Amazon Flexible Payments Service
Getting Started Guide
API Version 2010-08-28

Amazon Web Services
Copyright © 2012 - 2013 Amazon Web Services LLC or its affiliates. All rights reserved.

The following are trademarks or registered trademarks of Amazon: Amazon, Amazon.com, Amazon.com Design, Amazon DevPay, Amazon EC2, Amazon Web Services Design, AWS, CloudFront, EC2, Elastic Compute Cloud, Kindle, and Mechanical Turk. In addition, Amazon.com graphics, logos, page headers, button icons, scripts, and service names are trademarks, or trade dress of Amazon in the U.S. and/or other countries. Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon.

All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

FPS Getting Started Guide

Welcome

This guide provides a conceptual overview of the Amazon Flexible Payments Service, helps you to set up required accounts, and leads you through the steps of making a payment. The major sections of this guide are described in the following table.

Amazon Flexible Payments Service (FPS) is a web service that enables developers to accept payments on their website. The payments can be for selling goods or services, raising donations, executing recurring payments, and sending payments. For more information about this product, go to Amazon Flexible Payments Service (Amazon FPS).

> **Note**
>
> The Amazon Payments service has been designed and developed for use within a web browser only. Our service cannot be used within a native application (including, without limitation, iOS, Android, RIM and Windows operating systems). Amazon Payments reserves the right to suspend the Payment Account of any user of our services that has implemented our Services within a native application.

How Do I...?

How do I…?	Topics
Learn about Amazon FPS	Introduction to Amazon Flexible Payments Service
Get started with Amazon FPS API	Getting Set Up
Process an Amazon FPS payment transaction	Making a Payment
Find Amazon FPS sample code and libraries	Download Amazon FPS SDK
Learn which Amazon FPS Quick Start guides to read	New Product Development

Introduction to Amazon Flexible Payments Service

This introduction to Amazon FPS provides a high-level overview of this web service. After reading this section, you should understand the basics you need to work through the examples in this guide.

Overview of Amazon FPS

Amazon Flexible Payments Service is the first payments service designed for developers. This set of web service APIs differs from other Amazon Payments products, such as Checkout by Amazon, because it allows the development of highly customized payment solutions for a variety of businesses. The foundation of Amazon FPS is Amazon's reliable and scalable payments infrastructure. The service provides a convenient way for the tens of millions of Amazon customers to use their current account information to pay for items using Amazon Payments.

Because Amazon FPS is an extensive service, we've divided the functionality into the following five Quick Start implementations:

- Amazon FPS Basic Quick Start—For single payments

  Enable payments for e-commerce, digital content, services, or other one-time use cases. This includes the flexibility to charge consumers immediately when the order is fulfilled, or later when they return to the website after confirming the payment. Amazon FPS supports multiple payment methods (for example, credit cards, bank accounts, or Amazon Payments account balances). For more information, go to the Amazon FPS Basic Quick Start.

- Amazon FPS Marketplace Quick Start—For facilitating transactions between a buyer and a third party seller

  Facilitate transactions between a buyer and a third party seller, take a cut of the transaction, and have control over who pays the transaction processing fees. For more information, go to the Amazon FPS Marketplace Quick Start.

- Amazon FPS Advanced Quick Start—For multiple or recurring payments

  Allows you to handle payments for subscriptions, and situations where you charge a buyer more than once based on the buyer's initial authorization. For more information, go to the Amazon FPS Advanced Quick Start.

- Amazon FPS Account Management Quick Start—For programmatic access to account activity

  This guide gets you started using the Amazon FPS Basic Quick Start implementation. There's not a separate Getting Started Guide for each of the Quick Start

implementations. However, you can apply the basic concepts and code from this guide to any of the other Quick Start implementations. For more information about use cases for each implementation, see “Which Parts of Amazon FPS Should You Use?”

Amazon FPS Concepts

This section describes Amazon FPS basic concepts and terminology.

Sender, Recipient, and Caller Actions

Participants involved in an Amazon FPS transaction perform one or more of the following actions:

- The sender sends money

  The buyer, known as the sender in an Amazon FPS transaction, makes the payment for purchasing goods or services.

- The recipient receives money

  The merchant (or seller), also known as the recipient in an Amazon FPS transaction, receives payment for the goods or services sold to the sender.

