

# Subscribe with Amazon Getting Started Guide

Last updated May 15, 2017

## Change log

5/15/2017 : Added business reports section.

5/15/2017 : Added customer service section.

4/10/2017 : Updated Seller Central references.

10/24/2016 : Added supported HTML notes.

12/14/2016 : Added API call throttling notes.

10/24/2016 : Added Sample Seller section.

10/24/2016 : Added Sandbox section.

9/1/2016 : Updated Get Subscription API response format.

7/14/2016 : Updated LWA setup instructions.

6/15/2016 : Updated API endpoint to v3. Introduced **beneficiaryType** in API calls and the **beneficiaries** parameter in API responses.

6/15/2016 : Updated push notifications to **notificationVersion 2**. Introduced the **beneficiaries** parameter in push notifications.

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Overview</b>   | <b>5</b>  |
| 1.1      | What is Subscribe with Amazon? . . . . .  | 5         |
| 1.2      | How does Subscribe with Amazon work? . . . . .  | 5         |
| <b>2</b> | <b>Getting Started</b>  | <b>6</b>  |
| 2.1      | Integration checklist . . . . .   | 6         |
| 2.2      | Bank account and tax setup . . . . .  | 6         |
| 2.3      | Adding your Product listing details . . . . .   | 6         |
| 2.3.1    | Create a Product listing . . . . .  | 7         |
| 2.3.2    | Add Product information . . . . .   | 7         |
| 2.3.3    | Add Tiers & Offers . . . . .  | 8         |
| 2.3.4    | Add images . . . . .  | 9         |
| 2.3.5    | Add a Compare Tiers chart . . . . .   | 10        |
| 2.3.6    | View your product detail page in preview mode . . . . .                                     | 10        |
| 2.4      | Setting up your Login with Amazon application . . . . .                                     | 11        |
| 2.5      | Setting your push notification preferences . . . . .  | 11        |
| 2.6      | (Optional) Setting up Sample Seller, an example storefront . . . . .                        | 12        |
| <b>3</b> | <b>Integrating with Login with Amazon</b>   | <b>12</b> |
| 3.1      | Adding the Login with Amazon button to your website . . . . .                               | 12        |
| 3.2      | Adding the Login with Amazon button to your iOS app . . . . .                               | 12        |
| 3.3      | Adding the Login with Amazon button to your Android app . . . . .                           | 12        |
| 3.4      | Integrating with your existing account system . . . . .                                     | 12        |
| <b>4</b> | <b>Integrating with Subscribe with Amazon</b>   | <b>13</b> |
| 4.1      | Website and/or app integration with the SWA APIs: Overview . . . . .                        | 13        |
| 4.1.1    | How a subscription gets created . . . . .   | 13        |
| 4.1.2    | Caching Subscriptions from SWA . . . . .  | 14        |
| 4.1.3    | Required changes to user account management . . . . .                                       | 15        |
| 4.1.4    | CS/CRM/Operations integration . . . . .   | 15        |
| 4.2      | Website and/or app integration with the SWA APIs: Full specification . . . . .              | 15        |
| 4.2.1    | Security API . . . . .  | 15        |
| 4.2.2    | Pagination . . . . .  | 16        |
| 4.2.3    | Response codes . . . . .  | 17        |
| 4.2.4    | The subscription object . . . . .   | 17        |
| 4.2.5    | List Subscriptions API . . . . .  | 18        |
| 4.2.6    | Get Subscription API . . . . .  | 20        |
| 4.2.7    | Get Pricing Terms API . . . . .   | 20        |
| 4.3      | Website and/or app integration with the SWA APIs: Push notification specification . . . . . | 22        |
| 4.3.1    | Push notification overview . . . . .  | 22        |
| 4.3.2    | Why use push notifications? . . . . .   | 22        |
| 4.3.3    | Security . . . . .  | 22        |
| 4.3.4    | Retries . . . . .   | 23        |
| 4.3.5    | Message format for the ACTIVATED event . . . . .  | 23        |
| 4.3.6    | Message format for the RENEWED event . . . . .  | 23        |
| 4.3.7    | Message format for the CANCELLED event . . . . .  | 23        |

|          |   |           |
|----------|---|-----------|
| <b>5</b> | <b>Testing and Submitting your Product</b>                        | <b>24</b> |
| 5.1      | (Optional) Testing in the Sandbox . . . . .                       | 24        |
| 5.1.1    | Making a direct Sandbox API call . . . . .                        | 24        |
| 5.1.2    | Simulating a customer login with the Sandbox . . . . .            | 25        |
| 5.2      | Testing your integration . . . . .                                | 25        |
| 5.2.1    | Test scenarios . . . . .  | 25        |
| 5.3      | Submitting your Product for approval . . . . .                    | 26        |
| 5.3.1    | Submission process overview . . . . .                             | 26        |
| 5.3.2    | How to submit your Product for approval . . . . .                 | 26        |
| <b>6</b> | <b>Business Reports and Customer Service</b>                      | <b>27</b> |
| 6.1      | Business reports . . . . .  | 27        |
| 6.1.1    | Transaction report . . . . .                                      | 27        |
| 6.1.2    | Payment report . . . . .  | 27        |
| 6.1.3    | Tax report . . . . .  | 27        |
| 6.1.4    | Subscription report . . . . .                                     | 27        |
| 6.2      | Customer service . . . . .  | 27        |
| 6.2.1    | Use cases resolved by Amazon customer service . . . . .           | 28        |
| 6.2.2    | How to refer a customer to Amazon customer service . . . . .      | 28        |
| 6.2.3    | Amazon customer service referrals to your CS department . . . . . | 28        |

# 1 Overview

## 1.1 What is Subscribe with Amazon?

Subscribe with Amazon (SWA) is a cross-industry subscription management service targeted at a broad variety of sellers engaged in subscription commerce. It offers a convenient way for customers to sign up for and manage subscriptions on Amazon.com, and then use their Amazon account information to access those subscriptions on the sellers sites and/or apps via the Login with Amazon (LWA) button.

## 1.2 How does Subscribe with Amazon work?

SWA allows customers to subscribe to your products on Amazon.com using their Amazon account. Amazon manages the billing relationship with customers, as well as managing subscription lifecycle events such as activation, renewal, and cancellation. Sellers are responsible for allowing customers to log in with their Amazon accounts and providing service to active subscribers based on the subscription status maintained by Amazon.



Figure 1: First, the customer purchases the subscription on Amazon.com from an SWA seller.

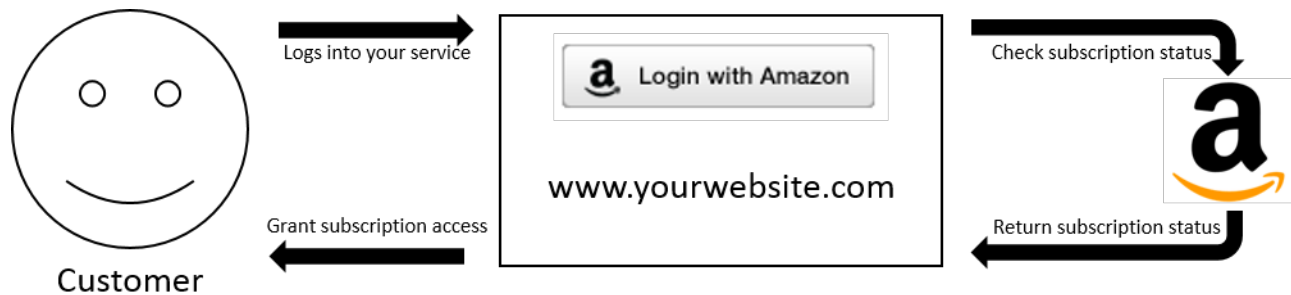


Figure 2: Then, the customer logs into the subscription provider's website or app using Login with Amazon. The subscription provider checks if a subscriber has active SWA subscriptions and serves the appropriate content or experience.

