

Amazon Pay
定期購入機能 インテグレーションガイド

amazon pay

Document Version 1.1.0 JP

Amazon Pay
Recurring Integration Guide

Copyright © 2017 Amazon.com, Inc. or its affiliates. All rights reserved.

目次

イントロダクション.....	8
重要な前提条件.....	8
Amazon Pay はどのように動作しますか?	9
キーコンセプト.....	9
ボタンウィジェット.....	9
アクセストークン.....	12
Amazon Pay オブジェクト.....	12
ウィジェット.....	13
アプリケーションプログラムインターフェイス (API)	15
インスタント支払通知 (IPN)	15
購入者エクスペリエンスの概要.....	16
Amazon ログインの購入者エクスペリエンス.....	16
Amazon Pay の購入者エクスペリエンス.....	21
インテグレーションの概要.....	26
購入フローでの利用.....	26
支払処理	28
必要条件	30
Amazon Pay を Web サイトに設定.....	30
Web サイトをアプリケーションに追加.....	31
Amazon Pay ウィジェットを利用するための準備.....	32
ベストプラクティス.....	33
処理前のデータエンコード.....	33
非同期 JavaScript	33
Step1：購入者認証用のボタンウィジェット追加	34
ボタンウィジェットパラメータ	35
YOUR_MERCHANT_ID.....	35
ENTER_TYPE_PARAMETER.....	35
ENTER_COLOR_PARAMETER	36
ENTER_SIZE_PARAMETER.....	36
ENTER_SCOPES	37
ENTER_POPUP_PARAMETER	38

YOUR_REDIRECT_URL.....	39
エラーハンドリングコード.....	40
ログアウトオプションの追加.....	40
Step2 : 個人情報の取得 (オプション)	40
サンプル	41
PHP サンプル.....	41
Java サンプル.....	42
Step3 : アドレス帳とお支払い方法ウィジェットの追加.....	42
Amazon Pay から配送先住所を取得.....	47
Step4 : リダイレクト認証エクスペリエンスの利用 (オプション)	48
処理方法.....	48
クロスサイトリクエストフォージェリ	49
購入者が拒否した場合.....	50
Step5 : 定期支払での購入者認証の取得	51
Step6 : 選択された配送先住所と支払方法の表示 (オプション)	53
Step7 : BillingAgreement の詳細のセットと同意の確認	55
Billing Agreement の詳細をセット.....	56
Billing Agreement Details の確認.....	57
定期支払を確認するための購入者メールのセットアップ.....	59
Step8 : Billing Agreement の確認	59
Step9 : オーソリのリクエスト	60
オーソリオブジェクト.....	61
Step10 : オーソリステータスの取得	61
オーソリの保持期間.....	62
Step11 : 売上請求のリクエスト	62
売上請求オブジェクト.....	62
Step12 : 売上請求ステータスの取得	63
売上請求成功の購入者メール.....	64
Step13 : 返金のリクエスト (オプション)	64
返金オブジェクト.....	64
Step14 : 返金ステータスの取得 (オプション)	65
返金が成功した場合の購入者メール.....	65

提供可能な購入者向けのメールコンテンツ	65
AuthorizeOnBillingAgreement 処理の同期と非同期モード	66
非同期モード	67
同期モード	68
アドレス帳ウィジェット無しのインテグレーション	69
スマートフォン用の折りたためるウィジェットの利用	70
高度な支払シナリオの可能性	72
アップグレードまたはダウングレードサービス	72
キャンセルまたはサスペンドサービス	73
支払方法の変更	74
Amazon Pay の Web サイトから支払方法を変更	74
サイトから支払方法を変更	74
配送先住所の変更	75
ワンタイムと定期の同時支払	75
支払失敗のハンドリング	76
オーソリの失敗	76
同期モード	76
非同期モード	78
売上請求の失敗	79
返金の失敗	79
システムと Amazon Pay の同期	80
Amazon Pay オブジェクト状態の遷移	80
API 呼び出しの最初のレスポンス	80
インスタント支払通知メッセージハンドリングのベストプラクティス	81
取得 API 呼び出しの実行	81
セキュアなインスタント支払通知 (IPN) 処理	81
Amazon Pay オブジェクト状態遷移のモニタリング	81
インスタント支払通知 (IPN) のサンプル	81
SANDBOX 環境でのインテグレーションテスト	89
テスト購入者アカウント	89
SANDBOX と本番環境の違い	90
SANDBOX シミュレーション	90

エラーハンドリング	97
Amazon Pay API 処理からのエラーハンドリング	97
Amazon Pay ウィジェットからのエラーハンドリング	99
ボタンギャラリー	103
スタイルガイドライン	103
サイトに Amazon アカウントでお支払いボタンを追加	103
ボタンのガイダンスを追加	103
その他のサインインインテグレーションの提供	104
顧客アカウントを Amazon とリンク	105
サイトに Amazon アカウントでお支払いボタンを追加	105
商品詳細ページに Amazon Pay を追加	106
ゲスト購入後に顧客にログインを尋ねる	107

変更履歴

バージョン	変更日	変更内容
1.0.0 JP	2016/08/06	インテグレーションガイドとして 1.0.0 を新規作成
1.0.1 JP	2016/10/11	誤字修正、および、英語表記の統一
1.0.2 JP	2016/10/13	Widgets.js URL の誤字修正
1.0.3 JP	2016/12/16	P105 の profile:postal_code の表記ミスを修正し、postal_code に修正
1.1.0 JP	2017/02/22	「Amazon ログイン&ペイメント」、および、「Amazon ペイメント」の名称変更により「Amazon Pay」に修正

イントロダクション

当資料は Amazon Pay US 環境の情報を翻訳したものです。仕様上は日本環境も同じですが、画像、および、サンプルコードの一部は US 環境のままで記載しております。適宜 Amazon Pay 日本環境に読み替えて頂く必要があります。

Amazon Pay には、通常購入と定期購入の2つのインテグレーション方法があります。当資料は定期購入機能について説明しております。通常購入については「*Amazon Pay* インテグレーションガイド」を参照してください。また、定期購入機能は自動払い機能の1つになります。*Amazon Pay* 定期購入機能は自動支払の1つの利用形態になります。

Amazon Pay は、Amazon のお客様が、セキュアで安心して便利にお買い物ができるログインと支払方法を販売事業者のサイトに提供します。購入者は Amazon ログインを利用してプロフィール情報（名前、メールアドレス、郵便番号）を共有し、Amazon アカウントに登録されている完全な配送先住所と支払方法にアクセスできます。

このガイドは、定期販売の定期支払または他の通常支払のシナリオの利用方法の詳細を提供します。Amazon Pay の API ベースのインテグレーションを利用して、支払ニーズに対処するためにより大きな柔軟性とカスタマイズを提供します。

定期支払機能では、購入者の購入のための支払を事前に許可できます。これは購入者の Amazon Pay アカウントに規則的な定期販売を請求することと、購入者が支払いするたびにオーソリ要求すること無しに請求を行うことが可能です。

注意： Amazon Pay サービスは Web ブラウザ上でのみ利用することでデザイン、および、開発されています。我々のサービスはネイティブアプリケーション（iOS、Android、RIM、Windows オペレーティングシステムを含みますがこれに限りません）の中で利用することができます。

重要な前提条件

以下の内容は Amazon Pay サービスの前提条件です。

- アプリケーションに HTML、JavaScript、XML を利用した経験がある。
- Web サービスを利用した経験がある。
- セラーセントラルについてよく知っていること。
- 支払いにおける非同期処理を扱うことができること。詳しい情報は「システムと Amazon Pay の同期」を参照してください。
- Amazon Pay 販売事業者のためのセラーID（マーチャント ID と呼ばれる）を保有していること。Amazon Pay 販売事業者アカウントは事前に Amazon Pay サービスを準備していないければなりません。

- Amazon マーケットプレイス Web サービス (AmazonMWS) のアクセスキーとシークレットキーにアクセスします。これらのキーを取得するためには、AmazonMWS に登録する必要があります。
- 購入者のブラウザは JavaScript と Cookie を利用できる必要があります。

Amazon Pay はどのように動作しますか？

Amazon Pay は存在する Web サイトに直接導入され、すべての購入者は導入された Amazon Pay のウィジェットを介します。よって、購入者はあなたのサイトから離れることはありません。

Amazon Pay サービスは密接に関連した技術要素と同一のプラットフォーム上で動作する 2 つの部分で構成されています。

- Amazon ログインはお客様が Amazon ログイン認証にログインすることであなたのお客様として登録することを許可します。それから Amazon Pay はローカルの顧客アカウントを生成するために顧客情報を転送し、実在する顧客のメールアドレスを受け取り、直接お客様にコンタクトすることができます。
- Amazon Pay から提供されたウィジェットを完全に導入されたサイトで購入処理を確立します。この Web サイト内の Amazon 購入処理では、購入者はすでに Amazon ユーザーアカウントに登録されている配送先住所と支払方法を選択します。購入者が前もって Amazon ログインを利用したかにかかわらず、ゲスト購入や購入時に会員登録するように Amazon Pay を提供することができます。後者の場合は、お客様の購入意欲から外れるかもしれません。

Amazon Pay を利用することで、購入時に多くの情報を入力することなく、数回クリックしてお客様を会員にすることができます。

キーコンセプト

エンドツーエンドの Amazon Pay エクスペリエンスは、購入者にとってワンタイム購入、または、定期購入を簡単に安全に行うようにデザインされており、支払処理は簡単で安全であります。インテグレーションを開始する前に、あなたはいくつかのこのサービスに関してのキーコンセプトに慣れている必要があります。

ボタンウィジェット

Amazon Pay をインテグレーションする場合は、サイトに購入者のためのログインや購入するためのボタンウィジェットを埋め込みます。選択したボタンウィジェットによって、サイトで購入を開始する前、途中、後から、いつでも名前、メールアドレス、郵便番号の顧客情報を入手することができます。以下の表はそれぞれのボタンウィジェットのオプションを説明します。

ボタン	ユースケース
-----	--------

Amazon ログイン



このボタンを利用してサイトに顧客がログインし、Amazon アカウントを利用して支払います。顧客は購入プロセスを開始する前、途中、後からログインすることができます。顧客は個人情報（名前、メールアドレス、郵便番号）を共有することができます、サイトで購入を完了するために Amazon アカウントに登録されている彼らの配送先と支払方法にアクセスします。

顧客が Amazon アカウント認証でログインした後

- Amazon の顧客情報として彼らの名前、メールアドレスを受け取れます。この情報は顧客のお買い物エクスペリエンスをカスタマイズするために利用できます。
- 顧客がログインしてから 60 分の間に Amazon アカウントに登録されている配送先と支払方法にアクセスできます。60 分後に認証トークンは期限切れになり、再度購入者にログインするように依頼する必要があります。
- 顧客が配送先住所を選択した時は、すぐに完全な配送先住所にアクセスできます。

このボタンは次のサンプルのように利用されるべきです。

- 顧客は注文履歴の確認や、特別な取り決めのサインアップが必要です。
ログインしてから 60 分間で、顧客は多くの商品を Amazon Pay に登録されている配送先と支払情報を利用して、2 クリックで購入することができます。
- 購入プロセスの間で顧客は購入ボタンをクリックし、Amazon ログインで認証します。それから、2 クリックのみで購入処理を完了させるために Amazon アカウントに登録されている配送先と支払情報にアクセスします。
- 顧客の注文履歴や商品の配送状況を確認するためにサイトの購入プロセスの最後で会員登録するオプションを顧客に提示します。
- 顧客はアカウント設定ページに遷移し、以前に注文された時に関連付けされた配送先と支払情報の更新を行いたい場合があります。その為には、顧客は登録されている Amazon アカウントに登録されている配送先や支払情報にアクセスし情報を更新するのに、Amazon ログインボタンを利用することができます。

顧客が Amazon アカウントでああなたのサイトに初めてログインした場合は同意画面が表示されます。同意画面はそれぞれの顧客の権限を共有するために次の情報を要求します。

	<ul style="list-style-type: none"> 顧客の名前、メールアドレス、郵便番号 注文が完了する前に選択された顧客の配送先住所 <p>顧客が情報共有を拒否した場合は、認証トークンは access_denied のエラーレスポンスメッセージを返します。顧客は Amazon Pay 経由でサイトから顧客情報を登録することができず、彼らの Amazon アカウントから支払と配送先情報にアクセスすることができません。当然、顧客は Amazon Pay 以外の顧客アカウント登録プロセスを完了することはできません。</p>
<p>Amazon Pay (ゲスト購入)</p> 	<p>顧客がトランザクションを完了する前に多くの個人情報の入力や問合せをせずに、Amazon アカウントに登録されている配送や支払情報を利用して、通常購入 (ワンタイム) または定期購入を完了するために、このボタンを利用します。</p> <p>顧客がサインインし、Amazon アカウントから配送先住所を選択した場合は、送料を計算する目的のために顧客の都道府県、国、郵便番号のみ受け取れます。</p> <p>顧客が Amazon アカウントを使用して購入プロセスを完了した後は、あなたは顧客の名前、メールアドレス、完全な配送先住所を受け取れます。顧客が購入プロセスを完了しなかった場合は、顧客情報を受け取れません。</p> <p>Amazon Pay を利用してサインインした場合は、顧客は同意画面を参照されません。</p>
<p>Amazon Pay (完全な住所にアクセス)</p> 	<p>顧客がトランザクションを完了する前に Amazon 認証を利用してサイトにログインし、トランザクションを完了する前に完全な配送先住所を共有するために、このボタンを利用します。</p> <ul style="list-style-type: none"> Amazon の顧客プロフィール情報に含まれる名前とメールアドレスを受け取れます。この情報により顧客のお買い物エクスペリエンスをカスタマイズできます。 顧客が Amazon Pay ウィジェットをサイトで利用することで、彼らがログインしてから 60 分間は、登録されている Amazon アカウントの配送先と支払情報にアクセスすることができます。60 分後に認証トークンは期限切れになり、再度顧客にログインするように依頼する必要があります。 顧客が配送先住所を選択した時は、すぐに完全な配送先住所にアクセスできます。

	<p>このボタンは次の例のように利用されるべきです。</p> <ul style="list-style-type: none"> 顧客は支払処理をカートページから開始し、彼らの Amazon アカウントに登録されている配送先と支払情報にアクセスします。 顧客は Amazon アカウントに登録されている配送先と支払情報を使い、定期支払のための配送先と支払方法を選択して注文をセットアップします。 配送するための完全な配送先住所にアクセスすることや、トランザクションを完了する前に配送料の計算をすることが必要です。 <p>顧客が Amazon アカウントでサイトに初めてサインインした場合は同意画面が表示されます。同意画面はそれぞれの顧客の権限を共有するために次の情報を要求します。</p> <ul style="list-style-type: none"> 顧客の名前、メールアドレス、郵便番号 注文が完了する前に選択された顧客の配送先住所
--	---

アクセストークン

サイトにログインした場合は、アクセストークンは認証サーバから生成されます。アクセストークンはクライアント、ユーザー、アクセススコープが特定します。クライアントは顧客プロフィールデータを獲得し配送先と支払情報へのアクセスを許可するためにアクセストークンを利用しなければなりません。

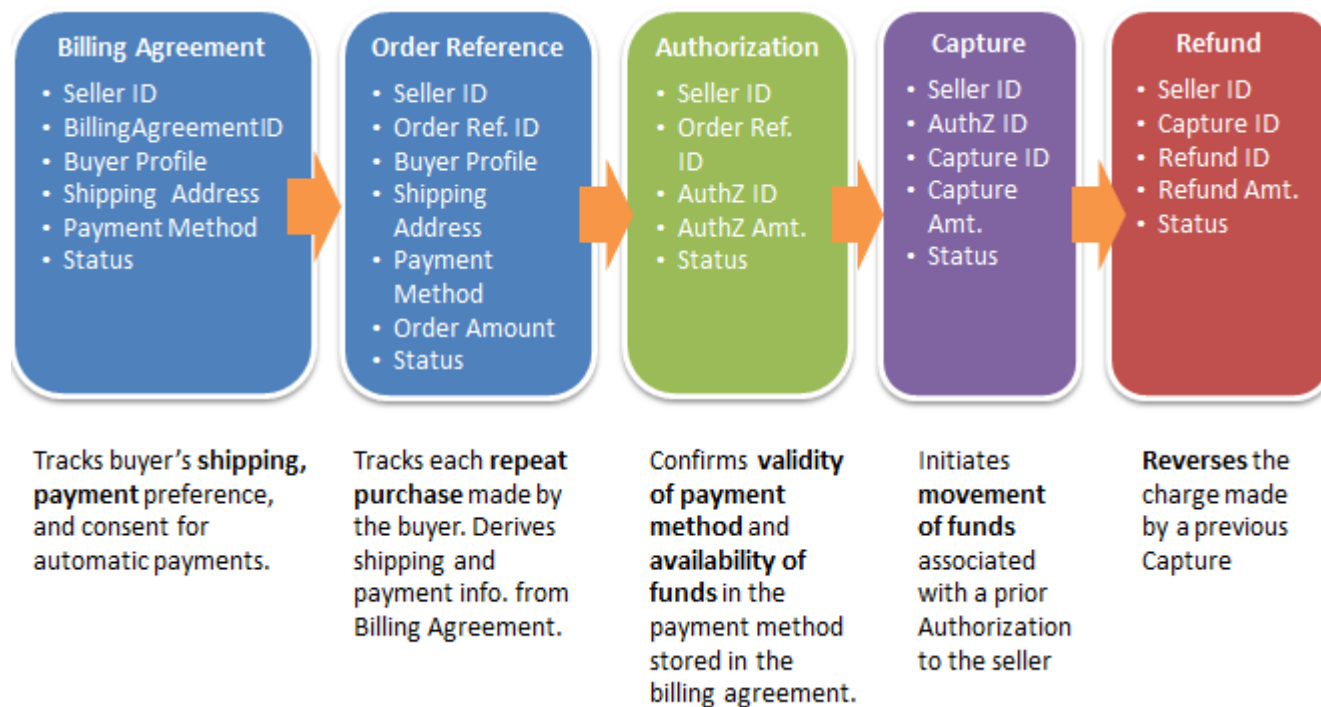
Amazon Pay オブジェクト

Amazon Pay オブジェクトは Amazon Pay をサポートするための基礎的な要素です。次のオブジェクトはこのガイドでいろいろなところで説明されます。

- Billing Agreement** オブジェクトー このオブジェクトは購入者が選択した支払方法、選択した配送先住所、定期支払の同意の記録です。購入者はこれらのいくつかの購入プロセスで選択した属性（例えば、配送先住所と支払方法）を決め、あなたは注文または定期販売の情報と一緒にいくつかの属性を設定します。
- Order Reference** オブジェクトー このオブジェクトはそれぞれの購入で購入者により作成された記録です。定期支払では、このオブジェクトは定期購入の一部としての支払を処理する場合に生成されます。支払方法と配送先住所は Billing Agreement オブジェクトから得られます。
- Authorization** オブジェクトー このオブジェクトは選択された支払方法での有効な支払を意味し、将来の売上請求する場合に確保しておくものです。

- **Capture** オブジェクト— このオブジェクトは購入者の支払方法からあなたの Amazon Pay セラーアカウントに資金が移動されることを意味します。
- **Refund** オブジェクト— 返金イベント内で、このオブジェクトは以前に売上請求した資金を購入者に戻すことを意味します。

これらのそれぞれのオブジェクトは個別の ID で識別され、ステータスを保有します。このガイドの次の章では、これらのオブジェクトの使い方について知ることができます。



ウィジェット

ウィジェットは Amazon Pay 側でコントロールしており、サイトに埋め込むことができます。ウィジェットは購入者がサイトにログインすることを可能にし、Amazon 認証を使って確認し、Amazon アカウントに登録されている住所と支払方法を参照し、購入プロセスの間で彼らが必要なものを1つ選択します。選択された住所と支払方法の情報は **Billing Agreement** オブジェクトに保存され、**Order Reference** オブジェクトにコピーされます。

Amazon Pay を導入する場合に利用できるものとして次の5つのウィジェットがあります。

- **Amazon アカウントでログイン** ボタンウィジェット：このウィジェットは購入者が購入を開始する前、途中、後で彼らの Amazon アカウントを利用して、サイトにログインすることを可能にします。Amazon Pay はこのボタンをサイト上の目立つ場所に配置することを提案します。次のイメージは **Amazon アカウントでログイン** ボタンウィジェットを表します

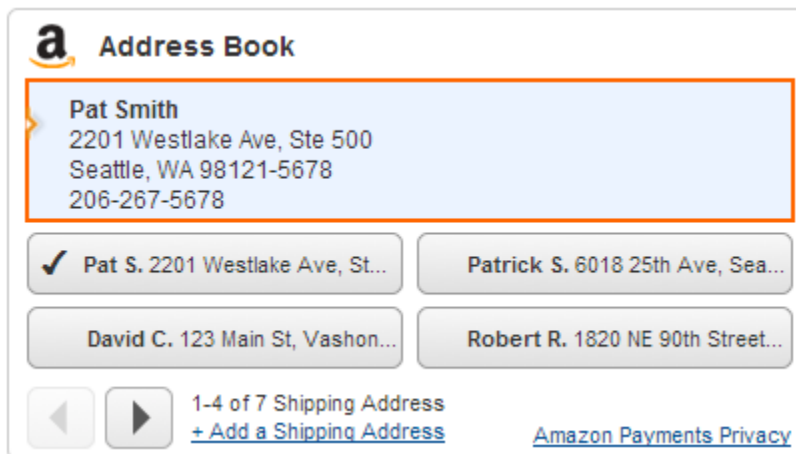
す。



- **Amazon アカウントでお支払いボタン (Amazon アカウントでログインボタンを利用して**いる場合はオプション) : これは購入者が支払を Amazon Pay に選択することと Amazon 認証で確認することが可能です。購入者が購入プロセスを開始する一般的なページ上に配置します、例えば、ショッピングカート、または、商品詳細ページです。次のイメージは Amazon アカウントでお支払いボタンウィジェットを表します。

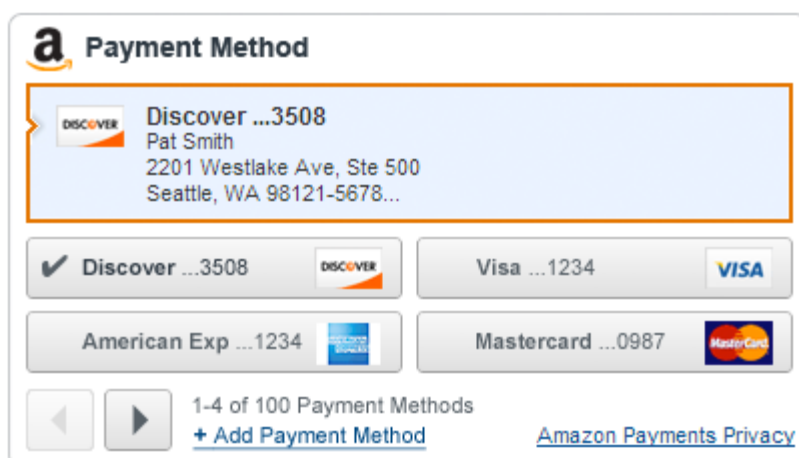


- **アドレス帳ウィジェット (オプション) :** アドレス帳ウィジェットは購入者の Amazon アカウントに登録されている住所を表示します。このウィジェットで購入者からの住所を取得し利用することで、サイトの手動入力フォームを変更することができます。次のイメージはアドレス帳ウィジェットを表します。

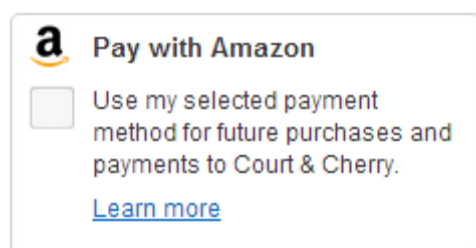


- **お支払い方法ウィジェット (必須) :** お支払い方法ウィジェットは購入者の Amazon アカウントに登録されている支払方法を表示します。このウィジェットで購入者からの支払方法を取得し利用することで、サイトの手動入力フォームを変更することができます。次の

イメージはアドレス帳ウィジェットを表します。



- **定期支払同意**ウィジェット（定期支払では必須）：定期支払同意ウィジェットは購入者が定期販売または定期請求に同意することに利用します。Amazon Payはこのウィジェットをサイトの**定期契約**または**注文する**ボタンの前に配置することを推奨します。次のイメージは**定期支払同意**ウィジェットを表します。



アプリケーションプログラムインターフェイス（API）

Amazon Pay API は、Amazon Pay と EC システム間での情報や命令をやり取りする動作を呼び出します。例えば、顧客個人情報を取得、購入者への請求を Amazon Pay へ要求、返金への対応、購入者の配送先住所の取得、Billing Agreement のキャンセルなどのためにこれらの命令を呼び出します。詳しい内容は Amazon Pay API リファレンスガイドを参照してください。

インスタント支払通知（IPN）

Amazon Pay は多くの場合、支払リクエスト（オーソリ、売上請求、返金リクエスト）を非同期方法で処理します。Amazon Pay がリクエストを処理した後でのリクエストの最終状態については、インスタント支払通知（IPN）と呼ばれる通知を受け取れます。

さらに、Amazon Pay オブジェクトのステータスはリクエストの提示された結果、もしくは、Amazon Pay の内部処理の内容によって変更することがあります。Amazon Pay オブジェクトのステータスが変更された場合は、Amazon Pay は IPN を送信し、EC システムと Amazon Pay を同期することができます。詳しくは、システムと Amazon Pay の同期を参照してください。

購入者エクスペリエンスの概要

Amazon Pay は購入者が Web サイトで彼らの Amazon アカウント認証で確認することが可能であり、サイト上でほんの数クリックでワンタイムや定期購入が完了します。

このガイドは Amazon Pay の定期支払機能を活用する方法にフォーカスします。次のサンプルは Amazon Pay を利用した定期購入商品のサインアップについての購入者エクスペリエンスのデモです。サイト上でデジタル商材の定期購入、定期支払請求を可能にするために同様のアプローチで利用することができます。

注意：下記のイメージは **SANDBOX** 環境でのデモです。**SANDBOX** はエンドツーエンドインテグレーションのテストを本番環境の準備がされるまで Amazon Pay の購入者インターフェイスを公開すること無しにすぐに開始するテスト環境です。**SANDBOX** モードでは全てのウィジェットは**"Sandbox"**または**"S"**のマークを表示します。このマークは本番環境では表示されません。

Amazon ログインの購入者エクスペリエンス

1. まだ認証されていない購入者を認証するためにサインインリンクをクリックします。

COURT & CHERRY

Get the Amazon Payments Newsletter
Enter your email address SIGN UP

SIGN IN YOUR ACCOUNT

0

Signature Blend Coffee Pods
24-count pack

With a medium body, perfectly balanced, and wonderfully smooth, this Signature Blend is perfect for any occasion, any time of day.

Made with 100% prized Arabica Beans. Pre-measured and sealed inside their own filter for maximum freshness. These 1-cup coffee pods are made from biodegradable paper making them garden-friendly.

Compatible with most coffee pod brewers. Meets standard sizing of 9-10 grams and approximately 61 mm.

