VeriTrans 4G

MDK トークン　開発ガイド

Ver. 1.0.9（2024 年 3 月～）

目次

第1章 MDKトークン導入の前に

1-1 本ガイドの内容

本ガイドは、店舗様の EC サイトが株式会社 DG フィナンシャルテクノロジーの提供する VeriTrans4G の MDK トークンを利用して決済を行うために必要な情報について記載しています。

1-2 対象者

本ガイドは、MDK トークンを利用する店舗様 EC サイトの開発者を対象としております。

MDK トークンを連携できる決済サービスについては、「2-4 対象サービス」を参照して下さい。

タイトル	概要	企画者	開発者	運用者
MDK トークン 開発ガイド	本ガイド MDK トークンの概要およびトークンを取得するための API が記載されています。		◎	
VeriTrans4G 開発ガイド	VeriTrans4G 導入にあたって、MDK（Merchant Development Kit）の導入方法について記載されています。		◎	
VeriTrans4G インターフェース詳細 ～クレジットカード決済～ ～MPI ホスティング～	決済のためのインターフェース詳細が記載されています。MDK トークンを連携する API はこちらをご覧ください。		◎	

表 1-2-1 ドキュメント一覧

1-3 制限事項

消費者様がご利用いただける環境（ブラウザ）は以下の条件を満たす必要があります。

- JavaScript が使用可能
- Cross-Origin Resource Sharing に対応している
- SHA-2 証明書 及び TLS1.2 以上での通信に対応している

※フィーチャー・フォンは Cross-Origin Resource Sharing に対応していないため、トークンのご利用はできません。

1-4 著作権、および問い合わせ先

[お問い合わせ先]
株式会社 DG フィナンシャルテクノロジー ベリトランス テクニカルサポート
電子メール： tech-support@veritrans.jp

1-5 改訂履歴

2017/2 ：Ver1.0.0 リリース

MDK トークン開発ガイド

2017/4 ：Ver1.0.1
「4-2 サンプルコードについて」の記載を追記
2017/7 ：Ver1.0.2
2-1～2-4 の章タイトルの見直し（内容に変更はありません）
2-2 の図のレイアウトを微修正
2-4 の以下の文章を削除
「当社本人認証機能は、VISA/MASTER/JCB の 3 つの国際カードブランドの本人認証（3-D Secure）に対応しています。」
2018/2 ：Ver1.0.3
3.1.2 に、セキュリティコードの必須チェック機能に関する説明を追加
「4-3 セキュリティコードの必須チェック機能」を追加
2018/6 ：Ver1.0.4
3.1.1 の暫定環境（SSL3.0/TLS1.0 有効）に関する記載を削除。推奨ブラウザを追加。
2019/4 ：Ver1.0.5
1-3 の TLS1.1 以上の通信に未対応の場合の記載を削除。
2020/7 ：Ver1.0.6
「1-3 制限事項」を最新の情報に更新（TLS1.1 の廃止に伴う更新）。
「3.1.1 アクセス URL」を最新の情報に更新（TLS1.1 の廃止に伴う更新）。
2021/7 ：Ver1.0.7
「3.1.1 アクセス URL」の推奨ブラウザの表記を変更。
「3.1.2 要求電文項目」に cardholder_name を追加。
「3.1.3 要求電文サンプル」に cardholder_name を追加。
「4-2 サンプルコードについて」に cardholder_name を追加。
2023/2 ：Ver1.0.8
「2-3 注意点」に本人認証（mpi-none）を利用する際の注意点を追加。
2024/3 ：Ver1.0.9
「3.1.1 アクセス URL」のトークンサーバー接続用の URL を変更

