- ・ PayPal 決済の支払い手続き開始 IF 呼出し用リダイレクトページのテンプレートの説明を追加
- ・ 呼出しサンプルの一覧に PayPal 決済を追加

### 2010年6月16日 - 1.03版

---

- ・ WebMoney 決済の支払い手続き開始 IF 呼出し用リダイレクトページのテンプレートの説明を追加
- ・ 呼出しサンプルの一覧に WebMoney 決済を追加

### 2012年11月20日 - 1.04版

---

- ・ Mobile Edy から楽天 Edy に表記を変更

### 2015年11月24日 - 1.05版

---

- ・ 1.1 必要条件 「JRE/JDK 1.7.0 以上」を追加

### 2017年5月8日 - 1.06版

---

- ・ 「2.2.設定ファイルおよびテンプレートファイルの配置」に gmoconnection.properties を追加
- ・ 「3.5.接続コネクション定義(gmoconnection.properties)」を追加

### 2018年3月12日 - 1.07版

---

- ・ 「1.1 必要条件」の Java 環境を一部変更

### 2020年4月14日 - 1.08版

---

- ・ 「1.1 必要条件」の Java 環境に java のバージョンについて追加

## 1. 動作環境について

### 1.1. 必要条件

モジュールタイプ Java 版を動作させるために必要な環境は以下の通りです。

- ・ Java 環境                    JRE/JDK 1.7.0 以上  
                                  JRE/JDK 1.8.0 以上  
                                  java7  
                                  java8
- ・ OS                            Java クラスライブラリとして提供されるため、Java 動作環境対応 OS 上で稼働し、特定の OS に依存しません。

※JRE/JDK 1.7 系をご利用頂く場合は、モジュールの起動オプションに以下を追加する必要があります。

-Dhttps.protocols=TLSv1.2

## 2. インストール手順

### 2.1. Java クラスライブラリの配置

以下のファイルをクラスパスの通ったディレクトリに配置してください。

- ・ gpay\_client-3.0.jar 決済モジュール

【 配置ディレクトリ例 】

/usr/local/tomcat/webapps/payment/WEB-INF/lib

### 2.2. 設定ファイルおよびテンプレートファイルの配置

以下のファイルをクラスパスの通ったディレクトリに配置してください。

(“conf/xxxxxxx.properties”としてパスを探すため、“~/conf”配下に配置してください。)

- ・ connector.properties 接続情報定義
- ・ logging.properties ロギング定義(※1)
- ・ RedirectPage.html 本人認証用リダイレクトページのテンプレート (クレジット)
- ・ DoPaypalStartRedirect.html 支払い手続き開始 IF 呼出し用リダイレクトページのテンプレート  
※PayPal 決済
- ・ DoWebmoneyStartRedirect.html 支払い手続き開始 IF 呼出し用リダイレクトページのテンプレート  
※WebMoney 決済
- ・ gmoconnection.properties 接続コネクション定義

【 配置ディレクトリ例 】

/usr/local/tomcat/webapps/payment/WEB-INF/classes/conf

※1 : Tomcat を利用の場合、\$CATALINA\_HOME/conf 配下の logging.properties を優先的に読み込む可能性がありますのでご注意ください。

