



Advanced Credit Card Integration Guide

目次

1	概要	3
1.1	実装概要	3
2	Payment Experience	3
2.1	クレジットカード決済	3
3	PayPal アカウント登録・API 認証情報取得	3
3.1	PayPal ビジネスアカウントを作成する	3
3.2	テスト用 PayPal アカウントを作成する	3
3.3	REST API アプリの作成	5
3.4	API 認証情報を取得する	6
4	実装方法	6
4.1	JavaScript SDK	8
4.2	クレジットカード決済	8
4.3	クレジットカード情報の保存機能 (Vault)	21
4.4	保存したカード情報での決済	24
4.5	保存したカード情報での継続課金	29
4.6	シーケンス図	31
4.7	キャンセル・返金	31
4.8	処理結果	32
4.9	エラー処理	37
4.10	ベストプラクティス	37
5	ドキュメント	38

1 概要

1.1 実装概要

実装していただく内容は以下のとおりです。

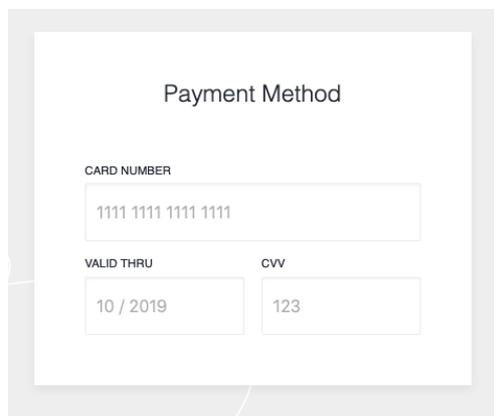
- クレジットカード決済

以下の技術を使用します。

- PayPal JavaScript SDK
- Orders API v2
- Payments API v2

2 Payment Experience

2.1 クレジットカード決済



The image shows a 'Payment Method' form with the following fields:

- CARD NUMBER: 1111 1111 1111 1111
- VALID THRU: 10 / 2019
- CVV: 123

Advanced credit and debit card payments は、PayPal JavaScript SDK を使用してクレジットカードでの支払いを安全に受け付ける方法であり、デスクトップとモバイルのウェブサイトのチェックアウト UI のスタイルをコントロールすることができます。Advanced credit and debit card payments を使用して収集されたデータは、お客様のクライアントから PayPal に直接送信されるため、お客様の生の支払い情報がお客様のサーバーに触れることはありません。

3 PayPal アカウント登録・API 認証情報取得

3.1 PayPal ビジネスアカウントを作成する

<https://www.paypal.com/jp/business> で PayPal ビジネスアカウントを作成してください。

3.2 テスト用 PayPal アカウントを作成する

PayPal Sandbox 環境で取引をシミュレーションできるように、テスト用のバイヤーとマーチャントのアカウントを作成します。(デフォルトで自動作成されているアカウントを使用することもできます。)

PayPal Sandbox とは

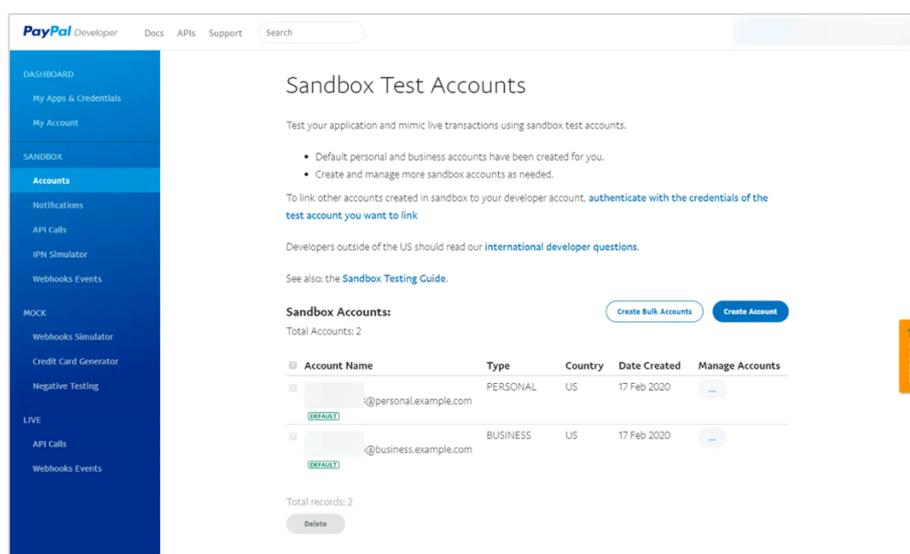
PayPal Sandbox <https://sandbox.paypal.com> は、本番の PayPal 環境 <https://paypal.com> のテスト版です。ここでは、本番に入る前にトランザクションをテストし、実装が期待通りに動作していることを確認します。

Environment	Base URL	Test here
Sandbox	https://api-m.sandbox.paypal.com	www.sandbox.paypal.com
Live	https://api-m.paypal.com	www.paypal.com

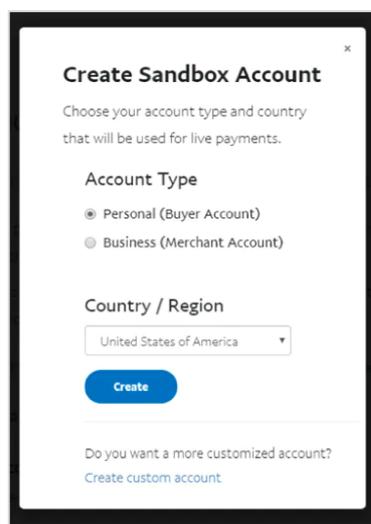
Sandbox アカウントの作成

Sandbox のテストアカウントを作成するには

1. Dashboard にログインします。 <https://developer.paypal.com/developer/accounts/>
2. Sandbox で Accounts を選択します。
3. Create Account (アカウントの作成) を選択します。



4. Personal または Business を選択し、テストの買い手または売り手の国/地域を選択します。
5. Create を選択します。PayPal の残高、クレジットカード、銀行口座情報などが記載された Sandbox メールアカウントが生成されます。



6. Sandbox > Accounts で作成したビジネスアカウントの右端の「…」から View/Edit Account をクリックします。Profile > Account Type の Upgrade to Pro をクリックします。
7. 下記リンクから本人確認 (ダミー) を完了してください。
<https://www.sandbox.paypal.com/policydashboard/process/compliance/cip>

Sandbox アカウントのデフォルト設定を変更するには、以下の手順に従います。

1. Manage Accounts で (...) を選択し、View/Edit Account を選択します。
2. Edit を選択し、変更を行います。
3. 完了したら、Save を選択します。

Sandbox にログインするには

1. <https://www.sandbox.paypal.com/> にアクセスします。
2. テスト購入者または販売者のメールアドレスとパスワードを使用してログインします。
3. 実装するには、テストバイヤーとセラーのアカウントを交互に使用して、本番前に取引が期待通りに動作しているかどうかを確認することができます。

3.3 REST API アプリの作成

- **Live 環境用の REST App を作成するためには、PayPal アカウントのメールアドレスの確認が完了している必要があります。**まだ確認を行っていない場合は、PayPal アカウント作成時にメールが送信されていますのでご確認ください。以下の方法で確認メールを再送することも可能です。

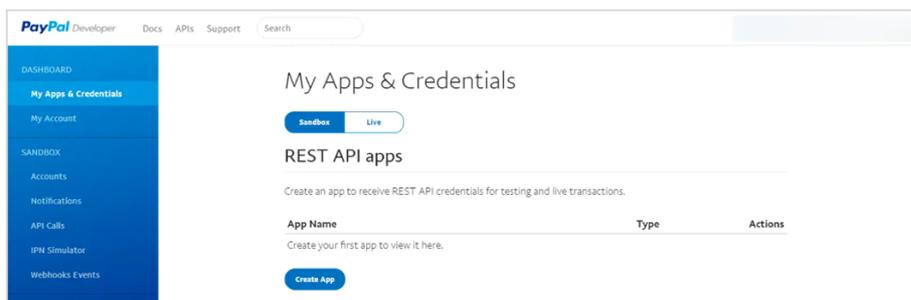
メールアドレスを確認する方法

<https://www.paypal.com/jp/smarthelp/article/%E3%83%A1%E3%83%BC%E3%83%AB%E3%82%A2%E3%83%89%E3%83%AC%E3%82%B9%E3%82%92%E7%A2%BA%E8%AA%8D%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95%E3%82%92%E6%95%99%E3%81%88%E3%81%A6%E3%81%8F%E3%81%A0%E3%81%95%E3%81%84%E3%80%82-faq1485>