\$19.00 / pack

One-time delivery

Automatic monthly delivery

Qty: 1 of 24-count pack

ADD TO CART

You might also like

Medium-roast capsules \$17 **ADD TO CART**

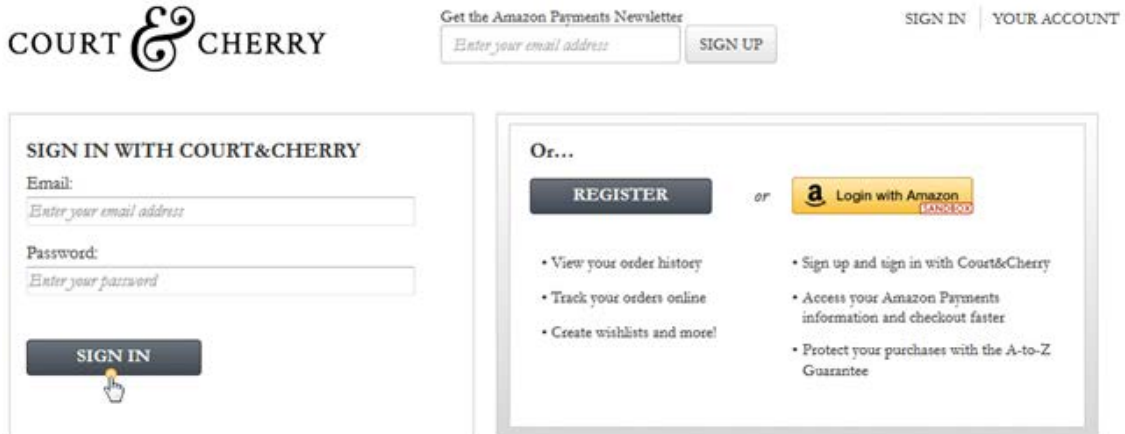
Another package type \$19 **ADD TO CART**

Whole bean Italian \$12 **ADD TO CART**

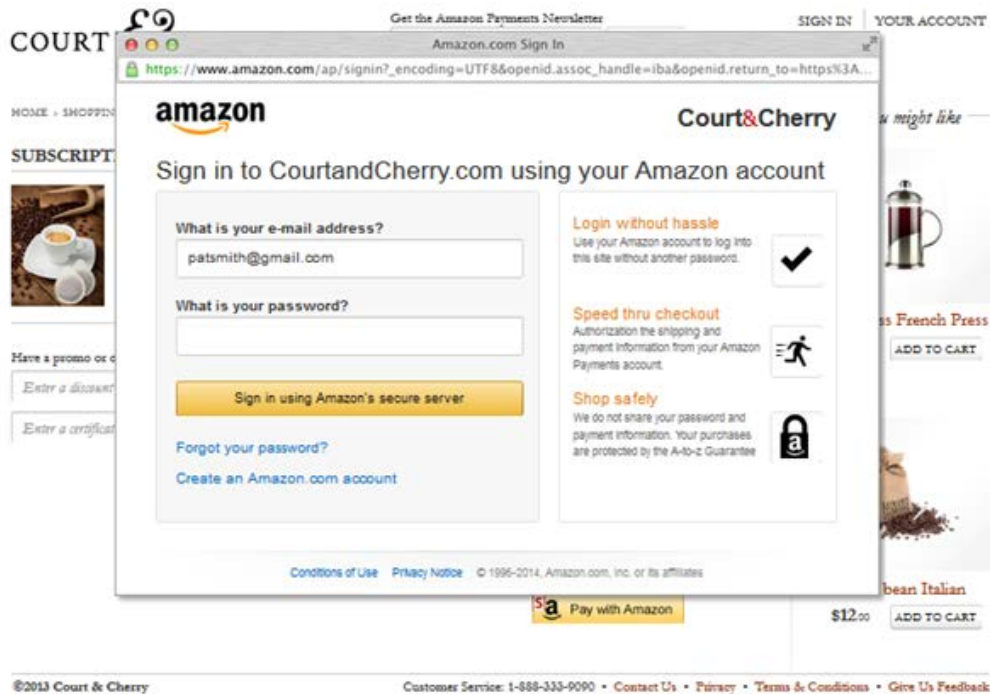
Stove-top espresso \$37 **ADD TO CART**

Stainless French Press \$31 **ADD TO CART**

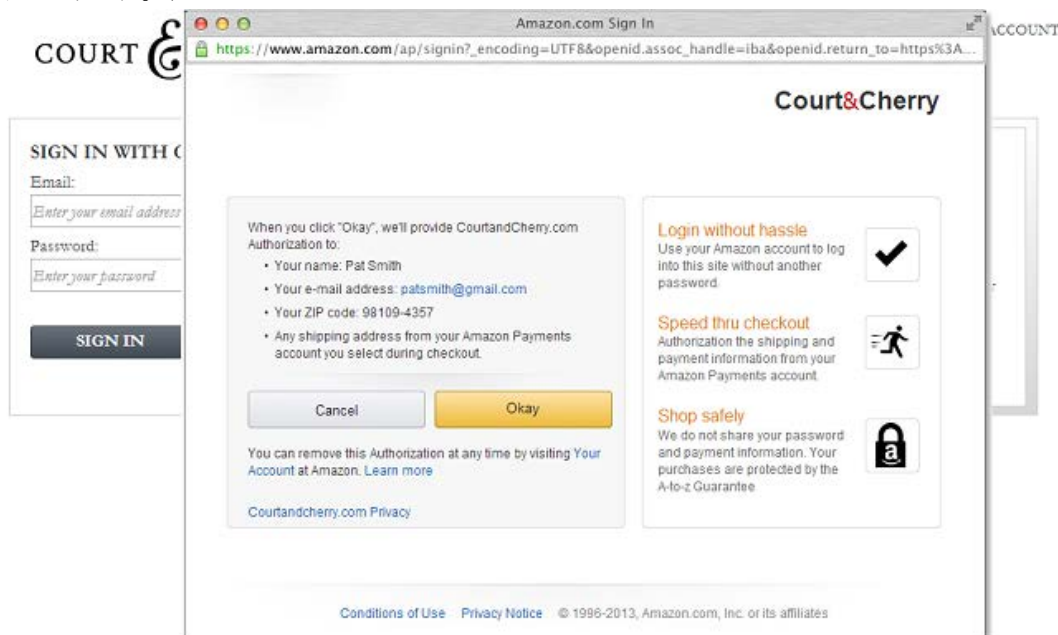
2. 購入者が Amazon アカウントを利用してサイトにログインするために Amazon アカウントでログインボタンを選択します。



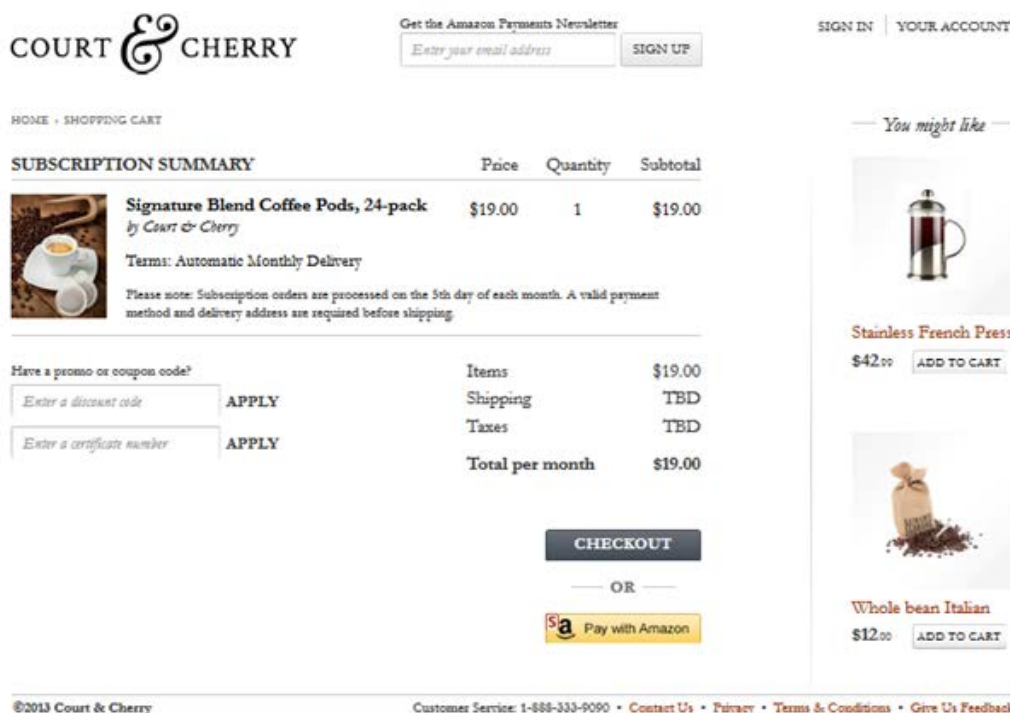
3. 購入者は Amazon のユーザー名とパスワードを利用してサインインします。



4. 購入者は情報を共有するために同意します。（彼らがサイトへ最初にログインするときのみ必要です。）



5. 購入者がサイトを参照しカートにアイテムを追加します。購入者は **Amazon アカウント** で **お支払いボタン**か**購入ボタン**のどちらかをクリックします。



6. 購入者は**アドレス帳**ウィジェットから**配送先住所**を選択し（適用していれば）、**お支払い方法**ウィジェットから**支払方法**を選択します。購入者はすでに **Amazon アカウント** 認証で

ログインしているので、再度ログインする必要はありません。

COURT & CHERRY

Hello Pat!

HOME › SHOPPING CART › CHECKOUT

SHIPPING ADDRESS

Address Book

Pat Smith
2205 Pinzon Ave, Ste 500
Seattle, WA 98199-5678
206-901-2345

Pat & 2205 Pinzon Ave, Ste... Patrick & 994 Coloss Ave W...
Grayson F. 212 Glacier Ave... Morgan T. 7107 Seidnik St, R...

1-4 of 7 [+ Add New](#) Amazon Payments Privacy

PAYMENT INFORMATION

Payment Method

VISA Visa ...3508

Visa ...3508 VISA American Exp ...7110
American Exp ...1234 Mastercard ...1234

1-4 of 5 Amazon Payments Privacy

Pay with Amazon

Use my selected payment method for future purchases and payments to this merchant.

©2013

SUBSCRIPTION SUMMARY

Signature Blend Coffee Pods, 24-pack
by Court & Cherry
Terms: Automatic Monthly Delivery

Items: 1	\$19.00
Shipping	FREE
Taxes	Included
TOTAL PER MONTH	\$19.00

Please Note: Subscription orders are processed on the 5th day of the month. A valid payment method and delivery address are required before shipping.

Pay with Amazon

Use my selected payment method for future purchases and payments to this merchant.

[Learn more](#)

PLACE ORDER

Customer Service: 1-888-333-9090 • [Contact Us](#) • [Privacy](#) • [Terms & Conditions](#) • [Give Us Feedback](#)

7. 購入者は提供された定期購入と請求詳細を確認し、**定期支払同意**ウィジェット内のチェックボックスをチェックすることで定期支払を承認します。それから、購入者は**購入するま**

たは同等のボタンをクリックしてお買い物を完了します。

COURT & CHERRY Hello Pat!

HOME • SHOPPING CART • CHECKOUT

SHIPPING ADDRESS

Address Book

Pat Smith
2205 Pinzon Ave, Ste 500
Seattle, WA 98199-5678
206-901-2345

Pat S. 2205 Pinzon Ave, Ste... Patrick S. 994 Obidos Ave W...
Grayson F. 212 Glacier Ave... Morgan T. 7107 Selkirk St. R...

1-4 of 7 [+ Add new](#) Amazon Payments Privacy

PAYMENT INFORMATION

Payment Method

VISA Visa ...3508

Visa ...3508 VISA American Exp ...7110 AMEX
American Exp ...1234 MASTERCARD Mastercard ...1234

1-4 of 5 [+ Add new](#) Amazon Payments Privacy

SUBSCRIPTION SUMMARY

Signature Blend Coffee Pods, 24-pack
by Court & Cherry
Terms: Automatic Monthly Delivery

Items: 1 \$19.00
Shipping FREE
Taxes Included

TOTAL PER MONTH \$19.00

Please Note: Subscription orders are processed on the 5th day of the month. A valid payment method and delivery address are required before shipping.

Pay with Amazon

Use my selected payment method for future purchases and payments to this merchant.
[Learn more](#)

PLACE ORDER

©2013 Court & Cherry Customer Service: 1-888-333-9090 • [Contact Us](#) • [Privacy](#) • [Terms & Conditions](#) • [Give Us Feedback](#)

8. 購入者は注文完了を受け取ります。

COURT & CHERRY Hello Pat!

THANK YOU FOR SUBSCRIBING

We'll send an email confirmation with your subscription details shortly.
Court&Cherry Ref#: 123-XYZ
Amazon Payments Ref#: SB23-1234567-1234567
Go to [payments.amazon.com](#) to see your payment history and other account information.

SUBSCRIPTION SUMMARY

Signature Blend Coffee Pods, 24-pack
by Court & Cherry
Terms: Automatic Monthly Delivery

Please note: Subscription orders are processed on the 5th day of each month. A valid payment method and delivery address are required before shipping.

MONTHLY TOTAL

Price	\$19.00
Qty	1
Shipping	FREE
Taxes	Included
Total per month	\$19.00

SHIPPING ADDRESS

Address Book

Pat Smith
2205 Pinzon Ave, Ste 500
Seattle, WA 98199-5678
United States
206-901-2345

Amazon Payments Privacy

PAYMENT INFORMATION

Payment Method

VISA Visa ...3508

Amazon Payments Privacy

You might like

Stainless French Press
\$42.00 [ADD TO CART](#)

Whole bean Italian
\$12.00 [ADD TO CART](#)

©2013 Court & Cherry Customer Service: 1-888-333-9090 • [Contact Us](#) • [Privacy](#) • [Terms & Conditions](#) • [Give Us Feedback](#)


Amazon Pay の購入者エクスペリエンス

1. 前述の Amazon ログイン例と異なったケースとして、未認証の購入者が Amazon アカウントでお支払いボタンをクリックして購入処理を開始します。

COURT & CHERRY Get the Amazon Payments Newsletter [SIGN IN](#) | [YOUR ACCOUNT](#)

HOME • SHOPPING CART

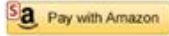
SUBSCRIPTION SUMMARY

	Price	Quantity	Subtotal
 Signature Blend Coffee Pods, 24-pack <i>by Court & Cherry</i>	\$19.00	1	\$19.00
Terms: Automatic Monthly Delivery			
Please note: Subscription orders are processed on the 5th day of each month. A valid payment method and delivery address are required before shipping.			


Have a promo or coupon code?


Items	\$19.00
Shipping	TBD
Taxes	TBD
Total per month	\$19.00

OR

 Pay with Amazon

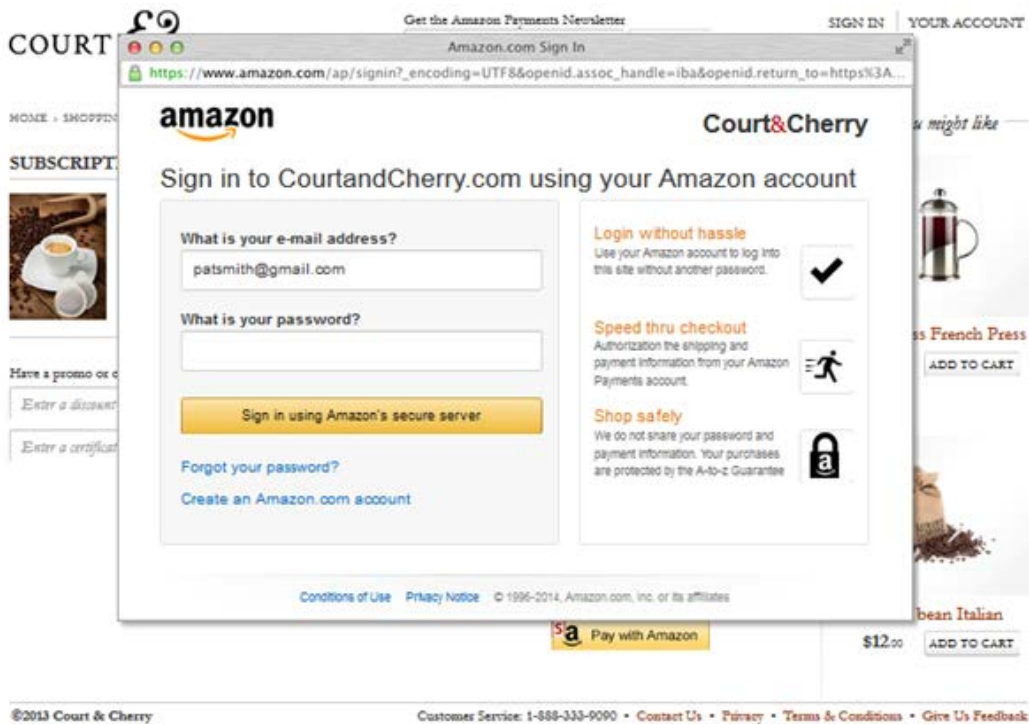
— You might like —

 **Stainless French Press**
\$31.00

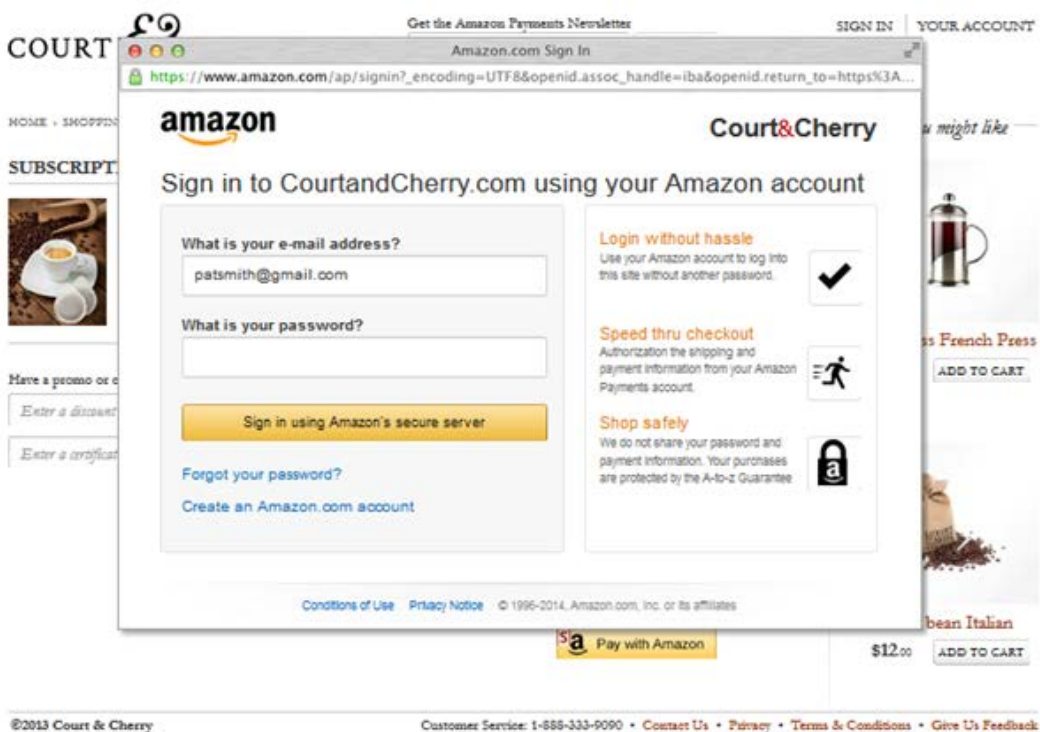
 **Stainless French Press**
\$12.00

©2013 Court & Cherry Customer Service: 1-888-333-9090 • [Contact Us](#) • [Privacy](#) • [Terms & Conditions](#) • [Give Us Feedback](#)

- 購入者が Amazon アカウント認証で確認されていない場合は、購入者は Amazon ユーザー名とパスワードを利用してサインインすることを要求されます。



- 購入者が情報共有に同意した場合（彼らがサイトで初めてログインした場合のみ）は、購入者の完全な配送先住所を取得できます。Amazon アカウントでお支払いボタンをゲスト購入で利用している場合は、購入者はこの画面は表示されません。



4. 購入者はアドレス帳ウィジェットから配送先住所を選択し、お支払い方法ウィジェットから支払方法を選択し、注文を行います。

COURT & CHERRY Hello Pat!

HOME › SHOPPING CART › CHECKOUT

SHIPPING ADDRESS

Address Book

Pat Smith
2205 Pinzon Ave, Ste 500
Seattle, WA 98199-5678
206-901-2345

✓ Pat S. 2205 Pinzon Ave, Ste... Patrick S. 994 Oolios Ave W...
Grayson F. 212 Glacier Ave... Morgan T. 7107 Seidnik St, R...

1-4 of 7 [+ Add new](#) Amazon Payments Privacy

PAYMENT INFORMATION

Payment Method

VISA Visa ...3508

✓ Visa ...3508 VISA American Exp ...7110
American Exp ...1234 Mastercard ...1234

1-4 of 5 Amazon Payments Privacy

Pay with Amazon

Use my selected payment method for future purchases and payments to this merchant. [Learn more](#)

SUBSCRIPTION SUMMARY

Signature Blend Coffee Pods, 24-pack
by Court & Cherry
Terms: Automatic Monthly Delivery

Items: 1	\$19.00
Shipping	FREE
Taxes	Included
TOTAL PER MONTH	\$19.00

Please Note: Subscription orders are processed on the 5th day of the month. A valid payment method and delivery address are required before shipping.

PLACE ORDER

©2013 Customer Service: 1-888-333-9090 • [Contact Us](#) • [Privacy](#) • [Terms & Conditions](#) • [Give Us Feedback](#)

5. 購入者は提供された定期購入と請求詳細を確認し、定期支払同意ウィジェット内のチェックボックスをチェックすることで定期支払を承認します。それから、購入者は購入するま

たは同等のボタンをクリックしてお買い物を完了します。

HOME | SHOPPING CART | CHECKOUT

SHIPPING ADDRESS

Address Book

Pat Smith
2205 Pinzon Ave, Ste 500
Seattle, WA 98199-5678
206-901-2345

Pat 5. 2205 Pinzon Ave, Ste... Patrick S. 994 Obidos Ave W...

Grayson F. 212 Glacier Ave... Morgan T. 7107 Seldnik St, R...

1-4 of 7 [+ Add new](#) Amazon Payments Privacy

PAYMENT INFORMATION

Payment Method

VISA Visa ...3508

Visa ...3508 VISA American Exp ...7110

American Exp ...1234 Mastercard ...1234

1-4 of 5 [+ Add new](#) Amazon Payments Privacy

SUBSCRIPTION SUMMARY

	Signature Blend Coffee Pods, 24-pack <i>by Court & Cherry</i> Terms: Automatic Monthly Delivery
Items: 1	\$19.00
Shipping	FREE
Taxes	Included
TOTAL PER MONTH	\$19.00

Please Note: Subscription orders are processed on the 5th day of the month. A valid payment method and delivery address are required before shipping.

Pay with Amazon

Use my selected payment method for future purchases and payments to this merchant.

[Learn more](#)


PLACE ORDER

6. 購入者は注文完了を受け取ります。

COURT & CHERRY

THANK YOU FOR SUBSCRIBING

We'll send an email confirmation with your subscription details shortly.
Court&Cherry Ref#: 123-XYZ
Amazon Payments Ref#: SB23-1234567-1234567
Go to payments.amazon.com to see your payment history and other account information.

SUBSCRIPTION SUMMARY	MONTHLY TOTAL
 Signature Blend Coffee Pods, 24-pack <i>by Court & Cherry</i>	Price \$19.00
Terms: Automatic Monthly Delivery	Qty 1
Please note: Subscription orders are processed on the 3th day of each month. A valid payment method and delivery address are required before shipping.	Shipping FREE
	Taxes Included
	Total per month \$19.00

SHIPPING ADDRESS

Address Book

Pat Smith
2205 Pinzon Ave. Ste 500
Seattle, WA 98199-5678
United States
206-901-2345

Amazon Payments Privacy

PAYMENT INFORMATION


Payment Method


VISA Visa ...3008

Amazon Payments Privacy

2 Hello Pat

You might like


Stainless French Press
\$42.00 [ADD TO CART](#)


Whole bean Italian
\$12.00 [ADD TO CART](#)

©2013 Court & Cherry Customer Service: 1-888-333-9090 • [Contact Us](#) • [Privacy](#) • [Terms & Conditions](#) • [Give Us Feedback](#)

前述のシナリオの概要では、顧客は購入プロセスを開始し **Amazon アカウント**でログインボタンまたは **Amazon アカウント**でお支払いボタンをクリックしました。それから、彼らは Amazon アカウントでサインインし、個人情報（氏名、メールアドレス、郵便番号）を共有し、トランザクションを完了するために彼らの支払と配送情報にアクセスしました。

これらのシナリオでは、購入者は Amazon アカウント認証を共有し、Web サイト上に顧客情報またはアカウントを作成するために利用できます。Amazon Pay でインテグレーションする場合は、Web サイト上にアカウント作成せずに顧客の支払を開始できるゲスト購入のように Amazon アカウントでお支払いボタンのオプションがあります。このボタンを Web サイトの顧客が購入プロセスを開始する前、途中、後などアクセスする多くのページに追加することができます。顧客がすでに Amazon ログインでログインしている場合は、購入ワークフローを始めるための購入ボタンのように、**Amazon アカウント**でお支払いボタンを利用できます。顧客は再度サインインする必要はありません。

サイトにて **Amazon アカウント**でログインボタンを利用する利点について詳しい情報はボタンウィジェットを参照してください。

イメージ配置の例や Amazon アカウントでログインボタンの利用については、「スタイルガイドライン」を参照してください。

インテグレーションの概要

この章は定期支払である Amazon Pay を Web サイト上にセットアップする代表的な方法の概要です。方法には購入フローでの利用と支払処理の2つのグループになっております。ガイド後半で、これからの詳細についてより詳しく参照できます。

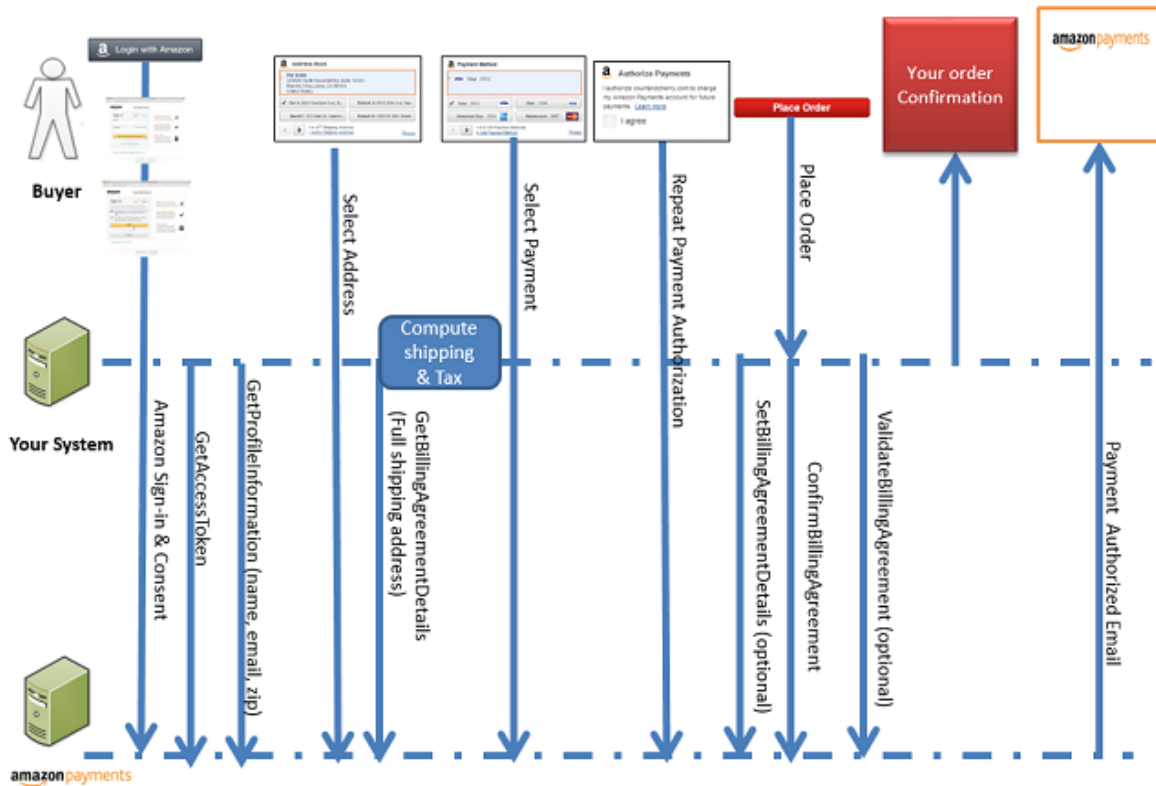
高度なシナリオでの利用について詳しい情報は、「高度な支払シナリオでの利用」を参照してください。