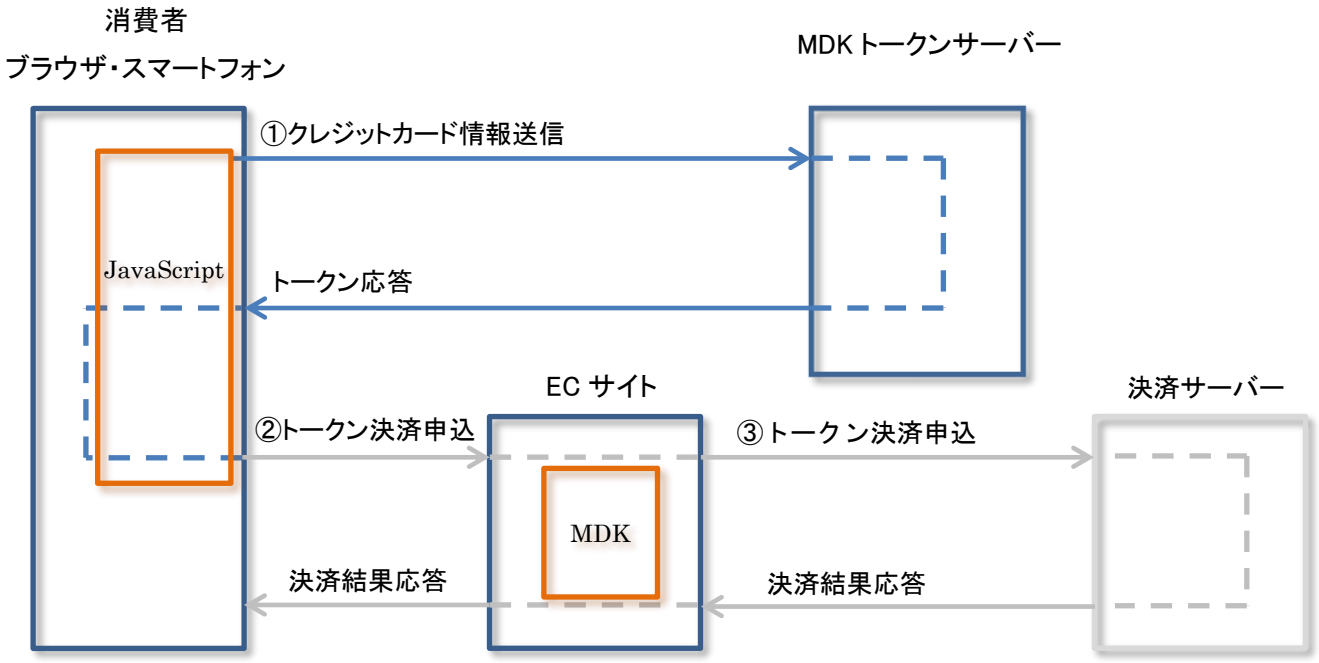
第2章 MDKトークン概要

2-1 概要

- MDK トークンサーバー（以下、トークンサーバー）は、カード情報非保持・非通過のシステムを実現するために必要な「トークン」を発行します。
- トークンサーバーに対してクレジットカード情報を送信することで、決済に利用するためのトークンを取得できます。
- トークンを取得する際は、JavaScript を用いてクレジットカード情報を消費者ブラウザから直接トークンサーバーに送信します。決済要求時には、クレジットカード番号の代わりにトークンを送ることで、EC サイト側のシステムではクレジットカード情報を通過させることなく決済が可能です。
- トークンを利用可能な決済サービスは「2-4 トークンで利用できる決済サービス」を参照して下さい。

2-2 決済処理の流れ

トークンを利用した決済処理概略は以下の通りです。



※②以降の処理は各決済サービスの開発ガイドを参照して下さい。

2-3 注意点

- トークンサーバーから取得したトークンは一度しかご利用いただけません（ワンタイムトークン）。
- 取得したトークンには有効期限があり、有効期限を過ぎるとそのトークンはご利用いただけなくなります。
  詳しくは、「3.2.1 応答電文」の応答電文の"token_expire_date"の項目を参照して下さい。
- トークンを取得する際には、マーチャント ID に紐づくトークン API キーをご使用下さい。
  トークン取得時と決済時でマーチャント ID が異なる場合、クレジットカード情報の識別ができません。

- 本人認証（3-D Secure）のサービスオプションタイプを”mpi-none（認証のみ）”とした場合は、後続のクレジットカードの与信用に別のトークンが必要になります。
  - トークンの取得処理を 2 回実行し、取得した 2 つのトークンの値を加盟店様サーバに連携し、1 つを 3D セキュア認証、もう 1 つをカード与信の電文に設定するという実装が必要となります。

2-4 対象サービス

トークンを用いた各決済サービスの詳細は、各決済サービスの開発ガイドを参照して下さい。

決済サービス名	説明
クレジットカード決済	クレジットカードによる決済が利用できるサービスです。 各種ブランド、カード会社等発行のクレジットカードでのお支払いが可能です。
クレジットカード決済 （本人認証有り）	本人認証（3-D Secure）機能付きのクレジットカード決済が利用できるサービスです。
クレジットカード決済 （会員 ID 決済）	決済と同時に会員とカード情報を登録して、次回から会員を指定することで、カード情報の入力を必要とせず決済できるサービスです。 消費者自身でカード情報管理（追加・削除）を行うことも可能です。 決済時に会員 ID を指定することでカード情報入力画面を省略して、すでに登録済のクレジットカードで決済することも可能です（クイックチェックアウト）。 ※会員 ID 決済機能はワンクリック継続課金サービス（オプションサービス）の機能となります。ご利用いただくためには、別途お申し込みが必要となります。

