

旧システムトークン決済_EMV3-Dセキュア導入について

1.はじめに

1.1 本ドキュメントについて

現在開発中のEMV3-Dセキュア2.0を旧システムで利用する際に発生する、想定される作業内容を整理した仮ドキュメントです。

開発状況、内容によって変更が発生する可能性があります。

1.2 3-Dセキュア認証のタイミング

ラッパー機能を利用している場合、カード情報をトークン化する時のみ、カード登録の3-Dセキュア認証を実行します。

upcmemberidを指定した以降のジョブにおいて、3-Dセキュア認証が実行されることはありません。

1.3 3-Dセキュア認証に必要な追加項目

カードホルダーの電話番号・メールアドレスが必須項目となります。

住所の項目は現時点では任意ですが、2025年3月以降、必須になる可能性があります。

1.4 注意事項

3-Dセキュア認証にはカードホルダーの電話番号・メールアドレス・(住所)が必要になる為、トークン作成時に情報がない場合、またはダミー情報をセットしている場合は3-Dセキュア認証が失敗する可能性があります。

※決済時にカードホルダーの電話番号・メールアドレスをセットされる運用の場合は、カード情報のトークン化時にセットする変更が必要となります。

2. 必要作業

2.1 参照資料

旧システムトークン方式サンプルコード

<https://link.upc-app.com/js.html>

旧システム クレジット決済接続仕様書(変更後)

本ドキュメントと併せてお送りした3-Dセキュア対応実装後の仕様書をご確認ください。

2.2 トークン作成時の項目追加

カードホルダーの電話番号・メールアドレス・(住所)を取得する必要があります。

トークン作成時のコードに項目の追加が必要になります。

※住所情報は、1つでも項目に値が入る場合、全ての住所関連項目が必須になります。

※決済時にカードホルダーの電話番号・メールアドレスをセットされる運用の場合は、カード情報のトークン化時にセットする変更(決済時にはセットしないようにする)が必要になります。

2. 店舗IDの識別と、各種フィールド名を定義するJavaScript

```
● Multipayment.init('100001'); //当社発行の店舗ID
● Multipayment.getMember({ //決済フォームより取得した情報
●   cardno: cardno, //カード番号
●   securitycode: securitycode, //セキュリティコード
●   expire: expire, //カード有効期限
●   holderfirstname: holderfirstname, //カードホルダー名
●   holderlastname: holderlastname, //カードホルダー姓
●   memberid: memberid, //会員番号
●   membercode: membercode, //会員コード
●   addressLine1: addressLine1, //住所(番地・号)
●   addressLine2: addressLine2, //住所(ビル名等)
●   addressCity: addressCity, //住所(市区町村)
●   addressState: addressState, //住所(都道府県)
●   addressZip: addressZip, //住所(郵便番号)
●   addressCountry: addressCountry, //住所(国)
●   phonenumCountryCode: phonenumCountryCode, //国際電話の国番号
●   email: email, //メール ※任意から必須へ変更
●   phonenum: phonenum //電話番号 ※任意から必須へ変更
```