- The caller makes Amazon web service calls to enable money transfer

  The developer, also known as the caller in an Amazon FPS transaction, can transfer money between a sender and a recipient in a transaction. A caller can also perform the role of a sender or a recipient. A caller must have an AWS account as well as an FPS developer account. For information about getting your accounts, see “Getting Set Up.”

  Important
  Amazon FPS does not allow a participant to be both the sender and the recipient in a transaction. However, the caller and the recipient can be the same.

Amazon Payments Account Types

When you use Amazon FPS, you and your customers use Amazon Payments accounts to send and receive money. The following table describes the types of Amazon Payments accounts.

Account Type	Description
Personal Account	A Personal account holder can send or receive money using bank accounts or Amazon Payments account balances. This account is for users making online purchases. A Personal account holder can send money using credit cards but cannot receive payments from credit cards. An Amazon Payments account is automatically created for users who already have an Amazon.com account, which lets users make purchases using their credit cards already on file with Amazon.com. There is no separate registration process. This Amazon Payments account is activated when the Amazon.com account holder makes their first payment on any third-party website that accepts Amazon Payments.
Business Account	In addition to sending and receiving payments using bank accounts and Amazon Payments account balance, a Business account holder can also receive payments using credit cards. Business accounts are for recipients who rely on credit card payments for the sale of their goods or services.
FPS Developer Account	This is a Business account with the added ability to make Amazon FPS web service calls (for this reason, this type of account is also referred to as a caller account).
FPS Sandbox Account	The sandbox enables you to make Amazon FPS web service calls to a test environment without incurring any charges. For more information, see “Sign Up for an Amazon FPS Sandbox Account.”

You, as the developer, need an FPS Sandbox account, an AWS account, and a FPS developer account. For information about getting your accounts, see “Getting Set Up.”

Two Interfaces to Integrate With

Your website or application must integrate with two interfaces:

- Co-branded service API—To obtain the buyer's authorization of the payment. For more information, see “Making a Co-Branded Service Request”
- Amazon FPS API—To process the payment itself. For more information, see “Making a Pay Request and Handling the Response”

This guide presents code for making requests to both interfaces, and code for handling the responses. Code is presented in several programming languages, including C#, Java, Perl, and PHP.

Sandbox

Amazon FPS provides an environment called the sandbox for testing your applications. In the sandbox you can try out your requests without incurring charges or making purchases. We recommend that you test all of your requests in the sandbox before exposing them on your website.

To learn how to get a sandbox account, see “Sign Up for an Amazon FPS Sandbox Account.”

Getting Set Up

This section describes how to set up the accounts and download the code samples you need to get started using Amazon FPS.

Sign Up for an Amazon FPS Developer Account

First, sign up for your Amazon FPS developer account:

1. Go to http://payments.amazon.com.
2. Click on the **Developers** tab.
3. Click on the **Sign Up For Amazon FPS** button.
4. Follow the instructions on the subsequent pages to set up a developer account.

Amazon Payments Business Account

An Amazon Payments business account is a requirement for using some Amazon FPS products. You create a business account, which enables you to receive payments from Amazon, as the part of creating a developer account.

Sign Up for an Amazon Web Services Account

You sign up for an Amazon Web Services (AWS) account as the part of creating an Amazon FPS developer account. The AWS account allows you to make API calls to the Amazon FPS Sandbox and production environment.

You can also sign up for an AWS account at the AWS website at http://aws.amazon.com.

The following security credentials are automatically associated with your AWS account:

- AWS Access Key ID—You use this to identify yourself when you send requests to the Co-branded service or when you send REST requests to Amazon FPS.
- AWS Secret Access Key—You use this to generate URL signatures that provide tamper-proof requests.

> **Important**
> You can use these security credentials in both the Sandbox and the production environment.

To view your Access Key ID and Secret Access Key

1. Go to the Amazon Security Credentials page at http://aws.amazon.com/security-credentials. If you are not logged in, you will be prompted for your user name and password.
2. Your Access Key ID is displayed on the **Security Credentials** page in the **Access Credentials** area. Your Secret Access Key remains hidden as a further precaution as shown in the following figure.
3. To display your Secret Access Key, on the **Access Keys** tab, under **Secret Access Key**, click **Show**.

The AWS Security Credentials page also enables you to manage a second set of credentials for rotation. This feature enhances the security of your account.

For more information, see “Access Key Rotation” in the Amazon FPS Basic Quick Start Developer Guide.