## 2 Getting Started

In this guide we will show you how to set up a Subscribe with Amazon Product detail page on Amazon.com, add the Login with Amazon button to your website and/or app, complete integration so you can authenticate and grant access to SWA subscribers, and publish your Product live to Amazon.com.

### 2.1 Integration checklist

As you proceed through the setup and integration process, please consult the following checklist of integration activities:

- Provide the required banking and 1099K tax information (section 2.2).
- Create your Product listing(s) and reviewed the detail page in preview mode (section 2.3).
- Add the Login with Amazon button to **all** locations in your website and app(s) where a customer can log in (section 2.4, 3.1 - 3.3).
- Enable customers with existing accounts on your website or app to link those accounts to their Amazon accounts (section 3.4).
- Set your push notification preferences and provided a valid HTTPS endpoint if you choose to receive push notifications (section 2.5).
- Perform technical integration with the SWA APIs to grant access to Amazon customers when they log in (section 4.1 - 4.3).
- Modify your account management links to direct the customer to Amazon’s subscription manager (section 4.1.3).
- Update your CS/CRM/Operations tools so they can look up the account status of the customer (section 4.1.4).
- Test your LWA and SWA integration (section 5).
- Submit your Product to Amazon for approval (section 5.3).
- Ensure that your internal customer service processes and public-facing help content are updated to support SWA customers (section 6.2).

### 2.2 Bank account and tax setup

If you are a new SWA seller, you will need to provide bank account and 1099K tax information before you can publish a Product listing. Log in to your SWA seller account at <https://sellercentral.amazon.com>, and then click the “Provide banking and tax information” button and follow the instructions on the screen.

### 2.3 Adding your Product listing details

As an SWA seller, you will create one or more subscription Product listings. A listing is represented by a product detail page on Amazon.com, where customers can discover, learn about, and purchase your subscription offerings. Customers can find your listings by searching or browsing the Amazon catalog, viewing on-site recommendations, or navigating directly to a product detail page URL. Creating a listing also allows your Amazon customers to view their subscriptions in **Your Digital Subscriptions**. A complete listing contains a Product, Tier, and Offers.

### 2.3.1 Create a Product listing

Log in to your SWA seller account at <https://sellercentral.amazon.com>, and then click the yellow **Get started** button in the “Add product details” section to create a Product listing. You must provide the information below to create a complete Product listing.

### 2.3.2 Add Product information

A Subscribe with Amazon listing starts with a Product. Generally, a Product is a grouping that represents an overall subscription service or app that may contain different Tiers (i.e. service levels). Customers will see the Product name in search results, product detail pages, orders, and in **Your Digital Subscriptions**. Creating a Product generates an Amazon.com product detail page, to which you will later add Tiers, Offers, images, and product information. Follow the instructions on the screen to provide the required Product information as described below.

**Product name:** Your Product name will appear at the top of the Amazon.com product detail page, in search results, in **Your Memberships & Subscriptions**, and on order confirmation screens. Your Product name should match how you refer to your subscription offering on your website or app, if applicable. Do not include details about specific Tiers or Offers in your Product name.

**Login URL:** The default webpage where users log in to your website to access their subscription, once their account has been successfully created or linked. This page will be displayed during the post-purchase flow and also accessible from **Your Memberships & Subscriptions**. If you prefer to use a separate log in page for mobile devices, please provide this URL by checking the box next to **I have a dedicated page for mobile devices**.

**Category and subcategory:** Categorization will affect placement in search and browse, and help Amazon customers discover your listings. Each category contains a different set of subcategories, so make sure to select the right category before browsing subcategories. You should choose the most specific available subcategory that matches your Product, though note that not all categories include subcategories. If you cannot find an appropriate category or subcategory, select **Other**.

**Product description:** This description will appear in the **Product details** section of the Amazon detail page. It is recommended that you utilize this field to describe the Product and its benefits. Your Product description may not contain any URLs. This field supports the following HTML tags: `<br>`, `<ul>`, `<ol>`, `<li>`, `<p>`, `<sub>`, `<sup>`, `<i>`, `<em>`, `<strong>`, `<b>`, and `<u>`.

**The fine print/important restrictions:** Use this field to describe any access restrictions for your Product (e.g. “requires a minimum 3G connection for mobile access”). By default, your Product will only be available on Amazon.com to US-based customers, so you should not repeat that restriction here. Do not use this field to describe Product benefits or features. This field supports the following HTML tags: `<br>`, `<ul>`, `<ol>`, `<li>`, `<p>`, `<sub>`, `<sup>`, `<i>`, `<em>`, `<strong>`, `<b>`, and `<u>`.

**Communication preferences URL:** A webpage where customers customize their communication preferences and unsubscribe from emails.

**Terms of use URL:** A webpage where customers access the terms of use for your Product. Must not require customers to log in.

**Privacy policy URL:** A webpage where customers access the privacy policy for your Product. Must not require customers to log in.

**Customer service email address:** An email address where SWA subscribers can access customer service for your Product. Amazon will instruct customers to contact your customer service department only when necessary. You must provide an email address or a phone number, or both.

**Customer service phone number:** A phone number where SWA subscribers can access customer service for your Product. Amazon will instruct customers to contact your customer service department only when necessary. You must provide an email address or a phone number, or both.

At any time, you may click **Save without publishing** in the bottom left corner to return to the Seller Central home page. When finished with the **Product** tab, click **Continue** to proceed to the **Tiers & Offers** tab.

### 2.3.3 Add Tiers & Offers

Once you've created a Product, you will add one to three Tiers. A Tier is a specific level of service that a customer can subscribe to within your subscription Product. For example, you might offer a "Premium Tier" and a "Basic Tier" that provide different sets of benefits to subscribers. Customers will be able to select from your available Tiers on a single Product detail page (maximum three Tiers).

You must add one to three Offers to each Tier. An Offer is a specific pricing and payment option for customers to subscribe to a Tier (maximum three Offers per Tier). For example, you might offer your Premium Tier for either \$99/year or \$10/month, each of which is represented as an Offer within that Tier. Customers will be able to choose from your available Offers from the Buy Box in the upper right section of the Amazon.com product detail page after selecting a Tier. An Offer includes a base price (required), a free trial (optional), and an introductory pricing period (optional).

Follow the instructions on the screen to provide the required Tier and Offer information as described below.

#### Tiers

**Tier name:** Your Tier name will be selectable on the Amazon detail page and viewable in search results. Your Tier name should match how you refer to this level of service on your website or app, if applicable. Do not include details about pricing in the Tier name. At times, Amazon will represent a Tier by concatenating the Product name with the Tier name (i.e. "Product Name: Tier Name") so please do NOT include the Product name within the Tier name.

**productId:** The Tier productId is the 10-character alphanumeric identifier that Amazon will use in API calls when referencing a customer's subscription. A productId must be unique across Products and Tiers within your SWA seller account, as they describe the subscription you grant to customers when they log in to your website or app. Please see section 4.2.4 for more details.

**Tier description:** The Tier description will appear immediately below the main product images on the Amazon.com product detail page. It is recommended that you utilize this field to describe the Tier and its benefits. This field supports the following HTML tags: <br>, <ul>, <ol>, <li>, <p>, <sub>, <sup>, <i>, <em>, <strong>, <b>, and <u>.

**Product tax code (optional):** The tax code that Amazon will use to calculate sales tax on sales of this Tier (only if you have enabled Amazon Tax Collection Services; refer to <https://sellercentral.amazon.com/gp/tax-manager/dashboard.html> for more details).



**Supported devices:** For each Tier, you must select supported devices, software, and platforms. Supported devices will appear in a table on your product detail page and may also be used by customers as filters.