API 認証情報を取得する最初のステップは、REST API アプリを作成することです。アプリは、PayPal API への呼び出しを認証するために使用される保存された設定のセットです。アカウントに関連付けられた複数のアプリを持つことができます。

REST API アプリを作成するには、以下の手順に従います。

1. ダッシュボード
<https://www.paypal.com/signin?returnUri=https%3A%2F%2Fdeveloper.paypal.com%2Fdeveloper%2Fapplications> にログインします。
2. Create app を選択します。
3. App Name を入力します。
4. Create app を選択します。
5. LIVE (または SANDBOX) APP SETTINGS > Accept payments > Vault にチェックを入れてください。

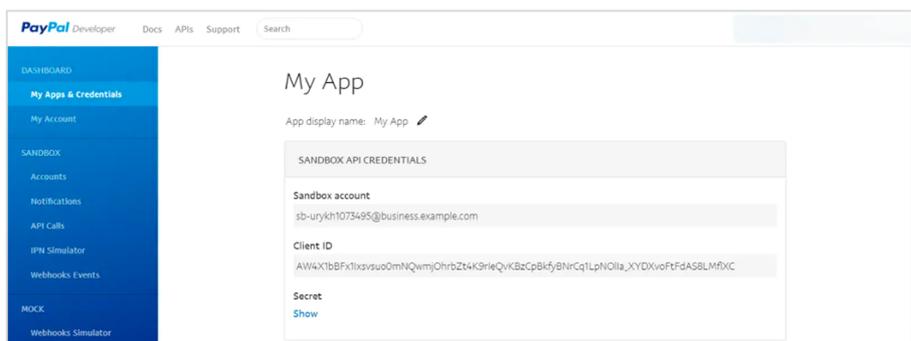


3.4 API 認証情報を取得する

API 認証情報は、クライアント ID とシークレットで構成されており、サンドボックスとライブ環境で API 呼び出しを行い、トランザクションを処理するために必要となります。

サンドボックス API 認証情報を取得するには、以下の手順に従います。

1. ダッシュボード
<https://www.paypal.com/signin?returnUri=https%3A%2F%2Fdeveloper.paypal.com%2Fdeveloper%2Fapplications> にログインします。
2. My Apps & Credentials で、リストから REST API アプリを選択します。
3. Client ID をコピーします。
4. Secret で Show を選択して、Secret を表示します。



4 実装方法

REST API URL

PayPal REST API と通信するには、サンドボックス環境または本番環境のいずれかで、JSON 形式の HTTP リクエストを API エンドポイントに渡します。

Environment	Base URL
Sandbox	https://api-m.sandbox.paypal.com
本番環境	https://api-m.paypal.com

API べき等性

<https://developer.paypal.com/api/rest/reference/idempotency/>

サーバーがリソースに対するアクションを複数回作成したり完了したりすることを気にすることなく、何度でもコールを行うことができます。ネットワークタイムアウトや HTTP 5xx ステータスコー

ドで失敗した呼び出しを、サーバが ID を保存している限り再試行することができます。べき等性を使用すると、リクエストのペイロードをレスポンスのペイロードと相関させ、重複したリクエストを排除し、失敗したリクエストやレスポンスが不明瞭なリクエストを再試行することができます。

REST API の POST 呼び出しで Idempotency を強制するには、PayPal-Request-Id リクエストヘッダーを使用します。このヘッダーには、サーバが一定期間保存する一意のユーザー生成 ID が含まれています。

注意: すべての API がこのヘッダーをサポートしていません。お使いの API がこのヘッダーをサポートしているかどうか、およびサーバが ID を保存する期間に関する情報については、お使いの API のリファレンスを参照してください。

たとえば、以前に指定した PayPal-Request-Id ヘッダーをリクエストに含めると、PayPal は、同じヘッダーを使用した以前のリクエストの最新のステータスを返します。逆に、リクエストから PayPal-Request-Id ヘッダーを省略すると、PayPal はリクエストを複製します。

注意: 同じ PayPal-Request-Id ヘッダーを持つ 2 つの API リクエストを同時に送信すると、PayPal は最初のリクエストを処理し、2 目目のリクエストに失敗する可能性があります。

使用上の注意

- PayPal-Request-Id ヘッダー値は、各リクエストと API コールタイプの両方で一意でなければなりません。たとえば、authorize payment や capture authorized payment などです。
- PayPal は、元のリクエストのステータスではなく、現在の時点でのリクエストのステータスを提供します。

Rate Limit

PayPal は、加盟店をサポートするために、サイトの可用性とセキュリティに主眼を置いています。

レートリミットポリシーは公表していませんが、乱用と思われるトラフィックを確認した場合、一時的に API 使用制限を行うことがあります。API 使用制限は、その行為が PayPal、加盟店、またはお客様にとって問題がないと確信できるまで行います。

最大限の保護とサイトの安定性を確保するために、トラフィックの急増と減少を常に評価し、ポリシーを調整しています。お客様やお客様の顧客が HTTP 429 Unprocessable Entity - RATE_LIMIT_REACHED というステータスコードを受け取った場合、リクエストが多すぎて異常なトラフィックを示している可能性があるため、サイトの安定性を確保するために API 使用制限を行います。

<https://developer.paypal.com/api/rest/reference/rate-limiting/>

Sandbox では、1 分以内に同一 IP アドレスから 50 以上のリクエスト（すべてのエンドポイントの累積ヒット数）を受信すると、次の 5 分間その IP をブロックします。

IP アドレスが一時的にブロックされるのを防ぐには、1 分間のリクエスト数を 50 以下に制限してください。

<https://www.paypal.com/us/smarthelp/article/why-did-i-get-http-error-code-429-ts2115>

注意:

貴社が PCI-DSS に準拠しており、貴社サーバの同一 IP よりクレジットカードフォームの SDK を利用したリクエストをするケースでは、本番環境においても Rate Limit の制限を受け、またカードアタッ

クのリスク判定エラーが発生する可能性があります。システムの制約などの事情により同一 IP にて SDK 経由でのリクエストを実施されたい場合は必要に応じて設定を行いますので PayPal 担当者にご連絡ください。

なお、REST API によるバッチ処理については同一 IP からのリクエストであっても本番環境においてこれらの Rate Limit は発生しません。

4.1 JavaScript SDK

JavaScript SDK は、あなたのページに PayPal がサポートしている支払い方法を表示し、バイヤーにパーソナライズされた合理的なチェックアウト体験を提供します。

JavaScript SDK スクリプトにクエリとスクリプトのパラメータを渡すことで、実装をカスタマイズできます。これらのパラメータは、PayPal が最適な決済方法とボタンを決定して、バイヤーに表示するのに役立ちます。

<https://developer.paypal.com/sdk/js/configuration/>

<https://developer.paypal.com/sdk/js/reference/>

<https://developer.paypal.com/sdk/js/performance/>

4.2 クレジットカード決済

<https://developer.paypal.com/docs/checkout/advanced/>

概要

1. 以下のサンプルコードをコピーして、チェックアウトページに貼り付けてください。
2. コード中の YOUR_SB_CLIENT_ID を、使用する Client ID に置き換えてください。
3. 必要に応じて JavaScript SDK に渡すパラメータを明示的に指定して、デフォルトの値を上書きしてください。各パラメータのデフォルト値については以下をご参照ください。
<https://developer.paypal.com/sdk/js/configuration/>
4. 決済情報を送信するためのサーバ側コードを実装してください。createOrder と onApprove のメソッド内からサーバ側へリクエストしてください。サーバ側の実装は以降の説明をご参照ください。

対応通貨

AUD, BRL, CAD, CHF, CZK, DKK, EUR, GBP, HKD, HUF, ILS, JPY, MXN, NOK, NZD, PHP, PLN, SEK, SGD, THB, TWD, USD

(American Express - AUD, CAD, CHF, CZK, DKK, EUR, GBP, HKD, HUF, JPY, NOK, NZD, PLN, SEK, SGD, USD のみ)