購入フローでの利用

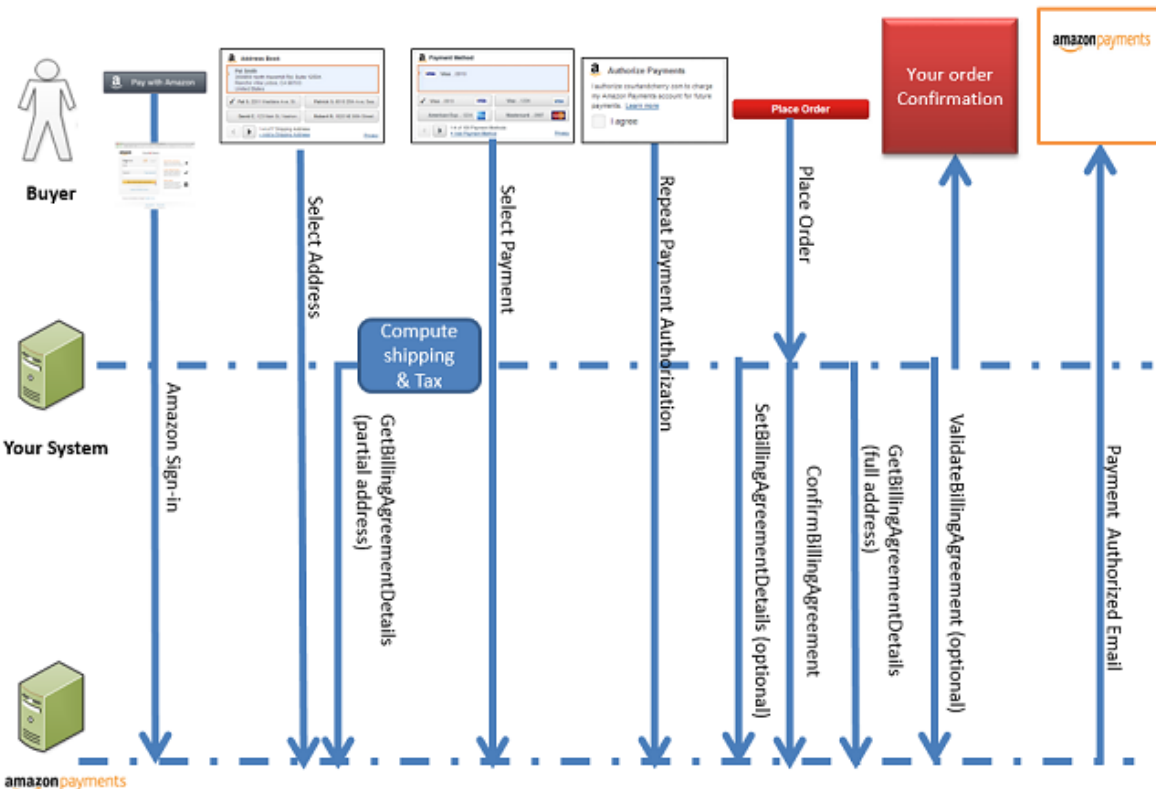
1. 購入者が購入フローを開始する場所（例えば、アカウント作成ページ、ショッピングカート、商品詳細ページ）に **Amazon アカウント** でログインボタンまたは **Amazon アカウント** でお支払いボタンを追加します。**Amazon アカウント** でログインボタンは Amazon Pay で購入者を認証するために有効であり、**Amazon アカウント** でお支払いボタンは認証と支払のために利用できます。
2. 購入者の認証に成功した後で、Amazon Pay は **アクセストークン** を提供します。このトークンを利用して購入者の氏名、メールアドレス、郵便番号を取得することがオプションとして利用できます。**Amazon アカウント** でログイン、または、**Amazon アカウント** でお支払い（完全な配送先住所にアクセス）ボタンのタイプを利用している場合にのみ、購入者の個人情報を取得されることに注意してください。
3. 購入者がアドレス帳ウィジェットから住所を選択している場合は、配送料を計算するために **GetBillingAgreementDetails** 処理を呼び出して配送先住所を取得します。その後で購入者に最終的な金額を表示します。
4. **SetBillingAgreementDetails** 処理を呼び出すことで、購入者の購入に関するメモや Amazon Pay の注文や定期販売を識別するための詳細を追加で提供します。
5. 将来の支払について購入者の同意を得るために **定期支払同意** ウィジェットを表示します。
6. オプションとしてアドレス帳とお支払い方法ウィジェットを注文確認ページに参照バージョンで表示します。
7. 購入者が支払の承認と注文確認を行った後、**ConfirmBillingAgreement** 処理を呼び出して、Amazon Pay の購入を確認します。
8. 購入者が選択した有効な支払方法を確実にするために、オプション的に **ValidateBillingAgreement** 処理を呼び出します。
9. 購入者へ注文して頂いたことを完了ページに表示します。購入者へ注文完了メールを送信します。Amazon Pay も同様に「支払方法のセットアップ」メールを購入者へ送信します。

注意：「お支払方法の設定」メールは Amazon Pay で **BillingAgreement** を承認した時に購入者へ送信されます。この購入者の通知は、彼らの将来の支払と購入のための支払方法を Amazon Pay にセットアップされたことを示します。

次のイメージは Amazon Pay を利用して購入を可能にする方法を示します。



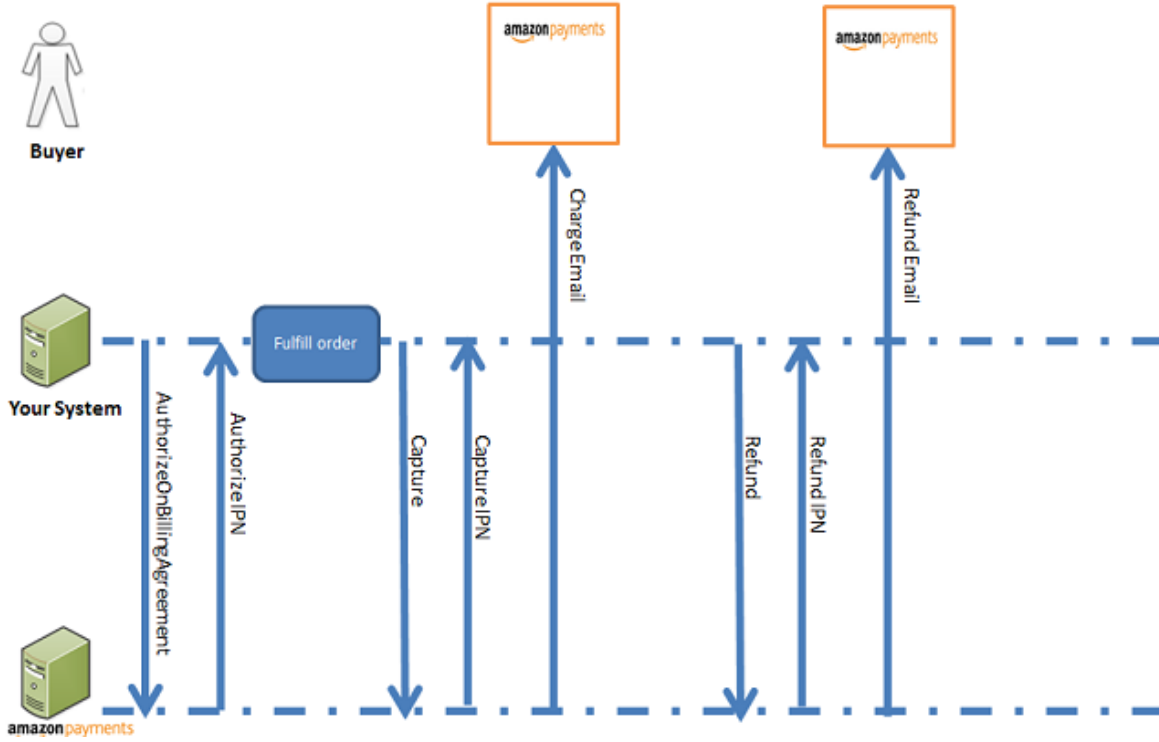
次のイメージは Amazon Pay を利用してゲスト購入オプションを可能にする方法を示します。



支払処理

1. 購入者の支払認証を行うために **AuthorizeOnBillingAgreement** 処理の呼び出しを Amazon Pay に要求し、購入のための支払方法を確実にするために与信を確保します。次の3つのステップで説明されているとおりに、**Capture** 処理を利用することで、これらの資金をその後で売上請求できます。注文を実行する前に支払認証する必要がない場合は、この処理で資金のオーソリと売上請求の両方を行うことができます。
注意：Amazon Pay は同期と非同期モードの2種類の **AuthorizeOnBillingAgreement** 処理を提供しています。同期モードについては、最後の処理ステータスを **AuthorizeOnBillingAgreement** リクエストと同期的に受け取ります。非同期モードでは、最初に **Pending** ステータスでオーソリオブジェクトを受け取ります。その後の処理が完了し、Amazon Pay はインスタント支払通知 (IPN) 経由で最後の処理ステータスを通知します。詳しい情報は「**AuthorizeOnBillingAgreement** 処理の同期と非同期モード」を参照してください。
2. 非同期モードでは、オーソリリクエストの処理ステータスである **Authorize** インスタント支払通知 (IPN) を待機します。
3. 前もってオーソリされた支払を売上請求したい場合は **Capture** 処理を呼び出します。
4. 売上請求リクエストについての処理ステータスを確認するためには **Capture IPN** を待機します。売上請求が正常に処理された場合は、Amazon Pay が購入者に「ご請求内容のお知らせ」メールを送信します。
5. 返金する必要がある場合は、**Refund** 処理を呼び出すことでそれを実行できます。
6. 返金リクエストについての処理ステータスは **Refund IPN** を待機します。返金が正常に処理された場合は、Amazon Pay が購入者に「ご返金内容のお知らせ」メールを送信します。

次のイメージは支払処理方法を表しています。



必要条件

注意：このガイドはテスト環境の Amazon Pay でインテグレーションする方法を説明します。Amazon Pay の SANDBOX であるこのテスト環境はインテグレーションを本番環境に移行する前にエンドツーエンドのテストを行うために有効です。

SANDBOX 環境についての詳しい情報は「SANDBOX 環境でのインテグレーションテスト」を参照してください。

SANDBOX 環境から本番環境へ切り替えるためには、次の内容を変更します。

Amazon MWS エンドポイント	
SANDBOX	https://mws.amazonservices.jp/OffAmazonPayments_Sandbox/2013-01-01/
本番環境	https://mws.amazonservices.jp/OffAmazonPayments/2013-01-01/
Widgets.js URL	
SANDBOX	https://static-fe.payments-amazon.com/OffAmazonPayments/jp/sandbox/lpa/js/Widgets.js
本番環境	https://static-fe.payments-amazon.com/OffAmazonPayments/jp/lpa/js/Widgets.js
Profile API エンドポイント	
SANDBOX	https://api.sandbox.amazon.co.jp/user/profile
本番環境	https://api.amazon.co.jp/user/profile

Amazon Pay を Web サイトに設定

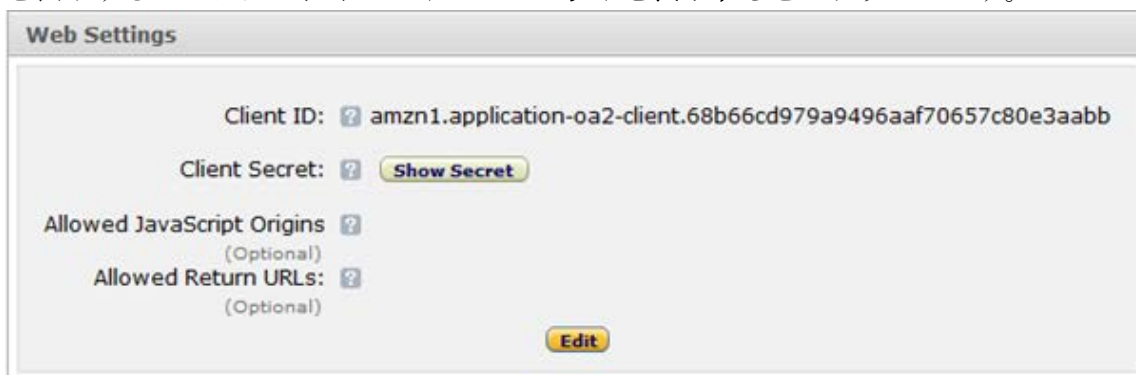
Amazon Pay は重要な購入者エクスペリエンスを提供して動作します。ボタンオプションである Amazon ログインや Amazon Pay のどのボタンを利用するかにかかわらず、行わなければならない Amazon ログインの基本的な設定がいくつかあります。Amazon ログインの設定はセラーセントラルの Amazon ログインページにアクセスして設定します。

1. アプリケーション登録は、Amazon Pay のセラーセントラルにログインして画面上部から「Amazon ログイン」を選択して、**新しいアプリケーションを登録する** ボタンをクリックします。アプリケーション登録フォームが表示されます。
2. アプリケーション詳細ページには、Web サイトについての基本情報を追加します。これらの詳細は Web サイトで利用されます。
 - a. **名前** ユーザーが情報を共有することに同意する画面に表示されるサイトの名前です。この名前はアプリケーションである Web サイトに提供されます。
 - b. **詳細** Amazon ログインを利用する Web サイトの説明です。

- c. **プライバシー規約 URL** プライバシー規約 URL は会社のプライバシーポリシーが記載されている URL です。このリンクはユーザーがアプリケーションへ最初にログインした時に表示されます。（例えば、<http://www.example.com/privacy.html>）
 - d. **ロゴデータ** このロゴは購入者が Web サイトやモバイルアプリにログインした時のサインインと同意画面に表示されます。ロゴは自動的に 50 x 150 ピクセルにリサイズされます。受け取れるフォーマットは PNG、JPEG、GIF になります。
3. 設定が完了しましたら、変更を保存するために**保存する**をクリックしてください。

Web サイトをアプリケーションに追加

1. Amazon ログイン画面の**ウェブ設定**を参照（クリック）します。クライアント ID とクライアントシークレットは自動的に設定されます。クライアント ID は Web サイトに、クライアントシークレットは認証する状況において、Web サイトを確かめるために利用します。パスワードに似ているクライアントシークレットは機密です。クライアントシークレットを表示するためには**クライアントシークレットを表示する**をクリックします。



Web Settings

Client ID: [?](#) amzn1.application-oa2-client.68b66cd979a9496aaf70657c80e3aabb

Client Secret: [?](#) [Show Secret](#)

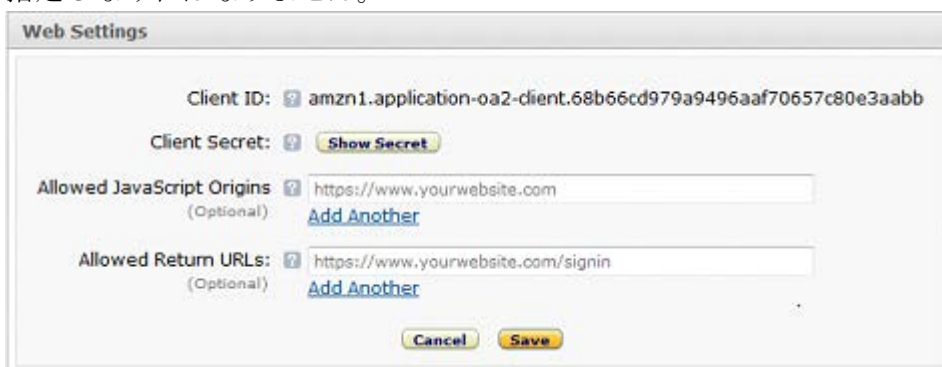
Allowed JavaScript Origins [?](#)
(Optional)

Allowed Return URLs: [?](#)
(Optional)

[Edit](#)

2. アプリケーションの **JavaScript の種類** と **リダイレクト URL** を追加するためには**編集する**をクリックします。

注意：Web サイトで Amazon ログインを使うためには、**JavaScript の種類**（購入者のログイン画面がポップアップ）か **リダイレクト URL**（購入者のログイン画面が画面遷移）のどちらかを指定しなければなりません。Amazon Pay を利用する場合は、オプションの 1 つを指定しなければなりません。



Web Settings

Client ID: [?](#) amzn1.application-oa2-client.68b66cd979a9496aaf70657c80e3aabb

Client Secret: [?](#) [Show Secret](#)

Allowed JavaScript Origins [?](#)
(Optional) [Add Another](#)

Allowed Return URLs: [?](#)
(Optional) [Add Another](#)

[Cancel](#) [Save](#)

3. アドレスはプロトコル、ドメイン名、ポートの組み合わせです。（例としては、`https://www.example.com:8443`）許可されているアドレスは HTTPS プロトコルを使わなければなりません。標準的なポート（443 番ポート）を利用している場合は、ドメイン名を含めるのみで十分です。（例としては、`https://www.example.com`）ログイン処理中に直接 Web サイトと通信するためのドメイン名を JavaScript の種類に追加します。Web ブラウザは、一般的にクロスサイト間の通信するスクリプトは許可されたものでない限りブロックします。
アドレスを追加するには、**さらに追加する**をクリックします。
4. Web サイトが Amazon ログイン認証サービスを呼び出すのに HTTPS で構築し、応答で `redirect_uri` を指定するのであれば、それらのリダイレクト URI をリダイレクト URL に追加します。戻りの URL にはプロトコル、ドメイン、パス、クエリー文字列を含めます。（例としては、`https://www.example.com/login.php`）
戻りの URL を追加するためには、**さらに追加する**をクリックします。
5. **保存する**をクリックすると Amazon ログインの設定が完了されます。

Amazon Pay ウィジェットを利用するための準備

サイト上に埋め込まれたウィジェットを正確に表現するために、`onAmazonLoginReady` コールバックのためのハンドラを提供し、Web サイトのソースコードに Amazon Pay の JavaScript ファイルである `Widgets.js` を追加しなければなりません。Amazon Pay ウィジェットを表示したい場所でこれを実行しなければなりません。

`onAmazonLoginReady` コールバックを定義する方法、Amazon Pay の JavaScript ファイルに追加する方法は、次のサンプルコードに示しているような行を追加します。

ベストプラクティスとして、ボタン、住所、支払ウィジェットが含まれる Web ページの `head` セクションにこのコードを配置することを Amazon Pay は推奨します。さらに、ローカルコピーを作成せず、Amazon から提供している JavaScript の内容を変更しないでください。これは意図しないウィジェットの表示や機能的なエラーなどの結果を引き起こすことがあります。

```
<head>
// your head section here
<script type='text/javascript'>
  window.onAmazonLoginReady = function() {
    amazon.Login.setClientId('YOUR_CLIENT_ID');
  };
</script>

<script async='async' type='text/javascript'
src='https://static-na.payments-
amazon.com/OffAmazonPayments/us/sandbox/js/
  Widgets.js'>
</script>
// your head section here
</head>
```

Amazon Pay の JavaScript ファイルがソースコードに追加された後は、Amazon Pay のウィジェットを導入できる状態になります。

ベストプラクティス

処理前のデータエンコード

HTML、JavaScript、URL の中など多くのフォームの中に出力する前に、Amazon から届いた多くのデータをエンコードすることを推奨します。悪意のあるスクリプトや多く投入された実行可能ファイルをあなたのサイト上で実行できないように出力したエンコードを確実にします。

[オープンウェブアプリケーションセキュリティプロジェクト \(OWASP\) API である ESAPI](#) は、ローリスクのアプリケーション開発を容易にするためにオープンソースウェブアプリケーションセキュリティコントロールライブラリをフリーで提供しています。アウトプットを確実にするためにこの標準的なライブラリを利用してください。

いくつかの最も一般的な種類のアウトプットのデータをエンコードする方法を次のサンプルで表します。

- HTML:
`String safe = ESAPI.encoder().encodeForHTML(request.getParameter("input"));`
- width、名前、値などの HTML 属性の値:
`String safe = ESAPI.encoder().encodeForHTMLAttribute(request.getParameter("input"));`
- URL パラメータの値:
`String safe = ESAPI.encoder().encodeForURL(request.getParameter("input"));`
- JavaScript データの値:
`String safe = ESAPI.encoder().encodeForJavaScript(request.getParameter("input"));`

非同期 JavaScript

Amazon Pay はすばやく Web ページに表示するための JavaScript の非同期ローディングを限定的にサポートします。非同期ローディングを行う唯一の方法は、Amazon Pay のスクリプトタグ内の `async` 属性です。

非同期の Amazon Pay ボタンをロードするためには

1. ページのヘッダー内の、`onAmazonLoginReady` コールバック内にクライアント ID をセットします。
2. ページの body に、`LoginWithAmazon` div 要素をボタンやウィジェットを表示する JavaScript の前に配置します。
3. `window.onAmazonPaymentsReady` 関数を定義します。多くの JavaScript リファレンスをこの `OffAmazonPayments` 関数の中に移動します。

4. 以下の定義のように `async` 属性と同じに `Widgets.js` スクリプトタグを追加します。

タイミングの問題が起こらないように、スクリプトタグは上記の順番に配置する必要があります。

次は非同期ローディングのサンプルです。

```
// In the head of the web page:
<head>
  <script type='text/javascript'>
    window.onAmazonLoginReady = function() {
      amazon.Login.setClientId('YOUR-CLIENT-ID');
    };
  </script>
</head>

// In the body of the page:
<div id="LoginWithAmazon"></div>

<script>
  window.onAmazonPaymentsReady = function(){
    // render the button here
    OffAmazonPayments.Button('LoginWithAmazon',
                              '<Your-Seller-ID>', {
      ...
    }
  }
</script>

<script async='async' type='text/javascript' src=' https://static-
na.payments-amazon.
com/OffAmazonPayments/us/sandbox/js/Widgets.js'>
</script>
```

注意：上記のサンプルは `SANDBOX` 用です。本番環境に移行する場合はそれに応じた URL に変更が必要です。

スクリプトタグの `async` 属性についての詳しい情報は [Mozilla Developer Network script tag Summary](#) を参照してください。

Step1：購入者認証用のボタンウィジェット追加

必要条件：このインテグレーションステップを開始する前に必要条件の全ての要求事項を完了してください。

埋め込みが必要で購入者に表示する最初のウィジェットは、**Amazon Pay ボタン**ウィジェットです。購入者がサイト上での注文を支払うために **Amazon Pay** アカウントを使うことができることを知ってもらうためにこのウィジェットボタンを表示します。

Web サイトの head セクション内に必要な（必要条件内で記述されている通りに）JavaScript コードを追加している場合は、適切なコードを追加することで選択したボタンを埋め込むことができます。サンプルの Amazon Pay ボタンコードは以下の内容を参照してください。

```
<div id="AmazonPayButton"/>
<script type="text/javascript">
  var authRequest;
  OffAmazonPayments.Button("AmazonPayButton", "YOUR_MERCHANT_ID ", {
    type: "ENTER_TYPE_PARAMETER",
    color: "ENTER_COLOR_PARAMETER",
    size: "ENTER_SIZE_PARAMETER",

    authorization: function() {
      loginOptions =
        {scope: "ENTER_SCOPES", popup: "ENTER_POPUP_PARAMETER"};
      authRequest = amazon.Login.authorize (loginOptions,
        "YOUR_REDIRECT_URL ");
    },
    onError: function(error) {
      // your error handling code
    }
  });
</script>
```

Web ページで上記の定義したコードサンプルは、ボタンを表示するために利用されます。ボタンを表示させたい HTML 内にこのコードを配置してください。

ボタンウィジェットパラメータ

サイトへボタンコードを追加する場合は、コードサンプルのパラメータを販売事業者の値に変更する必要があります。次のセクションはコードサンプル内のテキスト変更方法を記述します。



YOUR_MERCHANT_ID

アカウント登録時に得られた出品者 ID を YOUR_MERCHANT_ID に置き換えます。出品者 ID が不明な場合は、セラーセントラルにログインし、インテグレーションメニューの MWS Access Key ページで表示することで得られます。

ENTER_TYPE_PARAMETER

type パラメータは Web ページで選択したいボタンイメージを指定するオプションパラメータです。**type** に値を指定しなかった場合は、Amazon ログインボタンはデフォルトの値がセットされますので注意してください。

次の表は有効な **type** パラメータの値、ボタンの説明、サンプルのボタンイメージを表します。

Type コード	ボタンの説明	ボタンサンプル
LwA	Amazon ログインボタンを指定します。 type が指定されていない場合は、これがデフォルトボタンです。	
PwA	Amazon Pay ボタンを指定します。	

ENTER_COLOR_PARAMETER

color パラメータはボタンの色を選択するためのオプションパラメータです。この表は有効な color コードの値、色の説明、色のサンプルを表します。


Color コード	色の説明	色サンプル
Gold	Gold (デフォルト)	
LightGray	Light gray	
DarkGray	Dark gray	


ENTER_SIZE_PARAMETER

size パラメータはボタンのサイズを選択するためのオプションパラメータです。有効な size コードの値は次の通りです。

標準	高詳細
<ul style="list-style-type: none"> • small • medium (デフォルト) 	<ul style="list-style-type: none"> • large • x-large

選択されたボタンタイプで実際に表示されるサイズ

パラメータ	
タイプ	LwA—Amazon ログイン ボタンを指定します。
サイズ	<ul style="list-style-type: none"> • small(156px x 32px) • medium(174 x 46px) • large(312px x 64px) • x-large(390px x 92px)

パラメータ	
タイプ	PwA—Amazon Pay ボタンを指定します。
サイズ	<ul style="list-style-type: none"> • small(148px x 30px) • medium(200 x 45px) • large(296px x 60px) • x-large(400px x 90px)

ENTER_SCOPES

scope パラメータの値は、呼び出したレスポンス内で返ってきた内容と、個人情報を安全に共有する権限を表示する Amazon Pay の同意画面のタイプの両方に影響します。

次の scope パラメータの組み合わせを利用できます。

Scope パラメータ	説明	要件
profile	ログイン後に全てのユーザー情報（ユーザー名、メールアドレス、ユーザーID）へのアクセスを与えます。	
profile:user_id	ログイン後にユーザー情報からユーザーID だけへのアクセスを与えます。	
postal_code	ユーザーの既定の住所から郵便番号を与えます。	postal_code スコープへアクセスした場合で、profile や profile:user_id スコープと単独、または、組み合わせでアクセス要求時は、ユーザーは

		個人情報を共有するために同意しなければなりません。
payments:widget	ページ上で Amazon Pay ウィジェット（アドレス帳とお支払い方法ウィジェット）を表示するために要求します。パラメータ無しで利用した場合は、 <code>ConfirmOrderReference</code> 呼び出し後に完全な配送先住所にアクセスすることを許可されます。	
payments:shipping_address	アドレス帳ウィジェットで選択された住所をすぐに <code>GetOrderReferenceDetails</code> の API 呼び出し経由で完全な配送先住所にアクセスすることを与えます。	必須の範囲パラメータ <code>payments:widget</code> <code>AddressConsentToken</code> (=AccessToken) を <code>GetOrderReferenceDetails</code> 呼び出すの中でセットします。

組み合わせ例 : `scope: profile payments:widget payments:shipping_address`

`payments:shipping_address` は購入者の配送先住所へのアクセスを提供します。完全な配送先住所を取得するためには、`getOrderReferenceDetails` の API を呼び出す必要があります。詳しい情報は、「Step3: アドレス帳とお支払い方法ウィジェットの追加」の中の「Amazon Pay から配送先住所を取得」、または、Amazon Pay API リファレンスガイドの `GetOrderReferenceDetails` を参照してください。

ENTER_POPUP_PARAMETER

`popup` パラメータは購入者に認証をポップアップウィンドウで表示するのか、代わりに Amazon Pay 認証ページにリダイレクトするのか決定します。

次の `popup` パラメータの 1 つを利用します。

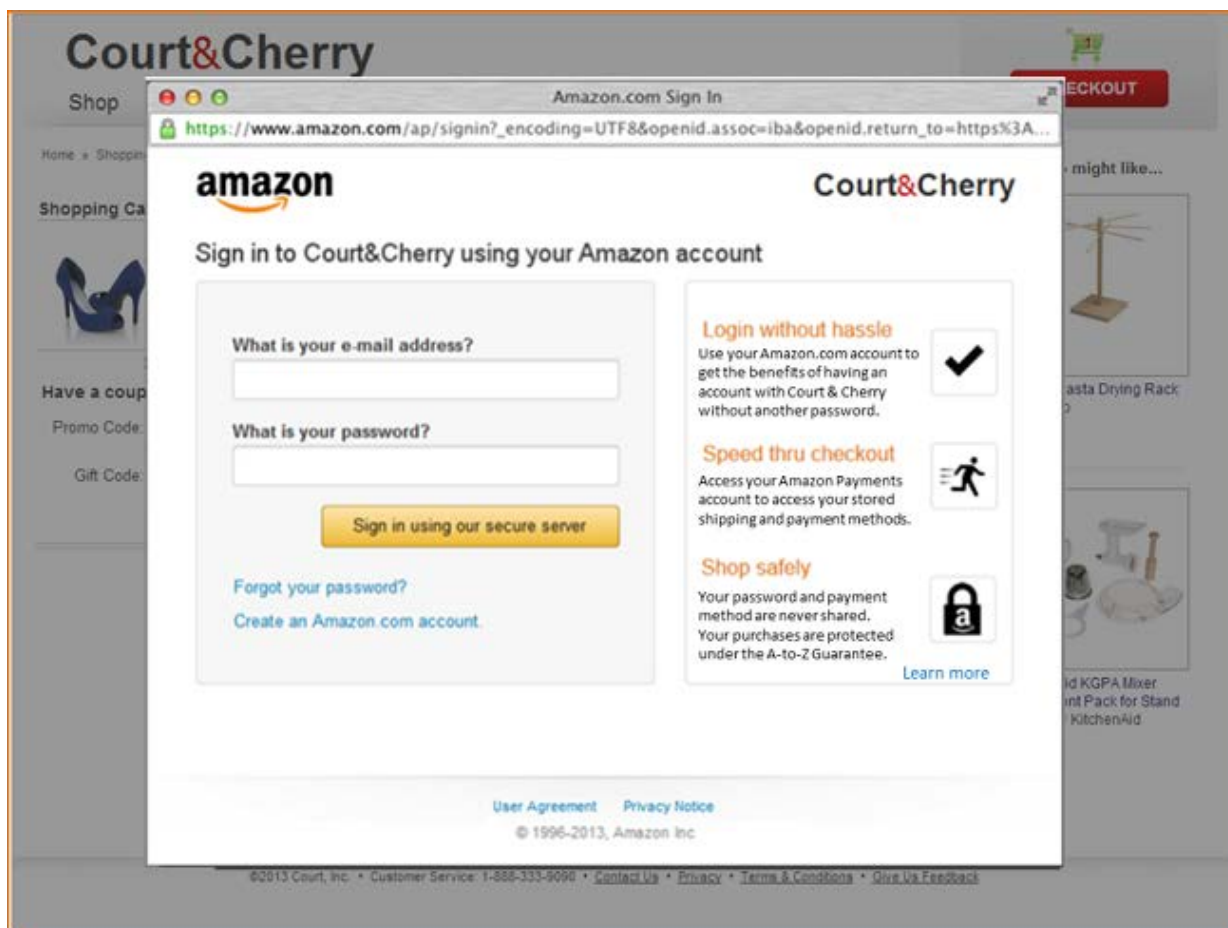
- `true` — Amazon Pay の認証画面は、ポップアップダイアログが表示され Web サイトから離れずに購入者が認証できます。
この値はデスクトップでボタンウィジェットがセキュアページ上で表示されることが必要です。このオプションは正しい TLS/SSL 証明書がある HTTPS ページ上に存在することを理解してください。
- `false` — 購入者は同じブラウザウィンドウ内で Amazon Pay ページがリダイレクトされます。
この値はスマートフォン向けに最適化されたデバイスや非セキュアページ上でボタンウィ

