

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ないので、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', 'SDKとサンプル', and '開発補助ツール'. The '開発補助ツール' section 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, and the entire '開発補助ツール' section is also highlighted with a red dashed box.

**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 販売事業者様お申し込み

### インテグレーションガイド

- ワンタイムペイメント**  
Amazon ログインとAmazon Payを導入するためのガイドです。
- 定期支払い**  
Amazon Payの定期支払い(Auto Pay機能)について説明したガイドです。
- Amazon Pay APIリファレンス**  
Amazon Payで利用できるAPIのリファレンスです。
- Amazon Pay よくあるご質問**  
インテグレーションに関してよくいただく質問をまとめました。(PDF)

### SDKとサンプル

- php
- Java
- Ruby
- python
- C#

### 開発補助ツール

- Scratch Pad**  
APIへのRequestや、APIからのResponseを試せるツールです。
- Code Generator**  
サンプルコードを自動生成するツールです。
- セルフチェックシート**  
テストケースの事前確認と実装レベルの確認にご利用ください。
- ユーザーエクスペリエンスガイド～実装例集～**  
Amazon Payの導入にあたって、より深い導入効果が期待できる推奨実装事例を紹介いたします。

## > 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 input 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 radio buttons for button colors (Gold, Light gray, Dark gray), types (Login with Amazon, Login, Pay with Amazon, Pay, A), and sizes (small, medium, large, x-large). It also includes checkboxes for display targets (Login button, Address bar widget, etc.) and scopes (Profile, Widget).
- 5**: A code editor area on the right showing the generated HTML and JavaScript code. A "Select HTML" button is visible at the top right of this area.

| 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

1 ボタンの色

2 表示対象

3 ウィジェットサイズ

4 ボタンの種類

5 スcope

6 リダイレクト先URL/Javascript関数

7 ボタンサイズ

8 POPUP

9 実装種別

10 Script async 属性

ボタンの色

Gold

Light gray

Dark gray

表示対象

ログインボタン

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

カスタム

ボタンの種類

Amazonアカウントでログイン

Amazonアカウントでお支払い

ボタンサイズ

small

medium

large

x-large

POPUP

Popup

実装種別

ワンタイムペイメント

定期支払い

Script async 属性

Asynchronous

スcope

Profile

Widget

Shipping address

Billing address

リダイレクト先URL/Javascript関数

URL

Javascript

callback.html

カスタム

Width

400

Height

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<br>(「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<br>レスポンシブ：画面サイズに応じてサイズ変更<br>スマートフォン：Smart Phone向け<br>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)<br>※Billing address=クレジットカードに紐づく請求先住所を<br>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<br>※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<br>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<br>(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。<br>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。 |

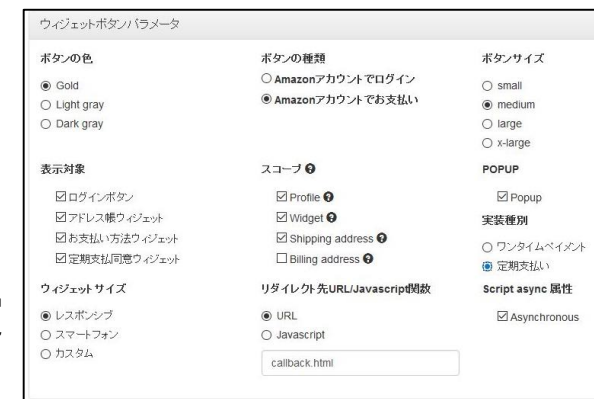
## ➤ 動作確認方法（例）

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