※赤字が追加内容になります

※メールアドレスが無ければ当システム側でダミー情報をセットします

3.3-Dセキュア認証の処理の流れ

3.1 新規トークン作成時の決済

新規のトークンを作成して決済処理を行う場合、トークン作成時に**カード登録の3-Dセキュア認証**を行います。

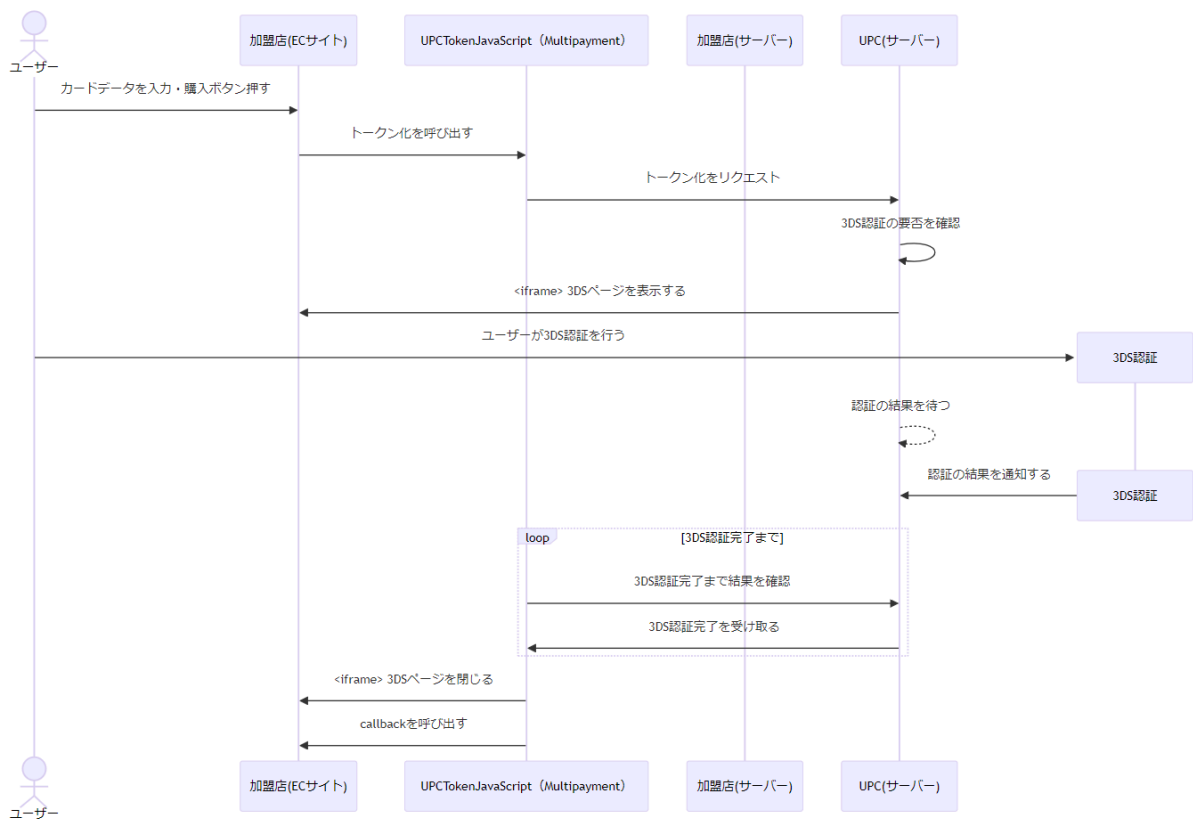
※トークン作成時に電話番号・メールアドレス・住所の項目をセットして、決済時にはセットしない想定となります

トークン作成時、iframe上で強制的に3-Dセキュア認証ページを表示します。

加盟店サーバーでiframeを展開できる環境であることを確認してください。

メールアドレス・電話番号を除き、決済リクエストなどのジョブの実行、以降のパラメータ変更は必要ありません。

3.2 シーケンス図



3.3 決済リクエスト

決済のリクエスト方法に変更はありません。

※旧システムトークン方式サンプルコード参照

5. サーバ間通信を行うプログラム

ここでは一例としてphpを使い、 仮実働時処理をする記述を紹介しています。実装時には接続仕様書とゲートウェイ接続サンプルを参照の上、加盟店サイトのビジネスモデルに適したジョブ（処理）内容に編集してください。

```
プレーン表示 クリップボードにコピー 印刷
01. require_once 'HTTP/Request.php';
02.
03. /*
04.     sid: 店舗ID
05.     svid: サービス種別 (1: 固定)
06.     ptype: 処理種別 (1: Gateway接続)
07.     job: 決済ジョブタイプ (CAPTURE: 仮実同時処理)
08.     upcmemberid: トークン
09.     rt: 結果返信方法 (2: レスポンス)
10.     sod: 店舗オーダー番号 (店舗独自のパラメータ)
11.     siam1: 商品金額
12.     sisf1: 商品送料
13. */
```

3.4 決済結果の受取り

決済の結果、成否の受取りに変更はありません。

※旧システムトークン方式サンプルコード参照

```
44.
45.     if ( (int)$target_array[1] == 1 ) {
46.         /* 決済処理成功の場合はここに処理内容を記載 */
47.         print ( '決済成功' );
48.     } else {
49.         /* 決済処理失敗の場合はここに処理内容を記載 */
50.         print ( '決済失敗' );
51.     }
```

3.5 決済結果通知

決済結果通知URLへキックバック(ウェブフック)通知を行います。

※旧システム クレジット決済接続仕様書(変更後)52ページを参照

4. コールバックのresultcodeの追加

3-Dセキュア認証を追加するにあたり、コールバックの結果内のresultcodeを追加します。
追加されるコードは下記表の通りです。

※旧システム クレジット決済接続仕様書(変更後)22ページ(5章 カード情報のトークン化)を
参照

コード	日本語(ja)	英語(en)
320	住所(国)が不正です。	Address's Country is Invalid.
321	電話番号(国コード)が不正です。	Phone Number Country Code is Invalid.
322	住所情報が不十分です。	Address is incomplete.
323	住所は必須です。	Address is required.
324	許可されていない国のカードです。	Card's Country is forbidden.
325	3-Dセキュア認証に失敗しました。	3D Secure verification has failed.
326	3-Dセキュア未登録カードです。3-Dセキュア登録後、再決済を行うか、別のカードを使用してください。	Card is not enrolled into 3D Secure, please enroll it or use another card.
327	住所(都道府県)が不正です。	Address State is Invalid.