Sign Up for an Amazon FPS Sandbox Account

The Amazon FPS Sandbox is a test environment that you can use to test all the features of Amazon FPS without moving real money or paying any fees. We recommend that you test all of your requests in the Sandbox before exposing them on your production website.

You can use the Amazon FPS Sandbox to:

- Make web service API and Co-branded service requests
- Test credit cards and bank accounts in your test transactions without any prior verification
- Simulate certain error conditions that could appear in a real transaction to test your application's error handling
- Test the end-to-end user experience without incurring expenses or making purchases

To create an Amazon FPS Sandbox account

1. Go to http://aws.amazon.com/fps.
2. Scroll down, and on the left side of the screen, click **Enter the Sandbox**.
3. Sign in with your Amazon Web Services (AWS) login email address and password. Your Sandbox account is created.
4. Click **Continue**. The **Amazon Flexible Payments Service Sandbox** page appears.
5. If you do not already have an Amazon FPS Sandbox business account, create a Sandbox business account by clicking the link for a business account in the middle of the page.

Important Sandbox URLs

You can use the following URL endpoints to test your applications in the Amazon FPS Sandbox.

If you need to	Use
View the Sandbox account management UI	Go to https://payments-sandbox.amazon.com and then click the **Your Account** tab.
Make Co-branded service requests	https://authorize.payments-sandbox.amazon.com/cobranded-ui/actions/start
Make web service API requests	https://fps.sandbox.amazonaws.com/

Download an Amazon FPS SDK

To use the example code in this guide, you must have one of the SDKs in the following list. Download the SDK of your choice and extract it to a convenient location. Then read the ReadMe file in the SDK root folder for specific instructions on how to set up the SDK for use.

- C# - Amazon FPS Quick Starts: Standalone Library in C#
- Java - Amazon FPS Quick Starts: Standalone Library in Java
- Perl - Amazon FPS Quick Starts: Standalone Library in Perl
- PHP - Amazon FPS Quick Starts: Standalone Library in PHP

For detailed information on the samples, see “Code Samples” in the Amazon Flexible Payments Service Basic Quick Start.

The central location for Amazon FPS sample code is http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=123.

The following table lists the minimum required dependencies for each SDK.

Language and Interface	Software	Download Location
C#: REST	NET Framework 2.0	http://msdn.microsoft.com/en-us/netframework/default.aspx
Java: REST	JDK: J2SE 5.0 JDK 1.5.0	http://java.sun.com/javase/downloads/index.jsp
	Apache ANT 1.6.5	http://ant.apache.org/bindownload.cgi
	Apache Tomcat 5.5	http://tomcat.apache.org/
Perl: REST	Perl 5.6.0	http://www.perl.org/get.html
	Digest::SHA package	http://www.opencsw.org/packages/pm_digestsha
	Crypt::OpenSSL:: RSA package	http://search.cpan.org/~cdrake/Crypt-OpenSSL-RSA-0.22/RSA.pm
	Crypt::X509 package	http://search.cpan.org/~leo/Convert-X509-0.1/X509.pod
	Crypt::OpenSSL:: X509 package	http://search.cpan.org/~dland/Crypt-SSLeay-0.57/SSLeay.pm
PSP: REST	PHP 5.2.2	http://www.php.net/downloads.php
	Apache Web Server 2.0	http://httpd.apache.org/

Making a Payment

This section describes how a payment works and guides you through a use case scenario by demonstrating how you can make a payment transaction using the Amazon FPS.

Payment Transaction Use Case

For this use case, let's assume that DigitalDownload is a licensed seller of popular music downloads. John visits the DigitalDownload website to purchase a song. John can set up a payment authorization that allows DigitalDownload to charge his payment instrument for his purchase.

In this use case:

- DigitalDownload is the caller who makes web service calls to execute payment transactions. To use FPS, DigitalDownload must have an Amazon FPS developer account to make web service calls.
- DigitalDownload also plays the role of a recipient who sells the content and receives payments for the sales.
- John is the sender who pays for the content. He must have an Amazon Payments account (either a Personal account or a Business account).

For an overview of how making a payment on the DigitalDownload website would work, see "How a Payment Works."

The following table describes the tasks required to process John's payment.

Process for Making a Payment

1. Make a request to the co-branded service to get John's authorization for the payment. For more information, see "Making a Co-branded Service Request."
2. Make a Pay request to the Amazon FPS API to transfer the money from John's account to DigitalDownload. For more information, see "Making a Pay Request and Handling the Response."