(JCB - JPY のみ)

4.2.1 Access Token を作成する

<https://developer.paypal.com/api/rest/authentication/>

サンプルリクエスト

```
curl -v https://api-m.sandbox.paypal.com/v1/oauth2/token \
-H "Accept: application/json" \
-H "Accept-Language: en_US" \
-u "<client_id>:<secret>" \
-d "grant_type=client_credentials"
```

サンプルレスポンス

3D セキュアを適用するためには、HostedField の submit 処理内に contingencies パラメータを追加します。

また、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します。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。

```
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"SCA_ALWAYS", "SCA_WHEN_REQUIRED"があります。
"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 3D セキュアが適用され、下図のようなポップアップが表示されます。ユーザーはワンタイムパスワード等を入力し送信します。



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/>

フォームのカスタマイズは、下記を参照してください。

<https://developer.paypal.com/docs/checkout/advanced/customize/card-field-style/>

注意:

Internet Explorer ではクレジットカード決済をご利用いただけません。以下のような注意文を掲載してください。

```
Internet Explorer ではクレジットカード決済をご利用いただけません。
Microsoft Edge, Google Chrome, Safari などのブラウザをご利用ください。
```

また、Android 端末で Chrome を使っている場合で Chrome のライトモードをオンに設定している場合に、カード情報の入力ができないという事象が発生することがあります（ライトモードだと必ず発生するという事象でもないようです）。この場合にはライトモードをオフにさせていただく必要があります。

サンプルフォーム

```
<!doctype html>
<head>
  <script
    src="https://www.paypal.com/sdk/js?components=hosted-fields&client-
id=<client_id>&currency=JPY&intent=capture"
    data-client-token="<client_token">
  </script>
</head>

<body>
  <form id="my-sample-form">
    <label for="card-number">Card Number</label>
    <div id="card-number"></div>

    <label for="expiration-date">Expiration Date</label>
    <div id="expiration-date"></div>

    <label for="cvv">CVV</label>
    <div id="cvv"></div>

    <button value="submit" id="submit" class="btn">Pay with Card</button>
  </form>

  <script>
    // Check if card fields are eligible to render for the buyer's country. The card fields are not eligible in all
    countries where buyers are located.
    if (paypal.HostedFields.isEligible() === true) {
      paypal.HostedFields.render({
        createOrder: function (data, actions) {
          // logic to return an order ID from your server
          return fetch('/my-server/create-order', {
            method: 'POST'
          }).then(function (res) {
            return res.json();
          }).then(function (data) {
            return data.id;
          });
        },
        styles: {
          'input': {
            'font-size': '14px',
            'font-family': 'Product Sans',
            'color': '#3a3a3a'
          },
          ':focus': {
            'color': 'black'
          },
          '.invalid': {
            'color': '#FF0000'
          }
        }
      });
    }
  </script>
</body>
</html>
```

```

    }
  },
  fields: {
    number: {
      selector: '#card-number',
      placeholder: 'Credit Card Number',
    },
    cvv: {
      selector: '#cvv',
      placeholder: 'CVV',
    },
    expirationDate: {
      selector: '#expiration-date',
      placeholder: 'MM/YYYY',
    }
  }
}).then(function (hf) {

  document.querySelector('#my-sample-form').addEventListener('submit', event => {
    event.preventDefault();
    hf.submit({

      // Trigger 3D Secure authentication
      contingencies: ['SCA_ALWAYS']

    }).then(function (payload) {
      // Make a call to capture the order (payload.orderId) here
    });
  });
});
}
else {
  /*
   * Handle experience when
   * Custom Card Fields is not eligible
   */
}
}
</script>

</body>
</html>

```

4.2.4 決済を作成する

サンプル HTML フォームの `/my-server/create-order` で下記 `/v2/checkout/orders` を貴社サーバからコールし決済を作成します。

https://developer.paypal.com/api/orders/v2/#orders_create

Intent に `CAPTURE` または `AUTHORIZE` を設定します。JavaScript のクエリパラメータで指定した intent と一致する必要があります。intent パラメータを `Capture` にすると決済が確定されます。Authorize の場合には、与信のみを取得して決済の確定は行われません。

以下のパラメータを追加してください。ギフト配送など購入者とは異なる配送先に商品を配送する場合は、purchase_units.shipping には購入者の住所ではなく実際の配送先の住所を指定してください。3D セキュアには購入者の氏名、住所、電話番号またはメールアドレスが必要です。

- purchase_units.shipping: 貴社サイト上で指定した配送先住所
- invoice_id: 貴社で発番した一意の注文 ID
- payer.name: 購入者の氏名
- payer.address: 購入者の住所
- payer.phone: 購入者の電話番号
- payer.email_address: 購入者のメールアドレス

また、必要に応じカードフォームで氏名と住所を入力していただき取得・送信する方法もあります。

<https://developer.paypal.com/docs/checkout/advanced/sdk/v1/#link-integratebackend>

サンプルリクエスト

```
curl -v -X POST https://api-m.sandbox.paypal.com/v2/checkout/orders \
-H "Content-Type: application/json" \
-H "Authorization: Bearer <access token>" \
-d '{
  "intent": "CAPTURE",
  "payer": {
    "name": {
      "given_name": "太郎",
      "surname": "青山"
    },
    "email_address": "buyer@example.com",
    "address": {
      "address_line_1": "青山 1-2-3",
      "address_line_2": "ペイパルビル 1001",
      "admin_area_2": "港区",
      "admin_area_1": "東京都",
      "postal_code": "107-0001",
      "country_code": "JP"
    },
    "phone": {
      "phone_type": "MOBILE",
      "phone_number": {
        "national_number": "08034567890"
      }
    }
  },
  "purchase_units": [
    {
      "amount": {
        "currency_code": "JPY",
        "value": "1000",
        "breakdown": {
          "item_total": {
            "currency_code": "JPY",
            "value": "1000"
          }
        }
      }
    }
  ],
  "items": [
```

```

    {
      "name": "Sample Goods",
      "unit_amount": {
        "currency_code": "JPY",
        "value": "1000"
      },
      "quantity": "1"
    }
  ],
  "shipping": {
    "name": {
      "full_name": "太郎 青山"
    },
    "address": {
      "address_line_1": "青山 1-2-3",
      "address_line_2": "ペイパルビル 1001",
      "admin_area_2": "港区",
      "admin_area_1": "東京都",
      "postal_code": "107-0001",
      "country_code": "JP"
    }
  },
  "invoice_id": "your-own-uniqueid-for-an-order",
  "custom_id": "貴社指定のカスタムパラメータ値"
}
]
}'

```

サンプルレスポンス

```

{
  "id": "3W9401969M757672V",
  "status": "CREATED",
  "links": [
    {
      "href": "https://api.sandbox.paypal.com/v2/checkout/orders/3W9401969M757672V",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "https://www.sandbox.paypal.com/checkoutnow?token=3W9401969M757672V",
      "rel": "approve",
      "method": "GET"
    },
    {
      "href": "https://api.sandbox.paypal.com/v2/checkout/orders/3W9401969M757672V",
      "rel": "update",
      "method": "PATCH"
    },
    {
      "href": "https://api.sandbox.paypal.com/v2/checkout/orders/3W9401969M757672V/capture",
      "rel": "capture",
      "method": "POST"
    }
  ]
}
]

```

```
}
```

4.2.5 3D セキュア認証結果を判定する

ユーザーが 3D セキュアの 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 セキュア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 セキュア適用の場合のみ決済受付を行い、Liability_shift が POSSIBLE または YES 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/>

※ 即時決済の場合は以下の「決済を実行する」を行う

4.2.6 決済を実行する

サンプル HTML フォームの /my-server/capture-order で下記 /v2/checkout/orders/{order_id}/capture を貴社サーバからコールし決済を実行します。

https://developer.paypal.com/api/orders/v2/#orders_capture

以下のヘッダを追加してください。

```
"PayPal-Request-Id: <Unique-Session-Identifier>"
```

※注意点

クレジットカード決済（PayPal アカウント決済でない）でカード会社からの拒否エラーであった場合、API のレスポンスは以下ようになります。

[レスポンスヘッダ]

```
HTTP レスポンスコード：2xx 台
```

[レスポンスボディ]