### 3. 設定ファイルについて

#### 3.1. 接続情報定義(connector.properties)

決済用サーバへ接続するための情報を定義します。

```
# Connect url properties
com.gmo_pg.g_pay.client.tran.EntryTran=https://pt01.smbc-gp.co.jp/payment/EntryTran.idPass
com.gmo_pg.g_pay.client.tran.ExecTran=https://pt01.smbc-gp.co.jp/payment/ExecTran.idPass
com.gmo_pg.g_pay.client.tran.SecureTran=https://pt01.smbc-gp.co.jp/payment/SecureTran.idPass
com.gmo_pg.g_pay.client.tran.AlterTran=https://pt01.smbc-gp.co.jp/payment/AlterTran.idPass
com.gmo_pg.g_pay.client.tran.ChangeTran=https://pt01.smbc-gp.co.jp/payment/ChangeTran.idPass

com.gmo_pg.g_pay.client.tran.SearchTrade=https://pt01.smbc-gp.co.jp/payment/SearchTrade.idPass
com.gmo_pg.g_pay.client.tran.TradedCard=https://pt01.smbc-gp.co.jp/payment/TradedCard.idPass

com.gmo_pg.g_pay.client.tran.SaveMember=https://pt01.smbc-gp.co.jp/payment/SaveMember.idPass
com.gmo_pg.g_pay.client.tran.UpdateMember=https://pt01.smbc-gp.co.jp/payment/UpdateMember.idPass
com.gmo_pg.g_pay.client.tran.SearchMember=https://pt01.smbc-gp.co.jp/payment/SearchMember.idPass
com.gmo_pg.g_pay.client.tran.DeleteMember=https://pt01.smbc-gp.co.jp/payment/DeleteMember.idPass
com.gmo_pg.g_pay.client.tran.SaveCard=https://pt01.smbc-gp.co.jp/payment/SaveCard.idPass
com.gmo_pg.g_pay.client.tran.SearchCard=https://pt01.smbc-gp.co.jp/payment/SearchCard.idPass
com.gmo_pg.g_pay.client.tran.DeleteCard=https://pt01.smbc-gp.co.jp/payment/DeleteCard.idPass

com.gmo_pg.g_pay.client.tran.EntryTranSuica=https://pt01.smbc-gp.co.jp/payment/EntryTranSuica.idPass
com.gmo_pg.g_pay.client.tran.ExecTranSuica=https://pt01.smbc-gp.co.jp/payment/ExecTranSuica.idPass
com.gmo_pg.g_pay.client.tran.EntryTranEdy=https://pt01.smbc-gp.co.jp/payment/EntryTranEdy.idPass
com.gmo_pg.g_pay.client.tran.ExecTranEdy=https://pt01.smbc-gp.co.jp/payment/ExecTranEdy.idPass
com.gmo_pg.g_pay.client.tran.EntryTranCvs=https://pt01.smbc-gp.co.jp/payment/EntryTranCvs.idPass
com.gmo_pg.g_pay.client.tran.ExecTranCvs=https://pt01.smbc-gp.co.jp/payment/ExecTranCvs.idPass
com.gmo_pg.g_pay.client.tran.EntryTranPayeasy=https://pt01.smbc-gp.co.jp/payment/EntryTranPayEasy.idPass
com.gmo_pg.g_pay.client.tran.ExecTranPayeasy=https://pt01.smbc-gp.co.jp/payment/ExecTranPayEasy.idPass

com.gmo_pg.g_pay.client.tran.SearchTradeMulti=https://pt01.smbc-gp.co.jp/payment/SearchTradeMulti.idPass
```

※：下線部分を環境に応じて変更してください。

## 3.2. ロギング定義(logging.properties)

ログ出力のための情報を定義します。

```
#####
# Logging Configuration File
#
~~~~ 中略 ~~~~
handlers= java.util.logging.FileHandler
~~~~ 中略 ~~~~
.level= INFO
~~~~ 中略 ~~~~
java.util.logging.FileHandler.pattern = ./gpay_client.log
java.util.logging.FileHandler.limit = 50000
java.util.logging.FileHandler.count = 1
java.util.logging.FileHandler.formatter = com.gmo_pg.g_pay.client.common.log.LogFormatter
~~~~ 中略 ~~~~
# WARNING --> for Log#warn()
# CONFIG --> for Log#debug()
# INFO --> for Log#info()
# SEVERE --> for Log#error()
# ALL --> for ALL
com.gmo_pg.client.level = INFO
```

※：下線部分を環境に応じて変更してください。

※：詳細につきましては、「java.util.logging」パッケージの解説を参照してください。

### 3.3. 本人認証用リダイレクトページのテンプレート

本人認証サービスの認証画面へ遷移するための HTML ページ(雛形)を定義します。

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=Windows-31J">
</head>
<body OnLoad='OnLoadEvent();'>
  <form name="ACSCall" action="{ACSUrl}" method="POST">
    <noscript>
      <br>
      <br>
      <center>
        <h2>
          3-D セキュア認証を続けます。<br>
          ボタンをクリックしてください。
        </h2>
        <input type="submit" value="OK">
      </center>
    </noscript>
    <input type="hidden" name="PaReq" value="{PaReq}">
    <input type="hidden" name="TermUrl" value="{TermUrl}">
    <input type="hidden" name="MD" value="{MD}">
  </form>
</script>
<!--
  function OnLoadEvent() {
    document.ACSCall.submit();
  }
  //-->
</script>
</body>
</html>

```

※：下線部分については、動作を損なわないように注意して変更してください。

※：インタフェース仕様書も合わせて参照してください。

### 3.4. 支払い手続き開始 IF 呼び出し用リダイレクトページのテンプレート

支払い手続き開始 IF へ遷移するための HTML ページ(雛形)を定義します。

#### 3.4.1. PayPal 決済

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=Windows-31J">
</head>
<body onLoad='OnLoadEvent();'>
  <form name="PaypalStart" action="{PaypalStartUrl}" method="POST">
    <noscript>
      <br>
      <br>
      <center>
        <h2>
          Paypal 支払開始を続けます。<br>
          ボタンをクリックしてください。
        </h2>
        <input type="submit" value="続行">
      </center>
    </noscript>
    <input type="hidden" name="ShopID" value="{ShopID}">
    <input type="hidden" name="AccessID" value="{AccessID}">
  </form>
  <script >
  <!--
    function OnLoadEvent() {
      document.PaypalStart.submit();
    }
  //-->
</script>
</body>
</html>