表 2-4-1 決済サービス一覧

第3章 インターフェース詳細

本章では、MDK トークンを取得する際に使用する電文について説明します。

決済に関する電文の説明は各決済サービスの開発ガイドを参照ください。

3-1 要求電文

要求電文の仕様は以下の通りです。

プロトコル		HTTP1.1
メソッド		POST
リクエストヘッダ	**Accept**	application/json
	Content-Type	application/json
ポート番号		443
文字コード		UTF-8
URL		「3.1.1 アクセス URL」を参照
パラメータ		「3.1.2 要求電文項目」を参照

3.1.1 アクセス URL

アクセス URL	
トークンサーバー接続用	https://api3.veritrans.co.jp/4gtoken

＜接続要件＞

この URL にアクセスするためには、ユーザーが利用するブラウザが次の 2 点を満たしている必要があります。

- TLS1.2 以上での通信に対応している
- Cross-Origin Resource Sharing に対応している

＜推奨ブラウザ＞

Microsoft Edge

Chrome

Firefox

Safari

※各ブラウザの最新安定版を推奨

※推奨はしておりませんが、Microsoft Internet Explorer 11 でも動作確認済みです。

＜対応 OS＞

Windows7 以上

Android5 以上

iOS5 以上

OS X v10.9 以上

3.1.2 要求電文項目

MDK トークンを取得する際の要求電文に指定できる項目の一覧を以下に示します。

- 「設定」欄の内容は以下の通りです。
  - 必須項目：〇
  - 任意項目：△

要求電文 ： 消費者ブラウザ → トークンサーバー				
フィールド名	項目名	書式・制限	説明	設定
card_number	カード番号	半角数字 19 桁以内	数字のみで指定	〇
card_expire	カード有効期限	半角数字記号 5 桁	MM/YY （月 + "/" + 年）の形式 （例 "08/18"）	〇
security_code	セキュリティコード	半角数字 3 桁 or 4 桁	セキュリティコード	△注1
cardholder_name	カード保有者名	半角英数字記号 2-45 桁	カード保有者名	△注2
token_api_key	トークン API キー	半角英数記号 36 桁以内	店舗様毎で発行されたトークン取得に使う鍵	〇
lang	言語	右記参照	応答電文の"message"の言語を指定 "en"： 英語 "ja"： 日本語 ※指定なしの場合 Accept-Language の値を使用	△

注1）「セキュリティコードの必須チェック機能（4-3 参照）」を有効化している場合は、セキュリティコードが未設定の場合にエラーとなり、以下の結果を返戻します。

HTTP ステータス	400
code	missing_security_code

注2）カード保有者名は 3-D Secure 2.0 を利用する場合にのみ必要となるフィールドです。

3.1.3 要求電文のサンプル

```
{
  "card_number":"4111111111111111",
  "card_expire":"01/20",
  "security_code":"123",
  "cardholder_name":"TARO YAMADA",
  "token_api_key":"test-token-api-key",
  "lang":"en"
}
```

3-2 応答電文

応答電文の仕様は以下の通りです。

レスポンスヘッダ	**Content-Type**	application/json
文字コード		UTF-8
パラメータ		「3.2.1 応答電文項目」を参照

3.2.1 応答電文項目

トークンサーバーからの応答電文の項目の一覧を以下に示します。

■ 「設定」欄の内容は以下の通りです。

- 必ず返戻：〇
- 処理成功時のみ返戻：△

応答電文 ： トークンサーバー → 消費者ブラウザ				
フィールド名	**項目名**	**書式・制限**	**説明**	**設定**
token	トークン	半角英数記号 36 桁	決済時にクレジットカード情報の識別に用いるトークンの値	△
token_expire_date	トークン有効期限	半角数字 14 桁	YYYYMMDDhhmmss の形式 (発行後 15 分)	△
req_card_number	要求カード番号	半角数字記号 19 桁以内	要求電文に設定した値 上 6 桁下 2 桁のみ数字表示され、 その他は ″*″(アスタリスク)に変換されます。 (例 ″411111********11″)	△
status	ステータス	右記参照	″success″：成功 ″failure″：失敗	〇
code	結果コード	半角英字記号	処理結果を詳細に表すコード	〇
message	結果メッセージ	文字列	処理結果を英語又は日本語で表示	〇

※決済サーバー側でクレジットカード情報の取得が失敗した際に、トークンの有効期限切れが原因かどうか EC サイト側でご確認いただけるように、″token_expire_date″の値を保管するようにして下さい。