```
status: COMPLETED
```

```
purchase_units.payments.captures.status: DECLINED (⇔非 COMPLETED)
```

そのため、決済成功を HTTP レスポンスコードが 2xx 台か、や、レスポンスボディの status が COMPLETED であるかだけで判定すると、正しく判定できないため、

purchase_units.payments.captures.status: COMPLETED となっているかも判定するようご注意ください。

なお、PayPal アカウント決済の場合は、Authorize もしくは Capture で却下されると HTTP ステータスは 422 エラーが返されます。この点の挙動がクレジットカード決済と異なります。

サンプルリクエスト

```
curl -v -X POST 'https://api-m.sandbox.paypal.com/v2/checkout/orders/{order_id}/capture' \  
-H 'Content-Type: application/json' \  
-H 'Authorization: Bearer <access-token>' \  
-H 'PayPal-Request-Id: 7c3df866-90b7-4173-83e6-f2d2b53c8480'
```

サンプルレスポンス

```
{  
  "id": "1S536099LR689001U",  
  "status": "COMPLETED",  
  "payment_source": {  
    "card": {  
      "last_digits": "8229",  
      "brand": "VISA",  
      "type": "CREDIT"  
    }  
  },  
  "purchase_units": [  
    {  
      "reference_id": "default",  
      "shipping": {  
        "name": {  
          "full_name": "太郎 青山"  
        },  
        "address": {  
          "address_line_1": "青山 1-2-3",  
          "address_line_2": "ペイパルビル 1001",  
          "admin_area_2": "港区",  
          "admin_area_1": "東京都",  
          "postal_code": "107-0001",  
          "country_code": "JP"  
        }  
      },  
      "payments": {  
        "captures": [  
          {  
            "id": "47V78187KH288311P",  
            "status": "COMPLETED",  
            "amount": {  
              "currency_code": "JPY",  
              "value": "10"  
            },  
            "final_capture": true,  
            "disbursement_mode": "INSTANT",  
            "seller_protection": {  
              "status": "NOT_ELIGIBLE"  
            },  
            "seller_receivable_breakdown": {
```

```

    "gross_amount": {
      "currency_code": "JPY",
      "value": "1000"
    },
    "paypal_fee": {
      "currency_code": "JPY",
      "value": "76"
    },
    "net_amount": {
      "currency_code": "JPY",
      "value": "924"
    }
  },
  "links": [
    {
      "href": "https://api.sandbox.paypal.com/v2/payments/captures/47V78187KH288311P",
      "rel": "self",
      "method": "GET"
    },
    {
      "href":
"https://api.sandbox.paypal.com/v2/payments/captures/47V78187KH288311P/refund",
      "rel": "refund",
      "method": "POST"
    },
    {
      "href": "https://api.sandbox.paypal.com/v2/checkout/orders/1S536099LR689001U",
      "rel": "up",
      "method": "GET"
    }
  ],
  "create_time": "2021-01-25T07:56:18Z",
  "update_time": "2021-01-25T07:56:18Z",
  "processor_response": {
    "response_code": "0000"
  }
}
]
}
},
"links": [
  {
    "href": "https://api.sandbox.paypal.com/v2/checkout/orders/1S536099LR689001U",
    "rel": "self",
    "method": "GET"
  }
]
}
}

```

※ オートソリを使用する場合は以下の「オートソリを実行する」「決済を実行する」を行う

4.2.7 オートソリを実行する

下記 `/v2/checkout/orders/{id}/authorize` を貴社サーバからコールし決済を実行します。

https://developer.paypal.com/api/orders/v2/#orders_authorize

以下のヘッダを追加してください。

"PayPal-Request-Id: <Unique-Session-Identifier>"

※注意点

クレジットカード決済（PayPal アカウント決済でない）でカード会社からの拒否エラーであった場合、API のレスポンスは以下のようになります。

[レスポンスヘッダ]

HTTP レスポンスコード：2xx 台

[レスポンスボディ]

status: COMPLETED

purchase_units.payments.authorizations.status: DENIED (⇐非 CREATED)

そのため、決済成功を HTTP レスポンスコードが 2xx 台か、や、レスポンスボディの status が COMPLETED であるかだけで判定すると、正しく判定できないため、purchase_units.payments.authorizations.status: CREATED となっているかも判定するようご注意ください。

なお、PayPal アカウント決済の場合は、Authorize もしくは Capture で却下されると HTTP ステータスは 422 エラーが返されます。この点の挙動がクレジットカード決済と異なります。

サンプルリクエスト

```
curl -v -X POST 'https://api-m.sandbox.paypal.com/v2/checkout/orders/{order_id}/authorize' \  
-H 'Content-Type: application/json' \  
-H 'Authorization: Bearer <access-token>' \  
-H 'PayPal-Request-Id: 7c3df866-90b7-4173-83e6-f2d2b53c8480'
```

サンプルレスポンス

```
{  
  "id": "5KA56930N3113540J",  
  "status": "COMPLETED",  
  "payment_source": {  
    "card": {  
      "last_digits": "8229",  
      "brand": "VISA",  
      "type": "CREDIT"  
    }  
  },  
  "purchase_units": [  
    {  
      "reference_id": "default",  
      "shipping": {  
        "name": {  
          "full_name": "太郎 青山"  
        },  
        "address": {  
          "address_line_1": "青山 1-2-3",  
          "address_line_2": "ペイパルビル 1001",  
          "admin_area_2": "港区",  
          "admin_area_1": "東京都",  
          "postal_code": "107-0001",  
        }  
      }  
    }  
  ]  
}
```

```

    "country_code": "JP"
  }
},
"payments": {
  "authorizations": [
    {
      "status": "CREATED",
      "id": "1T3898K57614G8S79",
      "amount": {
        "currency_code": "JPY",
        "value": "1000"
      },
      "seller_protection": {
        "status": "NOT_ELIGIBLE"
      },
      "processor_response": {
        "response_code": "0000"
      },
      "expiration_time": "2021-02-23T08:07:50Z",
      "links": [
        {
          "href":
"https://api.sandbox.paypal.com/v2/payments/authorizations/1T3898K57614G8S79",
          "rel": "self",
          "method": "GET"
        },
        {
          "href":
"https://api.sandbox.paypal.com/v2/payments/authorizations/1T3898K57614G8S79/capture",
          "rel": "capture",
          "method": "POST"
        },
        {
          "href":
"https://api.sandbox.paypal.com/v2/payments/authorizations/1T3898K57614G8S79/void",
          "rel": "void",
          "method": "POST"
        },
        {
          "href":
"https://api.sandbox.paypal.com/v2/payments/authorizations/1T3898K57614G8S79/reauthorize",
          "rel": "reauthorize",
          "method": "POST"
        },
        {
          "href": "https://api.sandbox.paypal.com/v2/checkout/orders/5KA56930N3113540J",
          "rel": "up",
          "method": "GET"
        }
      ],
      "create_time": "2021-01-25T08:07:50Z",
      "update_time": "2021-01-25T08:07:50Z"
    }
  ]
}
},
],

```

```
"links": [
  {
    "href": "https://api.sandbox.paypal.com/v2/checkout/orders/5KA56930N3113540J",
    "rel": "self",
    "method": "GET"
  }
]
```

※ オートソリを使用する場合は上の「オートソリを実行する」、下の「決済を実行する」を行う

4.2.8 決済を実行する

下記 /v2/payments/authorizations/{authorization_id}/capture を貴社サーバからコールし決済を実行します。

https://developer.paypal.com/api/payments/v2/#authorizations_capture

以下のヘッダを追加してください。

"PayPal-Request-Id: <Unique-Session-Identifier>"

また、以下のように金額を指定してオートソリ金額の一部金額を売上処理することが可能です。後で残りも売上処理する場合は final_capture に false を指定します。

```
{
  "amount": {
    "value": "700",
    "currency_code": "JPY"
  },
  "invoice_id": "your-own-uniqueid-for-an-order",
  "final_capture": true
}
```

サンプルリクエスト

```
curl -v -X POST 'https://api-sandbox.paypal.com/v2/payments/authorizations/1T3898K57614G8S79/capture' \
-H 'Authorization: Bearer <Access-Token>' \
-H 'Accept: application/json' \
-H 'Content-Type: application/json' \
-H 'PayPal-Request-Id: 1cc9e87b-0410-4999-9fef-76d732a94bc9'
```

サンプルレスポンス