How a Payment Works

The following process describes the steps involved in making a payment using Amazon FPS.

> **Note**
> In the process, the caller (the fictitious company DigitalDownload) is also the recipient.

A buyer who is ready to authorize a purchase clicks the **Buy Now** button. The button redirects the buyer from the DigitalDownload site to the Amazon Payments Co-Branded UI (CBUI) using a signed URL

The signed URL contains all the parameters required to make this transaction, a signature to secure the request, and a returnURL to redirect the customer back to the DigitalDownload destination page.

The CBUI's authorization pipeline asks the buyer to sign in, specify a personal payment instrument (like a credit card), and authorize the purchase.

> **Note**
> To make for a better buying experience, DigitalDownload uses the co-branding feature of the Co-Branding service to include their branding on the CBUI payment authorization web pages. The **Buy Now** button powered by Amazon FPS redirects buyers away from the DigitalDownload website to Amazon's. But because of the co-branding on Amazon's CBUI web pages, buyers don't feel as if they've completely left the DigitalDownload website. The CBUI provides continuity between the checkout and payment authorization experience.

The Co-branded service returns the buyer to the DigitalDownload site. In the return URL, the service passes the sender token, which represents the buyer's authorization of the payment. For more information, see “Making a Co-branded Service Request.”

DigitalDownload then sends a Pay request to the Amazon FPS API and includes the sender token in the request. The response from Amazon FPS is an XML document containing the status of the request. If the request succeeds, DigitalDownload lets the buyer download the music. For more information, see “Making a Pay Request and Handling the Response.”

Making a Co-Branded Service Request

Before you can make a Pay request, you must first create a payment token, which represents the sender's authorization of the payment. You do this by sending a Co-Branded service request. The response contains a sender token ID, which is a reference to the payment token. The following diagram shows the process for the DigitalDownload use case.

1. On the DigitalDownload website, the sender (John) selects a song or a video and clicks **Buy Now**.
2. DigitalDownload (the caller) sends a Co-branded service request, which directs John to the Amazon-hosted Co-branded User Interface (CBUI) web pages. A sample Co-branded service request follows this procedure.
3. John signs in on the CBUI pages with his Amazon Payments login name and password. The Payment Authorization page appears.
4. John selects a payment method and clicks the **Continue** button. A confirmation page displays the payment details.
5. John clicks the **Continue** button after reviewing the details.
6. Amazon FPS redirects John away from the Amazon-hosted CBUI pages to the URL DigitalDownload specified in the returnURL parameter in the Co-branded service request.

Obtaining a Payment Token

To get the payment token, you send a Co-branded service request. The request contains all the information required to enable Amazon Payments to issue a payment token. Following is an example request for the DigitalDownload use case. Note that the endpoint is for the Amazon FPS sandbox.

> **Note**
> The sender logs in with their Amazon.com account credentials. The sender's login account must be different than the caller Amazon account.

```
https://authorize.payments-sandbox.amazon.com/cobranded-
ui/actions/start? callerKey=[Access Key ID]
&callerReference=[Unique caller reference identifier]
&currencyCode=USD &paymentMethod=ABT,ACH,CC
&paymentReason=Now%20and%20Forever%20-%20Richard%20Marx
&pipelineName=SingleUse &returnURL=[URL to return to]
&signature=[calculated signature value]
&signatureMethod=HmacSHA256 &signatureVersion=2
&transactionAmount=0.90 &version=2010-08-28
```

To get the senderTokenId, the sender must sign into a valid Amazon account, and go through the CBUI.

The following table describes the parameters in this request.

Parameter	Definition
callerKey	DigitalDownload's AWS Access Key ID.
callerReference	A unique value DigitalDownload generated to identify the sender token for future references.
currencyCode	Indicates the token's monetary currency, such as USD (U.S. Dollars).
paymentMethod	Payment methods that DigitalDownload supports. In this case the payment methods include an Amazon Payments account balance, bank transfers, and credit cards.
paymentReason	Description of this transaction. John can see this on the **Payment Authorization** page.
pipelineName	Name of the particular CBUI authorization pipeline DigitalDownload requested. The value SingleUse refers to the pipeline that creates a payment token that is to be used only once. For information about payment tokens that can be used more than once (e.g., for recurring payments), go to the Amazon FPS Advanced Quick Start.
returnURL	The destination website John is redirected to after completing the payment authorization. This is a location on DigitalDownload's site.
signature	A value Digital Download calculated using an HmacSHA256 or HmacSHA1 encryption. For information about how to create the signature, go to the Amazon FPS Basic Quick Start.
signatureVersion	A value that specifies the signature format.
signatureMethod	A value that specifies the signing method. For information on signing your request, see “Working with Signatures” in the Amazon FPS Basic Quick Start.
transactionAmount	The total purchase price, including tax and shipping.
version	The version of the Co-branded service API to use. This should always be set to 2009-01-09.

