

Sample Code Guide

Amazon Pay Checkout v2

2021/06/14

Contents

はじめに	1
この資料の位置づけ	1
当サンプルコードで確認できること	1
環境準備	1
サンプルコードの取得	1
設定に必要な情報の取得	3
初期設定	3
①Private Key の配置	3
②ap_sample_config.php の更新	3
③Signature の生成	4
④各 API の設定値更新	4
updateCheckoutSession.php	4
cart.html(apb.html)	4
tester_updateCheckoutSession.html	5
tester_buttonOnetime.html, tester_buttonRecurring.html, tester_buttonSignIn.html, tester_buttonAPB.html	5
⑤テストアカウントの用意	6
疎通確認	6
当サンプルコードの使い方	6

はじめに

当ガイドは 2021 年 6 月 14 日現在の Amazon Pay Checkout v2(API version2) に対しての資料となります。

正式バージョンリリース時に一部の仕様変更される可能性があります。その旨ご理解の上お取り扱いいただけますようお願いいたします。

この資料の位置づけ

この資料は動作をご理解いただくためのサンプルコードを利用する為のガイドです。別途ご用意しておりますサンプルコードと合わせてご確認ください。なお、サンプルコードは PHP の SDK を使用することを想定しており、PHP が利用できる環境をお持ちであることを前提条件といたします。PHP の SDK については、開発者向け情報ページからご取得ください。

また、V2 に関する全体の概要や、開発時のよくあるご質問 (FAQ) については、開発者向け情報ページにまとめておりますので、こちらをご参照ください。

<https://developer.amazon.com/ja/docs/amazon-pay/intro.html>

当サンプルコードで確認できること

当サンプルコードを利用することで、以下の 2 つの事を確認することができます。

- 1) 基本的な Amazon Pay の購入フローの確認
- 2) ボタンの表示・各 API リクエストをする際に必要なパラメータやリクエストの確認

1) につきましては、`cart.html(abp.html)` にアクセスすることで Amazon Pay で最も一般的な購入フローをご確認いただけます。

2) につきましては、`tester_resultPage.html` にアクセスすることで、各 API のパラメータを設定し、リクエストをテストするページに移動することができます。

まずは、次の章でサンプルコードの環境準備を行ってください。

環境準備

サンプルコードの取得

開発者向け情報ページからサンプルコードをご取得ください。

下表はサンプルコード内のフォルダとファイルの構成です。PHP が動作するテスト用の WEB サーバー等に配置してください。

```
./
  apb.html
  apb_order_confirmation.html
  apb_order_error.html
  cart.html
  composer.json
  composer.lock
  order_confirmation.html
  order_error.html
  review_order_page.html
  tester_buttonAPB.html
  tester_buttonOnetime.html
  tester_buttonRecurring.html
  tester_buttonSignIn.html
  tester_cancelCharge.html
  tester_captureCharge.html
  tester_closeChargePermission.html
  tester_completeCheckoutSession.html
  tester_createCharge.html
  tester_createCheckoutSession.html
  tester_createRefund.html
  tester_getBuyer.html
  tester_getCharge.html
  tester_getChargePermission.html
  tester_getCheckoutSession.html
  tester_getRefund.html
  tester_resultPage.html
  tester_updateChargePermission.html
  tester_updateCheckoutSession.html

—css
  common.css
  logo-pay.png
  uikit-rtl.css
  uikit-rtl.min.css
  uikit.css
  uikit.min.css

—js
  common.js
  uikit-icons.js
  uikit-icons.min.js
  uikit.js
  uikit.min.js

—keys
—php
  ap_sample_cancelCharge.php
  ap_sample_captureCharge.php
  ap_sample_closeChargePermission.php
  ap_sample_completeCheckoutSession.php
  ap_sample_config.php
  ap_sample_createCharge.php
  ap_sample_createCheckoutSession.php
  ap_sample_createRefund.php
  ap_sample_getBuyer.php
  ap_sample_getCharge.php
```

```
ap_sample_getChargePermission.php
ap_sample_getCheckoutSession.php
ap_sample_getRefund.php
ap_sample_updateChargePermission.php
ap_sample_updateCheckoutSession.php
buttonSignature.php
buttonSignatureForapb.php
buttonSignatureForcart.php
createCheckoutSession.php
result_return.php
result_return_apb.php
updateCheckoutSession.php
```