```
{
  "id": "OVG7SW2984469872W",
  "status": "COMPLETED",
  "links": [
    {
      "href": "https://api.sandbox.paypal.com/v2/payments/captures/OVG7SW2984469872W",
      "rel": "self",
      "method": "GET"
    },
    {
      "href": "https://api.sandbox.paypal.com/v2/payments/captures/OVG7SW2984469872W/refund",
      "rel": "refund",
      "method": "POST"
    }
  ]
}
```

```

    },
    {
      "href": "https://api.sandbox.paypal.com/v2/payments/authorizations/1T3898K57614G8S79",
      "rel": "up",
      "method": "GET"
    }
  ]
}

```

4.3 クレジットカード情報の保存機能 (Vault)

現時点までに実装が完了している環境に下記の 4 つの手順を追加してください。なお、本機能を実行するためには設定が必要ですので、技術担当者に REST APP の Client ID を共有ください。

<https://developer.paypal.com/limited-release/vault-payment-methods/vault-sdk/>

4.3.1 ユーザーにカード情報の保存の有無を選択させるラジオボタンを表示する

```

<div>
  <input type="checkbox" id="vault" name="vault">
  <label for="vault">Save your card</label>
</div>

```

The screenshot shows a form with the following elements: a text input field labeled 'card number', a text input field labeled 'MM/YYYY', a text input field labeled 'security code', and a checkbox labeled 'Save your Card'.

4.3.2 ユーザーがカード情報の保存を選択したことを確認できた場合、カード情報保存のフラグを true にするロジックの追加

```

document.querySelector('#my-sample-form').addEventListener('submit', (event) => {
  event.preventDefault();
  hf.submit({

    // pass the value of the checkbox when submitting the advanced credit and debit card payments
    instance
    vault: document.querySelector('#vault').checked

  }).then(function (payload) {
    // Make a call to capture the order (payload.orderId) here
  });
});

```

なお、サンドボックスで保存できるカード番号は下記に限定されていますので、ご注意ください。ただし、4111111111111111 や 5555555555554444 などの単純なカード番号は多くの方に使用される回数が多く登録が正常にできなくなることがありますので、エラーとなる場合はリスト中の別のカード番号でテストを行ってください。

<https://developer.paypal.com/limited-release/vault-payment-methods/vault-sdk#link-testcards>

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 5206181900115543 5470432115202579	01/2025	123

4.3.3 保存されたカード情報を取得する

決済が成功するとカード情報が保存されますので、以下の方法によりその情報を取得します。レスポンスには、ユーザーID に紐付く `payment_method_token` が返却されます。 `payment_tokens.id` を以降の決済で使用しますので、保存しておきます。

カード情報の保存は決済とは非同期で実行されます。 `payment_tokens.id` が取得できなかった場合は5秒ほど空けてリトライしてください。

サンプルリクエスト

```
curl -v -k -X GET 'https://api-m.sandbox.paypal.com/v2/vault/payment-tokens?customer_id=customer_12345&total_required=true' \
-H 'Authorization: Bearer <access_token>' \
-H 'Content-Type: application/json'
```

サンプルレスポンス

```
{
  "customer_id": "customer_12345",
  "payment_tokens": [
    {
      "id": "6km842",
      "status": "CREATED",
      "source": {
        "card": {
          "brand": "VISA",
          "last_digits": "1234"
        }
      }
    }
  ],
  "links": [
    {
      "rel": "self",
      "href": "https://api.paypal.com/v2/vault/payment-tokens/6km842",
      "method": "GET"
    },
    {
      "rel": "delete",
      "href": "https://api.paypal.com/v2/vault/payment-tokens/6km842",
      "method": "DELETE"
    },
    {
      "rel": "edit",
      "href": "https://api.paypal.com/v2/vault/payment-tokens/6km842",
      "method": "PATCH"
    }
  ]
}
```

```

}
],
"links": [
  {
    "rel": "self",
    "href": "https://api.paypal.com/v2/vault/payment-
tokens?customer_id=customer_12345&page=1&page_size=5&total_required=false",
    "method": "GET"
  },
  {
    "rel": "first",
    "href": "https://api.paypal.com/v2/vault/payment-
tokens?customer_id=customer_12345&page=1&page_size=5&total_required=false",
    "method": "GET"
  },
  {
    "rel": "last",
    "href": "https://api.paypal.com/v2/vault/payment-
tokens?customer_id=customer_12345&page=1&page_size=5&total_required=false",
    "method": "GET"
  }
]
}

```

/v2/vault/payment-tokens/ API を実行すると、保存されたカードのメタ情報（カードブランド、有効期限、下 4 桁）を取得することが可能です。

サンプルリクエスト

```

curl -v -k -X GET 'https://api-m.sandbox.paypal.com/v2/vault/payment-tokens/3pqky3r' \
-H 'Authorization: Bearer <access_token>' \
-H 'Content-Type: application/json'

```

サンプルレスポンス

```

{
  "id": "3pqky3r",
  "status": "CREATED",
  "customer_id": "testuser001",
  "source": {
    "card": {
      "expiry": "2020-11",
      "brand": "VISA",
      "last_digits": "0026"
    }
  },
  "links": [
    {
      "href": "https://api.paypal.com/v2/vault/payment-tokens/3pqky3r",
      "rel": "self",
      "method": "GET",
      "encType": "application/json"
    },
    {
      "href": "https://api.paypal.com/v2/vault/payment-tokens/3pqky3r",
      "rel": "delete",

```

```
"method": "DELETE",
"encType": "application/json"
}
]
}
```

4.3.4 保存されたカード情報を削除する

保存されたカード情報を削除するには、同 API を以下のように DELETE メソッドで実行します。

サンプルリクエスト

```
curl -v -k -X DELETE 'https://api-m.sandbox.paypal.com/v2/vault/payment-tokens/3pqky3r' \
-H 'Authorization: Bearer <access_token>' \
-H 'Content-Type: application/json'
```

サンプルレスポンス

HTTP ステータスとして 204 No Content が返され、Body は空となります。

保存されたカード情報を編集する場合は、編集用の API は用意されていないため、一旦削除してから新規に保存してください。

4.4 保存したカード情報での決済

※ 継続課金については次セクションを参照してください

4.4.1 決済を作成する

下記 /v2/checkout/orders を貴社サーバからコールし決済を作成します。

Intent に CAPTURE または AUTHORIZE を設定します。intent パラメータを Capture にすると決済が確定されます。Authorize の場合には、与信のみを取得して決済の確定は行われません。

payment_source.card.vault_id に、以前保存した payment_tokens.id を指定してください。カードフォームの表示は不要です。

以下のヘッダを追加してください。

"PayPal-Request-Id: <Unique-Session-Identifier>"

サンプルリクエスト

```
curl -v -X POST https://api-m.sandbox.paypal.com/v2/checkout/orders \
-H "Content-Type: application/json" \
-H "Authorization: Bearer <access token>" \
-H 'PayPal-Request-Id: 7c3df866-90b7-4173-83e6-f2d2b53c8480' \
-d '{
  "intent": "CAPTURE",
  "application_context": {
    "return_url": "https://example.com",
    "cancel_url": "https://example.com"
  },
  "payer": {
    "name": {
```

```
"given_name": "太郎",
"surname": "青山",
},
"email_address": "buyer@example.com",
"address": {
  "address_line_1": "青山 1-2-3",
  "address_line_2": "ペイパルビル 1001",
  "admin_area_2": "港区",
  "admin_area_1": "東京都",
  "postal_code": "107-0001",
  "country_code": "JP"
},
"phone": {
  "phone_type": "MOBILE",
  "phone_number": {
    "national_number": "08034567890"
  }
}
},
"payment_source": {
  "card": {
    "vault_id": "6km842"
    "attributes": {
      "verification": {
        "method": "SCA_ALWAYS"
      }
    }
  }
}
},
"purchase_units": [
  {
    "amount": {
      "currency_code": "JPY",
      "value": "1000",
      "breakdown": {
        "item_total": {
          "currency_code": "JPY",
          "value": "1000"
        }
      }
    }
  }
],
"items": [
  {
    "name": "Sample Goods",
    "unit_amount": {
      "currency_code": "JPY",
      "value": "1000"
    },
    "quantity": "1"
  }
],
"shipping": {
  "name": {
    "full_name": "太郎 青山"
  }
},
```