Generating the Signed Co-Branded Service Request

Download one of the following SDKs to get code samples for generating a Co-Branded service request in the language of your choice. The ReadMe file in the SDK root folder of each SDK provides step-by-step instructions about how to get set up and how to generate a Co-Branded service request.

- C# - Amazon FPS Quick Starts: Standalone Library in C#
- Java - Amazon FPS Quick Starts: Standalone Library in Java

- Perl - Amazon FPS Quick Starts: Standalone Library in Perl
- PHP - Amazon FPS Quick Starts: Standalone Library in PHP

Validating the Co-Branded Service Response

After you send a Co-Branded service request, the service redirects the buyer to the web page you specified in the returnURL request parameter. The Co-Branded service appends parameters to that URL to give you information about the status of the request. One of the parameters is a signature, which you should validate to confirm the request came from the Co-Branded service. The SDKs contain code samples to show how to parse the URL and validate the signature.

Making a Pay Request and Handling the Response

This section explains how to make a Pay request, which uses the senderTokenId you received in the Co-Branded service response. The Pay request initiates the Amazon FPS transfer of money from the sender's payment instrument to the recipient's Amazon Payments account.

> **Note**
> The sender must login with their Amazon.com account credentials. The sender account must be a different account than recipient Amazon account.

If you have not already done so, download one of the following SDKs to get code samples in the language of your choice. The ReadMe file in the SDK root folder of each SDK gives step-by-step instructions on how to make a Pay request and handle the response.

- C# - Amazon FPS Quick Starts: Standalone Library in C#
- Java - Amazon FPS Quick Starts: Standalone Library in Java
- Perl - Amazon FPS Quick Starts: Standalone Library in Perl
- PHP - Amazon FPS Quick Starts: Standalone Library in PHP

Resending a Request

Your Pay request or the response from Amazon FPS might not reach its destination due to underlying network failures. If this happens, you can resend the same request within 7 days, and Amazon FPS returns the result of the previous request. Amazon FPS doesn't execute a new transaction.

> **Important**
> All the parameters in the request you resend should have exactly the same parameters and values (inclusive of date and time) that were in the initial request. Any change in the request will result in Amazon FPS returning an error or processing this request as a new transaction.

Sample Response

When you send the Pay request, you receive a response similar to the following.

```
<PayResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <PayResult>
     <TransactionId>
          13N8UPFET32I4I7FCF9T4ZKFETETINTK56Q
       </TransactionId>
     <TransactionStatus>
          Pending
       </TransactionStatus>
  </PayResult>
  <ResponseMetadata>
     <RequestId>
          b415f09d-5924-4315-b31a-21c977c85c39:0
       </RequestId>
  </ResponseMetadata>
</PayResponse>
```

1. TransactionId - The transaction ID for this transaction.
2. TransactionStatus - The status of the payment transaction.
3. ResponseMetatdata - The request ID of the Pay action request.

You should store the transaction ID and request ID in your database and cross-reference them to the caller reference value you provided in the Pay request.

New Product Development

Now that you have completed the tasks of sending Co-Branded service and Amazon FPS requests and processing the results, you are ready to start designing your own application. Most applications built for Amazon FPS will not be as simple as the example in this guide, so this section touches on a few more helpful pieces of information.

Which Parts of Amazon FPS Should You Use?

The sample requests and responses in this guide provide only a very narrow slice of Amazon FPS functionality. The full product is divided into the five Quick Start implementations mentioned earlier in the guide (see “Introduction to Amazon Flexible Payments Service”). The following table gives guidance on which Quick Start implementation to use depending on your type of business and use case. You might use more than one.