## Offers

**What is the price for this Offer?** The amount that customers will be billed for this Offer at each billing period (also known as “base price”). For example, if you choose **\$10** as your price and **1 month** as your billing period, new customers will be billed \$10/month, and the subscription will autorenew at the end of each month unless the customer chooses to disable auto-renew. You may choose a billing period of daily, weekly, monthly, every 3 months, every 6 months, or annually.

**What type of promotion would you like to run?** An Offer may include either a Free Trial **OR** an introductory price. A Free Trial allows customers to access this Tier free of charge for a finite period of time after purchase. When a Free Trial expires, the subscription will autorenew at the base price and billing period. If the Offer includes a Free Trial, you must specify the length of time that the Free Trial will last before a customer is billed. An introductory price allows new subscribers to pay something other than the base price for a finite period of time. For example, an Offer might begin at \$5/month for the first three months, and then switch to the base price of \$10/month each month thereafter. When the introductory price expires, the subscription will auto-renew into the base price and billing period. If the Offer includes an introductory price, you must set up its price, billing period, and length. The introductory price billing period does not need to be the same as the base price billing period. Free Trials and introductory prices are only available to customers who have not previously purchased a subscription for this Product on Amazon.com.// At any time, you may click **Save without publishing** in the bottom left corner to return the dashboard. When finished with the **Tiers & Offers** tab, click **Continue** to proceed to the **Images** tab.

### 2.3.4 Add images

A Product listing requires three types of images: one icon image, one product description image, and one to three hero images for each Tier. Follow the instructions on the screen to upload the required images as described below.

**Icon image:** The icon is a small square image used to represent your Product. Customers will see your icon in search results, the product detail page, and **Your Digital Subscriptions**.

Dimensions: 500px W x 500px H

File format: PNG

Optimal file size: 30-70 kb

Minimum resolution: 72 dpi

**Product description image:** The Product description image is “below the fold” on your product detail page next to your Product description. The Product description image should represent the entire Product, not one specific Tier.

Dimensions: 600px W x 315px H

File format: PNG

Optimal file size: 30-70 kb

Minimum resolution: 72 dpi

**Tier hero images:** A hero image is one of the main images displayed on your product detail page. You must add at least one hero image (maximum of three) per Tier. Customers can cycle through Tier hero images by using the thumbnails.

Dimensions: 1265px W x 365px H

File format: JPG

Optimal file size: 30-70 kb  
Minimum resolution: 72 dpi

At any time, you may click **Save without publishing** in the bottom left corner to return to the Seller Central home page.

**If you have added more than one Tier:** When finished with the **Tiers & Offers** tab, click **Continue** to proceed to the **Compare Tiers** tab (section 2.3.5).

**If you have added only one Tier:** When finished with the **Tiers & Offers** tab, click **Submit Product details & exit** to publish your Product listing in preview mode and return to the Seller Central home page. Then skip to section 2.3.6. Your product detail page will not yet be discoverable by customers.

### 2.3.5 Add a Compare Tiers chart

Detail pages for Products with multiple Tiers include a Compare Tiers chart that allows customers to distinguish between different Tiers and Offers. The first three rows are auto-generated by information you've already provided. You may add up to seven custom rows. For each custom row, check the box underneath each Tier for which that row attribute applies. You must add at least one custom row or you will not be allowed to proceed. Note: the chart will be titled "Compare Plans" on the detail page.

When finished with the **Compare Tiers** tab, click **Submit product details & exit** to publish your Product listing in preview mode and return to the Seller Central home page. Your product detail page will not yet be discoverable by customers.

### 2.3.6 View your product detail page in preview mode

Once you click **Submit Product details & exit** your product detail page will be available in preview mode within approximately 4 hours. You will receive an email when the push to preview mode has begun; however, a detail page preview may not be available right away. When the preview is available, the Product name in the "Subscriptions in progress" section on the Seller Central home page will change to a clickable link.

While a Product listing is in preview mode, you will be able to:

- View the product detail page as a customer would see it.
- Make changes to the Product information by clicking the **Edit** button next to "Add Product details".  
*Note: any changes will take up to approximately 4 hours to be reflected on the product detail page in preview mode.*
- Add, remove, or replace images. *Note: any image changes will take up to approximately 4 hours to be reflected on the product detail page in preview mode.*

While a Product listing is in preview mode, you will NOT be able to:

- View the Product listing in search results or access it via any method besides a direct link from the Seller Central home page.
- Fulfill purchased subscriptions from the detail page.

Once you have submitted your Product details and returned to the dashboard, you may proceed to the **Add technical details** section, where you will create your Login with Amazon application, retrieve your API credentials, and set your push notification preferences.

## 2.4 Setting up your Login with Amazon application

Subscribe with Amazon requires that you implement Login with Amazon (LWA) on your website and/or apps so SWA customers can use their Amazon.com accounts to access their subscriptions. Customers who are already signed in to Amazon.com will be automatically logged in when they visit your site or open your app. Setting up Login with Amazon will also generate the API credentials that you will use to call the SWA APIs in order to grant subscription access to these customers.

The first step to adding the Login with Amazon button to your website and/or app is to register an application with Login with Amazon. An application contains information the authorization service needs for your website or app to grant access to customers and retrieve their profiles. For each Product listing that you create, you may use an existing LWA application or create a new one. If you create multiple Products that customers access on their same websites or apps, have similar branding or names, and use the same privacy policy, you may want to use the same LWA application across these Products. If not, it may be preferable to create a new LWA application for each Product. Click “Get started” in the **Add technical details** section for your Product to begin this process; the first tab will guide you through LWA setup.

If you are new to LWA, you will need to set up your LWA account at login.amazon.com. Follow the instructions on the screen to complete this process.

Once you’ve completed LWA account setup, you will need to register an application. Follow the instructions on your LWA app console to complete this process. In the application details section, add basic details about your application. These details will be used on your website and mobile apps (if applicable).

**Name shown to users:** This is the name displayed on the consent screen when the users agree to share the information with your application. This name applies to Android, iOS, and website versions of your application.

**Description:** A description of your application for Login with Amazon users.

**Privacy notice URL:** The privacy URL is the location of your company or application’s privacy policy. It is also displayed on the consent screen. This link is displayed to users when they first login to your application (for example, <https://www.example.com/privacy.html>).

**Logo image file:** This logo is displayed on the sign-in and consent screen when users log into your website or mobile app. The logo will be automatically resized to 50 x150 pixels. The following formats are accepted: PNG, JPEG, GIF.

After your basic application settings are saved, you can add settings for specific websites and mobile apps that will use this application to grant access to SWA customers (see sections 3.1, 3.2, and 3.3). Return to your app console at any time to update these settings by switching the dropdown in the top right to “Login with Amazon.”

When you’ve finished setting up your LWA application, switch the dropdown in the top right to “www.amazon.com” and return the Seller Central home page. From there, click “Edit” in the **Add technical details** section to continue to push notification setup.

## 2.5 Setting your push notification preferences

Push notifications are messages that Amazon sends when an event happens to a subscription that a customer purchased from you through SWA. Having these messages sent to you is optional; they should be used

primarily for caching, business reporting, and updating internal systems. On the **Push notifications** tab in the **Add technical details** section, follow the instructions on the screen to opt in to receiving notifications. If you opt in, you will need to provide the HTTPS endpoint to which Amazon will send notifications for this product. See section 4.3 for detailed push notification specifications.

At any time, you may click **Save & exit** in the bottom left corner to return to the dashboard. When you've set up your push notifications preferences, click **Submit and exit** to complete the **Add technical details** section.

## 2.6 (Optional) Setting up Sample Seller, an example storefront

Sample Seller is a web application that simulates a seller storefront. It follows this guide closely. You may use Sample Seller as a starting point for integrating LWA and the SWA APIs with your website or app. Setting up Sample Seller is purely optional.

