



Advanced Credit Card

3DS 実装概要

Global Professional Services

目次

1	はじめに.....	3
1.1	この文書について.....	3
2	3DS 実装概要.....	4
2.1	JavaScript にて contingencies フラグの追加.....	4
2.2	サーバサイドプログラムでの 3D セキュア認証の判定処理の追加.....	4
2.3	テスト用カード.....	5
3	token を利用した決済における 3DS 実装概要.....	5
3.1	token を利用した決済への 3DS 適用.....	6
3.2	token を利用した継続課金.....	9
4	ドキュメント.....	10

1 はじめに

1.1 この文書について

この文書は別紙「Solution Design - PayPal ACDC」に基づき、ACDCの実装が完了していることを前提として、3DS機能の実装方法について説明します。

2 3DS 実装概要

既に実装された **Custom Card Fields** に **3DS** を適用させるためには、次の追加実装を行います。

- 決済ページに、個人情報の利用目的について顧客の同意を取得するための文言とチェックボックスを実装。日本クレジット協会「**EMV 3-D セキュア導入ガイド**」 「**5. EMV 3-D セキュア導入加盟店における個人情報保護法の遵守に関する留意点**」をご参照下さい。
- **JavaScript** にて **contingencies** フラグの追加
- 決済時にカードホルダ氏名、住所、電話番号またはメールアドレスを送信 (**Advanced Credit Card** 実装ガイドを参照)
- サーバサイドプログラムでの **3DS** 認証の判定処理の追加

2.1 JavaScript にて contingencies フラグの追加

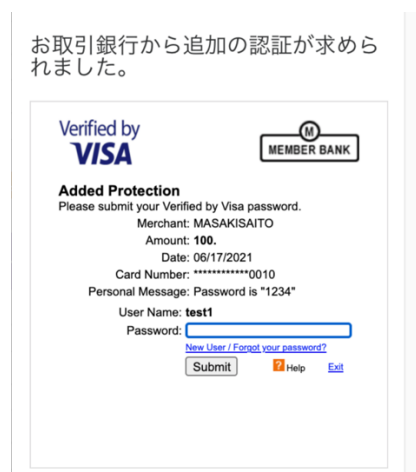
次のように **HostedField** の **submit** 処理内に **contingencies** パラメータを追加します。

```
hf.submit({  
  
    // Trigger 3D Secure authentication  
    contingencies: ['SCA_ALWAYS']  
  
}).then(function (payload) {
```

<https://developer.paypal.com/docs/checkout/advanced/customize/3d-secure/v1/sdk/>

contingencies パラメータには“**3D_SECURE**”, “**SCA_ALWAYS**”, “**SCA_WHEN_REQUIRED**”の3つがあります。“**3D_SECURE**”と“**SCA_ALWAYS**”, は、すべてのトランザクションに対して認証を行います。 “**SCA_WHEN_REQUIRED**”は、**PSD2** のような地域的なコンプライアンス義務が必要な場合にのみ認証を行います。

contingencies パラメータを追加することにより **3DS** が適用され、下図のようなポップアップが表示されます。ユーザーは自身が設定した **3DS** のパスワード等の回答を入力し **Submit** します。



2.2 サーバサイドプログラムでの 3D セキュア認証の判定処理の追加

ユーザーが **3DS** の **Submit** を行なった後で、**Show Order Details API** を使って、**3D セキュア**の認証を通っているかどうかの情報を取得します。**Show Order Details API** については、以下をご参照ください。

https://developer.paypal.com/docs/api/orders/v2/#orders_get

```
"authentication_result": {  
  "liability_shift": "POSSIBLE",  
  "three_d_secure": {  
    "enrollment_status": "Y",  
    "authentication_status": "Y"  
  }  
}
```

この API のレスポンス内の、`payment_source -> card -> authentication_result` の値で、引き続き売上確定 Capture Order API を実行するかどうかの判定をします。

判定条件については、以下のサイトにある表を参照ください。この表の中で「Continue with authorization.」となっている場合に引き続き売上確定 Capture Order API（もしくは Authorize API）を行うことを推奨しています。この推奨の判定条件では 3D 非適用であっても広く決済受付を行うような判定となっていますので、リスクを取りたくない場合は 3D 適用の場合のみ決済受付を行い、Liability_shift が POSSIBLE のケースのみ売上確定を行うようにしてください。

<https://developer.paypal.com/docs/checkout/advanced/customize/3d-secure/response-parameters/>

2.3 テスト用カード

3DS のテスト用のカードは正常系・異常系等のテストシナリオ毎に、カード番号と有効期限、CVV が用意されています。下記のサイトをご参照ください。

<https://developer.paypal.com/docs/checkout/advanced/customize/3d-secure/test/>

また、3D セキュアかつ Vault をテストするテスト用カード番号は以下を使用してください。(正常系のみ)

テストカード番号	有効期限	CVV
4507371021526764	01/2025	123
5206181900115543		
5470432115202579		

3 token を利用した決済における 3DS 実装概要

Vault したカードの token を利用した決済において 3DS を適用させるための実装方法について説明します。

前提として 3DS 未適用時の決済は次のように実施します。

- ① CreateOrder を実行します。 POST /v2/checkout/orders
- ② Authorize を実行します。 POST /v2/checkout/orders/{order-id}/authorize
Request Body

```
{
  "payment_source": {
    "card": {
      "vault_id": "mw2ymwm"
    }
  }
}
```