ジェットをレンダリングする場合に推奨されます。リダイレクト URL は HTTPS プロトコルを使用しなければならず、正しい TLS/SSL 証明書が必要であることを理解してください。

詳しい情報は、「Step4：リダイレクト認証エクスペリエンスの利用（オプション）」を参照してください。

デフォルト：popup: true

ポップアップパラメータに true をセットして Amazon Pay ボタンを埋め込む場合は、購入者は Amazon Pay アカウントのメールアドレスとパスワードが求められます。



前のスクリーンショットの中で、会社ロゴを認証ウィンドウに追加することができます。セラーセントラルの Amazon ログイン画面のアプリケーション情報でロゴデータを登録することができます。

YOUR_REDIRECT_URL

リダイレクト URL は購入者が認証に成功した後にリダイレクトされるページのパラメータです。例えば、認証が成功した後、配送先住所または支払方法の詳細を選択するための次にページに遷移することが必要です。ボタンコードの中で“popup:false”を利用している場合は、どの

方法をリダイレクトログインエクスペリエンスで利用するのか注意して、セラーセントラル内の Amazon ログイン設定でのウェブ設定のリダイレクト URL に全て許可する URL を登録してください。

エラーハンドリングコード

Amazon はコードに `onError` ハンドラの実装を推奨します。`onError` コードはオプションですが、インテグレーションのトラブルシューティングにおいて少なからずあなたの労力を減らすことができます。詳しい情報は「SANDBOX 環境でのインテグレーションテスト」を参照してください。

ログアウトオプションの追加

ユーザがログインした後、Web サイトにログアウトオプション（多くの場合はハイパーリンク）を追加する必要があります。ログアウトオプションはキャッシュされたトークンが削除され、Web サイトからユーザーの個人情報（彼らの名前など）を削除されます。それから、Web サイトは再度ログインボタンを提供することができます。

Amazon ログイン SDK for JavaScript を利用している場合は、多くのキャッシュされたトークンを削除するために `amazon.Login.logout` メソッドを呼び出すことができます。例えば、次の通りです。

```
<script type="text/javascript">
  document.getElementById('Logout').onclick = function() {
    amazon.Login.logout();
  };
</script>
```

デフォルトで、その次の `amazon.Login.authorize` 呼び出しでログイン画面を表示します。

Step2：個人情報の取得（オプション）

次に、ボタンウィジェットから返ってきたアクセストークンを利用して、Amazon Pay にアクセスしてユーザの個人情報を取得することができます。アクセストークンはユーザがサイトにログインした時に認証サーバから承諾されています。アクセストークンはクライアント、ユーザ、アクセススコープで特定しております。クライアントは顧客の個人データを取得するためにアクセストークンを使用しなければなりません。サーバサイドアプリケーションでは、`/handle_login.php` で作成したリクエストをハンドリングし、アクセストークンと `Profile REST` の API を利用して個人情報を取得します。

サンプル

PHP サンプル

```
$c
curl_init('https://api.sandbox.amazon.com/auth/o2/tokeninfo?access_token='
    . urlencode($_REQUEST['access_token']));
curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
$r = curl_exec($c);
curl_close($c);
$d = json_decode($r);
if ($d->aud != 'YOUR-CLIENT-ID') {
    // the access token does not belong to us
    header('HTTP/1.1 404 Not Found');
    echo 'Page not found';
    exit;
}

// exchange the access token for user profile
$c = curl_init('https://api.sandbox.amazon.com/user/profile');
curl_setopt($c, CURLOPT_HTTPHEADER, array('Authorization: bearer '
    . $_REQUEST['access_token']));
curl_setopt($c, CURLOPT_RETURNTRANSFER, true);
$r = curl_exec($c);
curl_close($c);
$d = json_decode($r);
echo sprintf('%s %s %s', $d->name, $d->email, $d->user_id);
```

Java サンプル

このサンプルコードを利用するためには Jackson と HttpClientComponents ライブラリをダウンロードしなければなりません。

```
import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.ObjectMapper;
import org.apache.http.client.fluent.Content;
import org.apache.http.client.fluent.Request;
import java.net.URLEncoder;
import java.util.Map;
...
// verify that the access token belongs to us
Content c = Request.Get(
    "https://api.amazon.com/auth/o2/tokeninfo?access_token=" +
    URLEncoder.encode(access_token, "UTF-8"))
    .execute()
    .returnContent();
Map m = new ObjectMapper().readValue(
    c.toString(), new TypeReference<map>(){});
if (!"YOUR-CLIENT-ID".equals(m.get("aud"))) {
    // the access token does not belong to us
    throw new RuntimeException("Invalid token");
}

// exchange the access token for user profile
c = Request.Get("https://api.amazon.com/user/profile")
    .addHeader("Authorization", "bearer " + access_token)
    .execute()
    .returnContent();
m = new ObjectMapper().readValue(
    c.toString(), new TypeReference<map>(){});
System.out.println(
    String.format("%s %s %s", m.get("name"),
        m.get("email"), m.get("user_id")));</map</map
```

Step3 : アドレス帳とお支払い方法ウィジェットの追加

購入者の認証が成功された後に、指定した場所に Amazon Pay のアドレス帳とお支払い方法ウィジェットを表示できます。アドレス帳とお支払い方法ウィジェットは、購入者の Amazon アカウトに保存された配送先住所と支払方法がそれぞれ表示されます。

Web サイトにアドレス帳ウィジェットを埋め込むには、次のサンプルコードを Web ページに追加します。

```

<!-- please put the style below inside your CSS file -->
<style type="text/css">
  #addressBookWidgetDiv{width: 400px; height: 228px;}
</style>

<div id="addressBookWidgetDiv">
</div>

<script>
  new OffAmazonPayments.Widgets.AddressBook({
    sellerId: 'YOUR_SELLER_ID_HERE',
    agreementType: 'BillingAgreement',

    onReady: function(billingAgreement) {
      var billingAgreementId = billingAgreement.
        getAmazonBillingAgreementId();
    },
    onAddressSelect: function(billingAgreement) {
      // Replace the following code with the action you that want to perform
      // after the address is selected.
      // The amazonBillingAgreementId can be used to retrieve
      // the address details by calling the GetBillingAgreementDetails
operation.
      // If rendering the AddressBook and Wallet widgets on the same page,
you
      // should wait for this event before you render the Wallet widget for
      // the first time.
      // The Wallet widget will re-render itself on all subsequent
      // onAddressSelect events, without any action from you. It is not
      // recommended that you explicitly refresh it.
    },
    design: {
      designMode: 'responsive'
    },
    onError: function(error) {
      // your error handling code
    }
  }).bind("addressBookWidgetDiv");
</script>

```

注意： *Billing Agreement* で `agreementType` の値を指定する場合は、Amazon Pay は **Billing Agreement** オブジェクトを *Draft* 状態で生成します。Draft の **Billing Agreement** オブジェクトは「Step7：Billing Agreement の詳細のセットと同意の確認」に記載している通りに購入者がサイトでボタンをクリックしてから3時間以内に完了しなければキャンセルになります。

注意： ワンタイム購入で可能にするために Amazon Pay を利用したい場合は、*Order Reference* に `agreementType` パラメータの値を指定しなければなりません。この方法で進めたい場合は、Amazon Pay インテグレーションガイドを参照してください。

アドレス帳とお支払い方法ウィジェットの両方を利用する場合は、ウィジェットコード内の1つのパラメータに `'onReady'` コールバック関数を指定しなければなりません。`'onReady'` 関数はウ

イジェクトが表示された時に **Billing Agreement** オブジェクトを生成します。 **Billing Agreement** オブジェクトで `getAmazonBillingAgreementId()` 関数を呼び出すことで `billingAgreementId` を取得できます。

注意：購入者から収集する住所に Amazon の **アドレス帳** を使う必要がない場合は、このガイドの「アドレス帳ウィジェット無しのインテグレーション」を参照してください。

あなたの Web サイトにお支払い方法ウィジェットを埋め込むには、Web ページに次のサンプルコードを追加します。

```
<!-- please put the style below inside your CSS file -->
<style type="text/css">
  #walletWidgetDiv{width: 400px; height: 228px;}
</style>

<div id="walletWidgetDiv">
</div>

<script>
  new OffAmazonPayments.Widgets.Wallet({
    sellerId: 'YOUR_SELLER_ID_HERE',
    // amazonBillingAgreementId obtained from the AddressBook widget
    amazonBillingAgreementId: amazonBillingAgreementId,
    onPaymentSelect: function(billingAgreement) {
      // Replace this code with the action that you want to perform
      // after the payment method is selected.
    },
    design: {
      designMode: 'responsive'
    },
    onError: function(error) {
      // your error handling code
    }
  }).bind("walletWidgetDiv");
</script>
```

CSS は Amazon が許可しているパラメータの高さと幅のレンジで規定する限りウィジェットの高さと幅はコントロールすることができます。その他のパーセント、REM、EM のような CSS ユニットは規定されているレンジ内の最小と最大の高さと幅である要素である限り使用できます。

レスポンシブサイトの場合は次のこれらのステップを参照してください。

1. スマートフォンに最適化された各ページの head セクションに meta タグを追加します。この meta タグは有効な画面サイズでウィジェットが調整するのに有効です。そのため、ユーザーが手でページのズーム操作をすることなしにスマートフォンで読みやすくなります。

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0"/>
```

2. 次のサンプルの通りにモバイルファーストデザインで構築するための CSS メディアクエリを使用することができます。

```

<style type="text/css">

/* Please include the min-width, max-width, min-height and max-height if
you plan to use a */
/* relative CSS unit measurement to make sure the widget renders in
theoptimal size allowed. */

#addressBookWidgetDiv {min-width: 300px; max-width: 600px; min-
height:228px; max-height: 400px;}
#walletWidgetDiv {min-width: 300px; max-width:600px; min-height: 228px;
max-height: 400px;}

/* Smartphone and small window */
#addressBookWidgetDiv {width: 100%; height: 228px;}
#walletWidgetDiv {width: 100%; height: 228px;}

/* Desktop and tablet */
@media only screen and (min-width: 768px) {
  #addressBookWidgetDiv {width: 400px; height: 228px;}
  #walletWidgetDiv {width: 400px; height: 228px;}
}

```

Amazon Pay のアドレス帳とお支払い方法ウィジェットを生成する場合は、width と height パラメータを指定しなければならず、別の方法ではウィジェットは表示されません。次の表は1カラムと2カラムウィジェットのwidth と height の有効なパラメータです。

	2カラムウィジェット		1カラムウィジェット	
パラメータ	推奨	有効	推奨	有効
width	400px	400px - 600px	300px	300px - 399px
height	228px	228px - 400px	228px	228px - 400px

認証に成功した後で、購入者に次のスクリーンショットのようにウィジェットを表示するように指定した場所にアドレス帳ウィジェットが表示されます。

HOME > SHOPPING CART > CHECKOUT

SHIPPING ADDRESS

Address Book

Pat Smith
2205 Pinzon Ave, Ste 500
Seattle, WA 98199-6678
206-901-2345

✓ Pat S. 2205 Pinzon Ave, Ste... Patrick S. 994 Obispo Ave W...

Grayson F. 212 Glacier Ave... Morgan T. 7107 Seldnik St, R...

1-4 of 7
[+ Add New](#)

Amazon Payments
Privacy

PAYMENT INFORMATION

Payment Method

VISA Visa ...3508

✓ Visa ...3508 VISA American Exp ...7110

American Exp ...1234 Mastercard ...1234

1-4 of 5
Amazon Payments
Privacy

Pay with Amazon

Use my selected payment method for future purchases and

©2013

SUBSCRIPTION SUMMARY

Signature Blend Coffee Pods, 24-pack
by Court & Cherry
Terms: Automatic Monthly Delivery

Items: 1	\$19.00
Shipping	FREE
Taxes	Included
TOTAL PER MONTH	\$19.00

Please Note: Subscription orders are processed on the 5th day of the month. A valid payment method and delivery address are required before shipping.

Pay with Amazon

Use my selected payment method for future purchases and payments to this merchant.

[Learn more](#)

PLACE ORDER

Customer Service: 1-888-333-9090 • [Contact Us](#) • [Privacy](#) • [Terms & Conditions](#) • [Give Us Feedback](#)

上記のスクリーンショットは Web サイト上にアドレス帳とお支払い方法ウィジェットが表示されています。

それぞれのウィジェットから住所と支払方法を選択されることを確認できるまで、購入ワークフローを有効にしないようにすることを Amazon Pay は推奨します。Amazon Pay は住所と支払方法の選択をそれぞれの `onAddressSelect` と `onPaymentSelect` コールバック関数経由で通知します。これらの通知に基づき、購入フローは次のステップへ進むことができます。例えば、注文確認ページを採用している場合は、Web ページ上の続けるボタンを有効にすることができます。

注意： ページ上にアドレス帳とお支払い方法ウィジェットを表示する場合は、最初にお支払い方法ウィジェットを表示する前に `onAddressSelect` イベントを確認します。

注意： Amazon のアドレス帳とお支払い方法ウィジェットの両方を使用する場合は、Billing Agreement を承認する前に、購入者は住所と支払方法の両方を選択しなければなりません。最初に表示された住所と支払方法が生成された後に購入者が住所選択を変更した場合は、購入者はそれから支払方法を再選択しなければなりません。この意味は、各選択住所に基づき `onPaymentSelect` を確認しなければなりません。（`onAddressSelect` 通知）アドレス帳ウィジェ

ットから住所情報を収集する必要がない場合は、「アドレス帳ウィジェット無しのインテグレーション」を参照してください。

Amazon Pay から配送先住所を取得

多くの場合では、送料を計算するために購入者の配送先住所を入手する必要があります。

`onAddressSelect` 通知を受け取った後は、購入者が選択した住所を受け取るために `GetBillingAgreementDetails` を呼び出すことができます。購入者がウィジェットからサポートされていない住所を選択した場合は、あなたはすぐにフィードバックを提供することができます。このシチュエーションでは、購入者がウィジェットと異なる配送先住所を選択する必要がある場合は、彼らにメッセージを表示する必要があります。

Amazon アカウントでお支払いボタン (ゲスト購入) を利用している場合は、部分的な配送先住所 (都道府県、郵便番号、国) にアクセスを行います。完全な配送先住所にアクセスするためには、**Amazon アカウントでログインボタン**か **Amazon アカウントでお支払いボタン** を利用し、Web ページ上に埋め込むために利用している JavaScript に `payments:shipping_address` スコープパラメータをセットします。スコープパラメータについての詳しい情報は「Step1: 購入者認証用のボタンウィジェット追加」内の `ENTER_SCOPES` セクションを参照してください。

すべてのケースでは、Amazon Pay の **Billing Agreement** を認証した後に `GetBillingAgreementDetails` から完全な配送先住所を受け取れます。詳しい情報は、「Step7: Billing Agreement の詳細のセットと同意の確認」を参照してください。

購入者が選択した住所を受け取るには、`onAddressSelect` 通知が始まった後に `GetBillingAgreementDetails` 処理を呼び出します。

注意: 全ての API を呼び出すには、正確に動作するための署名がされなければなりません。API リクエストを署名するためには、Amazon の MWS キーが必要になります。これらのキーをセラーセントラルにログインし、**インテグレーションメニュー**にある **MWS Access Key** サブメニューを参照することで入手できます。この Amazon の MWS キーは Amazon Pay API 処理での `AWSSessionToken` リクエストパラメータ内にセットしなければなりません。サービスに署名された API 呼び出しを生成するための設定方法についての詳しい情報は、Web で公開している Amazon MWS デベロッパーガイドを参照してください。

次のサンプルは `GetBillingAgreementDetails` 処理の呼び出し方法を表します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01
?AWSAccessKeyId=AKIAIU5C4FEZYEXAMPLE
&Action=GetBillingAgreementDetails
&AddressConsentToken=IQEBLzAtAhUAjagYW4Jrgw84pCaaIDjrKoEhZXsEXAMPLE
&AmazonBillingAgreementId=C23-1234567-1234567
&ContentType=JSON&JSONCallback=cb
&MWSAuthToken=amzn.mws.4ea38b7b-f563-7709-4bae-87aeaEXAMPLE
&SellerId=YOUR_SELLER_ID_HERE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2013-10-29T10%3A38%3A44.000Z
```

```
&Version=2013-01-01
&Signature=gP1loEBaaiQdASWsLDyid18Wn%2BB%2FKZQQtKgpHCtEXAMPLE
```

注文確認する前で完全な配送先住所を取得するために `payments:shipping_address` スコープパラメータを利用し、「Step1: 購入者認証用のボタンウィジェット追加」で取得できる2つパラメータを提示する必要があります。

1. アクセストークンは購入者がログイン承認された場合に返されます。
`&AddressConsentToken=YOUR_ACCESS_TOKEN`
2. Amazon の Billing Agreement ID はログインボタン、アドレス帳、お支払い方法ウィジェットから生成されます。
`&AmazonBillingAgreementId=BILLING_AGREEMENT_IDENTIFIER`

`GetBillingAgreementDetails` 処理のリクエストパラメータとレスポンス要素の内容についての詳しい情報は、AmazonAPI リファレンスガイドにある `GetBillingAgreementDetails` を参照してください。

Step4: リダイレクト認証エクスペリエンスの利用 (オプション)

いくつかケースとして、購入者がモバイルデバイスを利用している場合は、同じページ内で、購入者の Amazon 認証するためにリダイレクトしたいかもしれません。

このセクションでは次の情報を説明します。

- **処理方法** Web サイトでリダイレクト認証を可能にするために必要な変更内容です。
- **クロスサイトリクエストフォージェリ** 購入者の Web ブラウザを利用した攻撃から守る方法です。
- **購入者が拒否した場合** 認証中に購入者が Amazon Pay の個人情報をにアクセスする承諾を拒否した場合のハンドリング方法です。

処理方法

次のステップは、Web サイトでリダイレクト認証するために必要な変更内容を説明します。

1. ボタンウィジェット内の `popup` パラメータに `false` をセットします。
2. Web ページ上から `access token` を受け取るために `<head>` セクション内に JavaScript コードを要求されたウィジェット内に追加しなければなりません。
次のサンプルコードは、`access token` を受け取るための JavaScript を表します。

```

<script type='text/javascript'>
  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 +
      ";secure";
  }

  window.onAmazonLoginReady = function () {
    amazon.Login.setClientId('amzn1.application-oa2-
      client.d607ddd4957c44019e73086bc7cSAMPLE');

    amazon.Login.setUseCookie(true);
  };
</script>

<script src='https://static-na.payments-amazon.com/OffAmazonPayments/us/
  sandbox/js/Widgets.js?sellerId=YOUR_SELLER_ID'>
</script>

```

3. ウィジェットの cookie を削除します。Amazon ログイン SDK for JavaScript を利用している場合は、多くのキャッシュされているトークンを削除するために `amazon.Login.logout` メソッドを呼び出すことができます。リダイレクト認証を利用している場合は、アドレス帳ウィジェットとお支払い方法ウィジェットで利用される cookie を削除する必要があります。続いて呼び出す `amazon.Login.authorize` はデフォルトでログイン画面を表示します。次のサンプルコードは、キャッシュされたトークンを削除するために、Amazon ログイン SDK for JavaScript の `amazon.Login.logout` メソッドを利用します。

```

<script type="text/javascript">
  document.getElementById('Logout').onclick = function() {
    amazon.Login.logout();
    document.cookie = "amazon_Login_accessToken=;
      expires=Thu, 01 Jan 1970 00:00:00 GMT";
    window.location = 'REDIRECT_URL';
  };
</script>

```

クロスサイトリクエストフォージェリ

攻撃者が悪意のあるサイトへユーザーを誘導し認証リンクをクリックするように騙す場合に、クロスサイトリクエストフォージェリが起きます。

多くのコマンドが埋め込まれた悪意のあるリンクは自動的に実行されます、なぜなら、ユーザーは既にサイトで認証されているからです。よって、ユーザーはログイン画面や多くの悪意のある挙動の確認を見ません。Amazon ログインのケースでは、クロスサイトリクエストフォージェリがクライアントや認証サーバを真似て利用することができます。

Amazon ログインでクライアントを登録する場合は、クライアントはクライアント ID (`client_id`) とクライアントシークレット (`client_secret`) が割与えられます。クライアントの認証確認はアクセストークンリクエスト内の `client_id` と `client_secret` パラメータを利用します。(これらの値は攻撃者によって偽造されなければなりません)

Amazon ログインはクロスサイトリクエストフォージェリが起こらないようにするためにステートパラメータを利用します。クライアントは認証リクエストを始める時にステートパラメータの値をセットします。`client_id` と `client_secret` の値が異なっていれば、ステートの値はそれぞれの認証要求で異なっているべきです。認証サーバは認証コードとアクセストークンをクライアントとの通信に渡すために同じ値を返します。ステートパラメータが最初の呼び出しからの値と一致しない場合は、通信は無視されるべきです。

ステートパラメータの計算

クライアントは選択した任意の方法でステートの値を計算することができます。しかしながら、値が偽造ではなくセキュアであることを保証しなければなりません。Amazon ログインはステートを計算するために Hash-based Message Authentication (HMAC) を使うことを推奨します。HMAC 方法は RFC2104 に詳しく述べられています。詳しい情報は RFC2104 を参照してください。

ステートの値を計算するためにこの方法を利用します。クライアントのみが知っている `csrf_secret` と `authenticator` メッセージを持たなければなりません。

このステートの値は `HMAC(csrf_secret, authenticator)` で計算されます。それから `authenticator` の値はクライアントによって保存されます。

ステートが認証レスポンスによって返答された場合は、回収した `authenticator` の値と再度 `HMAC(csrf_secret, authenticator)` 計算を実行することによってステートを確認することができます。値が一致しなかった場合は、認証レスポンスを無視しなければなりません。

購入者が拒否した場合

初めて Web サイトで購入者がログインし認証されると、購入者は Amazon Pay の個人情報を共有するための同意を与えなければなりません。リダイレクト認証中に、購入者が同意画面でキャンセルボタンをクリックし同意を拒否した場合は、Amazon Pay は購入者をリダイレクト URL に進めますが、Access Token の差し戻す代わりに、エラー情報を返します。

エラー情報がクエリー文字列経由で返された場合は、Access Token は URL フラグメント内で返されることに注意してください。リダイレクト URL をレンダリングする前に、エラー情報がクエリー文字列に含まれるか確認が必要であり、見つかった場合は、ログイン処理が初期化されたページにリダイレクトします。

次のサンプルは Amazon Pay 個人情報を共有するための同意を与える URL フラグメントを表します。

```
https://www.sample-store.com/amazonReturn.html#access_token=Atza%7CIQEBljAsAhQWkVKdr_uRdbW7QpkRY8LIxgQIUczBscZcklUALuEd651Nd1_ulkU8WiXq7ZmzcAlp9lZ6Vf_pwaAGL1eVBVKx2x74TmQJkdZiaMdVOT99M34V3HvW0tPgW0xUqIgcgBoJ4R1Lena0Ig9ZMyU_YO0Ma4Xvsqg7R5fTozyZaVVv5PwGNRhrQq32RFU7x4Jr6riKsR1AWFoSW3ilarpDCmbF_P3CgVf5X7Y3Dmdkci7JTestAccessTo ken&token_type=bearer&expires_in=3600&scope=profile+payments%3Awidget
```

次のサンプルは、購入者が個人情報の共有を拒否した場合でのクエリー文字列内で返されるエラー情報です。

```
https://www.sample-store.com/amazonReturn.html?error_description=Access+not+permitted.&error=access_denied
```

Step5 : 定期支払での購入者認証の取得

定期支払を有効にするためには、**Amazon 定期支払同意** ウィジェットを表示しなければなりません。このウィジェットを利用することで、購入者は Amazon Pay アカウントで定期購買や請求支払のように将来の購入と支払について請求することを承認することができます。この同意は販売業者に提供されます。この同意は Amazon Pay の定期支払ポリシーを条件として期限なしである定期購買、一定量の請求、その他の定期支払シナリオをサポートする目的で、変動する総額を柔軟に請求することを提供します。

定期支払同意 ウィジェットを Web サイトに埋め込むためには、Web ページに次のサンプルコードを追加します。

```

<!-- please put the style below inside your CSS file -->
<style type="text/css">
  #consentWidgetDiv {width: 400px; height: 140px;}
</style>

<div id="consentWidgetDiv">
</div>
<script>
new OffAmazonPayments.Widgets.Consent({
  sellerId: 'YOUR_SELLER_ID_HERE',
  // amazonBillingAgreementId obtained from the Amazon Address Book widget.
  amazonBillingAgreementId: amazonBillingAgreementId,
  design: {
    designMode: 'responsive'
  },
  onReady: function(billingAgreementConsentStatus){
    // Called after widget renders
    buyerBillingAgreementConsentStatus =
      billingAgreementConsentStatus.getConsentStatus();
    // getConsentStatus returns true or false
    // true - checkbox is selected
    // false - checkbox is unselected - default
  },
  onConsent: function(billingAgreementConsentStatus) {
    buyerBillingAgreementConsentStatus =
      billingAgreementConsentStatus.getConsentStatus();
    // getConsentStatus returns true or false
    // true - checkbox is selected - buyer has consented
    // false - checkbox is unselected - buyer has not consented

    // Replace this code with the action that you want to perform
    // after the consent checkbox is selected/unselected.
  },
  onError: function(error) {
    // your error handling code
  }
}).bind("consentWidgetDiv ");
</script>

```

定期支払同意ウィジェットを表示する場合は、`amazonBillingAgreementId` を指定しなければなりません。`amazonBillingAgreementId` が指定されていない場合は、ウィジェットはエラーメッセージを表示します。

定期支払同意ウィジェットを生成する場合は `width` と `height` パラメータを指定しなければなりません、そうしなければウィジェットは表示されません。推奨または有効な `width` と `height` パラメータは次の表の通りに指定できます。

パラメータ	推奨	有効
width	400px	300px - 600px

height	140px	140px - 400px
--------	-------	---------------

Amazon は購入者が同意するためのウィジェットのチェックボックスがチェックされるまで購入ワークフローができなくすることを推奨します。Amazon は同意ステータスを `billingAgreementConsentStatus` コールバック関数経由で通知します。チェックボックスがチェックされている場合は、`buyerBillingAgreementConsentStatus` は `true` をセットし、その他の場合は `false` がセットされます。また、`GetBillingAgreementDetails` 処理を呼び出すことで `BillingAgreementConsentStatus` を得ることができます。詳しい情報は Amazon Pay API リファレンスガイドの `GetBillingAgreementDetails` を参照してください。

同意ステータスに基いて、購入フローの次のステップを可能にできます。例えば、注文確認画面を持っている場合は Web ページの続けるボタンを有効にできます、また、購入者が購入を完了するために同意するボタンを有効にすることができます。`BillingAgreement` は `buyerBillingAgreementConsentStatus` が `true` にセットされるまで承認されないことに注意してください。

Step6 : 選択された配送先住所と支払方法の表示 (オプション)

購入者が Amazon Pay ウィジェットから配送先住所と支払方法の両方を選択した後で、購入者が購入を完了する前に選択したものを確認するために参照のみのアドレス帳とお支払い方法ウィジェットを表示することができます。

注文確認ページを用意しており表示のみのウィジェットを表示することを決定している場合は、Amazon は購入者が完了画面を表示する前に選択された住所と支払方法を確認できることを推奨します。`GetBillingAgreementDetails` 処理を呼び出し、制約要素に含まれるレスポンスをチェックするためにこれを実行できます。それぞれの制約の詳細は適切な調整を行う必要がある情報を提供します。例えば、`ShippingAddressNotSet` か `PaymentPlanNotSet` 制約のどちらかを受け取った場合は、購入者にまだ配送先住所または支払方法の選択が必要であることを知らせる必要があります。

それとも、自動的に購入者に選択してもらうために適切なページにリダイレクトするようにサイトに組み込むことを行って問題ありません。購入者が配送先住所または支払方法を選択していない場合は、参照のみのウィジェットは選択する必要があることを購入者に通知します。

`Billing Agreement` に制約が返ってこなかった場合は、`ConfirmBillingAgreement` 呼び出しが成功したことになります。制約について詳しい情報は Amazon Pay API リファレンスの `GetBillingAgreementDetails` と `Billing Agreement` 制約を参照してください。購入者が購入を確認した後は、注文の実行と支払を続行できます。

参照のみの Amazon Pay のアドレス帳とお支払い方法ウィジェットを生成した場合は、width と height パラメータを指定しなければなりません、そうしないとウィジェットは表示されません。推奨と有効な width と height パラメータは次の表示に定義しています。

パラメータ	推奨	有効
width	400px	300px - 600px
height	185px	185px

参照のみのウィジェットを表示するためには、サイト上の元々表示されたウィジェットのコードの中に displayMode: "Read" のパラメータを追加して利用できます。編集可能なアドレス帳とお支払い方法ウィジェットと同じページ上で参照のみのウィジェットを表示したい場合は、onAddressSelect または onPaymentSelect 通知を記載して、その際に参照のみのウィジェットを再表示します。そうしなければ、参照のみのウィジェットは正しい情報を表示しません。次は追加されたパラメータのサンプルコードです。

```

<!-- please put the style below inside CSS file -->
<style type="text/css">
    #readOnlyAddressBookWidgetDiv {width: 400px; height: 185px;}
    #readOnlyWalletWidgetDiv {width: 400px; height: 185px;}
</style>

<div id="readOnlyAddressBookWidgetDiv">
</div>

<div id="readOnlyWalletWidgetDiv">
</div>

<script>
new OffAmazonPayments.Widgets.AddressBook({
  sellerId: 'YOUR_SELLER_ID_HERE',
  // amazonBillingAgreementId obtained from Address or Wallet widget
  amazonBillingAgreementId: amazonBillingAgreementId,
  displayMode: "Read",
  design: {
    designMode: 'responsive'
  },
  onError: function(error) {
    // your error handling code
  }
}).bind("readOnlyAddressBookWidgetDiv");
</script>
<script>
new OffAmazonPayments.Widgets.Wallet({
  sellerId: 'YOUR_SELLER_ID_HERE',
  // amazonBillingAgreementId obtained from Address or Wallet widget
  amazonBillingAgreementId: amazonBillingAgreementId,
  displayMode: "Read",
  design: {
    designMode: 'responsive'
  },


```

```
onError: function(error) {
  // your error handling code
}
}).bind("readOnlyWalletWidgetDiv");
</script>
```

パラメータが追加された場合は、参照のみバージョンのウィジェットは次のスクリーンショットに似たものを Web サイトに表示します。

The screenshot shows the Amazon Pay checkout page for a Court & Cherry subscription. The page includes the following elements:

- Header:** COURT & CHERRY logo and "Hello Pat!" greeting.
- Message:** "THANK YOU FOR SUBSCRIBING" and a confirmation message: "We'll send an email confirmation with your subscription details shortly. Court&Cherry Ref#: 123-XYZ. Amazon Payments Ref#: SB23-1234567-1234567. Go to payments.amazon.com to see your payment history and other account information."
- Subscription Summary Table:**

SUBSCRIPTION SUMMARY		MONTHLY TOTAL	
	Signature Blend Coffee Pods, 24-pack <i>by Court & Cherry</i>	Price	\$19.00
	Terms: Automatic Monthly Delivery	Qty	1
	Please note: Subscription orders are processed on the 5th day of each month. A valid payment method and delivery address are required before shipping.	Shipping	FREE
		Taxes	Included
		Total per month	\$19.00
- Shipping Address:** Address Book showing Pat Smith, 2205 Pinzon Ave, Ste 500, Seattle, WA 98199-5678, United States, 206-901-2345.
- Payment Information:** Payment Method showing VISA Visa 3508.
- Product Recommendations:** "You might like" section with two items: "Stainless French Press" (\$42.00) and "Whole bean Italian" (\$12.00), both with "ADD TO CART" buttons.
- Footer:** ©2013 Court & Cherry, Customer Service: 1-888-333-9090, Contact Us, Privacy, Terms & Conditions, Give Us Feedback.