```

※：下線部分については、動作を損なわないように注意して変更してください。

※：インタフェース仕様書も合わせて参照してください。

### 3.4.2. WebMoney 決済

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=Windows-31J">
</head>
<body OnLoad='OnLoadEvent();'>
  <u>form name="WebmoneyStart" action="{WebmoneyStartUrl}" method="POST">
  <noscript>
    <br>
    <br>
    <center>
      <h2>
        WebMoney 決済画面に遷移します。<br>
        ボタンをクリックしてください。
      </h2>
      <input type="submit" value="続行">
    </center>
  </noscript>
  <input type="hidden" name="AccessID" value="{AccessID}">
</form>
<script >
<!--
  function OnLoadEvent() {
    document.WebmoneyStart.submit();
  }
  //-->
</script>
</body>
</html>

```

※：下線部分については、動作を損なわないように注意して変更してください。

※：インタフェース仕様書も合わせて参照してください。

### 3.5. 接続コネクション定義(gmoconnection.properties)

決済用サーバ接続時のコネクション設定を定義します。

```
com.gmo_pg.g_pay.client.BaseTran.connecttimeout = 0  
com.gmo_pg.g_pay.client.BaseTran.readtimeout = 0
```

※：下線部分を環境に応じミリ秒で設定してください。

※：該当の設定ファイルがない場合、及び下線部分の指定がない場合は「0」がセットされます。

※：connecttimeout を設定し、接続が確立される前にタイムアウトが過ぎた場合は

java.net.SocketTimeoutException が発生しますが、モジュールは

com.gmo\_pg.g\_pay.client.common. PaymentException をスローします。

※：readtimeout を設定し、データが読み取り可能になる前にタイムアウトが過ぎた場合は

java.net.SocketTimeoutException が発生しますが、モジュールは ErrCode=E91、ErrInfo=E91060001

を返却します。

※：詳細につきましては、「java.net.URLConnection」パッケージの解説を参照してください。

## 4. 呼出しサンプル

モジュールタイプで提供しているクラスの利用方法を理解して頂くために、以下のサンプルプログラムをご用意しました。また、インタフェース仕様書と合わせて参照する事で、より各クラスの呼出し方法や結果の取得方法を理解しやすくなります。

表 4-1 サンプルプログラム一覧

No	サンプルプログラム名	説明
1	CallEntryTranServlet.java	取引登録
2	CallExecTranServlet.java	決済実行
3	CallAlterTranServlet.java	取引変更
4	CallChangeTranServlet.java	金額変更
5	CallSecureTranServlet.java	3D セキュア認証結果確認
6	CallCreateRedirectPageServlet.java	リダイレクトページ生成
7	CallEntryExecTranServlet.java	取引実行(Entry + Exec)
8	CallSaveMemberServlet.java	会員登録
9	CallUpdateMemberServlet.java	会員更新
10	CallSearchMemberServlet.java	会員照会
11	CallDeleteMemberServlet.java	会員削除
12	CallSaveCardServlet.java	カード登録/更新
13	CallSearchCardServlet.java	カード照会
14	CallDeleteCardServlet.java	カード削除
15	CallSearchTradeServlet.java	取引状態照会
16	CallTradedCardServlet.java	取引後カード登録
17	CallEntryTranSuicaServlet.java	モバイル Suica 取引登録
18	CallEntryTranEdyServlet.java	楽天 Edy 取引登録
19	CallEntryTranCvsServlet.java	コンビニ取引登録
20	CallEntryTranPayEasyServlet.java	Pay-easy 取引登録
21	CallEntryTranPaypalServlet.java	PayPal 取引登録
22	CallEntryTranWebmoneyServlet.java	WebMoney 取引登録
23	CallExecTranSuicaServlet.java	モバイル Suica 決済実行
24	CallExecTranEdyServlet.java	楽天 Edy 決済実行
25	CallExecTranCvsServlet.java	コンビニ決済実行
26	CallExecTranPayEasyServlet.java	Pay-easy 決済実行
27	CallExecTranPaypalServlet.java	PayPal 決済実行
28	CallExecTranWebmoneyServlet.java	WebMoney 決済実行
29	CallEntryExecTranSuicaServlet.java	モバイル Suica 取引実行(Entry + Exec)
30	CallEntryExecTranEdyServlet.java	楽天 Edy 取引実行(Entry + Exec)
31	CallEntryExecTranCvsServlet.java	コンビニ取引実行(Entry + Exec)
32	CallEntryExecTranPayEasyServlet.java	Pay-easy 取引実行(Entry + Exec)
33	CallEntryExecTranPaypalServlet.java	PayPal 取引実行(Entry + Exec)
34	CallEntryExecTranWebmoneyServlet.java	WebMoney 取引実行(Entry + Exec)
35	CallPaypalStartServlet.java	PayPal 支払い手続き開始 IF 呼出し

36	CallWebmoneyStartServlet.java	WebMoney 支払い手続き開始 IF 呼出し
37	CallCancelTranPaypalServlet.java	PayPal 決済取消
38	CallSearchTradeMultiServlet.java	マルチ決済取引状態照会