

amazon pay

Amazon Pay
Code Generatorの使用方法
Date 2017.12.08



Introduction

Amazon PayをECサイトに導入する際に、クライアント側ではログインボタンとアドレス帳/お支払いウィジェットを実装いただく必要があります。

実装を支援するためのツールとして、Code Generatorというサンプルコード自動生成ツールを提供しておりますので、ご利用ください。

なお、本ツールはサンプルコードを提供するもので、コードのサポートはさせていただきますが、動作を保証するものではございませんので、ご注意ください。

> Code Generatorへのアクセス方法

<https://pay.amazon.com/jp/developer/documentation>にアクセスすると、下記の通りCode Generatorへのリンクがございますので、そちらをクリックください。

The screenshot shows the Amazon Pay developer documentation page. The navigation bar includes links for 'amazon pay', '販売事業者様', '認定制度', 'イベント', 'ニュース', 'サポート', '購入者様', 'サインイン', '販売事業者様 | 購入者様', and '販売事業者様お申し込み'. The main content is organized into three sections:

- インテグレーションガイド** (Integration Guide): Contains four cards for 'ワンタイムペイメント', '定期支払い', 'Amazon Pay APIリファレンス', and 'Amazon Pay よくあるご質問'.
- SDKとサンプル** (SDK and Samples): Contains five cards for 'php', 'Java', 'Ruby', 'python', and 'C#'.
- 開発補助ツール** (Development Support Tools): Contains four cards: 'Scratch Pad', 'Code Generator', 'セルフチェックシート', and 'ユーザーエクスペリエンスガイド～実装例集～'. The 'Code Generator' card is highlighted with a red dashed box.

The 'Code Generator' card description reads: 'サンプルコードを自動生成するツールです。' (A tool that automatically generates sample code.)

Code Generatorの画面構成

Code Generatorは、下記の通り構成されております。

The screenshot shows the Amazon Pay Widgets Code Generator interface. It is divided into several sections:

- 1:** A "Button Preview" area at the top left.
- 2:** A "Language" dropdown menu at the top right, currently set to "日本語".
- 3:** The "クライアント/パラメータ" (Client/Parameters) section, containing fields for "セラーID" (your_seller_merchant_id) and "クライアントID" (your_client_id), and a "リージョン" (Region) dropdown set to "US".
- 4:** The "ウィジェットボタン/パラメータ" (Widget Button/Parameters) section, containing options for button color (Gold, Light gray, Dark gray), button type (Login with Amazon, Login, Pay with Amazon, Pay, A), button size (small, medium, large, x-large), display targets (Login button, Address bar widget, etc.), scope (Profile, Widget), and popup options.
- 5:** The code generation area on the right, showing HTML and JavaScript code for the widget.

No	説明
1	Amazonログインボタンがどのように表示されるか確認します。
2	表示される言語を選択します。
3	セラーセントラルの各種IDを設定します。
4	ボタンやウィジェットの見た目や挙動について、オプションを選択します。
5	No.3, No.4で設定された値を元に、HTML/JavaScriptコードが自動生成されます。

➤ No.3 クライアントパラメータ

セラーセントラルから参照可能な各種IDを設定します。

クライアントパラメータ

セラーID

your_seller_merchant_id **1**

クライアントID

your_client_id **2**

リージョン

US UK DE JP **3**

No	説明
1	「概要 - 出品者ID」を設定
2	「Login with Amazonアカウント情報 - クライアントID」を設定
3	リージョンを選択します。（日本⇒JP）