—vendor //当ディレクトリ内の構成物は省略

設定に必要な情報の取得

この後のステップで使用する以下の情報をセラーセントラルからご取得ください。

1	Public Key Id
2	Private Key
3	Store Id (別名:クライアント ID)
4	Merchant Id (出品者 ID)

取得方法の詳細について、ご不明な点がある場合は、FAQにてご確認ください。

初期設定

①Private Key の配置

Private Key を Keys ディレクトリに配置してください。

※このキーは厳重に管理し、漏洩しているリスクがある秘密キーは絶対に使わないでください。

②ap_sample_config.php の更新

ap_sample_config.php を開き、ご取得頂いた Public Key Id, Private Key を設定します。

```
<?php
    $amazonpay_config = array(
        'public_key_id' => 'XXXXXXXXXXXXXXXXXXXXXXXXX',
        'private_key' => './keys/XXXXXX.pem', //①の Private Key の名称
        'sandbox' => true,
        'region' => 'jp'
    );
?>
```

③Signature の生成

1)基本的な Amazon Pay の購入フローの確認の機能を利用するための signature を生成します。

buttonSignatureForcart.php を開き、payload の storeId に取得した値を設定してください。checkoutReviewReturnUrl, checkoutResultReturnUrl を貴社の環境に合わせて URL パスを更新してください。

今すぐ支払うフロー(APB フロー)にて実装を行う場合は、buttonSignatureForapb.php を開き、payload の storeId に取得した値を設定してください。checkoutResultReturnUrl を貴社の環境に合わせて URL パスを更新してください。

その後、php コマンドを用いて buttonSignatureForcart.php を実行し、signature の値を取得してください。

payload と signature の値は、④にて利用します。

④各 API の設定値更新

④-1: 1)基本的な Amazon Pay の購入フローの確認の機能を利用するために、以下のファイルの設定値を更新します。

updateCheckoutSession.php

updateCheckoutSession.php を開き、checkoutReviewReturnUrl に③で設定した checkoutReviewReturnUrl と同じ値を設定してください。今すぐ支払うフローの場合は、こちらの設定は不要です。

```
'webCheckoutDetails' => array(
    'checkoutResultReturnUrl' => 'https://XXXXXXX/php/result_return.php'
),
```

cart.html(apb.html)

cart.html を開き、payload と buttonsignature に③で取得した payload と signature を設定してください。今すぐ支払うフローの場合は、apb.html に対して設定をして下さい。

```
const payload = '{"storeId":"amzn1.application-oa2-client.XXXXXXX","webCheckoutDetails":{"checkoutReviewReturnUrl":"https://XXXXXXX/review_order_page.html","checkoutResultReturnUrl":" https://XXXXXXX/php/result_return.php"}}';
const buttonsignature = 'XXXXXXX';
```

merchantId に貴社の Merchant Id (出品者 ID)を設定してください。

```
merchantId: '####Set your MerchantId####',
```

createCheckoutSessionConfig の publicKeyId に貴社の Public Key Id (②で設定したもの)を設定してください。

```
createCheckoutSessionConfig: {  
  payloadJSON: payload, // payload generated in step 2  
  signature: buttonsignature, // signature generated in step 3  
  publicKeyId: '####Set your Public Key Id####'  
}
```

④-2: 2)ボタンの表示・各 API リクエストをする際に必要なパラメータやリクエストの確認の機能を利用するために、以下のファイルの設定値を更新します。

tester_updateCheckoutSession.html

tester_updateCheckoutSession.html を開き、checkoutReviewReturnUrl, checkoutResultReturnUrl に貴社の環境に合わせた URL パスを設定してください。今すぐ支払うフローの場合は、こちらの設定は不要です。

```
<td>checkoutReviewReturnUrl:</td>  
<td><input id="checkout_review_return_url" class="ap_textbox" style="width:440px;" type="text"  
placeholder="https://XXXXXXX/tester_resultPage.html "  
value="https://XXXXXXX/tester_resultPage.html"></td>  
  
<td>checkoutResultReturnUrl:</td>  
<td><input id="checkout_result_return_url" class="ap_textbox" style="width:440px;" type="text"  
placeholder="https://XXXXXXX/tester_resultPage.html "  
value="https://XXXXXXX/tester_resultPage.html"></td>
```

tester_buttonOnetime.html, tester_buttonRecurring.html, tester_buttonSignIn.html, tester_buttonAPB.html