Visit <https://github.com/amznlabs/swa-sample-seller> and follow the README to set up Sample Seller.

# 3 Integrating with Login with Amazon

## 3.1 Adding the Login with Amazon button to your website

After creating your LWA application (see section 2.4 you will need to add the Login with Amazon button to any websites where SWA customers can access their subscriptions for this Product. First, return to your app console by switching the dropdown in the top right to “Login with Amazon”, select the appropriate application, and refer to the **Web Settings** section. Then, follow the instructions at <https://login.amazon.com/website>.

## 3.2 Adding the Login with Amazon button to your iOS app

After creating your LWA application (see section 2.4 you will need to add the Login with Amazon button to any iOS apps where SWA customers can access their subscriptions for this Product. First, return to your app console by switching the dropdown in the top right to “Login with Amazon”, select the appropriate application, and refer to the **iOS Settings** section. Then, follow the instructions at <https://login.amazon.com/ios>.

## 3.3 Adding the Login with Amazon button to your Android app

After creating your LWA application (see section 2.4 you will need to add the Login with Amazon button to any Android apps where SWA customers can access their subscriptions for this Product. First, return to your app console by switching the dropdown in the top right to “Login with Amazon”, select the appropriate application, and refer to the **Android Settings** section. Then, follow the instructions at <https://login.amazon.com/android>.

## 3.4 Integrating with your existing account system

SWA customers may want to link their Amazon account with a separate set of credentials for your service so that they can log into your website or app with either account. There are several reasons that customers would need this functionality:

- They are buying the subscription for a friend or family member and do not want to provide their Amazon.com account information to the recipient.

- They previously had a subscription to your Product and want to retrieve their settings, history, content, etc.
- They previously had access to a free version of your Product and want to continue using the login credentials associated with that free account.

In order to support this functionality, you will need to enable an account linking solution on your website or app. The steps for implementing this functionality can be found at <https://login.amazon.com/documentation/combining-user-accounts>.

**As a part of account linking you may NOT disable Login with Amazon functionality, allow customers to opt out of Login with Amazon, or otherwise limit, block, or alter access to subscriptions accessed via Login with Amazon.**

## 4 Integrating with Subscribe with Amazon

### 4.1 Website and/or app integration with the SWA APIs: Overview

Once you've added the Login with Amazon button to your website and/or app(s), you need to set up the ability to check the subscription status of customers who use this button to log in. This section provides an overview of how a subscription gets created for an SWA customer, and how you will need to modify your systems in order to handle these customers and their subscriptions.

#### 4.1.1 How a subscription gets created

In order to provide service to active subscribers who purchased your Product through Amazon, your systems need be able to determine if a customer has an active subscription or not. This section describes the process by which the customer purchases a subscription, and how your systems and Amazon's systems will interact at every step of the way in order to ensure that the customer receives access to the subscription.

**Step 1: The customer signs up for a subscription.** The first step is for the customer to purchase the subscription on Amazon.com, which can be done from the detail page on a web or mobile browser.

**Step 2: Amazon creates an active subscription for the customer** Amazon updates our records to indicate that the customer has an active subscription based on the Product, Tier (`productId`), and Offer billing terms. The customer's Amazon.com account is associated with a unique combination of `beneficiaryId` (the LWA `user_id`) and `beneficiaryType` (`LWA_USER_ID`) that you will use to look up their subscriptions via the SWA APIs when they log in to your website or app. Amazon will send a push notification, if you have opted in to receive these notifications, containing the `id` and `type` in the `beneficiaries` field.

**Step 3: The customer logs in to your website or app.** At this point, the subscription has been successfully purchased and activated on Amazon.com. The customer may choose to use Login with Amazon to access your website or app, or they may log in with an existing account for your service that has been linked to their Amazon account in your database. In the Login with Amazon case, you will need to call the profile endpoint to retrieve the LWA `user_id` for that customer (see [https://developer.amazon.com/public/apis/engage/login-with-amazon/docs/customer\\_profile.html](https://developer.amazon.com/public/apis/engage/login-with-amazon/docs/customer_profile.html) for more details). In the linked account case, you will need to retrieve the customer's LWA `user_id` from your database. In both cases, you will then need to query Amazon for that customer's subscriptions using the `beneficiaryId` and a `beneficiaryType`. For example, for a customer with an LWA `user_id` of `amzn1.account.AHQLL56R7AI5GPDFJV6CA6SUVQUA`,

you would make the following request to retrieve his/her active subscriptions (note that you may choose to specify a `status`, but that is not a required parameter):

```
GET https://appstore-sdk.amazon.com/v3/subscriptions?
  beneficiaryId=amzn1.account.AHQLL56R7AI5GPDFJV6CA6SUVQUA&
  beneficiaryType=LWA_USER_ID&
  status=ACTIVE
```

The response will be a list of active subscriptions that SWA has on record for this user (if you had not specified a `status` the response would include both active and cancelled subscriptions). For example:

```
{
  "nextCursor": null,
  "subscriptions": [
    {
      "beneficiaries": [
        {
          "id": "amzn1.account.AHQLL56R7AI5GPDFJV6CA6SUVQUA",
          "type": "LWA_USER_ID"
        }
      ],
      "endDate": null,
      "productId": "001101-042",
      "receiptId": "nUjwZs3cNjBeimbJKnnqFhWqOI6vVLH38IGczYs9dM0=:5:11",
      "startDate": 1326439320,
      "status": "ACTIVE",
    }
  ]
}
```

Complete information on the meaning of each field can be found in the full API specifications (section 4.2), but the most important fields on the subscription that you'll want to check are the `productId`, the `beneficiaries`, and the `status`:

- `productId` indicates which product the customer should be granted access to in your system in the event of an active subscription. You will provide this ID to us at the time when you create your Product listing. If an active subscription has a `productId` that matches up to the product that the user is attempting to use, you should grant access to the user for that product.
- `beneficiaries` indicates which Amazon customers should be granted access to the product, specified by an `id` and a `type`.
- `status` indicates whether this subscription is currently `ACTIVE` or `CANCELLED`.

#### 4.1.2 Caching Subscriptions from SWA

To minimize the number of requests made to the SWA APIs, you might want to temporarily cache the active subscriptions that you pull from Amazon. For instance, when you query Amazon for the list of active subscriptions assigned to a particular user, if the response indicates that that user is an active subscriber, you could locally cache that that user is an active subscriber. The next time you need to check if the user is an active subscriber, as long as the cache has not expired, you would simply check the cache instead of making a request to the SWA API.

This cache should have a limited lifetime. We recommend a maximum of 24 hours, but this is up to your discretion. The lifetime of the cache corresponds to the maximum length of time when the cache could

potentially be stale. For instance, if your cache is 24 hours, it's possible that a customer cancels their subscription on Amazon immediately after you cache it as active. If this happens, then the customer could potentially receive 24 hours of unpaid access.

Please note that you should only cache active subscriptions. You should not establish a cache stating that a customer has no active subscriptions. If you did this, it would result in an unacceptable customer experience, since a customer whose account had its subscription status cached by you as “inactive” or “non-subscriber” who then goes to Amazon.com and purchases a subscription to your product would then not receive immediate access to their subscription content. A subscriber with a stale cache like this on their account would have to wait until the cache expires in order to use the subscription, which is not acceptable, as the subscriber should be able to get immediate access after purchasing on Amazon.com.

#### 4.1.3 Required changes to user account management

Amazon manages the account lifecycle (billing, cancellation, etc.) for the SWA subscriptions purchased by your customers. To accommodate these customers, you will have to modify the user interface of your account lifecycle management solution to redirect to the **Your Digital Subscriptions** page on Amazon.com (<https://www.amazon.com/subscriptionmanager/digital-subscriptions/>) for customers whose accounts have active subscriptions with SWA. This will provide customers with a direct path to account management functionality on Amazon.com. If you do not do this, customers will have no direct way from your website or service to manage these aspects of their account and the experience may appear broken, resulting in confusion and customer service contacts. Changes to your customer account view are required to participate in SWA.