※結果コード・結果メッセージの一覧は別紙「Token_ResultCodeList」を参照下さい。

3.2.2 応答電文のサンプル

・トークン取得成功時

```
{
  ″token″:″3a024e60-5951-4521-81e8-db0003752af8″,
  ″token_expire_date″:″20161012182641″,
  ″req_card_number″:″411111********11″,
  ″status″:″success″,
  ″code″:″success″,
  ″message″:″Token has been successfully created.″
}
```

・トークン取得失敗時（カード番号のフォーマットエラー）

```
{
    "status": "failure",
    "code": "invalid_card_number",
    "message": "Card number is invalid."
}
```

・トークン取得失敗時（マーチャント認証エラー）

```
{
    "status": "failure",
    "code": "unauthorized_merchant",
    "message": " Merchant authorization error occurred."
}
```

第4章 その他補足事項

4-1 テストの実施方法

各決済サービスの取引時に、トークンサーバーから取得した MDK トークンを要求電文項目に設定し、取引が行えることを確認して下さい。

各決済のテスト仕様につきましては『導入テストガイド』を参照して下さい。

※ご利用可能なテストカード情報についても同ガイドを参照して下さい。

[注意点]

テストマーチャント ID を用いてダミー取引を実施する際は、テストマーチャント ID に紐づくトークン API キーをご使用下さい。

4-2 サンプルコードについて

MDK トークンを取得するための JavaScript コードサンプルを提供しておりますので、ご活用下さい。

```
function submitToken(e) {
    var data = {};
    data.token_api_key = document.getElementById('token_api_key').innerText;
    if (document.getElementById('cc-number')) {
        data.card_number = document.getElementById('cc-number').value;
    }
    if (document.getElementById('cc-exp')) {
        data.card_expire = document.getElementById('cc-exp').value;
    }
    if (document.getElementById('cc-csc')) {
        data.security_code = document.getElementById('cc-csc').value;
    }
    if (document.getElementById('cc-name')) {
        data.cardholder_name = document.getElementById('cc-name').value;
    }
    data.lang = "ja";

    var url = document.getElementById('token_api_url').innerText;

    var xhr = new XMLHttpRequest();
    xhr.open('POST', url, true);
    xhr.setRequestHeader('Accept', 'application/json');
    xhr.setRequestHeader('Content-Type', 'application/json; charset=utf-8');
    xhr.addEventListener('loadend', function () {
        if (xhr.status === 0) {
            alert("トークンサーバーとの接続に失敗しました");
            return;
        }
        var response = JSON.parse(xhr.response);
        if (xhr.status == 200) {
            document.getElementById('cc-number').value = "";
            document.getElementById('cc-exp').value = "";
            document.getElementById('cc-csc').value = "";
            document.getElementById('cc-name').value = "";
            document.getElementById('token').value = response.token;
            document.forms[0].submit();
        }
        else {
            alert(response.message);
        }
    });
    xhr.send(JSON.stringify(data));
}
```

［注意点］

- このサンプルコードは、トークンサーバーとの通信処理実装の一例であり、加盟店様のサイトで確実に動作することを保証するものではございません。加盟店様のサイトの仕様に合わせてカスタマイズして頂き、必ず十分なテストを行って頂きますようお願いします。
- このサンプルコードは、別途提供中の MDK サンプルプログラムのアーカイブに含まれます。上記コード中の未定義の変数等は、MDK サンプルプログラムをダウンロードしてご確認ください。

4-3 セキュリティコードの必須チェック機能

本機能が有効化されている場合、加盟店様の EC サイトやアプリケーションをご利用のお客様がセキュリティコードを入力しなかった場合に、トークンサーバーからエラーを返戻します。加盟店様の実装での必須チェックも必要ですが、加盟店様システムでのチェックと併せて本機能をご利用いただくことで、セキュリティコードが指定されていない不正な要求をトークンサーバーがエラーにしますので、より安全に決済をご利用いただけます。

本機能は、マーチャント ID 毎に弊社側で有効化／無効化の切替えを行います。
有効化／無効化の初期設定は、マーチャント ID の発行時期によって異なりますので、以下の内容をご確認いただきご不明な際は弊社までお問い合わせください。

- ✓ 2018 年 3 月 31 日以前に発行されたマーチャント ID をご利用の場合、初期設定では本機能は無効化されています。有効化をご希望の加盟店様は、弊社までご連絡下さい。
- ✓ 2018 年 4 月 1 日以降に、新規に発行されるマーチャント ID につきましては、初期設定として本機能を有効化した状態で発行いたしますので、無効化を希望される新規の加盟店様は、弊社までご相談ください。