Step7 : BillingAgreement の詳細のセットと同意の確認

デフォルトでは、購入者を「Step1 : 購入者認証用のボタンウィジェット追加」で示したように Amazon Pay で認証した場合や、Billing Agreement ID を指定せずにアドレス帳またはお支払い方法ウィジェットを表示した場合は、Amazon Pay が定期機能呼び出して Billing Agreement オブジェクトを生成します。このオブジェクトは購入者が Amazon Pay を利用して生成する支払に関連する全ての属性を保存します。

Amazon Pay の購入処理中では、Billing Agreement オブジェクトは、購入者が選択している配送先住所、支払方法と定期支払の同意が事前設定されています。Billing Agreement オブジェクトに、処理を継続するために必要な Billing Agreement 関連のその他多くの詳細として追加属性をセットします。例えば、トランザクションの特定や注文メモや定期購入番号などの Amazon Pay から送信する購入者向けメールに表示させるための詳細をセットできます。

SetBillingAgreementDetails 処理を呼び出しこれらの属性をセットした後で、**ConfirmBillingAgreement** 処理を呼び出して Amazon Pay での購入を承認しなければなりません。Billing Agreement の承認後は、Amazon Pay はこの Billing Agreement で使われた支払方法を許可したことを購入者に通知します。

Billing Agreement の詳細をセット

Billing Agreement オブジェクトの詳細をセットするためには、**SetBillingAgreementDetails** 処理を呼び出します。Billing Agreement オブジェクトに含めたい定期支払についての多くの追加情報を提供することが可能です。この追加情報は Amazon Pay から購入者へのコミュニケーションや返ってきた支払処理情報の両方で利用されます。

Amazon Pay では、それぞれの Billing Agreement の 1 ヶ月ごとの最大請求額は 50,000 円までに制限されています。**GetBillingAgreementDetails** 処理を呼び出すことでこの制限を確認することができます。

次は **SetBillingAgreement** 処理の呼び出し方法のサンプルを表しています。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01?
AWSAccessKeyId=AKIAJKYFSJU7PEXAMPLE
&AmazonBillingAgreementId=C01-8824045-7416542
&Action=SetBillingAgreementDetails
&BillingAgreementAttributes.PlatformId=PLATFORM_ID_HERE
&BillingAgreementAttributes.SellerNote=APPROVE%20LITE%20APPROVE%20HEAVY
&BillingAgreementAttributes.SellerBillingAgreementAttributes.CustomInformation=Example%20Customer%20Info
&BillingAgreementAttributes.SellerBillingAgreementAttributes.StoreName=Test%20Store%20Name
&MWSAuthToken=amzn.mws.4ea38b7b-f563-7709-4bae-87aeaEXAMPLE
&SellerId=YOUR_SELLER_ID_HERE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2013-12-11T10%3A57%3A18.000Z
&Version=2013-01-01
&Signature=Z0ZVgWu0ICF4FLxt1mTjyK%2BjdYG6Kmm8JxLTfsQtKRY%3D
```

SetBillingAgreementDetails 処理、含まれるリクエストパラメータ、レスポンス要素についての詳細な情報は Amazon Pay API リファレンスの **SetBillingAgreementDetails** を参照してください。

Billing Agreement Details の確認

Billing Agreement オブジェクトの属性 (`SetBillingAgreementDetails` 処理呼び出しを經由して追加できる多くの追加属性を含む) をセットした後は、Billing Agreement オブジェクトを承認しなければなりません。このステップは購入者がサイトで注文を行ったことを Amazon Pay に通知します。Billing Agreement オブジェクトの承認は、資金の保証や保留中の資金の支払をしません。このため、支払と資金の売上請求を実行するために Amazon Pay へリクエストして実行依頼する必要があります。

注意：購入者が住所や支払方法を選択していない、または、**定期支払同意** ウィジェットから定期支払の承認をしていない場合は Billing Agreement を確認できません。

Billing Agreement が承認できなかった場合は、それに制約が含まれます。注文確認ページを利用している場合は、注文確認ページを表示する前に `GetBillingAgreementDetails` 処理を呼び出して制約を確認します。

住所、支払、または、制約に関する購入者の同意がある場合は、購入者が住所か支払手段を選ぶまで、Web サイト上の確認ボタンは無効にする必要があります。

Billing Agreement の承認を試みる場合は、特定の制約が確認されるエラーメッセージを受け取ります。

`ConfirmBillingAgreement` 処理が成功、または、制約なしでなければ、購入者に注文確認ページを表示するべきではありません。

制約についての詳しい情報は、*Amazon Pay API リファレンスガイド*の `GetBillingAgreementDetails` と Billing Agreement の制約を参照してください。

`ConfirmBillingAgreement` 処理を呼び出し Billing Agreement オブジェクトを承認します。

次のサンプルは `ConfirmBillingAgreement` 処理の呼び出し方法を表します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01?AWSAccessKeyId=AKIAJKYFJSJU7PEXAMPLE
&AmazonBillingAgreementId=C01-8824045-7416542
&Action=ConfirmBillingAgreement
&MWSAuthToken=amzn.mws.4ea38b7b-f563-7709-4bae-87aeaEXAMPLE
&SellerId=YOUR_SELLER_ID_HERE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2013-12-11T11%3A37%3A19.000Z
&Version=2013-01-01
&Signature=ET6V00R4fr2inSDky4olLrlS1XrQfdrV9Bj%2BiWebz1c%3D
```

`ConfirmBillingAgreement` 処理に含まれるリクエストパラメータ、レスポンス要素についての詳しい情報については、*Amazon Pay API リファレンスガイド*の `ConfirmBillingAgreement` を参照してください。

ConfirmBillingAgreement 処理の呼び出しに成功した後は、**Billing Agreement** オブジェクトは **Draft** 状態から **Open** 状態に遷移します。状態遷移の詳細な情報は、「システムと Amazon Pay の同期」を参照してください。

Amazon Pay ボタン（ゲスト購入）オプションを利用している場合は、**ConfirmBillingAgreement** 処理を呼び出して **Billing Agreement** を確認するまで部分的な配送先住所のみが共有されます。配送料計算のために、この部分的な住所を利用することができます。Amazon Pay は注文を処理するために顧客と完全な配送先住所の情報を提供することを承認します。

ConfirmBillingAgreement 処理を呼び出し **Billing Agreement** を承認した後は、**GetBillingAgreementDetails** 処理を呼び出して完全な配送先住所を入手しなければなりません、また、その住所に配送できるのか確かめる必要があります。

Amazon アカウントでお支払いボタン（ゲスト購入）オプションを利用する場合は次の顧客情報は **Billing Agreement** オブジェクトが承認された後に利用できる状態で生成されます。

購入者詳細	入手方法
購入者	ConfirmBillingAgreement 処理の呼び出しに成功した後に GetBillingAgreementDetails 処理を呼び出します。
購入者メールアドレス	
購入者電話番号 (利用可能であれば)	
受取人名	ConfirmBillingAgreement 処理の呼び出しに成功した後に GetBillingAgreementDetails 処理を呼び出します。
配送先住所	部分的な住所情報（都道府県、郵便番号、国コード）は、アドレス帳ウィジェットで選択された後にすぐに GetBillingAgreementDetails 処理を呼び出して入手することができます。完全な住所情報（市区町村や番地情報を含む）は ConfirmBillingAgreement 処理の呼び出しに成功した後に、 GetBillingAgreementDetails 処理を呼び出すことで含まれます。 注意： Billing Agreement を承認する前に完全な配送先住所にアクセスしたい場合は、 payments:shipping_address スコープを使用しなければなりません。詳しい情報は、このガイドの「Step1：購入者認証用のボタンウィジェット追加」を参照してください。
受取人電話番号 (利用可能であれば)	ConfirmBillingAgreement 処理の呼び出しに成功した後に GetBillingAgreementDetails 処理を呼び出します。

この時点で、Amazon Pay は、現在の購入者が参照している注文確認のビジネス手順（メール確認やランディングページ確認など）を処理することを強く推奨します。あなたが非同期モード

のオーソリを利用している場合は、支払方法のオーソリと売上請求を次に続くステップと同様に処理するための追加時間が取られます。

定期支払を確認するための購入者メールのセットアップ

Billing Agreement オブジェクトが **Open** 状態になるとすぐに、**Amazon Pay** は購入者の **Amazon Pay** アカウントで自動的に支払うことに同意されたことを「お支払い方法の設定」メールとして購入者に通知します。

注意：**Amazon Pay** によって送信されるメールの中に様々な種類の情報を指定することができます、例えば、販売事業者のカスタマーサービスのメール、電話番号、メモです。これらの詳しい情報は、このガイドの「提供可能な購入者向けのメールコンテンツ」を参照してください。

Step8 : Billing Agreement の確認

注意：支払のオーソリを行うのであれば、このステップはスキップすることができます、支払手段が有効であるかオーソリはチェックします。

オプションステップのように、購入者に注文完了ページを表示する前に

ValidateBillingAgreement 処理を呼び出すことで、購入者が選択した支払方法と **Billing Agreement** ステータスを確認できます。これはその時の購入者によって選択された支払方法を保証し、後の時点で購入者に請求する場合に発生する可能性のある支払の失敗を減らします。

次のサンプルは、**ValidateBillingAgreement** 処理を呼び出す方法を表します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01?AWSAccessKeyId=AKIAJKYFSJU7PEXAMPLE
&AmazonBillingAgreementId=C01-8824045-7416542
&Action=ValidateBillingAgreement
&MWSAuthToken=amzn.mws.4ea38b7b-f563-7709-4bae-87aeaEXAMPLE
&SellerId=YOUR_SELLER_ID_HERE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2013-12-11T10%3A38%3A44.000Z
&Version=2013-01-01
&Signature=gP1loEBaaiQdASwSLDyid18Wn%2BB%2FKZQQtKgpHctDHdc%3D
```

ValidateBillingAgreement 処理は、**ValidationResult** パラメータの値として **Success** または **Failure** を返します。**Failure** が返された場合は、理由コードも返っています。次は個々の理由コードを扱うための提案です。

- **InvalidPaymentMethod** は購入者によって選択された支払方法が正しくないことを示します（例えば、有効期限切れ、無効な請求住所など）。この場合は、**Billing Agreement** オブジェクトも **Suspended** 状態に遷移し、多くの **AuthorizeOnBillingAgreement** 処理は **Declined** に

なります。購入者に **Billing Agreement** オブジェクトに関連した支払方法を更新するようにリクエストする必要があります。

- **ValidationTimedOut** は Amazon Pay が **Billing Agreement** を数秒以内に確認出来なかったことを示し、**ValidationResult** に **Failure** を受け取ります。Amazon はいずれか経った後にこのリクエストをリトライすることを提案します。リクエストを数回リトライした後に成功しなかった場合は、購入者の支払方法に問題があることとなります。Amazon Pay の Web サイトかあなたの Web サイトから支払方法を更新することを購入者にリクエストし、購入者が支払方法を更新してからリクエストをリトライします。

Step9 : オーソリのリクエスト

Billing Agreement オブジェクトを承認し **Open** 状態である場合は、支払処理を開始できます。支払を処理するためには、**AuthorizeOnBillingAgreement** 処理を呼び出して注文処理中に選択した支払方法のオーソリを取得しなければなりません。オーソリは **Order Reference** に常に結びつけられます。**AuthorizeOnBillingAgreement** 処理は **Order Reference** を生成し、それから、オーソリを確定させる便利な処理です。**Order Reference** は **Billing Agreement** から支払方法と配送先住所情報を受け継ぎます。**Order Reference** は Amazon Pay の Web サイトの中で購入者のための一般情報として現れます。

次のサンプルは **AuthorizeOnBillingAgreement** 処理を呼び出す方法を表します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01?AWSAccessKeyId=AKIAJKYFJSJU7PEXAMPLE&Action=AuthorizeOnBillingAgreement&AmazonBillingAgreementId=C01-1234567-1234567&AuthorizationAmount.Amount=10&AuthorizationAmount.CurrencyCode=USD&AuthorizationReferenceId=test_authorize_1&MWSAuthToken=amzn.mws.4ea38b7b-f563-7709-4bae-87aeaEXAMPLE&SellerId=YOUR_SELLER_ID_HERE&SellerAuthorizationNote=ForNovemberOrder&SellerOrderAttributes.SellerOrderId=testSellerOrderId&SellerOrderAttributes.StoreName=testStore&SellerOrderAttributes.CustomInformation=ExampleInformation&InheritShippingAddress=true&SignatureMethod=HmacSHA256&SignatureVersion=2&Timestamp=2012-10-03T19%3A01%3A11Z&TransactionTimeout=60&Version=2013-01-01&Signature=WlQ708aqyHXMkoUBk69Hjxj8qdh3aDcqpY71hVgEXAMPLE
```

AuthorizeOnBillingAgreement 処理のリクエストパラメータに含める、または、レスポンス要素の詳しい情報は、Amazon Pay API リファレンスガイドの **Authorize** を参照してください。

オーソリオブジェクト

- Amazon Pay での **オーソリ** オブジェクトは **Billing Agreement** オブジェクトに関連した支払方法の資金を確保するものです。 **オーソリ** オブジェクトは次の 4 つステータスのどれかになります。
- **Pending** — **AuthorizeOnBillingAgreement** 処理は 2 つのモードである同期と非同期で働きます。 **Pending** ステータスは非同期モードを選択している場合にのみ適用します。詳しい情報は「AuthorizeOnBillingAgreement 処理の同期と非同期モード」を参照してください。非同期モードでは、Amazon はリアルタイムでオーソリを処理しません。 **オーソリ** オブジェクトは最初にオーソリリクエストを確定した時は常に **Pending** 状態です。
AuthorizeOnBillingAgreement 処理からこのステータスで示す同期レスポンスを受け取ります。オーソリオブジェクトは Amazon によって処理されまで **Pending** 状態で存続します。処理時間は 1 分程度で変更します。処理後は完了し、Amazon は最終の処理ステータスを通知します。最終のオーソリステータスを受け取る方法の詳しい情報は「Step10：オーソリステータスの取得」を参照してください。
- **Open** — **オーソリ** オブジェクトは支払方法のオーソリに成功すればすぐに **Open** 状態になります。オープン状態になってから 30 日までにこのオーソリに対する売上請求をリクエストすることができます。
- **Declined** — **オーソリ** オブジェクトは Amazon がオーソリリクエストを断った場合に **Declined** 状態になります。詳しい情報は「支払失敗のハンドリング」を参照してください。
- **Closed** — **オーソリ** オブジェクトは様々な理由でこの状態になることができます。
 - オブジェクトは本番環境モードでは 30 日間、**Sandbox** モードでは 2 日間は **Open** 状態です。
 - **Open** のオーソリに対してすべて、または、部分的な金額の売上請求が完了しました。Amazon は 1 つのオーソリに対して 1 つの売上請求のみ許可しています。オーソリは売上請求リクエストが成功した後にすぐに **Closed** になります。
 - オーソリを **Closed** にした。
 - **Billing Agreement** オブジェクトを **Canceled** にした。
 - Amazon がオーソリを **Closed** にした。

Step10：オーソリステータスの取得

注意：このステップは **AuthorizeOnBillingAgreement** 処理で非同期モード処理を利用する場合に必要です。

AuthorizeOnBillingAgreement 呼び出しのレスポンスは、**AuthorizationStatus** レスポンス要素を含みます。非同期モードの処理を選択している場合は **Pending** が常にセットされ、Amazon Pay 経由のインスタント支払通知 (IPN) サービスからオーソリリクエストの最後のステータス (例、**Open** や **Declined**) を後で受け取ります。インスタント支払通知 (IPN) サービスについての詳しい情報は、「システムと Amazon Pay の同期」を参照してください。

それから追加で **オーソリ** のレスポンスで返された **AmazonAuthorizationId** を利用して **GetAuthorizationDetails** 処理を呼び出すことで、オーソリオブジェクトの詳細を問合せることができます。このユニークな識別子をオーソリ成功した資金の売上請求を行うためにも必要になります。詳しい情報は、Amazon Pay API レファレンスガイドの **GetAuthorizationDetails** を参照してください。

オーソリの保持期間

AuthorizeOnBillingAgreement 処理の提示に成功した後、本番環境モードでは、オーソリに対して売上請求可能な期間は 30 日間あります。30 日を過ぎると自動的に **Closed** になります。SANDBOX 環境モードでは 2 日間になります。

オーソリ処理を待たせたい場合は、最大時間を指定するために **AuthorizeOnBillingAgreement** リクエストの **TransactionTimeout** パラメータを利用することができます。この時間内では処理されなかった (**Pending** 状態で残っている) 多くのオーソリは Amazon Pay によって **Declined** になります。インスタント支払通知 (IPN) サービス経由で **TransactionTimeout** の理由コードとともに **Declined** のメッセージを受け取ります。

TransactionTimeout に 0 をセットした場合は、オーソリが **Open** か **Declined** ステータスを常に返します。この場合は、**TransactionTimedOut** の理由コードとともに **Declined** になり、オーソリは同期処理することができません。詳しい情報は、「**AuthorizeOnBillingAgreement** 処理の同期と非同期モード」を参照してください。

Step11 : 売上請求のリクエスト

いつでも購入者に請求や定期ロジックごとに請求リクエストできます。**Billing Agreement** は運用ルールにて要求された頻度 (週次、月次、年次など) で調整可能な金額を柔軟に請求することを許可します。

AuthorizeOnBillingAgreement 処理の呼び出しに成功し、**オーソリ** オブジェクトが **Open** 状態になった後、オーソリに対応する資金を売上請求することができます。直ちに資金を売上請求するために、**AuthorizeOnBillingAgreement** 処理を **CaptureNow** パラメータに **True** をセットして呼び出すことができます。別の方法としては、個別に **売上請求** をリクエストしなければなりません。

売上請求オブジェクト

売上請求 オブジェクトは購入者からの資金をアカウントに移動することを意味します。すべての **売上請求** オブジェクトは前もって作成された有効なオーソリオブジェクトに関連されます。**売上請求** オブジェクトは次の 4 つのステータスの内のどれかになります。

- **Pending** – **売上請求** オブジェクトは Amazon によって処理されるまで **Pending** 状態中です。**売上請求** 処理の同期レスポンスは **Pending** 状態を返します、また、リクエストが Amazon

によって処理されるまで、このステータスが存続します。処理が完了した後は、Amazon は処理ステータスを通知します。最終の売上請求ステータスを受け取る方法の詳細な情報は、「Step12：売上請求ステータスの取得」を参照してください。

- **Declined** – Amazon が売上請求リクエストを断った時に、売上請求オブジェクトは **Declined** 状態です。詳しい情報は「支払失敗のハンドリング」を参照してください。
- **Completed** – 購入者の売上請求が成功した時に、売上請求オブジェクトは **Completed** 状態になります。
- **Closed** – Amazon が購入者アカウントに問題を確認した、売上請求の最大金額がすでに返金された、売上請求に対して 10 回の部分返金がリクエストされた場合に、**売上請求** オブジェクトは **Closed** 状態に遷移します。返金できる最大額は売上請求金額の 15% もしくは 8,400 円のどちらか低い額を加算した金額までです。

売上請求 オブジェクトの状態遷移についての詳しい情報は、「システムと Amazon Pay の同期」を参照してください。

オーソリ オブジェクトで指定された資金を回収するためには、**売上請求** 処理の呼び出しを本番環境モードではオーソリに成功してから 30 日以内に、Sandbox モードでは 2 日以内にしなければなりません。

次のサンプルは売上請求処理を呼び出す方法を表します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01
?AWSAccessKeyId=AKIAFBM3LG5JEEXAMPLE
&Action=Capture
&AmazonAuthorizationId=S23-1234567-1234567-0000001
&CaptureAmount.Amount=94.50
&CaptureAmount.CurrencyCode=USD
&CaptureReferenceId=test_capture_1
&SellerCaptureNote=Payment%20for%20Blue%20Shoes
&SellerId=YOUR_SELLER_ID_HERE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2013-11-19T19%3A01%3A11Z
&Version=2013-01-01
&Signature=WlQ708aqyHXMkoUBk69Hjxj8qdh3aDcqY71hVgEXAMPLE
```

売上請求 の処理、含まれるリクエストパラメータ、レスポンス要素についての詳しい情報は、Amazon Pay API リファレンスガイドの Capture を参照してください。

Step12：売上請求ステータスの取得

売上請求 リクエストを提示した場合は、Amazon Pay は直ちにそれを処理します。**売上請求** リクエストのためのレスポンスは **Completed** か **Declined** 状態を示します。

処理が完了した後、Amazon Pay はインスタント支払通知 (IPN) サービスを経由して処理状態を通知します。Amazon Pay で売上請求リクエストが完了できなかった場合は、**Declined** メッセ

ージを受け取ります。インスタント支払通知 (IPN) サービスについての詳しい情報は、「システムと Amazon Pay の同期」を参照してください。

売上請求レスポンスで返された **AmazonCaptureId** を利用して **GetCaptureDetails** 処理を呼び出すことで、売上請求オブジェクトの詳細について追加要求できます。完了した売上請求に対する資金を返金するために、この識別子が必要になります。詳しい情報は、*Amazon Pay API リファレンスガイド*の **GetCaptureDetails** を参照してください。

売上請求成功の購入者メール

売上請求が成功した場合いつでも、Amazon Pay は購入者に売上請求についての詳細を請求通知メールにて送信します。

注意： Amazon Pay は、送信されるメール内の各種情報を指定することを許可します。例えば、販売事業者のカスタマーサービスマール、電話番号、メモです。これらの詳しい情報は、このガイド内の「提供可能な購入者向けのメールコンテンツ」を参照してください。

Step13：返金のリクエスト (オプション)

返金処理を呼び出すことで以前に売上請求処理が成功したものに対して全額または一部返金を行うことができます。売上請求処理は**返金**処理を呼び出すために必要な **AmazonCaptureId** を生成します。

注意： 返金を開始するための制限時間はありません。しかしながら、期間が長い返金リクエストを行った場合は、すでに支払方法が無効になっている可能性が高く返金に失敗することがあります。

返金オブジェクト

Amazon Pay では、返金オブジェクトは以前に購入者アカウントから売上請求した資金をアカウントに戻したことを意味します。返金オブジェクトは次の3つ状態のどれかになります。

- **Pending** – Amazon は返金をリアルタイムで処理しません。最初に返金リクエストを提示した時、**返金**オブジェクトは常に **Pending** 状態です。**返金**オブジェクトは Amazon によってそれが処理されるまで **Pending** 状態で存続します。処理時間は可変で数時間を必要とします。処理が完了した後、Amazon は処理ステータスを通知します。最後の返金ステータスを受け取るための方法についての詳しい情報は、「Step14：返金ステータスの取得」を参照してください。
- **Declined** – Amazon によってリクエストが拒否された場合は、**返金**オブジェクトは **Declined** 状態になります。詳しい情報は「支払失敗のハンドリング」を参照してください。
- **Completed** – 返金の処理が成功した後、返金オブジェクトは **Completed** 状態になります。

次のサンプルは返金処理を呼び出す方法を表します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01
?AWSAccessKeyId=AKIAFBM3LG5JEEEXAMPLE
&Action=Refund
&AmazonCaptureId=S23-1234567-1234567-0000002
&RefundAmount.Amount=94.50
&RefundAmount.CurrencyCode=USD
&RefundReferenceId=test_refund_1
&SellerRefundNote=Refund%20for%20Blue%20Shoes
&SellerId=YOUR_SELLER_ID_HERE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2013-11-27T19%3A01%3A11Z
&Version=2013-01-01
&Signature=WlQ708aqyHXMkoUBk69Hjxj8qdh3aDcqpY71hVgEXAMPLE
```

返金処理に含まれるリクエストパラメータ、レスポンス要素についての詳しい情報は、Amazon Pay API リファレンスガイドの Refund を参照してください。

Step14 : 返金ステータスの取得 (オプション)

返金処理のレスポンスは **RefundStatus** と呼ばれるレスポンス要素を含み、最初は **Pending** にセットされます。インスタント支払通知 (IPN) サービス経由で Amazon Pay から返金リクエストの最終ステータス (例えば、**Completed** や **Declined**) を後ほど受け取ります。インスタント支払通知 (IPN) サービスについての詳しい情報は「システムと Amazon Pay の同期」を参照してください。

返金レスポンスで返された **AmazonRefundId** を利用して **GetRefundDetails** を呼び出し、返金オブジェクトの詳細を要求することができます。詳しい情報は Amazon Pay API リファレンスガイドの **GetRefundDetails** を参照してください。

返金が成功した場合の購入者メール

いつでも返金が成功したとき、Amazon Pay は購入者に返金について詳細として返金通知メールを送信します。

注意 : Amazon Pay は、送信されるメールの情報のいくつかの部分を設定することを許可しております。例えば、カスタマーサービスメール、電話番号、メモです。これらの詳しい情報は、このガイドの「提供可能な購入者向けのメールコンテンツ」を参照してください。