#### 4.1.4 CS/CRM/Operations integration

As Amazon controls billing management for the customer, your customer service, marketing, and operations systems may need to be modified in order to call the SWA APIs for customer subscription statuses. For example, if you need to track the status of a subscription so a customer service representative can look it up by email address, you will need to retrieve this data via API calls and use it to populate your systems appropriately. This work is required in order to ensure smooth operations and a frictionless resolution process for your customers who signed up through Amazon. There are numerous ways to do this, depending on your CS/CRM/Operations systems; below are a few suggestions:

- Modify your existing CRM tool to call the SWA APIs for the current status of the customer. These are the same APIs you will use for granting subscription access to customers when they log in to your website.
- Cache the subscription status information in your existing systems and use that as the authoritative source. The benefit of this approach is that it allows your system to decouple from calling the SWA APIs each time a customer needs to be looked up. However, you will need to ensure that your system does not become out of sync with Amazon.

## 4.2 Website and/or app integration with the SWA APIs: Full specification

### 4.2.1 Security API

In order to access the SWA APIs, you must provide a valid access token in the **Authorization** header of the request. This presence of a valid access token indicates that your systems have authorized the request.

To obtain the access token, you'll need your LWA client credentials: the `client_id` and `client_secret`. Use these credentials to issue a POST request to <https://api.amazon.com/auth/O2/token> with the following body:

```
grant_type=client_credentials&
  scope=swa:subscriptions&client_id=<YOUR_CLIENT_ID>&
  client_secret=<YOUR_CLIENT_SECRET>
```

If you provided valid credentials, you will receive a HTTP 200 response code and the data in the body will look like the following:

```
{
  "access_token": "Atc|MAEBIN1...",
  "scope": "swa_subscriptions",
  "token_type": "bearer",
  "expires_in": 3600
}
```

The body includes both the access token and its lifespan in seconds. The access token can be reused to authenticate an unlimited number requests as long as the lifespan has not elapsed.

To authenticate your requests with the access token, include it in the Authorization header. For example:

```
Authorization: Bearer Atc|MAEBIN1...
```

Calls to the SWA APIs are throttled to 1 transaction per second (TPS) per operation with a “burst” of 20 TPS. This means that you may make calls to the API endpoint once per second indefinitely. If in one of those seconds a burst is required due to a volume spike, you may make up to 20 additional calls in that second. Subsequently, you would be limited to 1 TPS for the next 20 seconds until a new 20-TPS burst becomes available. Call volumes that exceed the TPS limits as described above will result in 400 errors in response.

#### 4.2.2 Pagination

When you make a request for a list of objects, you may only receive only a subset of the total list of objects. To get the next subset of responses, or the next “page”, you need to use the `nextCursor` included in the response. The `nextCursor` can be specified as a `cursor` in a subsequent request to obtain the next page of results. If the `nextCursor` is `null` in the result, that means that there are no subsequent pages.

You may specify a `pageSize`, which determines the number of subscriptions that will appear on each page of results. For example, consider a scenario in which an API response contains 11 subscription results (indexed as subscriptions 0,1,2,...,10) while the `pageSize` is set to 5 and the `cursor` to the default value of 0. The response will include subscriptions 0 through 4 (5 total subscriptions) along with a `nextCursor` value of 5 to indicate the index of the next result. You would then make an identical API call with the `cursor` value set to 5 to retrieve the subscriptions 5 through 9 along with a `nextCursor` value of 10. Finally, a call with `cursor` set to 10 would return subscription 10 (the 11th and final subscription) along with a `null` value for `nextCursor`.

If you do not specify a `pageSize`, the value will default to 10 (i.e. 10 subscriptions per page of results). If you do not specify a `cursor` value, it will default to 0 (i.e. beginning with the first subscription in the set of results).



### 4.2.3 Response codes

Below is the list of HTTP response codes you may receive when calling our APIs:

| Code | Description   |
|------|---|
| 200  | Successful request.   |
| 403  | Access denied to invalid authorization token.                       |
| 404  | Receipt not found, if you query by a receipt id.                    |
| 500  | Internal error.   |
| 501  | unsupported operation (i.e. issue a POST request instead of a GET). |

### 4.2.4 The subscription object

If successfully called for a valid subscription or customer, the SWA APIs will return one or more subscription objects. A subscription includes a unique identifier (`receiptId`), as well as information about the Product and Tier associated with the subscription, the customer who purchased the subscription, the status of the subscription, and the start and end dates.

When an Amazon customer logs into your website or app, you will need to call the SWA APIs to retrieve the subscription(s) for that customer so you can check whether they have an active subscription, grant the correct level of access for that customer, and display the appropriate content. You should also use subscription data to integrate with your existing account management, CS, CRM, and operations systems as described in section 4.1.4.

Below is an example subscription object and a description of each field:

```
{
  "nextCursor": null,
  "subscriptions": [
    {
      "beneficiaries": [
        {
          "id": "amzn1.account.AHQLL56R7AI5GPDFJV6CA6SUVQUA",
          "type": "LWA_USER_ID"
        }
      ],
      "endDate": null,
      "productId": "001101-042",
      "receiptId": "nUjwZs3cNjBeimbJKnnqFhWq0I6vVLH38IGczYs9dM0=:5:11",
      "startDate": 1326439320,
      "status": "ACTIVE",
    }
  ]
}
```

| Field                      | Type               | Description   |
|----------------------------|--------------------|---|
| <code>receiptId</code>     | String             | The ID of the subscription. Has a maximum length of 64 characters. Each <code>receiptId</code> refers to a single unique subscription, but a single customer can have multiple <code>receiptIds</code> associated with their account.                 |
| <code>productId</code>     | String             | The unique identifier for the Tier that was purchased that resulted in the creation of this subscription. You will provide this value to us when you set up your Product listings. <b>Must be exactly 10 characters long.</b>                         |
| <code>beneficiaries</code> | List of structures | The Amazon customers to which this subscription is assigned. A beneficiary structure maps to a unique Amazon.com account, specified by the combination of <code>id</code> and <code>type</code> . Each string has a maximum length of 255 characters. |
| <code>status</code>        | String             | The current status of the subscription. Possible values are <code>ACTIVE</code> and <code>CANCELLED</code> .  |
| <code>startDate</code>     | long               | The timestamp at which the subscription was initially ordered by the customer. This only occurs once at the time of purchase and is not modified by subscription renewals. The <code>startDate</code> value is displayed in Unix Timestamp format.    |
| <code>endDate</code>       | long               | If the subscription has ended, this field will indicate the timestamp at which the subscription ended. It is null for active subscriptions. The <code>endDate</code> value is displayed in Unix Timestamp format.                                     |

#### 4.2.5 List Subscriptions API

GET [https://appstore-sdk.amazon.com/v3/subscriptions?](https://appstore-sdk.amazon.com/v3/subscriptions?beneficiaryId=<beneficiaryId>&beneficiaryType=<beneficiaryType>)

```
beneficiaryId=<beneficiaryId>&
beneficiaryType=<beneficiaryType>
```

The `List Subscriptions` API is the primary API you should call to determine which subscriptions a customer has purchased and their status. It returns a list of all subscriptions belonging to a particular customer, represented by a unique `beneficiaryId` and `beneficiaryType` for that customer. This is what allows you to grant the appropriate access level and display the appropriate content when a customer enters your site or app using Login with Amazon. The steps below describe a typical sequence for a web-based login experience:

1. A customer uses Login with Amazon to access your website.
2. Upon successful login, you use the Login with Amazon SDK to retrieve that customer's user profile (see section 4.1.1).
3. From the Login with Amazon user profile, you retrieve the customer's LWA `user_id`.
4. You use the customer's LWA `user_id` as the `beneficiaryId` parameter along with `LWA_USER_ID` as the `beneficiaryType` to call the `List Subscriptions` API.
5. The `List Subscriptions` API returns zero, one, or multiple subscriptions for that customer.
  - If the API returns **zero subscriptions**, your website defaults to the experience for a customer who has not already purchased a subscription on Amazon.com. This may be a "new user" experience, or any default behavior that you choose.
  - If the API returns **one subscription**, you should use the `productId` and `status` of that subscription to determine what the customer sees. For example, a customer with an `ACTIVE` subscription would receive access to the subscription that they purchased as indicated by the `productId` in the

API response. A customer with a CANCELLED subscription would see the “expired” or “inactive” subscription experience.