ご自身が確認したい tester_buttonXXXXX.html を開き、merchantId に貴社の Merchant Id (出品者 ID)を設定してください。

```
<td>merchant_id:</td><td><input id="merchant_id" class="ap_textbox" style="width:120px;"  
type="text" placeholder="####Your MerchantId####" value="####Your MerchantId####"></td>
```

payload に貴社の環境に合わせた情報(StoreId, URL パス, Scopes など)を設定してください。

例：tester_buttonOnetime.html の場合

```
const payload = '{"storeId":"amzn1.application-oa2-  
client.XXXXXXX","webCheckoutDetails":{"checkoutReviewReturnUrl":"https://XXXXXXX/tester_resultP  
age.html","checkoutResultReturnUrl":"https://XXXXXXX/tester_resultPage.html"},"scopes":["name","e  
mail","phoneNumber","postalCode","shippingAddress","billingAddress"]}';
```

createCheckoutSessionConfig の publicKeyId に貴社の Public Key Id (②で設定したもの) を設定してください。(tester_buttonSignIn.html の場合は signInConfig になります。)

```
createCheckoutSessionConfig: {  
  payloadJSON: payload, // payload generated in step 2  
  signature: buttonsignature, // signature generated in step 3  
  publicKeyId: '####Set your Public Key Id####'  
}
```

⑤テストアカウントの用意

動作を確認するためにサンドボックス環境のテスト用購入者アカウントを用意します。

テストアカウントの作成方法が不明な場合は FAQ にてご確認ください。

疎通確認

①～⑤の設定が完了しましたら、[cart.html\(apb.html\)](#)をブラウザなどで開いてください。

以下のフローで、Amazon Pay の購入フローが完了できましたら、設定は完了となります。

- 1.Amazon Pay のボタンが表示されたら、それをクリックする。
 - 2.サンドボックス環境のテスト用購入者アカウントでサインインし、支払方法などを変更せずに処理を進める。
 - 3.review_order_page.html ページに遷移した後に「Place Order」(購入)ボタンを押す。(今すぐ支払うフローの場合、確認ページを挟まない為当ステップはありません。)
 - 4.Amazon Pay のプロセスページにリダイレクトした後に order_confirmation.html ページが表示される。(今すぐ支払うフローでは、apb_order_confirmation.html ページ)
- ※うまく行かない場合は、初期設定の修正忘れや誤りがないか点検してください。合わせてブラウザのコンソールにどのようなエラーが出力されているか確認してください。

当サンプルコードの使い方

1)基本的な Amazon Pay の購入フローの確認

→[cart.html\(apb.html\)](#)をブラウザなどで開いてください。

前述の疎通確認と同じ手順でフローの確認にご利用ください。

2)ボタンの表示・各 API リクエストをする際に必要なパラメータやリクエストの確認

→[tester_resultPage.html](#) をブラウザなどで開いてください。

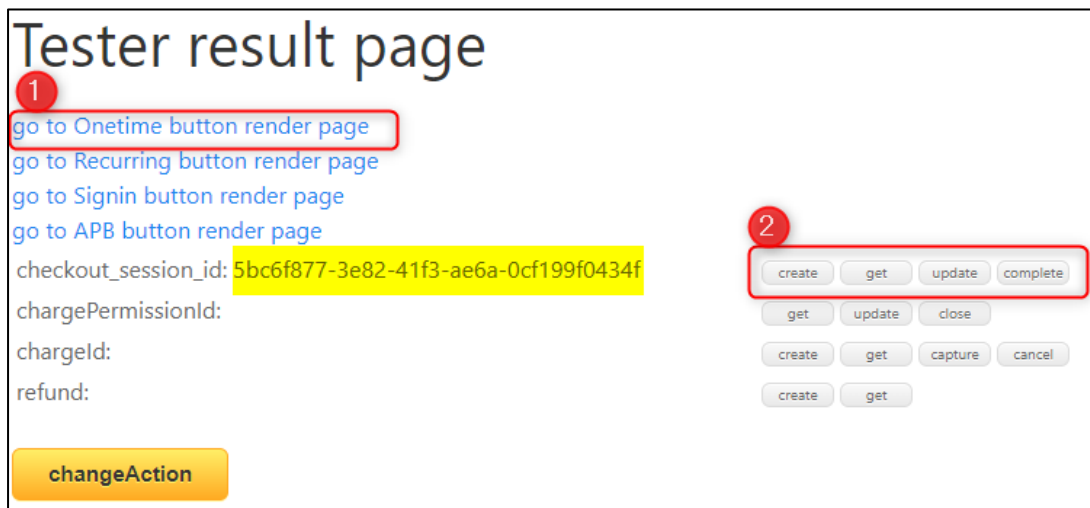
当サンプルコードでは、すべての API に対応するドライバページ(`tester_XXXXXXX.html`)を用意しています。それぞれの API の動きをドライバページからご確認ください。