提供可能な購入者向けのメールコンテンツ

次の状態で Amazon から購入者にメールが送信されます。

- **Billing Agreement** オブジェクトが承認された時

- **Billing Agreement** を利用して **Order Reference** が生成された時には、このメールは送信されません。
- 売上請求された時
- 返金した時
- 支払方法が変更された時
- 住所が変更された時
- **Billing Agreement** を Close にした時

Amazon では多くのメールコンテンツを提供しております、また、メッセージ本文に販売事業者が提供可能ないくつかの要素があります。次の表は購入者が参照するメールコンテンツのリストです。アカウント設定したものや API 処理のパラメータで定義できます。

メールフィールド	参照元
販売事業者カスタマーサービスメールアドレス	Amazon の販売事業者アカウントをセットアップした時に指定されたもの、または、セラーセントラルの出品用アカウント情報の設定タブで設定されたもの
販売事業者カスタマーサービス電話番号	
販売事業者店舗名	Amazon の販売事業者アカウントをセットアップした時に指定されたもの、または、セラーセントラルの出品用アカウント情報の設定タブで設定されたもの SetBillingAgreementDetails 処理の StoreName パラメータが優先されます。
販売事業者の Billing Agreement の識別子	SetBillingAgreementDetails 処理の SellerBillingAgreementId パラメータで指定されたもの
販売事業者メモ	<ul style="list-style-type: none"> • 支払方法セットアップ通知のために SetBillingAgreementDetails 処理の SellerNote パラメータで指定されたもの。 • 売上請求通知のために AuthorizeOnBillingAgreement 処理、または、Capture 処理の SellerCaptureNote パラメータで指定されたもの。 • 返金通知のために Refund 処理の SellerRefundNote パラメータで指定されたもの。

AuthorizeOnBillingAgreement 処理の同期と非同期モード

AuthorizeOnBillingAgreement 処理は次の 2 つの機能です。

- 非同期モード— このモードでは、**AuthorizeOnBillingAgreement** 処理は非同期レスポンス内の **オーソリ** オブジェクトについて常に **Pending** ステータスを返します。最終処理ステータス (**Open** または **Declined**) を得るためにインスタント支払通知 (IPN) を確認しなければなりません。
- 同期モード— このモードでは、**AuthorizeOnBillingAgreement** 処理は **オーソリ** オブジェクトのレスポンスで常に **Open** か **Declined** ステータスを返します。

AuthorizeOnBillingAgreement 処理の処理方法の選択は、運用ルールと技術的な観点から判断します。

非同期モード

非同期モードの処理については、値を設定しない、または、0 以外の値を **AuthorizeOnBillingAgreement** 処理の **TransactionTimeout** リクエストパラメータに設定することで選択できます。 **TransactionTimeout** リクエストパラメータの有効な値についての詳しい情報は、*Amazon Pay API* リファレンスの *Authorize* を参照してください。

次のコードサンプルは **AuthorizeOnBillingAgreement** 処理で非同期モードを有効にする方法を表しています。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01
?AWSAccessKeyId=AKIAJKYFSJU7PEXAMPLE
&Action=AuthorizeOnBillingAgreement
&AmazonBillingAgreementId=C01-1234567-1234567
&AuthorizationAmount.Amount=10
&AuthorizationAmount.CurrencyCode=USD
&AuthorizationReferenceId=test_authorize_1
&MWSAuthToken=amzn.mws.4ea38b7b-f563-7709-4bae-87aeaEXAMPLE
&SellerId=YOUR_SELLER_ID_HERE
&SellerAuthorizationNote=ForNovemberOrder
&SellerOrderAttributes.SellerOrderId=testSellerOrderId
&SellerOrderAttributes.StoreName=testStore
&SellerOrderAttributes.CustomInformation=ExampleInformation
&InheritShippingAddress=true
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2012-10-03T19%3A01%3A11Z
&TransactionTimeout=60
&Version=2013-01-01
&Signature=WlQ708aqyHXMkoUBk69Hjxj8qdh3aDcqpY71hVgEXAMPLE
```

非同期モードの処理では、**AuthorizeOnBillingAgreement** リクエストで生成された **オーソリ** オブジェクトのレスポンス内に、最初は **Pending** ステータスを約 2～3 秒以内で返されます。インスタント支払通知 (IPN) を経由して最終の処理ステータス (**Open** か **Declined**) が約 60 秒以内に返ります。ごく稀に、Amazon Pay が手動でトランザクションを調査した時に、最終の処理ステ

ータスを指定した **TransactionTimeout** 値まで、または、**TransactionTimeout** 値が指定されていない場合の 24 時間までは受け取れないかもしれません。

運用ルールと技術的な観点から 24 時間までにトランザクションを保持可能な場合は、Amazon は非同期モードを推奨します。オーソリのステータスが **Declined** で返ってきた場合は、購入者にトランザクションが失敗したことの通知が必要になり、Amazon Pay の Web サイトから支払方法の変更を彼らに要求し、代替りの支払方法を取得します。または、**Declined** の理由コードよりトランザクションをキャンセルします。詳しい情報は「支払失敗のハンドリング」を参照してください。

このモードを選択することで、より多くのトランザクション調査の時間を Amazon に提供し通常は低いオーソリ失敗率の結果になります。

同期モード

TransactionTimeout リクエストパラメータに 0 を設定することで、同期モードを選択できます。

次のコードサンプルは **AuthorizeOnBillingAgreement** 処理で同期モードを有効にする方法を表しています。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01
?AWSAccessKeyId=AKIAJKYFSJU7PEXAMPLE
&Action=AuthorizeOnBillingAgreement
&AmazonBillingAgreementId=C01-1234567-1234567
&AuthorizationAmount.Amount=10
&AuthorizationAmount.CurrencyCode=USD
&AuthorizationReferenceId=test_authorize_1
&MWSAuthToken=amzn.mws.4ea38b7b-f563-7709-4bae-87aeaEXAMPLE
&SellerId=YOUR_SELLER_ID_HERE
&SellerAuthorizationNote=ForNovemberOrder
&SellerOrderAttributes.SellerOrderId=testSellerOrderId
&SellerOrderAttributes.StoreName=testStore
&SellerOrderAttributes.CustomInformation=ExampleInformation
&InheritShippingAddress=true
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2012-10-03T19%3A01%3A11Z
&TransactionTimeout=0
&Version=2013-01-01
&Signature=WlQ708aqyHXMkoUBk69Hjxj8qdh3aDcqpY71hVgEXAMPLE
```

同期モードの処理では、**AuthorizeOnBillingAgreement** リクエストのレスポンス内でのオーソリオブジェクトは **Open** か **Declined** を約 6～8 秒以内に返します。Amazon がオーソリ同期を処理することが出来ない場合は、処理は **TransactionTimedOut** の理由コードと同時に **Declined** 返します。

購入者がサイトに滞在している間にオーソリまたは売上請求支払を行いたいシナリオを持っている場合は、Amazon Pay は同期モードを推奨します。例えば、デジタルダウンロードを提供したい場合、または、すぐの配送を保証している場合です。

このモードを選択することで、高いオーソリの失敗率で Amazon がいくつかのオーソリを **Pending** から **Declined** に遷移することをよく見るかもしれません。これらのオーソリ失敗は **TransactionTimedOut** の理由コードを確認することで判断できます。同期モードでは、購入者がサイトに滞在している間に失敗をハンドリングできます。詳しい情報は「支払失敗のハンドリング」を参照してください。

アドレス帳ウィジェット無しのインテグレーション

いくつかのシナリオでは、アドレス帳ウィジェットから購入者住所を取得する必要がないかもしれません。配送先住所が必要ない場合（例えば、デジタル商材を販売している場合）や、販売事業者が購入者アドレスを取得したい場合では、購入者情報を取得するためにアドレス帳ウィジェットを利用する必要がないかもしれません。

注意：購入者からの住所を取得するために Amazon のアドレス帳ウィジェットを利用しない場合は、トランザクションは Amazon Pay の支払保護ポリシーに準拠されません。Amazon Pay の支払保護ポリシーの詳しい情報は、Amazon Pay の Web サイトの Amazon Pay カスタマー契約を参照してください。

Amazon Pay の支払保護ポリシーは物品販売のみ適用します。

このケースでは、Billing Agreement オブジェクトを生成しなければなりません。これを行うには、ウィジェット呼び出しのパラメータの1つである **onReady** コールバック機能を指定する必要があります。onReady 機能はウィジェットが表示された時に **Billing Agreement** オブジェクトと同時に呼びだされます。**Billing Agreement** オブジェクト上の **getAmazonBillingAgreementId()** 関数を呼び出すことで **billingAgreementId** を取得できます。

Amazon Pay のインテグレーションのいたるところで **amazonBillingAgreementId** が必要な場合は、それを保存しなければなりません。

次のサンプルはアドレス帳ウィジェットを必要としないシナリオをサポートするためのお支払い方法ウィジェットの表示方法です。

```
<!-- put the style below inside your CSS file -->
<style type="text/css">
  #walletWidgetDiv {width: 400px; height: 228px;}
</style>
```

```

<div id="walletWidgetDiv">
</div>

<script>
  new OffAmazonPayments.Widgets.Wallet({
    sellerId: 'YOUR_SELLER_ID_HERE',
    onReady: function(billingAgreement) {
      var billingAgreementId =
billingAgreement.getAmazonBillingAgreementId();
    },
    agreementType: 'BillingAgreement',
    design: {
      designMode: 'responsive'
    },
    onPaymentSelect: function(billingAgreement) {
      // Replace this code with the action that you want to perform
      // after the payment method is selected.
    },
    onError: function(error) {
      // your error handling code
    }
  }).bind("walletWidgetDiv");
</script>

```

これは「Step3：アドレス帳とお支払い方法ウィジェットの追加」と似ており、CSSによってウィジェットのサイズを設定します。お支払い方法ウィジェットは提供されたスペースに適合するためにリサイズされます。ブラウザウィンドウのサイズでリサイズするためのウィジェットが必要なレスポンシブサイトを構築したい場合はメディアクエリーを利用できます。

スマートフォン用の折りたためるウィジェットの利用

すべての Amazon Pay のウィジェットはタブレットとスマートフォンのユーザビリティを改善するためにタッチスクリーン向けに最適化されています。しかしながら、スマートフォンに最適化された Web サイトを分けたい場合は、Web サイトで折りたためるウィジェットを利用することを Amazon は推奨します。

折りたためるアドレス帳とお支払い方法ウィジェットは 1 カラムユーザーインターフェイスに適合するためのデザインがされており、スクロールが不要な最小サイズに拡げることや折りたためることができます。最小化するためのウィジェットを割り当てるスペースを保持したい場合と購入者のデバイス上にて全画面エリアで利用したい場合では最適です。これは高さや横幅が固定されている、または、購入者によって拡げることや折りたためることができない標準のウィジェットとは異なります。標準のウィジェットは「イントロダクション」で説明しています。

折りたためるアドレス帳とお支払い方法ウィジェットはスマートフォン画面の横幅に適合するために調整されます。折りたためるウィジェットは、折りたたみ時では 135 ピクセルの高さで固定します。購入者は事前に選択された配送先住所または支払方法を変更するためにウィジェ

ットを拡げることができます。このケースでは、ウィジェットはデバイスの最大の高さと同幅で広がります。購入者が選択できた場合は、ウィジェットは自動的に折りたたみされ、購入者は注文を続けることができます。

折りたたみウィジェットをインテグレーションする方法は次のステップの通りです。

スマートフォンに最適化された各ページの head セクションに meta タグを追加します。この meta タグはスマートフォン上でユーザーがページズームの要求なしの参照可能なウィジェットになります。

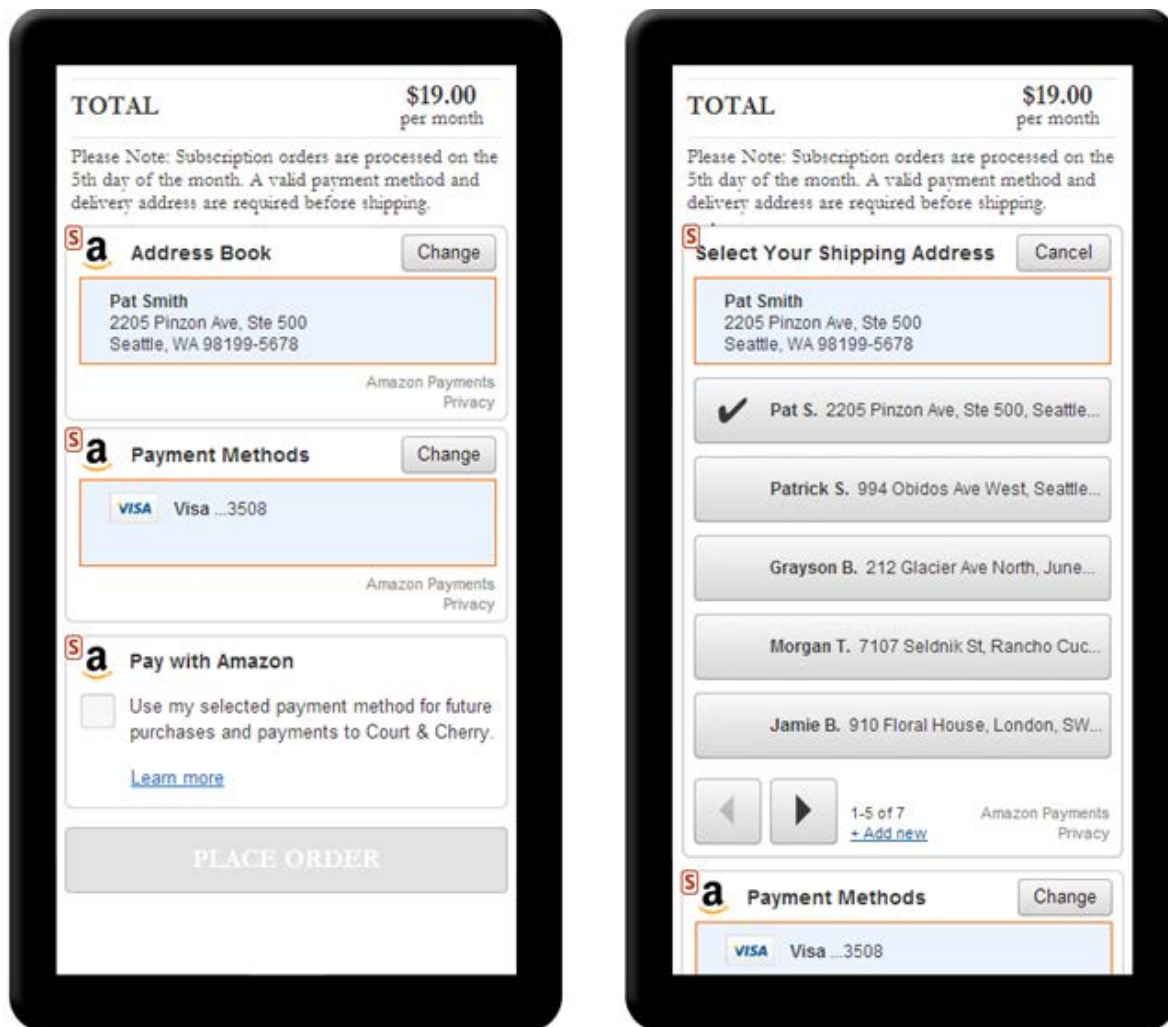
```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0"/>
```

designMode パラメータに *smartphoneCollapsible* をセットするために、**アドレス帳とお支払い方法** ウィジェットを生成するためのコードを次のサンプルのように変更します。

```
new OffAmazonPayments.Widgets.AddressBook({
  sellerId : 'YOUR_SELLER_ID_HERE',
  onReady: function(billingAgreement) {
    var billingAgreementId = billingAgreement.getAmazonBillingAgreementId();
  },
  design : {
    designMode: 'smartphoneCollapsible'
  },
  agreementType: 'BillingAgreement',
  onAddressSelect : function(billingAgreement) {
  },
  onError : function(error) {
  }
}).bind("addressBookWidgetDiv");
```

上記のサンプルは**アドレス帳**ウィジェットの変更方法を表します。**お支払い方法**ウィジェットと参照のみの**アドレス帳とお支払い方法**ウィジェットのために同じ変更が必要になります。折りたたみウィジェットの横幅と縦幅を指定することができません。

- 次の実例は購入者が確認できる折りたたみウィジェットの方法を表します。
- 最初、ウィジェットは折りたたまれています。ウィジェットは Amazon のデフォルトの配送先住所と支払方法を表示します。
- 購入者が配送先住所、または、支払方法を変更したい場合は、購入者は**変更**ボタンをクリックします。ウィジェットはそれから画面全体に広がります。



高度な支払シナリオの可能性

Billing Agreement では、様々な購買と高度な支払シナリオを容易に処理できます。次のシナリオがカバーされます。

アップグレードまたはダウングレードサービス

購入者が定期販売やサービスプランのアップグレードまたはダウングレードを要求する場合は、顧客に増額や減額を進めるための変更が必要であるかもしれません。また、購入者が請求サイクル間の途中で問合せした場合は、請求期間の最後まででの請求を按分する必要があるかもしれません。

Billing Agreement をこのシナリオを処理するための柔軟性を提供します。必要な情報と共に請求システムを更新した後、金額を充当するための売上請求オペレーションを呼び出すことで、購入者に請求するための要求を簡単に始めることができます。

注意：Amazon は暦月ごとに購入者へ請求可能な資金額の制限として 50,000 円を設定しています。

キャンセルまたはサスペンドサービス

定期支払を休止したいと購入者から問合せがあった場合は、再開されるまで Amazon への支払リクエストの承認を容易に止めることができます。購入者がこれらのサービスをキャンセルすると問合せがあった場合は、**CloseBillingAgreement** 処理を呼び出して Billing Agreement を Close することを Amazon は推奨します。Billing Agreement ステータスはそれから Amazon Pay の Web サイトのマイページ内で更新されます。

次のサンプルコードは **CloseBillingAgreement** 処理を要求する方法を表します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01?AWSAccessKeyId=AKIAJKYFSJU7PEXAMPLE
&AmazonBillingAgreementId=C01-8824045-7416542
&ClosureReason=Closing%20OR%20for%20Test
&Action=CloseBillingAgreement
&SellerId=YOUR_SELLER_ID_HERE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2013-12-11T12%3A32%3A42.000Z
&Version=2013-01-01
&Signature=yRpMpoDfGLu567t611z27v4yJ8SURIVMKcy26sJrwYc%3D
```

購入者は Amazon Pay の Web サイトで **Billing Agreement** を終了することができます。

Billing Agreement が **Closed** になった時、購入者は支払いを停止した案内メールを受け取ります。

注意：Amazon Pay は送信されるメール内のいくつかの情報を指定することを許可しています。例えば、カスタマーサービスメール、電話番号、メモです。これらの詳細についての詳しい情報は、「提供可能な購入者向けのメールコンテンツ」を参照してください。

これからのケース全てでは、インスタント支払通知 (IPN) を受け取ります。IPN の詳しい情報は、「システムと Amazon Pay の同期」を参照してください。

支払方法の変更

購入者が **Billing Agreement** に関連した支払方法を前回の支払方法の失敗やその他の理由で更新したい場合は、Amazon Pay Web サイトか販売事業者の Web サイトから支払方法を変更できます。

注意：購入者が支払方法を変更する場合、同意ウィジェットは表示しません。購入者が選択したものを更新する場合、同意の確認は必要ありません。ただし、購入者は Amazon Pay の Web サイト上で **Billing Agreement** の同意を編集するかもしれません。

Amazon Pay の Web サイトから支払方法を変更

Amazon Pay の Web サイトのアカウントセクションから **Billing Agreement** が **Open** または **Suspended** 状態の場合は、購入者は支払方法を更新できます。購入者が支払方法を変更する場合、販売事業者はインスタント支払通知 (IPN) を受け取ります。IPN の詳しい情報は、「システムと Amazon Pay の同期」を参照してください。次のオーソリは更新された支払方法で処理されます。前回のオーソリに関連される支払方法は更新されません。

サイトから支払方法を変更

購入者がサイトから支払方法の変更を選ぶ場合は、お支払い方法ウィジェットを利用できます。その為には次の通りです。

- サイト上にログインするための「Amazon アカウントでログイン」ボタンを配置して購入者がログインします。
- 購入者が支払方法を更新する場所に Amazon のお支払い方法ウィジェットを表示します。詳しくは、「Step3：アドレス帳とお支払い方法ウィジェットの追加」を参照してください。購入者が編集を希望する **Billing Agreement** を識別するために現在選択された **amazonBillingAgreementId** を確かめてください。
- 購入者はお支払い方法ウィジェットから支払方法を選択します。Amazon は **onPaymentSelect** コールバックをトリガーします。
- 最後の選択を決める前に購入者はウィジェット内のいくつかの支払方法を選択するかもしれません。購入者が購入完了、更新、または似たボタンをクリックすることで最終の選択を行った後で、**ConfirmBillingAgreement** を呼び出します。

ConfirmBillingAgreement 処理の呼び出しに成功した後に、Billing Agreement オブジェクトは **Draft** 状態から **Open** 状態に遷移します。**Draft** 状態でない Billing Agreement オブジェクトに **SetBillingAgreementDetails** を呼び出さないでください、また、追加の属性をセットしないでください。

支払方法が更新されたインスタント支払通知 (IPN) を受け取り、また、購入者は販売事業者更新メールを受信します。

配送先住所の変更

購入者が **Billing Agreement** に関連された配送先住所を更新したい場合は、**Web** サイトから **Amazon** のアドレス帳ウィジェットを利用できます。現在、購入者は **Amazon Pay** の **Web** サイトから配送先住所を更新できません。

配送先住所を変更するためには次の通りです。

- サイト上に「**Amazon** アカウントでログイン」ボタンを配置して購入者がログインします。
- 購入者が住所を更新するための場所に **Amazon** のアドレス帳ウィジェットを表示します。詳しくは、「**Step3** : アドレス帳とお支払い方法ウィジェットの追加」を参照してください。
- 購入者は**アドレス帳**ウィジェットから配送先住所を選択します。**Amazon** は **onAddressSelect** コールバックをトリガーします。
- **Amazon** は購入者に支払方法の選択を毎回要求します。**Amazon** のお支払い方法ウィジェットを提供し、購入者は住所に関連し許可された支払方法を選択します。毎回、購入者は支払方法を選択します。**Amazon** は**お支払い方法**ウィジェットからの **onPaymentSelect** 呼び出しをトリガーします。常に **onPaymentSelect** を次のステップである **Billing Agreement** の承認処理の前で待機してください。購入者が支払方法を選択する前に **Billing Agreement** の承認を試みた場合は、**ConfirmBillingAgreement** 処理は **PaymentMethodNotUpdated** エラーを返します。購入者が支払方法を変更した場合は、支払方法が変更された **IPN** を受け取ります。
- 購入者は最終選択を決める前にウィジェット内でいくつかの住所を選択するかもしれません。購入者が**更新**やよく似たボタンをクリックして最終選択を決めた後で、**ConfirmBillingAgreement** 処理を呼び出します。

ConfirmBillingAgreement 処理の呼び出しに成功した後に、**Billing Agreement** オブジェクトは **Draft** 状態から **Open** 状態に遷移します。**Draft** 状態でない **Billing Agreement** オブジェクトに **SetBillingAgreementDetails** を呼び出さないでください、また、追加の属性をセットしないでください。

住所が更新されたインスタント支払通知 (**IPN**) を受け取ります。購入者が支払方法の変更を行った場合は、販売事業者はその他の支払方法変更のインスタント支払通知 (**IPN**) も受け取ります。

- **ConfirmBillingAgreement** 処理を呼び出した後でのみ、新しい住所と支払方法について **BillingAgreement** は更新されることに注意してください。**ConfirmBillingAgreement** 処理を呼び出した後でのみ、**GetBillingAgreement** 処理を使って新しい住所を取得できます。
- 購入者は支払方法を更新した時に **Amazon** からメールを受信します。

ワンタイムと定期の同時支払

Billing Agreement は、定期支払と同様にワンタイムを処理することが可能です。例えば、雑誌の定期購読中（購入者が次の 24 ヶ月の間で毎月支払）に単品の書籍（ワンタイム支払）を購入す

ことができます。購入者が両方の商品を購入する場合は、インテグレーションエクスペリエンスは上で記述しているものと同じになります。Billing Agreement を保有している時は、定期支払とワンタイム購入の両方を購入者に請求することができます。

しかしながら、購入フローの最後のステップで可能ですが、購入者が定期購入を不安になることがあります。このケースでは、おそらく購入者は定期支払同意ウィジェットからの定期支払を承認されないでしょう。それによって、販売事業者は **Draft** 状態の Billing Agreement オブジェクトを承認することができないかもしれません。この状況下でも、Amazon Pay はワンタイム購入の Order Reference の生成は可能です。その為には、**ConfirmNow** パラメータに **true** の値を指定して、**CreateOrderReferenceForId** 処理を呼び出します。このオペレーションは、**Draft** の Billing Agreement からの支払と配送先住所を利用した Order Reference を承認します。Amazon は **Draft** の Billing Agreement から 1 つの Order Reference オブジェクトのみ生成することを許可することに注意してください。詳細は **Amazon Pay** の API リファレンスガイドの **CreateOrderReferenceForId** 処理を参照してください。

支払失敗のハンドリング

オーソリ、売上計上、返金では失敗することが起こります。**Amazon Pay** は、これらの失敗を処理するための支援、購入者が更新や支払方法の変更を行うためのいくつかの運用ルールがあります。

オーソリの失敗

AuthorizeOnBillingAgreement 処理の呼び出しが失敗した場合は、レスポンス内に以下の 4 つの理由コード内の 1 つが含まれ参照できます。

同期モード

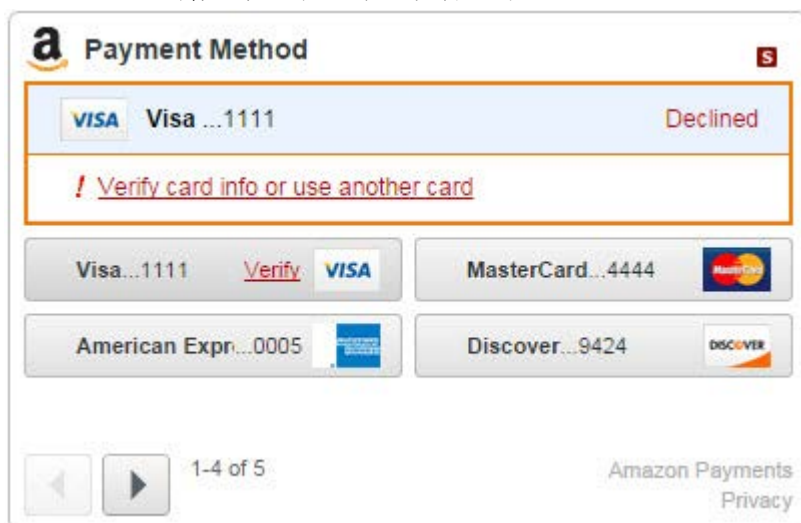
InvalidPaymentMethod

選択された支払方法が問題であることを **InvalidPaymentMethod** は示します。このシチュエーションでは、Billing Agreement は **Suspended** 状態に遷移しインスタント支払通知 (IPN) を送信します。購入者が Web サイト上での購入フローが完了するまでに、この失敗をハンドリングすることを Amazon は推奨します。

- Web サイト上の購入フロー内で、購入者に選択された支払方法が失敗したことを通知するためにエラーメッセージを表示します。
- ページにお支払い方法ウィジェットをレンダリングします。Web サイトでのサンプルコードは「**Step3** : アドレス帳とお支払い方法ウィジェットの追加」を参照してください。レンダリングした時にセットした **amazonBillingAgreementId** の値を現在の Billing Agreement の ID をお支払い方法ウィジェットに結びつけます。それを行った場合、Amazon Pay は前回選択した支払方法を **Declined** にマークします。購入者がウィジェットから支払方法やその他支払方法を選択できるようにするために、支

払情報を更新するための **Declined** リンクをクリックできるようテキスト情報を提供します。

以下のスクリーンショットは、お支払い方法ウィジェットが適切に失敗情報をレンダリングしたことを購入者へ表示する失敗メッセージのサンプルです。



- 購入者が支払方法の変更や新しい支払方法を選択した後は、Amazon はお支払い方法ウィジェットから **onPaymentSelect** コールバックをトリガーします。
- 購入者は最終選択を決定する前にウィジェット内でいくつかの支払方法をクリックするかもしれません。
購入者がお買い物の途中で**注文を完了する**、**更新**、またはよく似たボタンをクリックすることで最終選択を行った場合は、**ConfirmBillingAgreement** の API を呼び出します。
ConfirmBillingAgreement 処理の呼び出しに成功した後は、Billing Agreement オブジェクトは **Draft** 状態から **Open** 状態へ遷移します。
支払方法を更新したことを示すインスタント支払通知 (IPN) を受け取り、購入者は支払更新メールを受信します。
- 注文が **Open** 状態である場合は、**AuthorizeOnBillingAgreement** の API を利用して新しいオーソリを試みても問題ありません。呼び出しに成功した場合は、購入者は支払確認メールを受信します。詳しい情報は「提供可能な購入者向けのメールコンテンツ」を参照してください。

AmazonRejected

AmazonRejected は Amazon によって判断した結果 **AuthorizeOnBillingAgreement** 処理が失敗したことを示します。注文は **Closed** 状態に遷移し、トランザクションを失敗しなければなりません。

ProcessingFailure

ProcessingFailure は Amazon の内部処理エラーのためにトランザクションを処理できなかったことを示します。1、2分以内にリクエストをリトライしてください。注文は **Open** 状態になるかもしれません。リトライに成功しなかった場合はさらに数分以内に再度注文を行ってもらうように購入者に確認してください。

TransactionTimeout

TransactionTimedOut は Amazon がリクエストを 13 秒以内に処理できず、オーソリリクエストが失敗したことを示します。

通常、購入者アクションは新しい注文をリトライ、または、支払手段の更新では、この失敗理由を解決することはできません。非同期モード (Open で存続する注文) でリトライするか購入者のトランザクションをキャンセルするかのどちらかができます。

非同期モード

InvalidPaymentMethod

InvalidPaymentMethod は選択された支払方法に問題があることを示します。このシチュエーションでは、Billing Agreement は **Suspended** 状態に遷移し、インスタント支払通知 (IPN) を送信します。Amazon はこのシナリオをすでに購入者が Web サイトから離れていると仮定してハンドリングすることを推奨します。

- 購入者に支払方法を更新するようにメールか電話で問合せします。購入者は Amazon Pay の Web サイトに訪問し、販売事業者のオーソリを調べ、Web ページ上で支払方法を更新します。
- 購入者が支払方法を更新した後、Billing Agreement オブジェクトは **Suspended** 状態から **Open** 状態に遷移します。Amazon Pay はそれからインスタント支払通知 (IPN) を送信し、購入者はメールを受信します。

AmazonRejected

AmazonRejected は Amazon によって決定したことで **AuthorizeOnBillingAgreement** 処理が失敗したことを示します。注文は **Closed** 状態へ遷移され、あなたはトランザクションを失敗にしなければなりません。

ProcessingFailure

ProcessingFailure は Amazon の内部処理エラーのためにトランザクションを処理できなかったことを示します。1、2 分以内にリクエストをリトライしてください。注文は **Open** 状態になるかもしれません。リトライに成功しなかった場合はさらに数分以内に再度注文を行ってもらうように購入者に確認してください。

TransactionTimeout

TransactionTimedOut は、**AuthorizeOnBillingAgreement** 処理の呼び出しがデフォルトのタイムアウト時間の 24 時間か、TransactionTimeout パラメータ内で指定した時間内に処理できなかったことを示します。

売上請求の失敗

売上計上の呼び出しに失敗した場合は、レスポンスに以下の2つの理由コードの内の1つが含まれ参照できます。

AmazonRejected

AmazonRejected は売上請求の処理は Amazon によって失敗になったことを示します。Amazon が売上請求を拒否した場合は、オーソリオブジェクトと Billing Agreement オブジェクトは Closed になります。オーソリが Closed の場合は、売上請求リクエストは失敗になります。Billing Agreement がまだ **Open** 状態の場合は、新しいオーソリをリクエストし、それからオーソリから売上請求をリクエストします。

ProcessingFailure

ProcessingFailure は Amazon の内部処理エラーのためにトランザクションを処理できなかったことを示します。1、2分以内にリクエストをリトライしてください。

このケースでは、Billing Agreement オブジェクトはまだ **Open** であり、Amazon は新しい **AuthorizeOnBillingAgreement** 処理を呼び出し確認し、続いて新しい売上請求処理の呼び出しを推奨します。

返金の失敗

返金処理の呼び出しが失敗した場合は、レスポンスに以下の2つの理由コードの内の1つが含まれ参照できます。

AmazonRejected

AmazonRejected は Amazon によって返金処理が失敗になったことを示します。このケースでは、販売事業者は購入者に他の方法で返金の問題（例えば、登録されているクレジットカードを解決するために）について働きかけをしなければなりません。

ProcessingFailure

ProcessingFailure は次の1つを示します。

- Amazon は内部処理エラーのためにトランザクションを処理できませんでした。
- 購入者はすでに Amazon マーケットプレイス保証、または、チャージバックから返金を受け取っています。

レスポンス内で **ProcessingFailure** の理由コードを受け取っている場合は、最初にセラーセントラルのパフォーマンスをクリックして、注文の Amazon マーケットプレイス保証とチャージバックステータスをチェックしてください。購入者から Amazon マーケットプレイス保証またはチャージバックが発生しており、売上請求オブジェクトがまだ **Completed** 状態である場合は、リクエストを1～2分内でリトライします。リトライの試みが成功しなかった場合は、購入者に問合せし、他の方法で返金を解決します。

システムと Amazon Pay の同期

Amazon Pay オブジェクト状態の遷移

Amazon Pay オブジェクト（Billing Agreement オブジェクト、Order Reference オブジェクト、オーソリオブジェクト、売上請求オブジェクト、返金オブジェクト）は様々支払処理の特定のステータスに遷移します。これらの状態遷移はシステムや Amazon 内部のビジネスルールによって特定の処理が実行されます。

例えば、非同期モードでリクエストしたオーソリオブジェクトは、最初に **Pending** 状態になって処理が開始されます。処理が完了した後は、それは **Open** 状態に遷移し、資金を売上請求する準備になったことを示します。オーソリオブジェクトが売上請求された後は、それは **Closed** 状態に遷移し、オーソリに対して売上請求できる資金が無いことを示します。あるいは、30 日以内にオーソリを売上請求しなかった場合は、Amazon は **Closed** にします。インテグレーションエラーを防止するために、システムと Amazon のシステムの Amazon Pay オブジェクトの現在の状態を同期する必要があります。

Amazon Pay オブジェクト状態遷移についての詳細情報は、Amazon Pay API レファレンスガイド内のステータスと理由コードを参照してください。

さらに、Amazon は購入が完了した後に購入者が支払方法の更新や変更を許可しています。これは Billing Agreement を更新した結果かもしれません。カスタマーサービスとビジネス処理をサポートするためにこのような変更について知りたいかもしれません。

API 呼び出しの最初のレスポンス

最初のリクエストが実行された時のレスポンス結果はそれぞれの処理によって異なります。

- **AuthorizeOnBillingAgreement** — この呼び出しの非同期レスポンスは、常に **Pending** 状態を返します。同期の **AuthorizeOnBillingAgreement** リクエストは **Open** か **Declined** を返します。
- **Capture** — **売上請求** はリアルタイムで処理されます。この期間では、売上請求リクエストの同期レスポンスは **Completed** か **Declined** の状態を返します。
- **Refund** — 返金リクエストはリアルタイムで処理されません。返金リクエストの最初のレスポンスは常に **Pending** です。

AuthorizeOnBillingAgreement、**Capture**、**Refund** の API 呼び出しの最初のレスポンスを次の表で表します。

APC 呼び出し	API 呼び出しの最初のレスポンス
AuthorizeOnBillingAgreement	同期モード: Open か Declined のどちらか 非同期モード: 常に Pending

Capture	Completed 、 Pending 、 Declined のいずれか
Refund	常に Pending

インスタント支払通知メッセージハンドリングのベストプラクティス

取得 API 呼び出しの実行

Billing Agreement オブジェクトや **Order Reference** オブジェクト等の変更が行われた時に、Amazon は非同期のインスタント支払通知 (IPN) メッセージを送信します。多くの要求によってこれらの通知が送信され、システム内で状態を更新したり、フルフィルメントシステムで注文管理を行う場合に利用できます。インスタント支払通知 (IPN) を受け取った後でのベストプラクティスはそれぞれの通知を受け取ったオブジェクトの取得処理を実行することです。

セキュアなインスタント支払通知 (IPN) 処理

なりすまし攻撃を防御するためには、メッセージの安全性を確かめインスタント支払通知 (IPN) 署名を確認しなければなりません。

Amazon の SDK を利用している場合はこれらの確認は取り扱われています。利用していない場合は、AmazonSNS メッセージ署名の確認で記述している、または、SDK で導入されているのと同様の確認を利用してください。

Amazon Pay オブジェクト状態遷移のモニタリング

Amazon Pay オブジェクトの状態遷移をモニタリングするためにインスタント支払通知 (IPN) を利用できます。インスタント支払通知 (IPN) は Amazon Pay オブジェクトの状態が変更されると自動的に通知されます。インスタント支払通知 (IPN) を受け取った後は、あなたは Amazon Pay オブジェクトの完全な詳細を得るために、**GetBillingAgreementDetails**、**GetAuthorizationDetails**、**GetCaptureDetails**、**GetRefundDetails** 処理を利用して情報を取得できます。

インスタント支払通知 (IPN) のサンプル

多くの Amazon Pay オブジェクトや **Billing Agreement** オブジェクトの変更された状態を Amazon は通知として送ります。多くの要求によってこれらの通知が送信され、システム内で状態を更新したり、フルフィルメントシステムで注文管理を行う場合に利用できます。

セラーセントラルにアクセスして、設定のインテグレーション設定ページに通知エンドポイントをセットアップできます。

例えば、オーソリオブジェクトが **Pending** 状態から **Open** 状態へ遷移されたことを Amazon が通知します。この遷移は、**AuthorizeOnBillingAgreement** 処理の呼び出しが成功し、注文をフルフィルメント処理を進めることができることを確認したことになります。

それぞれの通知を受け取った後は、Amazon へ“200 OK”のレスポンスを直ちに送信するようにエンドポイントを構成しなければなりません。このレスポンスを送らなかった場合は、Amazon は 14 日の間に毎時間リトライを実行します。サーバがダウンしている場合は、4XX のエラーメッセージが送信され、Amazon からリトライされません。

注意：ほとんどの場合、それぞれの通知は、販売事業者のアプリケーションに正確に通達されます。また、Amazon の理由により一時的に重複したメッセージが送信されます。複数のインスタント支払通知 (IPN) メッセージを受信しても支払ワークフローが誤動作しないように、アプリケーションを設計してください。

NotificationData メンバーのコンテンツは次の一般的に直ぐに利用できる XSD によって記述されて、インスタント支払通知 (IPN) の中で返されます。

https://amazonpayments.s3.amazonaws.com/documents/payments_ipn.xsd.

全てのインスタント支払通知 (IPN) は Amazon によって署名されて返ります。

次は Amazon から利用できる通知のリストです。

- **BillingAgreementNotification**
- **OrderReferenceNotification**
- **AuthorizationNotification**
- **CaptureNotification**
- **RefundNotification**

これらのインスタント支払通知 (IPN) は Amazon Pay オブジェクトのステータスが更新された場合に発生します。さらに、購入者が支払方法、または、Billing Agreement に関連する住所を更新する場合もインスタント支払通知 (IPN) を送信します。**BillingAgreementNotification** の通知が送信された理由の説明として **EventType** パラメータを含みます。**EventType** パラメータは4つの可能な値を持っています。

- *StatusUpdated*
- *PaymentMethodUpdated*
- *AddressUpdated*
- *AddressAndPaymentMethodUpdated*

次のサンプルは **BillingAgreementNotification** を表します。