- ③ Capture を実行します。

3.1 token を利用した決済への 3DS 適用

- ① Request Body に payment_source を設定して CreateOrder を実行します。
POST /v2/checkout/orders

Request Body:

```
{
  "intent": "CAPTURE",
  "application_context": {
    "return_url": "https://example.com",
    "cancel_url": "https://example.com"
  },
  "purchase_units": [
    {
      "amount": {
        "currency_code": "JPY",
        "value": "100"
      }
    }
  ],
  "payment_source": {
    "card": {
      "vault_id": "cpk5h93",
      "attributes": {
        "verification": {
          "method": "SCA_ALWAYS"
        }
      }
    }
  }
}
```

- ② HTTP201 で以下のレスポンスが返ってきます。

Response : HTTP 201

Request Body:

```
{
  "id": "37905563G21548127",
  "status": "PAYER_ACTION_REQUIRED",
  "purchase_units": [
    {
      "reference_id": "default"
    }
  ],
  "links": [
    {
      "href": "https://api.paypal.com/v2/checkout/orders/37905563G21548127",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "https://www.paypal.com/webapps/helios?action=verify&flow=3ds&cart_id=37905563G21548127",
      "rel": "payer-action",
      "method": "GET"
    }
  ]
}
```

- ③ "rel": "payer-action" のリンクを画面表示します。このように 3DS の認証画面が表示されます。

Your bank has requested additional verification.



- ④ 入力が完了すると下記のように liability_shift のパラメータ付きで return URL に遷移します。
https://example.com/?state=undefined&code=undefined&liability_shift=POSSIBLE

- ⑤ Show Order Details API のレスポンスで authentication_result を確認します。

```

{
  "id": "37905563G21548127",
  "intent": "CAPTURE",
  "status": "CREATED",
  "payment_source": {
    "card": {
      "last_digits": "7109",
      "brand": "VISA",
      "type": "PREPAID",
      "authentication_result": {
        "liability_shift": "POSSIBLE",
        "three_d_secure": {
          "enrollment_status": "Y",
          "authentication_status": "Y"
        }
      }
    }
  },
  "purchase_units": [
    {
      "reference_id": "default",
      "amount": {
        "currency_code": "JPY",
        "value": "100"
      },
      "payee": {
        "email_address": "ppmassaito@gmail.com",
        "merchant_id": "Z6YFPJSWN6QYL"
      }
    }
  ],
  "create_time": "2021-05-24T04:54:32Z",
  "links": [
    {
      "href": "https://api.paypal.com/v2/checkout/orders/37905563G21548127",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "https://www.paypal.com/checkoutnow?token=37905563G21548127",
      "rel": "approve",
      "method": "GET"
    },
    {
      "href": "https://api.paypal.com/v2/checkout/orders/37905563G21548127",
      "rel": "update",
      "method": "PATCH"
    },
    {
      "href": "https://api.paypal.com/v2/checkout/orders/37905563G21548127/capture",
      "rel": "capture",
      "method": "POST"
    }
  ]
}

```

- ⑥ ステータスを確認した結果、決済を進めても問題がないと判断される場合は、**Capture** を実行します。

参考：<https://developer.paypal.com/docs/checkout/advanced/customize/3d-secure/response-parameters/>

3.2 token を利用した継続課金

強固な顧客認証（SCA）が必要な決済を処理する場合、支払い指標を使用して取引についての追加コンテキストを提供する必要があります。支払い指標を使用することで、購入者認証、保存されたカード、その他の要因が適切に処理されることが保証されます。これらの支払い指標を渡すことで、取引が拒否されることを避けることができます。取引に関する追加のコンテキストを提供するために、注文リクエストの作成時に `stored_credential` を含めてください。

- ① Request Body に `payment_source` を設定して `CreateOrder` を実行します。
POST /v2/checkout/orders

Request Body:

```
{
  "intent": "CAPTURE",
  "purchase_units": [
    {
      "amount": {
        "currency_code": "JPY",
        "value": "100"
      }
    }
  ],
  "payment_source": {
    "card": {
      "vault_id": "cpk5h93",
      "stored_credential": {
        "payment_initiator": "MERCHANT",
        "payment_type": "RECURRING",
        "usage": "SUBSEQUENT",
        "previous_transaction_reference": "95M63758F24067141"
      }
    }
  }
}
```

4 ドキュメント

3D Secure : Javascript SDK

<https://developer.paypal.com/docs/checkout/advanced/customize/3d-secure/sdk/>

3D Secure : Order API

<https://developer.paypal.com/docs/checkout/advanced/customize/3d-secure/api/>

3D Secure Sandbox テスト用のカード情報

<https://developer.paypal.com/docs/checkout/advanced/customize/3d-secure/test/>

※Sandbox でのテストは可能ではありますが、V プリカ等のカードをご用意いただき本番環境でのテストを行なっていただくことを推奨いたします。

Orders API

<https://developer.paypal.com/docs/api/orders/v2/>

Payments API

<https://developer.paypal.com/docs/api/payments/v2/>