Domain	Use Case	Quick Start to Use
E-commerce	Selling physical (shippable) goods on your website.	Amazon FPS Basic Quick Start
Magazines	Selling magazines or periodicals for a monthly subscription.	Amazon FPS Advanced Quick Start
Digital media and content	Selling digital goods or services on a pay-per-use basis. For example, selling MP3 songs on your website or selling points for your game.	Amazon FPS Basic Quick Start
Digital media and content	Selling digital goods or services on with a monthly subscription. For example, music streaming or monthly digital report subscriptions.	Amazon FPS Advanced Quick Start
Donations	Raising online donations for your non-profit organization.	Amazon FPS Basic Quick Start
Marketplace	Providing a marketplace where senders can pay multiple recipients.	Amazon FPS Marketplace Quick Start

Domain	Use Case	Quick Start to Use
Social networks	Enabling users on your social networks to pay each other. For example, collecting payments for meetings.	Amazon FPS Marketplace Quick Start
Social networks	Raising contributions for a project on your social network and charging a fee for it.	Amazon FPS Marketplace Quick Start
All	Checking your account activity, transaction history, or account balance.	Amazon FPS Account Management Quick Start

All of the preceding Quick Start implementations use the same WSDL, but each one covers a different feature set. Likewise, you use different Co-Branded service APIs with each implementation to create different kinds of payment tokens. In this getting started guide, you created a single-use payment token; that is, a token that can be used once, however you can also create multiple-use payment tokens such as Amazon FPS Advanced Quick Start offers.

Application Considerations

The Amazon Payments home page (payments.amazon.com) provides a user interface for senders and callers to get Amazon Payment accounts, view them, and to perform some actions with them, such as canceling purchases. Using the Amazon FPS API, you can also program these functions into your website. Your application needs to make sure that actions your website performs are in sync with the actions taken in the Amazon Payments user interface, and vice versa. Amazon Payments sends transaction notifications for actions taken in the Amazon Payments user interface and some actions executed by Amazon FPS web service requests. Your application needs to be able to receive these Instant Payment Notifications (IPN) and change your records accordingly. IPN is covered in all of the different Amazon FPS developer guides.

Your application needs to keep a database of sender information such as the sender's name, shipping information, and, to keep those values secure, sender login names and passwords. As you create a database of senders, you must store with each sender transaction the callerReference value that you generate and the transactionId Amazon FPS returns in each Pay and Reserve response. These identifiers enable you to implement the full range of Amazon FPS functionality, such as providing transaction status information.

The functionality you make available on your website or in your application should be sender-focused. Common sender tasks (beyond paying for a purchase) include canceling a purchase, asking for a refund, and getting the status of an order. Although senders can go to the Amazon

Payments website to view this information, they typically expect your website to also provide this functionality.

Not discussed in this guide is the order fulfillment side of your business. Because the order date and shipping date of products typically fall on different days, a common business practice is to charge a sender's payment instrument only upon product shipment. Remember that you need to update Amazon FPS regarding such events. In this case, you send a Settle request to charge the sender's payment instrument, causing Amazon FPS to initiate the transfer of money to the recipient. We recommend you implement this functionality in your order fulfillment software so that you never forget to send Amazon FPS a Settle request.

In this guide, you, the developer, also serve as the recipient of the money. Amazon FPS Marketplace Quick Start enables a different scenario in which you are neither the sender nor the recipient, but the caller that hosts the recipient's e-commerce store. For that service, you charge the recipient a fee. This unique feature available in Amazon FPS provides a range of application possibilities unavailable in other payment web services.

This guide describes how to redirect buyers to the Co-Branded UI where they authorize payment. You have the option of co-branding that UI with your company's logo. The value of co-branding the CBUI web pages is that your branding appears along with Amazon's at the top of the payment authorization web pages. This makes for a smoother transition between the shopping and payment authorization steps in the purchase. To co-brand the web pages, you must provide the URL of your company's branding in the Co-branded service request.

All of these topics are covered in greater detail in the Amazon FPS developer guides. Links to the guides are provided in the table in “Which Parts of Amazon FPS Should You Use?” and in “Amazon FPS Resources.”

Production Endpoints

After you have tested your applications using the sandbox, you can start using the production environment by changing the endpoints of the Co-branded service and Amazon FPS requests to the following URLs:

Co-branded service API— https://authorize.payments.amazon.com/cobranded-ui/actions/start

Amazon FPS API— https://fps.amazonaws.com/

> **Note**
> All transactions in the production environment require a real payment instrument (credit card, bank account, or Amazon Payments balance) and move real money.