```

</BillingAgreement>\n
</BillingAgreementNotification>","Timestamp":"2014-01-11T01:44:42Z"}
",
"Timestamp" : "2013-04-22T06:00:15.108Z",
"SignatureVersion" : "1",
"Signature" : "deako5R0...CVmPQOI=",
"SigningCertURL" : "https://sns.EXAMPLE.amazonaws.com/
SimpleNotificationService-f3ecfb7224c7233fe7bb5f59fEXAMPLE.pem",
"UnsubscribeURL" : "https://sns.EXAMPLE.amazonaws.com/
?Action=Unsubscribe
&SubscriptionArn=arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic:GUID"
}

```

次のサンプルは **OrderReferenceNotification** を表します。

```

POST / HTTP/1.1
x-amz-sns-message-type: Notification
x-amz-sns-message-id: 432f33bf-9f84-5004-815f-7a6cfEXAMPLE
x-amz-sns-topic-arn: arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic
x-amz-sns-subscription-arn:
    arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic: EXAMPLE
Content-Length: 961
Content-Type: text/plain; charset=UTF-8
Host: ec2-EXAMPLE.compute-1.amazonaws.com
Connection: Keep-Alive
User-Agent: Amazon Simple Notification Service Agent
{
  "Type" : "Notification",
  "MessageId" : "cf5543af-dd65-5f74-8ccf-0a410EXAMPLE",
  "TopicArn" : "arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic",
  "Message" :
    "{ \"NotificationReferenceId\" : \"32d195c3-a829-4222-b1e2-14ab2EXAMPLE\",
      \"NotificationType\" : \"OrderReferenceNotification\",
      \"SellerId\" : \"YOUR_SELLER_ID_HERE\",
      \"ReleaseEnvironment\" : \"Sandbox\",
      \"Version\" : \"2013-01-01\",
      \"NotificationData\" :
        \"<?xml version=\\\"1.0\\\" encoding=\\\"UTF-8\\\"?>
        <OrderReferenceNotification
          xmlns=\\\"https://mws.amazonservices.com/
            ipn/OffAmazonPayments/2013-01-01\\\">\\n
          <OrderReference>\\n
          <AmazonOrderReferenceId>
            S23-1234567-1234567
          <\\!/AmazonOrderReferenceId>\\n
          <OrderTotal>\\n
          <Amount>106.00<\\!/Amount>\\n
          <CurrencyCode>USD<\\!/CurrencyCode>\\n
          <\\!/OrderTotal>\\n
          <OrderReferenceStatus>\\n
          <State>CLOSED<\\!/State>\\n
          <ReasonCode>SellerClosed<\\!/ReasonCode>\\n
          <LastUpdateTimestamp>
            2013-04-01T10:49:59.532Z
        \"
    }

```

```

<\\LastUpdateTimestamp>\\n
<\\OrderReferenceStatus>\\n
<CreationTimestamp>2013-03-30T09:58:51.234Z<\\CreationTimestamp>\\n
<ExpirationTimestamp>
  2013-04-06T09:58:51.234Z
<\\ExpirationTimestamp>\\n
<\\OrderReference>\\n
<\\OrderReferenceNotification>\",
  \"Timestamp\": \"2013-04-22T06:00:14Z\"}],
\"Timestamp\" : \"2013-04-22T06:00:15.108Z\",
\"SignatureVersion\" : \"1\",
\"Signature\" : \"deako5R0...CVmPQOI=\",
\"SigningCertURL\" : \"https://sns.EXAMPLE.amazonaws.com/
  SimpleNotificationService-f3ecfb7224c7233fe7bb5f59fEXAMPLE.pem\",
\"UnsubscribeURL\" : \"https://sns.EXAMPLE.amazonaws.com/
  ?Action=Unsubscribe
  &SubscriptionArn=arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic:GUID\"
}

```

次のサンプルは **AuthorizationNotification** を表します。

```

POST / HTTP/1.1
x-amz-sns-message-type: Notification
x-amz-sns-message-id: 6f7e123e-49c9-5c9d-a389-5bed0EXAMPLE
x-amz-sns-topic-arn: arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic
x-amz-sns-subscription-arn:
arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic: EXAMPLE
Content-Length: 961
Content-Type: text/plain; charset=UTF-8
Host: ec2-EXAMPLE.compute-1.amazonaws.com
Connection: Keep-Alive
User-Agent: Amazon Simple Notification Service Agent
{
  \"Type\" : \"Notification\",
  \"MessageId\" : \"cf5543af-dd65-5f74-8ccf-0a410EXAMPLE\",
  \"TopicArn\" : \"arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic\",
  \"Message\" :
    \"{\"NotificationReferenceId\": \"32d195c3-a829-4222-b1e2-14ab28909513\",
      \"NotificationType\": \"PaymentAuthorize\",
      \"SellerId\": \"YOUR_SELLER_ID_HERE\",
      \"ReleaseEnvironment\": \"Sandbox\",
      \"Version\": \"2013-01-01\",
      \"NotificationData\":
        \"<?xml version=\\\"1.0\\\" encoding=\\\"UTF-8\\\"?>
          <AuthorizationNotification
            xmlns=\\\"https://mws.amazonservices.com/
              ipn/OffAmazonPayments/2013-01-01\\\">\\n
            <AuthorizationDetails>\\n
            <AmazonAuthorizationId>
              S23-1234567-1234567-0000001
            <\\AmazonAuthorizationId>\\n
            <AuthorizationReferenceId>
              9bbe88cd5ab4435b85d717fd8EXAMPLE
            <\\AuthorizationReferenceId>\\n

```

```

    <AuthorizationAmount>\\n
    <Amount>5.0<\\//Amount>\\n
    <CurrencyCode>USD<\\//CurrencyCode>\\n
    <\\//AuthorizationAmount>\\n
    <CapturedAmount>\\n
    <Amount>0.0<\\//Amount>\\n
    <CurrencyCode>USD<\\//CurrencyCode>\\n
    <\\//CapturedAmount>\\n
    <AuthorizationFee>\\n
    <Amount>0.0<\\//Amount>\\n
    <CurrencyCode>USD<\\//CurrencyCode>\\n
    <\\//AuthorizationFee>\\n
    <IdList/>\\n
    <CreationTimestamp>2013-04-22T05:59:38.186Z<\\//CreationTimestamp>\\n
    <ExpirationTimestamp>
      2013-05-22T05:59:38.186Z
    <\\//ExpirationTimestamp>\\n
    <AuthorizationStatus>\\n
    <State>Open<\\//State>\\n
    <LastUpdateTimestamp>
      2013-04-22T06:00:11.473Z
    <\\//LastUpdateTimestamp>\\n
    <\\//AuthorizationStatus>\\n
    <OrderItemCategories/>\\n
    <CaptureNow>>false<\\//CaptureNow>\\n
    <SoftDescriptor/>\\n
    <\\//AuthorizationDetails>\\n
    <\\//AuthorizationNotification>\",
    \"Timestamp\": \"2013-04-22T06:00:14Z\"}],
    \"Timestamp\" : \"2013-04-22T06:00:15.108Z\",
    \"SignatureVersion\" : \"1\",
    \"Signature\" : \"W/cfaDzC...5glwqJk=\",
    \"SigningCertURL\" : \"https://sns.EXAMPLE.amazonaws.com/
      SimpleNotificationService-f3ecfb7224c7233fe7bb5f59fEXAMPLE.pem\",
    \"UnsubscribeURL\" : \"https://sns.EXAMPLE.amazonaws.com/
      ?Action=Unsubscribe
      &SubscriptionArn=arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic:GUID\"
  }

```

次のサンプルは **CaptureNotification** を表します。

```

POST / HTTP/1.1
x-amz-sns-message-type: Notification
x-amz-sns-message-id: 64f5f75c-5799-53e5-b4c3-be8f1EXAMPLE
x-amz-sns-topic-arn: arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic
x-amz-sns-subscription-arn:
arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic: EXAMPLE
Content-Length: 961
Content-Type: text/plain; charset=UTF-8
Host: ec2-EXAMPLE.compute-1.amazonaws.com
Connection: Keep-Alive
User-Agent: Amazon Simple Notification Service Agent
{
  \"Type\" : \"Notification\",

```

```

"MessageId" : "cf5543af-dd65-5f74-8ccf-0a410EXAMPLE",
"TopicArn" : "arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic",
"Message" :
  "{ \"NotificationReferenceId\" : \"32d195c3-a829-4222-b1e2-14ab2EXAMPLE\",
    \"NotificationType\" : \"PaymentCapture\",
    \"SellerId\" : \"YOUR_SELLER_ID_HERE\",
    \"ReleaseEnvironment\" : \"Sandbox\",
    \"Version\" : \"2013-01-01\",
    \"NotificationData\" :
      \"<?xml version='1.0' encoding='UTF-8'>
        <CaptureNotification
          xmlns='https://mws.amazonservices.com/
            ipn/OffAmazonPayments/2013-01-01'>
          <CaptureDetails>
            <AmazonCaptureId>S23-1234567-1234567-0000002</AmazonCaptureId>
            <CaptureReferenceId>
              6f4d9dea0c234279a65e77994EXAMPLE
            </CaptureReferenceId>
            <CaptureAmount>
              <Amount>5.0</Amount>
              <CurrencyCode>USD</CurrencyCode>
            </CaptureAmount>
            <RefundedAmount>
              <Amount>0.0</Amount>
              <CurrencyCode>USD</CurrencyCode>
            </RefundedAmount>
            <CaptureFee>
              <Amount>0.0</Amount>
              <CurrencyCode>USD</CurrencyCode>
            </CaptureFee>
            <IdList/>
            <CreationTimestamp>2013-04-22T06:02:22.026Z</CreationTimestamp>
            <CaptureStatus>
              <State>Completed</State>
              <LastUpdateTimestamp>
                2013-04-22T06:02:25.227Z
              </LastUpdateTimestamp>
            </CaptureStatus>
            <SoftDescriptor>AMZ*softdescriptor</SoftDescriptor>
          </CaptureDetails>
        </CaptureNotification>\",
    \"Timestamp\" : \"2013-04-22T06:00:14Z\" }\",
"Timestamp" : "2013-04-22T06:00:15.108Z",
"SignatureVersion" : "1",
"Signature" : "dUWd9lrs...iNGKnR4=",
"SigningCertURL" : "https://sns.EXAMPLE.amazonaws.com/
  SimpleNotificationService-f3ecfb7224c7233fe7bb5f59fEXAMPLE.pem",
"UnsubscribeURL" : "https://sns.EXAMPLE.amazonaws.com/
  ?Action=Unsubscribe
  &SubscriptionArn=arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic:GUID"
}

```

次のサンプルは **RefundNotification** を表します。

```

POST / HTTP/1.1
x-amz-sns-message-type: Notification
x-amz-sns-message-id: 5f43584c-1f96-5880-9c98-119f5EXAMPLE
x-amz-sns-topic-arn: arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic
x-amz-sns-subscription-arn:
arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic: EXAMPLE
Content-Length: 961
Content-Type: text/plain; charset=UTF-8
Host: ec2-EXAMPLE.compute-1.amazonaws.com
Connection: Keep-Alive
User-Agent: Amazon Simple Notification Service Agent
{
  "Type" : "Notification",
  "MessageId" : "cf5543af-dd65-5f74-8ccf-0a410EXAMPLE",
  "TopicArn" : "arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic",
  "Message" :
    "{ \"NotificationReferenceId\" : \"32d195c3-a829-4222-b1e2-14ab2EXAMPLE\",
      \"NotificationType\" : \"PaymentRefund\",
      \"SellerId\" : \"YOUR_SELLER_ID_HERE\",
      \"ReleaseEnvironment\" : \"Sandbox\",
      \"Version\" : \"2013-01-01\",
      \"NotificationData\" :
        \"<?xml version='1.0' encoding='UTF-8'>
          <RefundNotification
            xmlns='https://mws.amazonservices.com/
              ipn/OffAmazonPayments/2013-01-01'>
            <RefundDetails>
              <AmazonRefundId>S23-1234567-1234567-0000003</AmazonRefundId>
              <RefundReferenceId>
                07fff0c4e05046958db7e47607e7db17
              </RefundReferenceId>
              <RefundType>SellerInitiated</RefundType>
              <RefundAmount>
                <Amount>5.0</Amount>
                <CurrencyCode>USD</CurrencyCode>
              </RefundAmount>
              <FeeRefunded>
                <Amount>0.0</Amount>
                <CurrencyCode>USD</CurrencyCode>
              </FeeRefunded>
              <CreationTimestamp>2013-04-22T06:07:34.617Z</CreationTimestamp>
              <RefundStatus>
                <State>Completed</State>
                <LastUpdateTimestamp>
                  2013-04-22T06:09:20.178Z
                </LastUpdateTimestamp>
              </RefundStatus>
              <SoftDescriptor>AMZ*softDescriptor</SoftDescriptor>
            </RefundDetails>
            </RefundNotification>\",
      \"Timestamp\" : \"2013-04-22T06:00:14Z\" }\",
  "Timestamp" : "2013-04-22T06:00:15.108Z",
  "SignatureVersion" : "1",
  "Signature" : "kjacl4DH...oQT6FbA=",
  "SigningCertURL" : "https://sns.EXAMPLE.amazonaws.com/
    SimpleNotificationService-f3ecfb7224c7233fe7bb5f59fEXAMPLE.pem",
  "UnsubscribeURL" : "https://sns.EXAMPLE.amazonaws.com/

```

```
?Action=Unsubscribe
&SubscriptionArn=arn:aws:sns:EXAMPLE:59860EXAMPLE:TestTopic:GUID"
}
```

次のサンプルは、オーソリオブジェクトのステータスを取得するための `GetAuthorizationDetails` 処理の呼び出す方法は表します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01
?AWSAccessKeyId=AKIAFBM3LG5JEEXAMPLE
&Action=GetAuthorizationDetails
&AmazonAuthorizationId=S23-1234567-1234567-0000001
&SellerId=YOUR_SELLER_ID_HERE
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2013-12-19T19%3A01%3A11Z
&Version=2013-01-01
&Signature=WlQ708agyHXMkoUBk69Hjxj8qdh3aDcqpY71hVgEXAMPLE
```

`GetAuthorizationDetails` 処理に含まれるリクエストパラメータ、レスポンス要素についての詳しい情報は、Amazon Pay API リファレンスガイドの `GetAuthorizationDetails` を参照してください。

SANDBOX 環境でのインテグレーションテスト

Amazon Pay のテスト環境は、本番環境で実行する前に、Amazon Pay のインテグレーションを徹底的にテストすることが可能です。SANDBOX モードで実装をテストする時は、Web サイト上の Amazon Pay ウィジェットを経由して購入者エクスペリエンスをシミュレーションできます。

SANDBOX モードでは、Amazon への API 処理の呼び出しが正しく設定され、注文全体をトラックする必要があるレスポンスに全ての Amazon Pay パラメータが含まれているか確実にするためのテストができます。開発環境は購入エクスペリエンス中のいろいろな障害イベントを管理するために、購入者エクスペリエンスの様々なエラー状態をシミュレーションできます。例えば、支払失敗やタイムアウトのシミュレーションができます。

テスト購入者アカウント

Amazon のセラーセントラルでは、インテグレーションのテストを支援するために、いくつかの購入者アカウントの設定を推奨します。これらのテストアカウントは多くのユースケースに合うように変更できます。例えば、販売事業者の運用ルールに違反するかもしれない追加の配送先住所を含めることができます。これらのアカウントをセラーセントラルのインテグレーションリンクのテストアカウントから設定できます。

セラーセントラルを通してカスタマイズは可能であるとともに、Amazon は複数のテスト購入者アカウントに適合した多くのテストシナリオを奨励します。

SANDBOX と本番環境の違い

Amazon Pay で利用される様々なオブジェクトは、生成された後のオブジェクトの有効期限に達したなどのビジネスルールに合わせて処理します。Amazon Pay 内で容易にテストを支援するために、次のビジネスルールは調整されます。

オブジェクト	本番環境ビジネスルール	SANDBOX ビジネスルール
Open 状態のオーソリアブジェクト	30 日で Closed に遷移します。	2 日で Closed に遷移します。

SANDBOX シミュレーション

Amazon Pay の SANDBOX は特定の支払テストシナリオをシミュレーションすることができます。これらのシミュレーションはレスポンスを生成することを許可し、販売事業者の運用ルールをテストするために利用できます。次の表はレスポンスと状態遷移をシミュレーションできる方法を説明します。状態と理由コードについての詳しい情報は、Amazon Pay API リファレンスガイドのイントロダクションセクションを参照してください。

注意：状態と理由コードは、シミュレーション文字列と一緒に利用することができます。望ましい状態と理由コードを Amazon Pay の API 処理の呼び出しで明示しなければなりません。無効な組み合わせは `InvalidSandboxSimulationSpecified` エラーの結果が返ります。

オブジェクト状態と理由コードのシミュレーション

Billing Agreement オブジェクト			
状態	理由コード	シミュレーション文字列	SANDBOX でのシミュレーション方法
Draft	--	N/A	SANDBOX の Amazon アカウントでお支払いボタンをクリックし、テスト購入者アカウント認証でサインインします。この生成で Billing Agreement オブジェクトは Draft 状態になります。
Open	--	N/A	<code>ConfirmBillingAgreement</code> 処理を呼び出して Draft の Billing Agreement

			オブジェクトを承認します。
Suspended	InvalidPaymentMethod	N/A	理由コードに InvalidPaymentMethod 利用してオーソリアブジェクトを Declined に遷移した後に Billing Agreement オブジェクトは Suspended 状態に遷移します。
Canceled	Stale	N/A	生成されてから 3 時間以内の Draft の Billing Agreement オブジェクトを承認してはいけません。 Billing Agreement オブジェクトは、それから Stale 状態になります。
	AmazonCanceled	N/A	シミュレーションできません。
Closed	AmazonClosed	<pre>{"SandboxSimulation": {"State": "Closed", "ReasonCode": "AmazonClosed"}}</pre>	Open の Billing Agreement に対して CloseBillingAgreement 処理の ClosureReason リクエストパラメータにこの値を指定します。
	SellerClosed	N/A	シミュレーション文字列に AmazonClosed の理由コードを指定せずに CloseBillingAgreement 処理を呼び出すことで Open の Billing Agreement オブジェクトを Close にします。