- if the API returns **multiple subscriptions**, you should build logic that examines each subscription and displays the right experience to the customer based on the set of **productIds** and **statuses**. For example, if the customer has a CANCELLED subscription but subsequently purchased a new ACTIVE subscription, you should honor the ACTIVE subscription. If the customer has multiple ACTIVE subscriptions, you should honor all of the subscriptions or otherwise ensure that the customer receives the appropriate access and content based on the full list of subscriptions. Ensure that you handle pagination in the API response properly in order to retrieve all subscriptions for the customer (see section 4.2.2).

6. Depending on if/how you are caching customer information, you may use the **List Subscriptions** API response to cache the subscription data for that customer or update other systems that you use for CS, CRM, or operations.

Using `curl` to initiate a sample **List Subscriptions** request might look like this:

```
curl -H "Authorization: Bearer your_token_here"
https://appstore-sdk.amazon.com/v3/subscriptions?
beneficiaryId=amzn1.account.AHQLL56R7AI5GPDFJV6CA6SUVQUA
&beneficiaryType=LWA_USER_ID
&pageSize=2
&cursor=1
```

The only required parameters are **beneficiaryId** and **beneficiaryType**. As shown in the example you can add other parameters to filter the result set. For the request above, you might receive a response that looks like this:

```
{
  "nextCursor": null,
  "subscriptions": [
    {
      "beneficiaries": [
        {
          "id": "amzn1.account.AHQLL56R7AI5GPDFJV6CA6SUVQUA",
          "type": "LWA_USER_ID"
        }
      ],
      "endDate": null,
      "productId": "001101-042",
      "receiptId": "nUjwZs3cNjBeimbJKnnqFhWq0I6vVLH38IGczYs9dM0=:5:11",
      "startDate": 1326439320,
      "status": "CANCELLED",
    }
  ]
}
{
  "beneficiaries": [
    {
      "id": "amzn1.account.AHQLL56R7AI5GPDFJV6CA6SUVQUA",
      "type": "LWA_USER_ID"
    }
  ],
  "endDate": null,
  "productId": "001101-099",
}
```

```

    "receiptId": "yBreZy3kMjBbkWJKolwFqSpGH9vVNB21IQpoFs6xL0=:2:14",
    "startDate": 1264616643,
    "status": "ACTIVE"
  }
]
}

```

#### 4.2.6 Get Subscription API

GET <https://appstore-sdk.amazon.com/v3/subscriptions/<receiptId>>

The `Get Subscription` API allows you to look up a single subscription by its `receiptId`. Unlike the `List Subscriptions` API, `Get Subscription` returns only one result at a time. In general, we do not recommend that you use `Get Subscription` in realtime when a customer logs in to your website or app, since in that scenario you would need to use the `beneficiaryId` and `beneficiaryType` to call the `List Subscriptions` API.

Instead, `Get Subscription` is most useful as a mechanism to update dependent systems (such as business reporting, CS, CRM, or operations) where you might be storing data for individual subscriptions by `receiptId`.

For example, you may use incoming push notifications to keep a log of all subscription purchases for your Products on Amazon.com, creating a new record each time you receive a push notification for a unique `receiptId`. If you track or report on this data, you would likely want to keep it updated with the correct subscription status for each `receiptId`. Thus, you may create a regular job that calls the `Get Subscription` API for each recorded `receiptId` at a daily/weekly/monthly frequency (depending on your business needs) and updates your records with the correct `ACTIVE` or `CANCELLED` status for each subscription.

Using `curl` to initiate a sample `Get Subscription` request might look like this:

```

curl -H "Authorization: Bearer your_token_here"
  https://appstore-sdk.amazon.com/v3/subscriptions/nUjwZs3cNjBeimbJKnnqFhWqOI6vVLH38IGczYs9dM0=:5:11

```

And the response you receive might look like this:

```

{
  "beneficiaries": [
    {
      "id": "amzn1.account.AHQLL56R7AI5GPDFJV6CA6SUVQUA",
      "type": "LWA_USER_ID"
    }
  ],
  "endDate": null,
  "productId": "001101-042",
  "receiptId": "nUjwZs3cNjBeimbJKnnqFhWqOI6vVLH38IGczYs9dM0=:5:11",
  "startDate": 1326439320,
  "status": "CANCELLED",
}

```

#### 4.2.7 Get Pricing Terms API

GET <https://appstore-sdk.amazon.com/v3/subscriptions/<receiptId>/pricingTerms>

The `Get Pricing Terms` API allows you to look up the pricing terms for a single subscription by its `receiptId`. Like the `Get Subscription` API, `Get Pricing Terms` returns only one result at a time. Unlike

the `Get Subscription` API and `List Subscriptions` API, the `Get Pricing Terms` API does not return subscription status for a customer or subscription. Rather, it allows you to determine the price and billing period for that subscription, including whether the customer is currently in a free trial or introductory pricing period.

In general, like the `Get Subscription` API, `Get Pricing Terms` may be most useful to update systems like reporting, CS, CRM, or operations where you might be interested to know which Offer a customer has subscribed to, whether he/she is currently a paying subscriber, when a free trial is scheduled to end, etc.

For example, you may use incoming push notifications to keep a log of all subscription purchases for your Products on Amazon.com, creating a new record each time you receive a push notification for a unique `receiptId`. If you track or report on this data set, you would likely want to keep it updated with the next billing amount for each `receiptId`. Thus, you may create a regular job that calls the `Get Pricing Terms` API for each recorded `receiptId` at a daily/weekly/monthly frequency (depending on your business needs) and updates your records with the correct price and billing period for each subscription.

Using `curl` to initiate a sample `Get Pricing Terms` request might look like this:

```
curl -H "Authorization: Bearer your_token_here"
  https://apstore-sdk.amazon.com/v3/subscriptions/
  nUjwZs3cNjBeimbJKnnqFhWqOI6vVLH38IGczYs9dM0=:5:11/pricingTerms
```

And the response you receive might look like this:

```
{
  "basePricingTerm": {
    "duration": {
      "period": 1,
      "unit": "year"
    },
    "endDate": null,
    "price": {
      "amount": 99.99,
      "currencyCode": "USD"
    },
    "startDate": 1476321601000
  },
  "currentPricingTerm": {
    "duration": {
      "period": 1,
      "unit": "month"
    },
    "endDate": 1476321601000,
    "price": {
      "amount": 0.0,
      "currencyCode": "USD"
    },
    "startDate": 1473729601000
  },
  "receiptId": "nUjwZs3cNjBeimbJKnnqFhWqOI6vVLH38IGczYs9dM0=:5:11"
}
```

The `currentPricingTerm` describes what the customer is currently paying (e.g. \$0 if he/she is in a free trial, or some nonzero amount if it is a paid subscription) while the `basePricingTerm` shows what he/she will be paying once any free trials/intro prices/promotions have elapsed. The `currentPricingTerm` and `basePricingTerm` will match if the customer is not in an introductory or free trial period.