```

    "address": {
      "address_line_1": "青山 1-2-3",
      "address_line_2": "ペイパルビル 1001",
      "admin_area_2": "港区",
      "admin_area_1": "東京都",
      "postal_code": "107-0001",
      "country_code": "JP"
    }
  },
  "invoice_id": "your-own-uniqueid-for-an-order",
  "custom_id": "貴社指定のカスタムパラメータ値"
}
]
}

```

サンプルレスポンス

```

{
  "id": "37905563G21548127",
  "status": "PAYER_ACTION_REQUIRED",
  "payment_source": {
    "card": {
      "name": "太郎 青山",
      "last_digits": "6764",
      "expiry": "2028-01",
      "brand": "VISA",
      "available_networks": [
        "VISA"
      ],
      "type": "CREDIT",
      "bin_details": {
        "bin": "412345",
        "bin_country_code": "JP"
      }
    }
  },
  "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"
    }
  ]
}

```

4.4.2 3D セキュア認証を実行する

"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。ユーザーはワンタイムパスワード等を入力し送信します。



処理後下記のように 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

4.4.3 3D セキュア認証結果を判定する

ユーザーが 3D セキュアの 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 セキュア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 セキュア適用の場合のみ決済受付を行い、Liablitiy_shift が POSSIBLE または YES 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/>

※ 即時決済の場合は以下の「決済を実行する」を行う

4.4.4 決済を実行する

下記 /v2/checkout/orders/{order_id}/capture を貴社サーバからコールし決済を実行します。

https://developer.paypal.com/api/orders/v2/#orders_capture

以下のヘッダを追加してください。

```
"PayPal-Request-Id: <Unique-Session-Identifier>"
```

サンプルリクエスト

```
curl -v -X POST 'https://api-m.sandbox.paypal.com/v2/checkout/orders/{order_id}/capture' \  
-H 'Content-Type: application/json' \  
-H 'Authorization: Bearer <access-token>' \  
-H 'PayPal-Request-Id: 7c3df866-90b7-4173-83e6-f2d2b53c8480'
```

※ オーソリを使用する場合は以下の「オーソリを実行する」「決済を実行する」を行う

4.4.5 オーソリを実行する

下記 /v2/checkout/orders/{id}/authorize を貴社サーバからコールしオーソリを実行します。

https://developer.paypal.com/api/orders/v2/#orders_authorize

以下のヘッダを追加してください。

```
"PayPal-Request-Id: <Unique-Session-Identifier>"
```

サンプルリクエスト

```
curl -v -X POST 'https://api-m.sandbox.paypal.com/v2/checkout/orders/{order_id}/authorize' \  
-H 'Content-Type: application/json' \  
-H 'Authorization: Bearer <access-token>' \  
-H 'PayPal-Request-Id: 7c3df866-90b7-4173-83e6-f2d2b53c8480'
```

※ オーソリを使用する場合は上の「オーソリを実行する」、下の「決済を実行する」を行う

4.4.6 決済を実行する

下記 /v2/payments/authorizations/{authorization_id}/capture を貴社サーバからコールし決済を実行します。

以下のヘッダを追加してください。

```
"PayPal-Request-Id: <Unique-Session-Identifier>"
```

また、以下のように金額を指定してオーソリ金額の一部金額を売上処理することが可能です。後で残りも売上処理する場合は final_capture に false を指定します。

```
{  
  "amount": {  
    "value": "700",  
    "currency_code": "JPY"  
  },  
  "invoice_id": "your-own-uniqueid-for-an-order",  
  "final_capture": true  
}
```

サンプルリクエスト

```
curl -v -X POST 'https://api-m.sandbox.paypal.com/v2/payments/authorizations/1T3898K57614G8S79/capture' \
-H 'Authorization: Bearer <Access-Token>' \
-H 'Accept: application/json' \
-H 'Content-Type: application/json' \
-H 'PayPal-Request-Id: 1cc9e87b-0410-4999-9fef-76d732a94bc9'
```

4.5 保存したカード情報での継続課金

4.5.1 決済を実行する

下記 `/v2/checkout/orders` を貴社サーバからコールし決済を実行します。

`payment_source.card.vault_id` に、以前保存した `payment_tokens.id` を指定してください。カードフォームの表示は不要です。

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。

以下のヘッダを追加してください。

```
"PayPal-Request-Id: <Unique-Session-Identifier>"
```

サンプルリクエスト

```
curl -v -X POST https://api-m.sandbox.paypal.com/v2/checkout/orders \
-H "Content-Type: application/json" \
-H "Authorization: Bearer <access token>" \
-H 'PayPal-Request-Id: 7c3df866-90b7-4173-83e6-f2d2b53c8480' \
-d '{
  "intent": "CAPTURE",
  "payer": {
    "name": {
      "given_name": "太郎",
      "surname": "青山"
    },
    "email_address": "buyer@example.com",
    "address": {
      "address_line_1": "青山 1-2-3",
      "address_line_2": "ペイパルビル 1001",
      "admin_area_2": "港区",
      "admin_area_1": "東京都",
      "postal_code": "107-0001",
      "country_code": "JP"
    },
    "phone": {
      "phone_type": "MOBILE",
      "phone_number": {
        "national_number": "08034567890"
      }
    }
  }
}
```

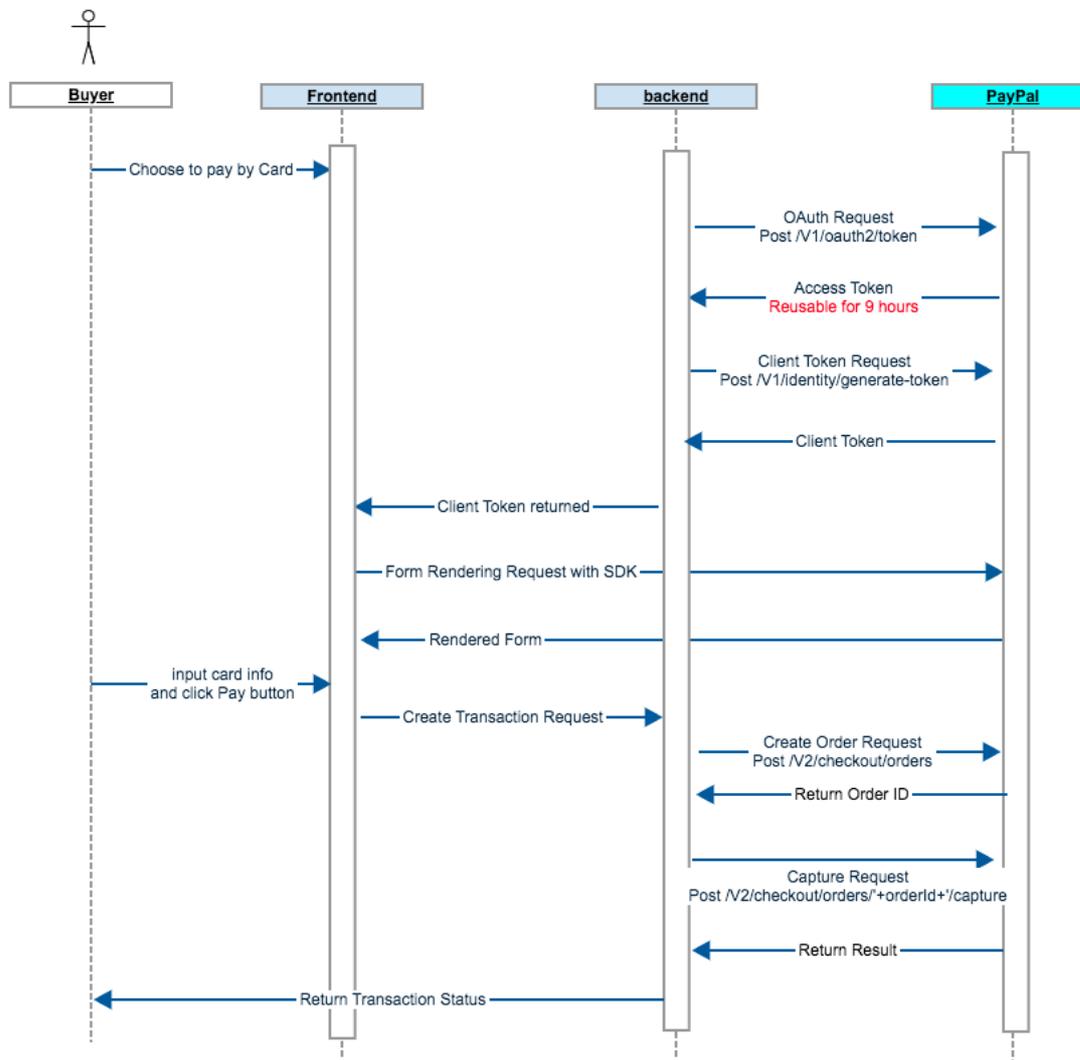
```