以下の流れで、ボタン表示 (CheckoutSessionId 取得) から CompleteCheckoutSession までの操作を行っていただけます。

※以降は OneTime 機能を事例として記載します。他の機能を試したい場合も流れは同様となるため、読み替えてご覧ください。

go to Onetime button render page リンク(下図①)から `tester_buttonOnetime.html` ページに遷移し、Button Render→テストアカウントでログインをしてください。

CheckoutSessionId を取得し、`tester_resultPage.html` に戻ってきます。



次に、`checkout_session_id` の行 (上図②) にある「update」ボタンを押して `tester_updateCheckoutSession.html` を開いてください。

Sample Code Guide (Amazon Pay Checkout v2)

checkoutSessionId:	<input type="text" value="8096ba89-dbac-4258-a25d-7455d69712ce"/>
checkoutReviewReturnUrl:	<input type="text" value="https://XXXXX/tester_resultPage.html"/>
checkoutResultReturnUrl:	<input type="text" value="https://XXXXX/tester_resultPage.html"/>
paymentIntent:	<input type="text" value="Authorize"/> Confirm / Authorize / AuthorizeWithCapture
canHandlePendingAuthorization:	<input type="text" value="false"/>
amount:	<input type="text" value="123"/>
currencyCode:	<input type="text" value="JPY"/>
totalOrderAmount(Only MultiAuth):	<input type="text" value="234"/>
totalOrdercurrencyCode(Only MultiAuth):	<input type="text" value="JPY"/>
softDescriptor:	<input type="text" value="softDescriptor"/>
merchantReferenceld:	<input type="text" value="2020-00000001"/>
merchantStoreName:	<input type="text" value="Store Name"/>
noteToBuyer:	<input type="text" value="Thank you for your order!"/>
customInformation:	<input type="text" value="custom information"/>
providerReferenceld:	<input type="text" value="234e234dwde23"/>
platformId:	<input type="text" value="platformID"/>
<input type="button" value="updateCheckoutSession"/>	

金額等を設定して CheckoutSession を更新することができます。この更新で必要項目が揃うと画面下部の webCheckoutDetails.amazonPayRedirectUrl の行に次のような URL が表示されるようになります。

```
https://apay-us.amazon.com/checkout/processing?amazonCheckoutSessionId=39137b37-afce-45b1-bb6f-8971a1e2baaf
```

上記の URL をブラウザなどで開くと、updateCheckoutSession の設定に応じて、オーソリ等の処理が進みます。

最後に、checkout_session_id の行（上図②）にある「complete」ボタンを押して tester_completeCheckoutSession.html を開いてください。

checkoutSessionId:	8096ba89-dbac-4258-a25d-7455d69712ce
amount:	123
currencyCode:	JPY
totalOrderAmount(Only MultiAuth):	123
totalOrderCurrencyCode(Only MultiAuth):	JPY
completeCheckoutSession	

updateCheckoutSession と同じ金額を amount に設定し、completeCheckoutSession を実行します。この結果、ChargePermissionId と ChargeId が取得できるようになりますので、tester_resultPage.html から同じ要領で各種 API を操作してください。

※API のリクエストを行う順番やインターフェースにつきましては、開発者向け情報ページからインテグレーションガイド・FAQにてご確認ください。