In the example above, the customer's current pricing term is \$0 for a period of one month (i.e. a one-month free trial), scheduled to end on October 13, 2016. On that date, the customer will begin the base pricing term of \$99.99 for a period of one year. Since that is a base pricing term, the \$99.99/year term will continue to renew after each one-year period.

### 4.3 Website and/or app integration with the SWA APIs: Push notification specification

#### 4.3.1 Push notification overview

There are three types of events for which Amazon will send a push notification:

1. **ACTIVATED** - occurs after a customer activates a new subscription.
2. **RENEWED** - occurs when an active subscription is renewed into a new contract (this does not necessarily correspond to any billing occurring).
3. **CANCELLED** - occurs when an active subscription is cancelled.

#### 4.3.2 Why use push notifications?

Push notifications enable your systems to immediately perform an action in response to a subscription event. For example, you could use push notifications as a trigger to:

1. Send a welcome email to the customer after they have purchased a subscription.
2. Feed an event into your internal metrics/reporting system.
3. Mark in your subscriptions cache that a customer has an active subscription.
4. Clear your subscriptions cache for a customer when the customer cancels their active subscription.

Amazon does not guarantee the delivery of push notification messages. As such, you should not rely on push notifications for mission-critical functions. For instance, you should never rely on push notifications as the exclusive source of subscription information and avoid calling the SWA APIs entirely. However, you may use push notifications to update your cache as described above and thereby decrease the number of calls that you need to make to the SWA APIs. Push notifications will be either enabled or disabled for each Product; at this time you may not customize the type(s) of push notifications that you receive.

#### 4.3.3 Security

**SSL:** When you opt in to receive push notifications, you will need to provide an HTTPS endpoint that Amazon can send the messages. This endpoint must be HTTPS, and not HTTP, in order to protect the messages from tampering and spoofing in transit. Your SSL certificate for this endpoint must be signed by a trusted Certificate Authority. For a list of trusted Certificate Authorities, see <https://docs.aws.amazon.com/sns/latest/dg/SendMessageToHttp.https.ca.html>.

**Basic access authentication:** SSL will protect against tampering in transit, but it does not protect against an unauthorized party sending its own messages to your HTTPS endpoint. To protect against the threat of spoofing, you need to check that the message is authentic and originated from Amazon.

You can do this by checking the `Authorization` header in the HTTP request containing the message. When you opt in to receive push notifications, you will be provided the username and password that Amazon will put in the `Authorization` header. This username and password combination will be encoded using the Basic Access Authentication scheme.

For example, if you configure `Aladdin` as the username and `open sesame` as the password then the header will be sent to you as follows:

```
Authorization: Basic QWxhZGRpbjpvYVUHNlc2FtZQ==
```

#### 4.3.4 Retries

Amazon will always send a push notification to you when an associated event occurs, but it's possible that your systems do not receive the message or do not process it properly.

If the message is not received properly by your systems, Amazon will attempt to resend the notification in intervals of  $n^4$  seconds between each retry. After 5 failed retries, Amazon will give up on trying to send the message to you. Note that in rare cases a duplicate message may be sent in order to ensure delivery. As a result you should design your system for receiving push notifications such that at any time you may receive messages out of order or duplicates; they must not be affected adversely when processing the same message more than once (i.e. idempotent).

#### 4.3.5 Message format for the ACTIVATED event

```
{
  "pushNotificationVersion": "2",
  "operation": "ACTIVATED",
  "subscription": {
    "beneficiaries": [{"type": "LWA_USER_ID", "id": "
      amzn1.account.K2LI23KL2LK2"}],
    "receiptId": "asdfgh0aoaYBYzDrxYYTE8IzuVvDJuzh2JV0j2_o6uY=:5:11",
    "productId": "001101-042"
  }
}
```

#### 4.3.6 Message format for the RENEWED event

```
{
  "pushNotificationVersion": "2",
  "operation": "RENEWED",
  "subscription": {
    "beneficiaries": [{"type": "LWA_USER_ID", "id": "
      amzn1.account.K2LI23KL2LK2"}],
    "receiptId": "asdfgh0aoaYBYzDrxYYTE8IzuVvDJuzh2JV0j2_o6uY=:5:11",
    "productId": "001101-042"
  }
}
```

#### 4.3.7 Message format for the CANCELLED event

```

{
  "pushNotificationVersion": "2",
  "operation": "CANCELLED",
  "subscription": {
    "beneficiaries": [{"type": "LWA_USER_ID", "id": "
      amzn1.account.K2LI23KL2LK2"}],
    "receiptId": "asdfgh0aoaYBYzDrxYYTE8IzuVvDJuzh2JV0j2_o6uY=:5:11",
    "productId": "001101-042"
  }
}

```

## 5 Testing and Submitting your Product

### 5.1 (Optional) Testing in the Sandbox

The Sandbox is a tool that allows you to create test subscription data that will be returned in an API call to the Sandbox endpoint. Once you've completed the **Add technical details** section for at least one Product, the Sandbox will become available in the **Quick links** section of the Seller Central home page, or directly at <https://sellercentral.amazon.com/swa/sandbox>. You can use the Sandbox to test three different scenarios:

- **Active subscription:** A customer uses Login with Amazon to access an active subscription.
- **Cancelled/inactive subscription:** A customer uses Login with Amazon to access an inactive (i.e. cancelled or expired) subscription.
- **No subscription:** A customer uses Login with Amazon with an account that has no associated subscription. This may occur when a non-SWA customer uses Login with Amazon to access your website for the first time.

There are two ways to test with Sandbox data: making a direct API call or simulating a customer login.

**To retrieve Sandbox data you must make API calls to the dedicated Sandbox endpoint:**  
<https://appstore-sdk-swa-sandbox.amazon.com/v3/subscriptions>.

#### 5.1.1 Making a direct Sandbox API call

The quickest way to test Sandbox data is to call the **List Subscriptions** or **Get Subscription** API using the Sandbox endpoint: <https://appstore-sdk-swa-sandbox.amazon.com/v3/subscriptions>.

1. Select a test case from the dropdown on the Sandbox page. You may leave the **Login URL** field at the top blank.
2. You will see the **Test subscription** table populate with data once a test case has been selected. Please refer to section 4.2.4 for details on the meaning of each column in the table. Note that the **beneficiaryId** refers to the Amazon.com account you've used to log into Seller Central.
3. If you've selected **Active subscription** or **Subscription cancelled**, you will need to provide a 10-character **productId** before creating a test subscription. Once you've entered a valid **productId**, click the yellow button beneath the table to create the test case.
4. With the test case successfully created, you may now make a **List Subscriptions** or **Get Subscription** API to the Sandbox endpoint using the **beneficiaryId** and **receiptId**, respectively, found in the **Test subscription** table (refer to section 4.2 for instructions for calling the production versions of these APIs). Verify that the response matches what you see in the table.



5. Switch the dropdown to the remaining test cases and verify that API calls continue to return correct data. Note that you can only test one case at a time.

### 5.1.2 Simulating a customer login with the Sandbox

If you've set up an LWA and SWA integration already (see sections 3 and 4), you may also use the Sandbox to simulate an actual customer login to your website or a test page . The steps below will allow you to test the end-to-end login process for a single customer, using the Amazon.com credentials you've used to access Seller Central and the Sandbox API endpoint (<https://appstore-sdk-swa-sandbox.amazon.com/v3/subscriptions>).