    }
  },
  "payment_source": {
    "card": {
      "vault_id": "6km842"
      "stored_credential": {
        "payment_initiator": "MERCHANT",
        "payment_type": "RECURRING",
        "usage": "SUBSEQUENT",
        "previous_transaction_reference": "95M63758F24067141"
      }
    }
  },
  "purchase_units": [
    {
      "amount": {
        "currency_code": "JPY",
        "value": "1000",
        "breakdown": {
          "item_total": {
            "currency_code": "JPY",
            "value": "1000"
          }
        }
      },
      "items": [
        {
          "name": "Sample Goods",
          "unit_amount": {
            "currency_code": "JPY",
            "value": "1000"
          },
          "quantity": "1"
        }
      ],
      "shipping": {
        "name": {
          "full_name": "太郎 青山"
        },
        "address": {
          "address_line_1": "青山 1-2-3",
          "address_line_2": "ペイパルビル 1001",
          "admin_area_2": "港区",
          "admin_area_1": "東京都",
          "postal_code": "107-0001",
          "country_code": "JP"
        }
      },
      "invoice_id": "your-own-uniqueid-for-an-order",
      "custom_id": "貴社指定のカスタムパラメータ値"
    }
  ]
}

```

4.6 シーケンス図

即時売り上げのケース



4.7 キャンセル・返金

4.7.1 決済のキャンセル・返金

下記 `/v2/payments/captures/{capture_id}/refund` を貴社サーバからコールし決済のキャンセル、返金を実行します。

https://developer.paypal.com/api/payments/v2/#captures_refund

PayPal アカウントの残高から返金されますので、実行するには十分な残高が必要です。返金可能な期間は取引日から 180 日間です。

金額を指定し一部金額を返金することが可能です。

4.7.2 オーソリのキャンセル・返金

下記 `/v2/payments/authorizations/{authorization_id}/void` を貴社サーバからコールしオーソリのキャンセルを実行します。

https://developer.paypal.com/api/payments/v2/#authorizations_void

4.8 処理結果

4.8.1 HTTP ステータスコード

<https://developer.paypal.com/api/rest/responses/#link-httpstatuscodes>

PayPal API コールは、HTTP ステータスコードを返します。

決済リクエストの結果は、http ステータス、決済ステータス、レスポンスコードを元に判別します。

200 系: 決済レコード (承認、および却下) が作成されています。カード会社から与信エラーなどを理由に却下された決済については、API レスポンスの `status` パラメータが `DECLINED` で返信されます。(オーソリで却下された場合は `status` パラメータが `DENIED` となります。)

<https://developer.paypal.com/api/orders/v2/#definition-capture>

400 系: 決済リクエストのバリデーションエラー。決済リクエストに問題があります。

500 系: 内部エラー。同一 PayPal-Request-ID と Invoice ID でリトライしてください。(※タイムアウトの場合もリトライ)

注意:

クレジットカード決済 (PayPal アカウント決済でない) でカード会社からの拒否エラーであった場合、API のレスポンスは以下ようになります。

[レスポンスヘッダ]

HTTP レスポンスコード: 2xx 台

[レスポンスボディ]

`status: COMPLETED`

`purchase_units.payments.captures.status: DECLINED` (⇔非 `COMPLETED`)

または

`purchase_units.payments.authorizations.status: DENIED` (⇔非 `CREATED`)

そのため、決済成功を HTTP レスポンスコードが 2xx 台か、や、レスポンスボディの `status` が `COMPLETED` であるかだけで判定すると、正しく判定できないため、`purchase_units.payments.captures.status: COMPLETED`、または `purchase_units.payments.authorizations.status: CREATED` となっているかも判定するようご注意ください。

なお、PayPal アカウント決済の場合は、Authorize もしくは Capture で却下されると HTTP ステータスは 422 エラーが返されます。この点の挙動がクレジットカード決済と異なります。

注意:

HTTP 5xx またはネットワークタイムアウトの場合でも、PayPal トランザクションが作成された可能性があります。トランザクションの結果を受けるには、以前と同じ PayPal-Request-Id HTTP ヘッダーを使用して、同じリクエストコールを少なくとも一回は繰り返すことをお勧めします。これを参照してください。

API idempotency

<https://developer.paypal.com/api/rest/reference/idempotency/>

却下の理由については、API レスポンスの `processor_response` パラメータに含まれるエラーコードを参照してください。エラーコードの一覧は下記の通りです。

Error Code	Description
0000	APPROVED
0100	REFERRAL
0800	BAD_RESPONSE_REVERSAL_REQUIRED
1000	PARTIAL_AUTHORIZATION
1300	INVALID_DATA_FORMAT
1310	INVALID_AMOUNT
1312	INVALID_TRANSACTION_CARD_ISSUER_ACQUIRER
1317	INVALID_CAPTURE_DATE
1320	INVALID_CURRENCY_CODE
1330	INVALID_ACCOUNT
1335	INVALID_ACCOUNT_RECURRING
1340	INVALID_TERMINAL
1350	INVALID_MERCHANT
1360	BAD_PROCESSING_CODE
1370	INVALID_MCC
1380	INVALID_EXPIRATION
1382	INVALID_CARD_VERIFICATION_VALUE
1384	INVALID_LIFE_CYCLE_OF_TRANSACTION
1390	INVALID_ORDER
1393	TRANSACTION_CANNOT_BE_COMPLETED
0500	DO_NOT_HONOR
5100	GENERIC_DECLINE
5110	CVV2_FAILURE
5120	INSUFFICIENT_FUNDS
5130	INVALID_PIN
5140	CARD_CLOSED
5150	PICKUP_CARD_SPECIAL_CONDITIONS. Try using another card. Do not retry the same card.
5160	UNAUTHORIZED_USER
5170	AVS_FAILURE
5180	INVALID_OR_RESTRICTED_CARD. Try using another card. Do not retry the same card.
5190	SOFT_AVS
5200	DUPLICATE_TRANSACTION
5210	INVALID_TRANSACTION
5400	EXPIRED_CARD
5500	INCORRECT_PIN_REENTER

5800	REVERSAL_REJECTED
5900	INVALID_ISSUE
5910	ISSUER_NOT_AVAILABLE_NOT_RETRIABLE
5920	ISSUER_NOT_AVAILABLE_RETRIABLE
6300	ACCOUNT_NOT_ON_FILE
7600	APPROVED_NON_CAPTURE
7700	ERROR_3DS
7710	AUTHENTICATION_FAILED
7800	BIN_ERROR
7900	PIN_ERROR
8000	PROCESSOR_SYSTEM_ERROR
8010	HOST_KEY_ERROR
8020	CONFIGURATION_ERROR
8030	UNSUPPORTED_OPERATION
8100	FATAL_COMMUNICATION_ERROR
8110	RETRIABLE_COMMUNICATION_ERROR
8220	SYSTEM_UNAVAILABLE
9100	DECLINED_PLEASE_RETRY. Retry
9500	SUSPECTED_FRAUD. Try using another card. Do not retry the same card.
9510	SECURITY_VIOLATION
9520	LOST_OR_STOLEN. Try using another card. Do not retry the same card.
9530	HOLD_CALL_CENTER. The merchant must call the number on the back of the card. POS scenario.
9540	REFUSED_CARD
9600	UNRECOGNIZED_RESPONSE_CODE
5930	CARD_NOT_ACTIVATED
PPMD	PPMD
PPCE	CE_REGISTRATION_INCOMPLETE
PPNT	NETWORK_ERROR
PPCT	CARD_TYPE_UNSUPPORTED
PPTT	TRANSACTION_TYPE_UNSUPPORTED
PPCU	CURRENCY_USED_INVALID
PPQC	QUASI_CASH_UNSUPPORTED
PPVE	VALIDATION_ERROR
PPVT	VIRTUAL_TERMINAL_UNSUPPORTED
PPDC	DCC_UNSUPPORTED
PPER	INTERNAL_SYSTEM_ERROR
PPIM	ID_MISMATCH
PPH1	H1_ERROR
PPSD	STATUS_DESCRIPTION

PPAG	ADULT_GAMING_UNSUPPORTED
PPLS	LARGE_STATUS_CODE
PPCO	COUNTRY
PPAD	BILLING_ADDRESS
PPAU	MCC_CODE
PPUC	CURRENCY_CODE_UNSUPPORTED
PPUR	UNSUPPORTED_REVERSAL
PPVC	VALIDATE_CURRENCY
PPS0	BANKAUTH_ROW_MISMATCH
PPS1	BANKAUTH_ROW_SETTLED
PPS2	BANKAUTH_ROW_VOIDED
PPS3	BANKAUTH_EXPIRED
PPS4	CURRENCY_MISMATCH
PPS5	CREDITCARD_MISMATCH
PPS6	AMOUNT_MISMATCH
PPRF	INVALID_PARENT_TRANSACTION_STATUS
PPEX	EXPIRY_DATE
PPAX	AMOUNT_EXCEEDED
PPDV	AUTH_MESSAGE
PPDI	DINERS_REJECT
PPAR	AUTH_RESULT
PPBG	BAD_GAMING
PPGR	GAMING_REFUND_ERROR
PPCR	CREDIT_ERROR
PPAI	AMOUNT_INCOMPATIBLE
PPIF	IDEMPOTENCY_FAILURE
PPMC	BLOCKED_MASTERCARD
PPAE	AMEX_DISABLED
PPFV	FIELD_VALIDATION_FAILED
PPII	INVALID_INPUT_FAILURE
PPPM	INVALID_PAYMENT_METHOD
PPUA	USER_NOT_AUTHORIZED
PPFI	INVALID_FUNDING_INSTRUMENT
PPEF	EXPIRED_FUNDING_INSTRUMENT
PPFR	RESTRICTED_FUNDING_INSTRUMENT
PPEL	EXCEEDS_FREQUENCY_LIMIT
PCVV	CVV_FAILURE
PPTV	INVALID_VERIFICATION_TOKEN
PPTE	VERIFICATION_TOKEN_EXPIRED

PPPI	INVALID_PRODUCT
PPIT	INVALID_TRACE_ID
PPTF	INVALID_TRACE_REFERENCE
PPFE	FUNDING_SOURCE_ALREADY_EXISTS
PPTR	VERIFICATION_TOKEN_REVOKED
PPTI	INVALID_TRANSACTION_ID
PPD3	SECURE_ERROR_3DS
PPPH	NO_PHONE_FOR_DCC_TRANSACTION
PPAV	ARC_AVS
PPC2	ARC_CVV
PPLR	LATE_REVERSAL
PPNC	NOT_SUPPORTED_NRC
PPRR	MERCHANT_NOT_REGISTERED
PPSC	ARC_SCORE
PPSE	AMEX_DENIED
PPUE	UNSUPPORT_ENTITY
PPUI	UNSUPPORT_INSTALLMENT
PPUP	UNSUPPORT_POS_FLAG
PPRE	UNSUPPORT_REFUND_ON_PENDING_BC

4.8.2 取引ステータス

Capture の結果は、`purchase_units.payments.captures.status` を確認してください。

COMPLETED - 売上実行済み

DECLINED - 売上却下

PARTIALLY_REFUNDED - 売上金額の一部金額が返金済み

PENDING - 売上の保留。詳細については、`status.details` を参照してください。

REFUNDED - 返金済み

Authorization の結果は、`purchase_units.payments.authorizations.status` を確認してください。

CREATED - オーソリ実行済み。売上は未実行。

CAPTURED - オーソリに対する売上が実行済み

DENIED - オーソリ却下

EXPIRED - オーソリ失効

PARTIALLY_CAPTURED - オーソリに対し、一部金額の売上が実行済み

VOIDED - オーソリ取り消し済み

PENDING - 保留。主なケースはリスク判断により一旦保留にするケースです。詳細については、`status.details` を参照してください。

なお、レスポンスの `status` の値は以下のとおりです。

CREATED - 注文が作成済み

SAVED - (今回は使用しません)

APPROVED - ユーザが PayPal 決済画面で支払いを承認済み
VOIDED - 取り消し済み
COMPLETED - オーソリまたは売上が実行済

4.8.3 Paypal-Debug-Id

レスポンスヘッダの Paypal-Debug-Id は必ず保存してください。API コールに問題などが発生した場合の調査に必要です。

4.9 エラー処理

4.9.1 擬似的にエラーのレスポンスを返す方法

エラーをシミュレートするには、Orders v2 および Payments v2 REST API 呼び出しで PayPal-Mock-Response リクエストヘッダーを使用します。