オーソリアブジェクト			
状態	理由コード	シミュレーション文字列	SANDBOX でのシミュレーション方法

Pending	--	N/A	<p>非同期モードで AuthorizeOnBillingAgreement 処理を呼び出し、オーソリをリクエストします。全てのオーソリオブジェクトは AuthorizeOnBillingAgreement リクエストを行った後、30 秒間は Pending 状態です。これは同期モードでシミュレーションできません。</p>
Open	--	N/A	<p>AuthorizeOnBillingAgreement 処理を呼び出し、オーソリをリクエストします。非同期モードでは、オーソリオブジェクトは Pending 状態で 30 秒経った後に Open 状態へ遷移します。同期モードでは、オーソリオブジェクトは速やかに Open 状態に遷移します。</p>
Declined	InvalidPaymentMethod	<pre> {"SandboxSimulation": {"State":"Declined", "ReasonCode":"InvalidPayment Method", "PaymentMethodUpdateTimeIn Mins":5}} </pre>	<p>AuthorizeOnBillingAgreement 処理の SellerAuthorizationNote リクエストパラメータにこの値を指定します。</p> <p>Order Reference は Open 状態から Suspended 状態に遷移します。Billing Agreement オブジェクトが Suspended 状態から Open 状態に戻った時に、PaymentMethodUpdateTimeInMins パラメータに時刻（1～240 分の間）を指定して利</p>

			用できます。このシミュレーションは購入者が無効な支払方法を更新し、Billing Agreement オブジェクトを Open 状態に戻します。
	AmazonRejected	<pre>{"SandboxSimulation": {"State": "Declined", "ReasonCode": "AmazonRejected"}}</pre>	AuthorizeOnBillingAgreement 処理の SellerAuthorizationNote リクエストパラメータにこの値を指定します。
	ProcessingFailure	N/A	シミュレーションできません。
	TransactionTimedOut	<pre>{"SandboxSimulation": {"State": "Declined", "ReasonCode": "TransactionTimedOut"}}</pre>	AuthorizeOnBillingAgreement 処理の SellerAuthorizationNote リクエストパラメータにこの値を指定します。
Closed	ExpiredUnused	<pre>{"SandboxSimulation": {"State": "Closed", "ReasonCode": "ExpiredUnused", "ExpirationTimeInMins": 1}}</pre>	AuthorizeOnBillingAgreement 処理の SellerAuthorizationNote リクエストパラメータにこの値を指定します。オーソリオブジェクトを Closed にしなければならない時間（1～60 分の間）を指定するために ExpirationTimeInMins パラメータを利用できます。
	MaxCapturesProcessed	N/A	Capture 処理を呼び出し Open のオーソリオブジェクトを売上請求します。
	AmazonClosed	<pre>{"SandboxSimulation": {"State": "Closed", "ReasonCode": "AmazonClosed"}}</pre>	AuthorizeOnBillingAgreement 処理の SellerAuthorizationNote リクエストパラメータ

			にこの値を指定します。
	SellerClosed	N/A	CloseAuthorization 処理を呼び出し Open のオーソリを Closed にします。

売上請求オブジェクト			
状態	理由コード	シミュレーション文字列	SANDBOX でのシミュレーション方法
Pending	--	<code>{"SandboxSimulation": {"State": "Pending"}}</code>	売上請求処理の SellerCaptureNote リクエストパラメータにこの値を指定します。売上請求オブジェクトはそれから 30 秒間は Pending 状態で残ります。
Declined	AmazonRejected	<code>{"SandboxSimulation": {"State": "Declined", "ReasonCode": "AmazonRejected"}}</code>	売上請求処理の SellerCaptureNote リクエストパラメータにこの値を指定します。
	ProcessingFailure	N/A	シミュレーションできません。
Completed	--	N/A	売上請求処理を呼び出し Open のオーソリに対して資金を売上請求します。
Closed	MaxAmountRefunded	N/A	返金処理を呼び出し売上請求された金額の 15%か 8,400 円のどちらか低い額まで加算した返金を行います。
	MaxRefundsProcessed	N/A	返金処理を対応する Completed の売上請求オブジェクトに対して呼び出しを 10 回発行します。

	AmazonClosed	<pre>{"SandboxSimulation": {"State":"Closed", "ReasonCode":"AmazonClosed"} }</pre>	売上請求処理の SellerCaptureNote リクエストパラメータにこの値を指定します。
--	---------------------	--	--

返金オブジェクト			
状態	理由コード	シミュレーション文字列	SANDBOX でのシミュレーション方法
Pending	--	N/A	返金処理を呼び出し、返金をリクエストします。全ての返金オブジェクトは返金リクエストが提示してから 30 秒間は Pending 状態になります。
Declined	AmazonRejected	<pre>{"SandboxSimulation": {"State":"Declined", "ReasonCode":"AmazonRejected"} }</pre>	返金処理の SellerRefundNote リクエストパラメータにこの値を指定します。
	ProcessingFailure	N/A	シミュレーションできません。
Completed	--	N/A	返金処理を呼び出し Completed の売上請求オブジェクトに対して資金の返金を行います。

ValidationResult のシミュレーションと ValidateBillingAgreement 処理の理由コード

ValidateBillingAgreement 処理			
Validation Result	理由コード	シミュレーション文字列	SANDBOX でのシミュレーション方法
Success	N/A	N/A	SANDBOX で Open の Billing Agreement へ ValidateBillingAgreement リクエストを提示します。

Failure	InvalidPaymentMethod	<pre> {"SandboxSimulation": {"Operation":"ValidateBillingAgreement", "ReasonCode":"InvalidPaymentMethod", "PaymentMethodUpdateTimeInMins":5}} </pre>	<p>SetBillingAgreementDetails 処理の SellerNote リクエストパラメータにこの値を指定します。一度、ValidateBillingAgreement リクエストを提示すると、Billing Agreement は Open 状態から Suspended 状態に遷移します。PaymentMethodUpdateTimeInMins パラメータに時刻（1～240 分の間）を指定して利用でき、その後、Billing Agreement オブジェクトは Suspended 状態から Open 状態へ戻ります。このシミュレーションは購入者が無効な支払方法を更新し、Billing Agreement オブジェクトを Open 状態に戻します。</p>
	ValidationTimedOut	<pre> {"SandboxSimulation": {"Operation":"ValidateBillingAgreement","ReasonCode":"ValidationTimedOut", "State":"Declined"}} </pre>	<p>SetBillingAgreementDetails 処理の SellerNote リクエストパラメータにこの値を指定します。</p>

次のサンプルは、**SellerAuthorizationNote** リクエストパラメータを利用して、**InvalidPaymentMethod** 理由コードを使いオーソリを **Declined** 状態にシミュレーションする方法を説明します。

```
https://mws.amazonservices.com/OffAmazonPayments_Sandbox/2013-01-01
?AWSAccessKeyId=AKIAFBM3LG5JEEXAMPLE
&Action=AuthorizeOnBillingAgreement
&AmazonBillingAgreementId=C12-1234567-1234567
&AuthorizationAmount.Amount=10
&AuthorizationAmount.CurrencyCode=USD
&AuthorizationReferenceId=test_authorize_1
&MWSAuthToken=amzn.mws.4ea38b7b-f563-7709-4bae-87aeaEXAMPLE
&SellerId=YOUR_SELLER_ID_HERE
&SellerAuthorizationNote=
    %7B%22SandboxSimulation%22%3A%7B%22State%22%3A%22Declined%22%2C
    %22ReasonCode%22%3A%22InvalidPaymentMethod%22%7D%7D
&SellerOrderAttributes.SellerOrderId=testSellerOrderId
&SellerOrderAttributes.StoreName=testStore
&SellerOrderAttributes.CustomInformation=ExampleInformation
&InheritShippingAddress=true
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2012-10-03T19%3A01%3A11Z
&TransactionTimeout=60
&Version=2013-01-01
&Signature=WlQ708aqyHXMkoUBk69Hjxj8qdh3aDcqpY71hVgEXAMPLE
```

エラーハンドリング

Amazon Pay のウィジェットと Amazon Pay の API 処理の両方はエラーを返します。Web サイト上で最高で実現可能な購入者エクスペリエンスを提供するためには、両者のほとんどのエラーを受け止めハンドリングしなければなりません。

Amazon Pay API 処理からのエラーハンドリング

Amazon Pay API 処理からのエラーを受け取った場合は、エラーの原因に応じて処理をリトライできるかもしれません。適切に構成された処理リクエストはべき等です。既に成功しているリクエストと同じ処理を呼び出した場合は、重複したトランザクションは生成されません。

Billing Agreement リクエストでのべき等のキーは **AmazonBillingAgreementId** です。

AuthorizeOnBillingAgreement、**Capture**、**Refund** リクエストでのべき等のキーは、それぞれ

AuthorizationReferenceId、**CaptureReferenceId**、**RefundReferenceId** です。例えば、一度、同じ **RefundReferenceId** を提示した場合は、それ以上に同じ売上請求オブジェクトに対して返金できません。次の表で説明している条件でエラーを確認できた場合は、この機能性は、多くの処理を安全にリトライするために許可します。

エラーを受け取った後に処理をリトライしたい場合は、最初のエラーレスポンスの後に直ぐにリトライできます。複数回リトライしたい場合は、Amazon は制限に対して「指数的に減らす」アプローチを行う実装を推奨します。それから、エラーを記録し、手動でのフォローアップ

プと調査を続行します。例えば、時間間隔を 1 秒、2 秒、4 秒、10 秒、30 秒としてリトライできます。実際に減らす時間と制限は、販売事業者の運用ルールによります。

次の表はエラーを説明します。

HTTP コード	HTTP ステータス	説明と調整アクション
500	InternalServerError	サーバはリクエストを実行中に予期しない状態に直面しました。 このエラーは SANDBOX セッションを本番環境のエンドポイント、またはその逆にリクエストする場合に発生します。
502	Bad Gateway	クライアントと接続先サーバの間を代理しているゲートウェイやプロキシのサーバがリクエストを拒否しました。多くのケースでは、時々途切れる問題であり、リトライしても安全です。問題が続く場合はネットワーク管理者に問合せしてください。
503	ServiceUnavailable RequestThrottled	これらはサーバサイド上の予期しない問題か、サーバはトランザクションクォータを超えた為にスロットルされました。クライアントはリクエストをリトライするか、リクエストの回数を減らさなければなりません。
504	Gateway Timeout	クライアントと接続先サーバの間で代理しているゲートウェイやプロキシのサーバがリクエストをタイムアウトしました。多くのケースでは、時々途切れる問題であり、リトライしても安全です。

詳しい情報は、Amazon Pay API リファレンスガイドのエラーコードセクションを参照してください。

Amazon Pay ウィジェットからのエラーハンドリング

「Step1：購入者認証用のボタンウィジェット追加」で述べたように、エラー状態を通知するためのボタン、アドレス帳、お支払い方法ウィジェットを設定できます。特定のインテグレーションエラーが作成されている場合は、これらのウィジェットはエラー通知を送信します。次のサンプルコードは関連したエラーとテキストフィールドに表示する `errorCode` と `errorMessage` を読むことができる方法を表します。これらのエラーコードは、本番環境上の潜在的な問題について通知を記録することで、インテグレーションのデバッグをより素早く支援します。

```
<label>Debug Error Code      :</label>
<div id="errorCode"></div>

<label>Debug Error Message  :</label>
<div id="errorMessage"></div>

<script>
  var authRequest;
  OffAmazonPayments.Button("AmazonPayButton", "YOUR_SELLER_ID", {
    type: "LwA",
    authorization: function() {
      loginOptions =
        {scope: "ENTER_SCOPES", popup: "ENTER_POPUP_PARAMETER"};
      authRequest  =
        amazon.Login.authorize(loginOptions,
"YOUR_REDIRECT_URL");
    },
    onError: function(error) {
      document.getElementById("errorCode").innerHTML = error.getErrorCode();
      document.getElementById("errorMessage").innerHTML =
error.getErrorMessage();
    }
  }).bind("payWithAmazonDiv");
</script>
```

次の表は様々なエラーコードとして購入者がウィジェット本体内で参照する Amazon Pay のウィジェットとそれに対応するエラーメッセージをリストしています。

エラーコード	説明	エラーメッセージ
AddressNotModifiable	Billing Agreement が与えられた状態では配送先住所を変更できません。	あなたはこの注文では配送先住所を変更できません。販売事業者の支援先にお問い合わせください。
BuyerNotAssociated	購入者は与えられた Billing Agreement に関連付けされていません。購入者はウィジェットが表示される前にサインインしなければなりません。	このセッションは無効です。 Amazon アカウントでお支払い ボタンをクリックして購入処理を再実行してください。

BuyerSessionExpired	購入者のセッションは Amazon では期限切れです。購入者はウィジェットが表示される前にサインインしなければなりません。	セッションは期限切れです。 Amazon アカウントでお支払い ボタンをクリックして再度サインインしてください。
ConfirmedBillingAgreement	指定された Billing Agreement は既に承認済です。すでに承認済の Billing Agreement の同意ウィジェットを表示することはできません。	定期支払の承認ウィジェットは表示されません。
InvalidAccountStatus	販売事業者のセラーアカウントはこのリクエストを実行するためには適切ではありません。例えば、それは停止になっているか、登録が完了しておりません。	ボタンウィジェットを表示する時にエラーが発生しウィジェットが表示されません。購入者にエラーメッセージは表示されません。アドレス帳やお支払い方法ウィジェットを表示する時にエラーが発生した場合は、購入者には次のメッセージが表示されます。 申し訳ございません。この Web サイトからあなたの支払処理に問題がありました。販売事業者の支援先までお問い合わせください。
InvalidBillingAgreementId	指定された Billing Agreement 識別子は無効です。	申し訳ございません。この Web サイトからあなたの支払処理に問題がありました。販売事業者の支援先までお問い合わせください。
InvalidParameterValue	指定されたパラメータに割り与えられた値は無効です。	ボタンウィジェットを表示する時にエラーが発生しウィジェットが表示されません。購入者にエラーメッセージは表示されません。アドレス帳やお支払い方法ウィジェットを表示する時にエラーが発生した場合は、購入者には次のメッセージが表示されます。 申し訳ございません。この Web サイトからあなたの支払処理に問題がありました。販売事業者の支援先までお問い合わせください。
InvalidSellerId	提示された販売事業者の識別子が無効です。正しい SellerId を指定してください。	ボタンウィジェットを表示する時にエラーが発生しウィジェットが表示されません。購入者にエラーメッセージは表示されません。アドレス帳やお支払い方法ウィジェットを表示する時にエラーが発生した場合は、購入者には次のメッセージが表示されます。

		申し訳ございません。この Web サイトからあなたの支払処理に問題がありました。販売事業者の支援先までお問い合わせください。
MissingParameter	指定されたパラメータは間違いです、正しいパラメータを提供しなければなりません。	ボタンウィジェットを表示する時にエラーが発生しウィジェットが表示されません。購入者にエラーメッセージは表示されません。アドレス帳やお支払い方法ウィジェットを表示する時にエラーが発生した場合は、購入者には次のメッセージが表示されます。 申し訳ございません。この Web サイトからあなたの支払処理に問題がありました。販売事業者の支援先までお問い合わせください。
PaymentMethodNotModifiable	Billing Agreement が与えられた状態では支払方法を変更できません。	この注文の支払方法は変更できません。販売事業者の支援先までお問い合わせください。
ReleaseEnvironmentMismatch	リリース環境内でのウィジェットが Billing Agreement と一致しないウィジェットの表示を試みました。リリース環境のウィジェットと Billing Agreement オブジェクトは一致しなければなりません。	申し訳ございません。この Web サイトでの支払処理に問題があります。販売事業者の支援先までお問い合わせください。
StaleBillingAgreement	指定された Billing Agreement は許可された時間内に承認されていない為、今キャンセルされました。キャンセルされた Billing Agreement の支払方法と住所を関連付けできません。	あなたのセッションは期限切れです。 Amazon アカウント でお支払いボタンをクリックして再度サインインしてください。
WidgetNotApplicable	セラーアカウントは現在 Billing Agreement の認証をしておりません。	申し訳ございません。この Web サイトでの支払処理に問題があります。販売事業者にお問い合わせください。
InvalidBillingAgreementStatus	指定された Billing Agreement 識別子は無効な状態です。	申し訳ございません。この Web サイトでの支払処理に問題があります。販売事業者にお問い合わせください。
UnknownError	サービスは原因不明のエラーです。	申し訳ございません。この Web サイトでの支払処理に問題があります。販売事業者にお問い合わせください。

ボタンギャラリー

Amazon はあなたが利用できるボタンの複数のバージョンとサイズギャラリーを用意します。ボタンサイズはアプリケーションが選択したボタンのカテゴリによります。ボタンはデスクトップとタッチ（タッチスタイルボタンの方が最適）の2つのカテゴリがあります。デスクトップとタッチカテゴリの中には、さらに進んだオプションとして Android、iOS、レティーナディスプレイなどがあります。

Amazon アカウントでお支払いボタンについては、Amazon Pay の Web サイトの開発ツールを参照してください。

Amazon アカウントでログインボタンについては、[Login with Amazon Button Guidelines](#) ページを参照してください。

スタイルガイドライン

これらのガイドラインは Amazon アカウントでログインボタンと Amazon アカウントでお支払いボタンを利用するためのベストプラクティスを提供します。Amazon からの事前承認なしでサイトやアプリでこれらのボタンを利用でき、これらのボタンを変更することなく提供されています。

サイトに Amazon アカウントでお支払いボタンを追加

Amazon アカウントでログインボタンは、提供されているサイトやアプリでログインする所はどこにでも配置されるべきです。これは新しい顧客と同様に既存の顧客（例えば購入中）がサインアップする場所を含みます。私たちはログインを促進するためにカスタマイズされたエクスペリエンスとして、小さいボタン（以下参照）をホームページ上に表示することを推奨します。ALT テキストに“Amazon アカウントでログイン”を利用します。次がサンプルです。



ユーザーがログインしている場合は、ログインボタンは表示させず、それをログアウトメッセージに置き換えるべきです。

ボタンのガイダンスを追加

Amazon はログイン画面に次のテキストを表示することを推奨します。

- <Your Site>では、Amazon.co.jp アカウントでログインできます。

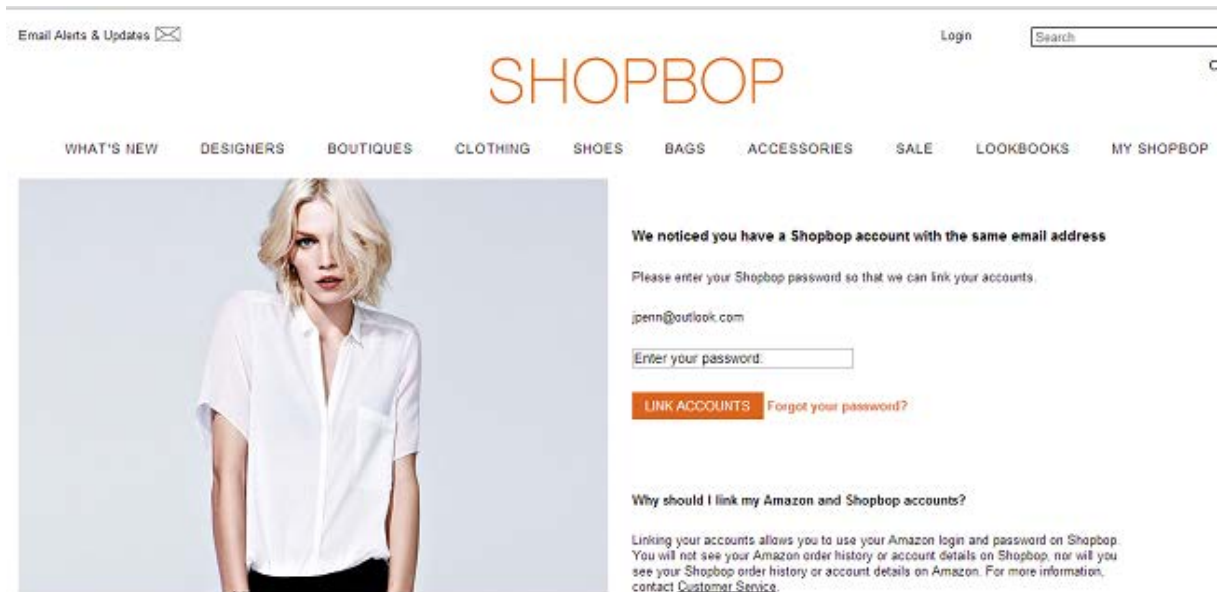
- お買い物をするために Amazon Pay アカウントでお支払いでき、Amazon の保護を受けられます。
- お支払い情報は Amazon に保存されているので安全です。


その他のサインインインテグレーションの提供

我々はシンプルなアイコンを提供しています。この画像はサイトやアプリで、異なるログインオプションをアイコン表示する時に役立ちます。

顧客アカウントを Amazon とリンク

現在の顧客が Amazon アカウントを利用してログインしている場合は、メールアドレスを識別でき、システムに登録されているアカウントとリンクできます。詳しい情報は存在するユーザーアカウントシステムとインテグレーションを参照してください。



Email Alerts & Updates 

Shopbop

WHAT'S NEW DESIGNERS BOUTIQUES CLOTHING SHOES BAGS ACCESSORIES SALE LOOKBOOKS MY SHOPBOP

We noticed you have a Shopbop account with the same email address

Please enter your Shopbop password so that we can link your accounts.

jpenn@outlook.com

Enter your password:

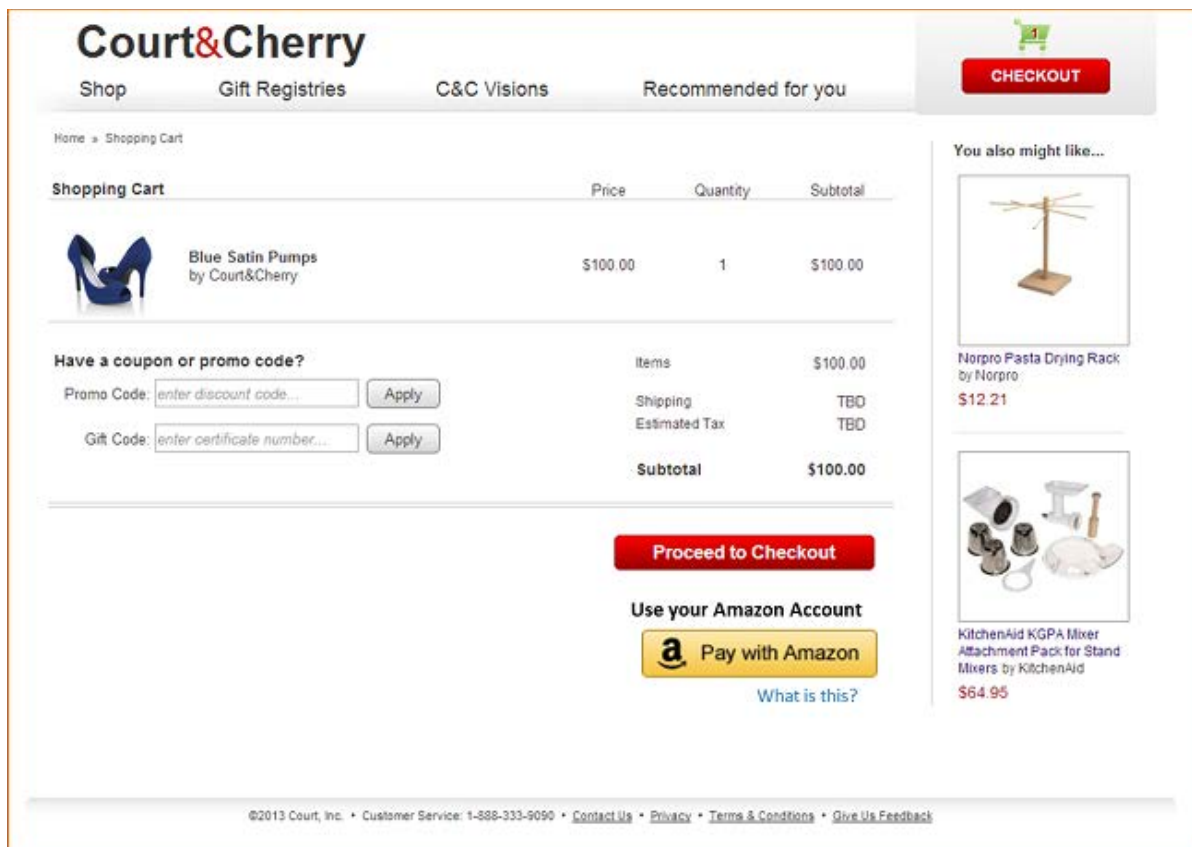
LINK ACCOUNTS [Forgot your password?](#)

Why should I link my Amazon and Shopbop accounts?

Linking your accounts allows you to use your Amazon login and password on Shopbop. You will not see your Amazon order history or account details on Shopbop, nor will you see your Shopbop order history or account details on Amazon. For more information, contact [Customer Service](#).

サイトに Amazon アカウントでお支払いボタンを追加

顧客がカートページにおり、購入処理を開始するための準備されている場合は、Amazon アカウントでお支払いボタンを次のスクリーンショットのように利用できます。



Amazon は、コンバージョン率を最大化するために、“What is this?” ポップアップを追加し、Amazon Pay とは何かを説明するために次のテキストを利用することを推奨します。



Amazon Pay では、<Your Site>の注文で Amazon アカウントに登録されている住所と支払情報を利用できます。簡単で信頼された支払方法である沢山の人が利用しております。お客様は多くの情報を再入力する必要はなく、クレジットカードの番号と支払詳細は共有されません。Amazon Pay を選択することで、Amazon アカウントで安全にサインイン要求されます。お客様は注文を完了するために Amazon の配送先と支払方法を安全にアクセスして提供されます。

<Your Site>でお買い物を楽しんでください。

商品詳細ページに Amazon Pay を追加

次のスクリーンショットで表示しているように、Amazon アカウントでお支払いボタンを追加することで、顧客はサイトのページから望み通りの商品を素早く購入できます。

Court & Cherry Your Account [Sign in](#) | [Manage Registry](#) | [Wishlist](#) | [Live Chat](#)

Shop Gift Registries C&C Visions Recommended for you **CHECKOUT** \$247.75

Home » Appliances » Mixers » Stand Mixers » KitchenAid

KitchenAid Artisan 5-Quart Stand Mixer

by KitchenAid

Was: ~~\$349.99~~
Now: **\$247.75**
You save: \$102.24 (29%)

Color:

Quantity:

Availability: In Stock

Add to cart

Buy now

Product Features

- 325-watt mixer with 10 speeds; 5-quart stainless steel bowl
- Tilt-back head for easy access to mixture
- 2-piece pouring shield with large chute for adding ingredients
- Includes flat beater, dough hook, and wire whip
- Measures 14 by 8-2/3 inches by 14 inches; 1-year warranty

From the Manufacturer

The Artisan mixer's strong 325-watt motor delivers the power to handle the heaviest mixtures, and mix large batches easily.

The 5-quart polished stainless steel bowl with ergonomic handle is big enough to handle large batches of heavy mixtures. Its ergonomically designed handle is contoured to fit the hand and makes lifting the bowl more comfortable.

Unique Mixing Action

The beater spins clockwise as the shaft spins counter clockwise; moving the beater to 67 different points around the bowl. This unique mixing action creates a thorough blend of ingredients and eliminates the need to rotate the bowl.

You also might

Norpro Pasta Drying Rack by Norpro
\$12.21

KitchenAid KGPA Mixer Attachment Pack for Stand Mixers by KitchenAid
\$64.95

Item added to cart
KitchenAid Artisan 5-Quart Stand Mixer by KitchenAid
Qty 1: \$247.75

©2013 Court and Cherry, Inc. • Customer Service: 1-888-333-9090 • [Contact Us](#) • [Privacy](#) • [Terms & Conditions](#) • [Give Us Feedback](#)

ゲスト購入後に顧客にログインを尋ねる

次のスクリーンショットで表示しているように、ゲスト購入後に素早く会員登録することを顧客に申し出ることができます。

[CHECKOUT](#)

Thank you for your order

We'll send you an order confirmation to patsmith@gmail.com shortly. We'll also be in touch with updates to your order status.

Your Order #: 123-XYZ

Amazon Payments Order#: 105-235434-363436

Go to payments.amazon.com to see your payment history and other account information.

This item will arrive via Express shipping in 2-3 business days



Blue Satin Pumps
by Court&Cherry

\$100.00

1

\$100.00

Login with Amazon to Court and Cherry and save 10% on your next purchase



- Login with your Amazon credentials
- Track your <site name> orders online
- View your <site name> order history

Recommended for you...



Norpro Pasta Drying Rack
by Norpro

\$12.21



KitchenAid KGPA Mixer
Attachment Pack for Stand
Mixers by KitchenAid

\$64.95