1. Enter a valid URL in the **Login URL** field at the top of the Sandbox page. This may be your actual website or a page you'd like to use for testing.
2. Select a test case from the dropdown.
3. You will see the **Test subscription** table populate with data once a test case has been selected. Please refer to section 4.2.4 for details on the meaning of each column in the table. Note that the **beneficiaryId** refers to the Amazon.com account you've used to log into Seller Central.
4. If you've selected **Active subscription** or **Subscription cancelled**, you will need to provide a 10-character **productId** before creating a test subscription. Once you've entered a valid **productId**, click the yellow button beneath the table to create the test case.
5. With the test case successfully created, you may now click the **Go to login URL** button to open the page you entered above. Assuming you've added a Login with Amazon button to this page, use that button to log in with the Amazon.com credentials you've used to access Seller Central.
6. If your integration is functioning properly, you should be granted access as a test customer and your experience should appropriately reflect the test case you've selected above.
7. Switch the dropdown to the remaining test cases and verify that logging in continues to reflect the correct subscription data. Note that you can only test one case at a time.

## 5.2 Testing your integration

### 5.2.1 Test scenarios

Once you've used the Sandbox to test basic API calls, you should ensure that your system is able to handle the scenarios below:

- A customer logs in with an active subscription
- A customer logs in with a cancelled subscription
- A customer cancels a subscription and then logs in with a repurchased active subscription
- A customer logs in with an active and a cancelled subscription
- A customer logs in with no subscriptions
- A customer logs in with multiple subscriptions (if applicable)
- A customer logs in with an active subscription that becomes cancelled while the user is still logged in
- A customer logs in with no subscriptions, but gains an active subscription while the user is still logged in
- A customer attempts to log in but network failures may prevent your API calls from reaching Amazon

## 5.3 Submitting your Product for approval

### 5.3.1 Submission process overview

All Products must be approved by Amazon before they are live for customers. During the approval process we will review your product detail page for content and quality, test the purchase and login experience, and ensure that SWA and LWA integration is working as expected. You will receive an email notifying you when your product has been approved and is live for customers on Amazon.com.

You will be ready to submit your Product for approval when:

- You have added your Product details and are satisfied with the appearance of your product detail page.
- You have added the Login with Amazon button to each location in your website or app where a customer can log in to access the Product.
- You have implemented SWA API integration to grant subscription access for Amazon customers who have logged into your website or app.
- You have successfully tested all three customer simulation scenarios in the Sandbox.
- You have modified your account management settings to direct SWA customers to the “Your Memberships & Subscriptions” page.
- You have integrated the LWA and SWA APIs with your customer service, CRM, and operations systems if necessary.
- Youve enabled an account linking solution on your website or app if necessary.
- You have configured your push notification settings if you would like to receive them.

### 5.3.2 How to submit your Product for approval

First, click the **Get started** button under the **(3) Publish** step. You will need to confirm that you have satisfied the requirements for submission, and then you may click the **Submit for approval and activation** button. Once you submit your Product, it will enter a queue for an Amazon representative to review the product detail page, perform a test purchase, and verify that LWA and SWA API integration has been implemented correctly. This process may take up to two business days.

**If approved:** You will receive an email notifying you that your Product is now available to customers. You can return to your dashboard to edit your Product details; add, deactivate, or remove the Product, Tiers, or Offers; and view business reports. Note: if you deactivate or remove any Products, Tiers, or Offers you must continue to honor logins and subscription access for existing SWA customers until their subscriptions are cancelled or expire.

**If not approved:** You will receive an email from Amazon notifying you of the requested corrections and/or changes before your Product can go live. After you comply with these requests you may re-submit by following the same process described above.

## 6 Business Reports and Customer Service

### 6.1 Business reports

Once your Product is live on Amazon.com, you may use the Subscribe with Amazon business reports to track subscriptions, individual transactions, sales tax details (if applicable), and disbursements to your bank account. Subscribe with Amazon offers four reports, all of which may be accessed from the Seller Central home page by clicking on **Business reports** from within the **Quick links** menu on the right. Please note that some reports are generated approximately in real-time (data may be up to 24 hours delayed) while others are generated on a monthly basis; please refer to the **Business reports** page or the following descriptions for more details.

In cases where you require information that is not available directly in the business reports, you may consider supplementing the reports with calls to the SWA APIs to pull in the relevant data. For example, if you wish to add `beneficiaryId` as a column to the standard subscription report, you may use the `receiptId` from the report to call the `Get Subscription` API to retrieve this data.

#### 6.1.1 Transaction report

This is a report of all new subscriptions, renewals, and cancellations. Transaction reports are **generated once per month**, between the 10th and 15th of each month. Selecting a date range for the report will return all transactions that occurred within that range. The transaction report is useful for tracking “historical data” about a subscription or a customer, such as the number of times a subscription has renewed or when a customer exited an introductory pricing period. You may also use the transaction report to calculate aggregate metrics like renewal rate and free-to-paid conversion.

#### 6.1.2 Payment report

This is a report of disbursements to your bank account. Payment reports are **generated once per month**, between the 27th and the last day of each month. Selecting a date range for the report will return all disbursements that occurred within that range.

#### 6.1.3 Tax report

This is a report of required data for completion of tax documents, in accordance with your Amazon Tax Collection Services settings (refer to <https://sellercentral.amazon.com/gp/tax-manager/dashboard.html> for more details). Data may take **up to 24 hours** to appear in the tax report. Note that tax reports do not include free trials.

#### 6.1.4 Subscription report

This is a report of your active and inactive subscribers. Data may take **up to 24 hours** to appear in the subscription report. Selecting a date range will return the **current status** of all subscriptions that began within that date range. For example, selecting “month to date” will return the current status of all subscriptions with a start date in the current month. It will NOT return any data for subscriptions with a start date before or after the selected date range. The subscription report is useful for getting a “snapshot” view of the subscriptions to your Product, including such aggregate metrics as current active subscriptions.

### 6.2 Customer service

Customers are encouraged to contact Amazon customer service as their first point of contact for questions related to subscriptions purchased through SWA. However, there will be instances when a customer contacts your customer service first and, in case of in-depth product or service-related questions, when we will refer

a customer to contact your customer service using the phone number and/or email address you provide when adding your Product. The following sections outline some common customer service scenarios and recommended actions to take.

### **6.2.1 Use cases resolved by Amazon customer service**

Amazon has a customer service team that is specifically trained to assist customers with general questions or issues related to their SWA subscription. For customers who have purchased their subscription through Subscribe with Amazon, the SWA customer support team is able to resolve questions about:

- General Amazon.com account inquiries
- Subscription purchases or plan modification
- Subscription cancellations or auto-renew status
- Login or subscription access issues
- How to use or manage Login with Amazon (LWA) settings
- Payment method or billing inquiries
- Product or device compatibility information provided on the Amazon.com Product detail page

### **6.2.2 How to refer a customer to Amazon customer service**

If a Subscribe with Amazon customer reaches out to your customer service department regarding any of the topics listed above or needs to be referred to Amazon customer service for another reason, please provide the customer with the SWA-specific customer service contact information below. **DO NOT refer the customer to the general Amazon customer service phone number or website.**

#### **Subscribe with Amazon customer service**

Hours of Operation: 3am-10pm PST

SWA CS Phone: 1-888-283-0524

SWA CS Email: [subscriptionservices-cs@amazon.com](mailto:subscriptionservices-cs@amazon.com)

SWA Contact Us URL: <https://www.amazon.com/gp/help/customer/contact-us/?ie=UTF8&initialIssue=cu-swa>

We recommend that you update your internal customer service training and, if applicable, public-facing help content with the information provided above before you submit your Product for approval.

### **6.2.3 Amazon customer service referrals to your CS department**

Amazon will instruct customers to contact your customer service department only when necessary using the phone number and/or email address you provide when adding your Product. Scenarios when a referral may be made include:

1. A customer has a product or service question that cannot be resolved by referring to the Amazon.com Product detail page
2. The customer has a technical question about the service that is not related to the SWA subscription purchase, plan modification, access, or billing.