```
-H "PayPal-Mock-Response: {"mock_application_codes" : "INSTRUMENT_DECLINED"}"
```

<https://developer.paypal.com/tools/sandbox/negative-testing/request-headers/>

HTTP ステータスコード

<https://developer.paypal.com/api/rest/responses/>

Orders v2 API エラーメッセージ

<https://developer.paypal.com/api/rest/reference/orders/v2/errors/>

Payments v2 API エラーメッセージ

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

4.9.2 Sandbox 環境におけるカード却下のテスト方法

以下の方法で Sandbox における決済却下のテストを行うことが可能です。決済実行またはオーソリ実行のレスポンスで purchase_units.payments.captures.status=DECLINED、または purchase_units.payments.authorizations.status=DENIED が返ります。(個別の processor_response ごとのテストはできません。)

/v2/checkout/orders をコールする際に、payer.name.given_name を CCREJECT-REFUSED のように指定します。

```
"payer":{
  "name":{
    "given_name":"CCREJECT-REFUSED ",
    "surname":"青山"
  }
}
```

4.10 ベストプラクティス

4.10.1 クライアントサイドのバリデーション処理を追加する

入力されたカード情報の形式の正否を判定するバリデーション処理をクライアントサイドに追加することで、カード情報入力時にその形式を判定し、不要なリクエストが実行されることを防ぐことができます。例えば下記のように処理を追加することで、正しい形式でカード情報が入力されるまで、サブミットボタンを `disabled` にすることができます。

1. Card Fields の submit ボタンの初期状態を `disabled` にします。

```
<button value="submit" id="submit" class="btn" disabled>Pay with Card</button>
```

2. 下記の処理を追加し、正しい形式でカード情報が入力された時にのみ submit ボタンが活性化されるようにします。

クレジットカードフォームの入力情報のバリデーション処理

```
paypal.HostedFields.render({ /* ... */ }).then(function (cardFields) {
  cardFields.on('validityChange', function (event) {
    var state = cardFields.getState();
    var formValid = Object.keys(state.fields).every(function (key) {
      return state.fields[key].isValid;
    });
    if (formValid) {
      // enable submit button
      document.querySelector("#submit").disabled = false;
    } else {
      // disable submit button
      document.querySelector("#submit").disabled = true;
    }
  });
});
```

4.10.2 複数回のクリックイベントが発生することを制御する

ユーザーが支払いボタンを連続クリックした際に `submit` の処理が複数回発生してしまうと、リクエストの不備につながる可能性があります。クリックイベントが複数回発生しないような制御を追加いただくことを推奨します。

5 ドキュメント

PayPal デベロッパーサイト

<https://developer.paypal.com/home/>

Sandbox (テスト環境)

<https://developer.paypal.com/tools/sandbox/>

※ Developer account は日本ではご利用になれません。通常のビジネスアカウントを作成ください。

概要

<https://developer.paypal.com/api/rest/>

Advanced credit and debit card payments

<https://developer.paypal.com/docs/checkout/advanced/>

Vaulting

<https://developer.paypal.com/limited-release/vault-payment-methods/>

3D Secure

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

Orders API

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

Payments API

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

API レスポンス

<https://developer.paypal.com/api/rest/responses/>

フォームのカスタマイズ

<https://developer.paypal.com/docs/checkout/advanced/customize/card-field-style/>