➤ No.4 ウィジェットボタンパラメータ

ボタンやウィジェットの見た目や挙動について、オプションを選択します。

ウィジェットボタンパラメータ

ボタンの色 **1**

Gold
 Light gray
 Dark gray

表示対象

ログインボタン **2**
 アドレス帳ウィジェット
 お支払い方法ウィジェット
 定期支払同意ウィジェット

ウィジェットサイズ

レスポンシブ
 スマートフォン **3**
 カスタム

ボタンの種類 **4**

Amazonアカウントでログイン
 Amazonアカウントでお支払い

スコープ **5**

Profile
 Widget
 Shipping address
 Billing address

リダイレクト先URL/Javascript関数 **6**

URL **6**
 Javascript

callback.html **10**

ボタンサイズ **7**

small
 medium
 large
 x-large

POPUP **8**

Popup

実装種別 **9**

ワンタイムペイメント
 定期支払い

Script async 属性

Asynchronous

カスタム

Width **4**

400

Height **4**

228

No	説明
1	ログインボタンの色
2	ソースコードを生成したい対象 (「定期支払同意ウィジェットは、実装種別「定期支払い」のみ表示)
3	ウィジェットのサイズ指定 レスポンシブ：画面サイズに応じてサイズ変更 スマートフォン：Smart Phone向け カスタム：手入力でWidthとHeightを指定。(単位：px)
4	ログインボタンに表示される文言
5	取得したい情報スコープ(基本的に変更不要) ※Billing address=クレジットカードに紐づく請求先住所を 取得する場合は、別途Amazon側で設定が必要のため依頼要
6	ログイン成功後の遷移後に呼び出されるURL、またはJavascript ※失敗時も同様
7	ボタンサイズ
8	ログイン画面がポップアップするか、自画面遷移するか
9	ワンタイム(都度購入)か、定期支払いを行うか
10	AmazonのJavascript(Widget.js)を同期型で読み込むか 非同期型で読み込むか(非同期型=Asynchronous推奨)

➤ No.5 HTML/JavaScriptコード自動生成

自動的に生成されたソースコードを確認、コピー可能です。

```
<html>
<body>
  <div id="loginButton"><button type="button" name="button" id="Logout">Logout</button>
  <div id="loginWidgetDiv" style="height:250px"></div>
  <div id="walletWidgetDiv" style="height:250px"></div>
  <div id="consentWidgetDiv" style="height:250px"></div>
  <script type='text/javascript'>
    // get access token
    function getURLParameter(name, source) {
      return decodeURIComponent((new RegExp('[?]&[#]' + name + '=' +
        '([^\&];+?)(\&|#|;|$)').exec(source) || [, ""])[1].replace(/\+/g, '%20')) || null;
    }

    var accessToken = getURLParameter("access_token", location.hash);
    if (typeof accessToken === 'string' && accessToken.match(/^Atza/)) {
      document.cookie = "amazon_login_accessToken=" + accessToken + ";path=/;secure";
    }

    window.onAmazonLoginReady = function() {

      amazon.Login.setClientId("your_client_id");
    };

    window.onAmazonPaymentsReady = function() {
      showLoginButton();
      showAddressBookWidget();
    };

    document.getElementById('Logout').onclick = function() {
      amazon.Login.logout();
      document.cookie = "amazon_login_accessToken=; expires=Thu, 01 Jan 1970 00:00:00 GMT; path=/";
      window.location.href = '/';
    };
  </script>
  <script type="text/javascript"
    src="https://static-fe.payments-amazon.com/OffAmazonPayments/jp/sandbox/lpa/js/widgets.js"
    async></script>

  <script type='text/javascript'>
    function showLoginButton() {
      var authRequest;
      OffAmazonPayments.Button("AmazonPayButton", "your_seller_merchant_id", {
        type: "PWA",
        color: "Gold",
        size: "medium",

```

Select HTML

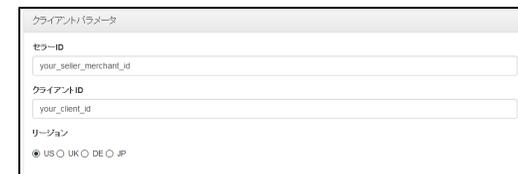
2

No	説明
1	自動生成されたソースコード (クライアントパラメータ、ウィジェットボタンパラメータのオプションを変更することで動的にソースコードが変化)
2	自動生成されたソースコードを、全て選択します。 その後、コピー&ペーストして、テキストに保存、もしくはご自身のソースコードに組み込んでください。

➤ 動作確認方法（例）

例として、カート画面([cart.html](#))にログインボタン配置、注文確認画面([callback.html](#))とした場合、下記手順を実施することで、動作確認が可能となります。

①クライアントパラメータを設定
（セラーセントラルを参照）



②【カート画面用】ウィジェットボタンパラメータを設定
⇒「表示対象」は、「ログインボタン」のみ。
「リダイレクト先URL」に、「[callback.html](#)」を設定



③生成されたソースコードを、[cart.html](#)に貼り付けて保存

④【注文確認画面用】ウィジェットボタンパラメータを設定
⇒「表示対象」は、「ログインボタン」以外の必要な
ウィジェット。

⑤生成されたソースコードを、[callback.html](#)に貼り付けて保存

⑥[cart.html](#)と[callback.html](#)を、HTTP(s)でアクセス可能なサーバに配置
（セラーセントラルの「ウェブ設定」→「JavaScriptの種類」にドメインを設定）
⇒「POPUP」パラメータがTRUEの場合

> 動作確認方法 (例)

- ⑦ HTTP(s)で、
`cart.html`にアクセス