Additional Resources

You might find these additional resources helpful when you design your own application.

Resource	Description
Amazon FPS resources	The primary web page for information about building on Amazon FPS, which includes links to the WSDL, documentation, code samples, and more.
Developer Forums	A community-based forum for developers to discuss Amazon FPS and interact with other developers.

Amazon FPS Resources

The following table lists related resources that you'll find useful as you work with this service.

Resource	Description
Amazon Flexible Payments Service Getting Started Guide	Gets you set up with Amazon FPS, and shows you how to implement a simple one-time payment using Amazon FPS Basic Quick Start.
Amazon Flexible Payments Service Marketplace Quick Start	Covers the marketplace functionality of Amazon FPS.
Amazon Flexible Payments Service Advanced Quick Start	Covers the multiple-payment functionality of Amazon FPS.
Amazon Flexible Payments Service Account Management Quick Start	Covers the account management functionality of Amazon FPS.
FAQs	Frequently asked questions about using Amazon FPS.
Release Notes	Provides a high-level overview of the current release, noting any new features, corrections, and known issues.
FPS Developer Resource Center	A starting point specifically for FPS documentation, code samples, release notes, and other information to help you build innovative applications.
Discussion Forums	A community-based forum for developers to discuss technical questions related to Amazon FPS.
Product information about Amazon FPS	The primary web page for information about Amazon FPS.
Contact Us	A central contact point for inquiries concerning AWS billing, accounts, events, abuse, and more.
Conditions of Use	Detailed information about Amazon.com copyright and trademark usage and other topics.

Document History

This documentation is associated with the 2010-08-28 release of the Amazon Flexible Payments Service. This guide was last updated on 16-September-2012.

Change	Description	Release Date
Enhancement	Added minor changes and typographical fixes applied from a maintenance edit.	In this release
Feature Deprecation	Amazon FPS has removed support for signature verification using signature version 1. If your application is using this feature, you must convert to a server-side call with the VerifySignature action. For more information, see Amazon Flexible Payments Service Basic Quick Start.	2011-02-10
Feature Deprecation	Amazon FPS has removed support for client-side signature verification using PKI. If your application is using this feature, you must convert to a server-side call with the VerifySignature action. For more information, see Amazon Flexible Payments Service Basic Quick Start.	2011-02-10
Maintenance Update	Added an expanded breadcrumb to the HTML version, and reduced the front matter content. In addition several minor fixes and corrections have been applied.	2010-11-01
Enhancement	The RecipientVerificationStatus data type (use by the GetRecipientVerificationStatus action) has been enhanced to enable you to determine whether a verified customer account is unlimited in the amount of money it can receive. For more information, see the Amazon Flexible Payments Service Advanced Quick Start or the Amazon Flexible Payments Service Marketplace Quick Start.	2010-11-01
Feature Extension	Amazon FPS has extended support for signature verification using signature version 1 to 10 February 2011. If your application is using this feature, you must convert to a server-side call with the VerifySignature action. For more information, see Amazon Flexible Payments Service Basic Quick Start.	2010-11-01
Feature Extension	Amazon FPS has extended support for client-side signature verification using PKI until 10 February 2011. If your application is using this feature, you must convert to a server-side call with the VerifySignature action. For more information, see Amazon Flexible Payments Service Basic Quick Start.	2010-09-14
Enhancement	Changed the values for soft descriptors from "AMZ*" and "AMZN PMNTS" to "ASI*" and "Amazon Payments", respectively. Also, applied minor changes and typographical fixes applied from a maintenance	2010-06-11

Document History

Change	Description	Release Date
	edit	
New Feature	Support for signature version 2, which completely replaced signature version 1 on 10 February, 2011. The enhanced security features include: • a more secure way of calculating signatures for inbound requests and outbound notifications. • support for SHA256 signing algorithm • the new VerifySignature FPS Action for server-side testing of return URL responses and IPN notifications. • support for PKI based authentication for client-side testing of return URL responses and IPN notification. **Note** This feature is deprecated as of 2010-09-14. The instructions in this guide touch on each of the new features of signature version 2. For more complete information, see the Amazon Flexible Payments Service Basic Quick Start.	2009-11-03
Enhancement	The Access Keys page has been renamed the Security Credentials page, located at http://aws.amazon.com/security-credentials. You must be logged in to view this page.	2009-09-09
Feature Deprecation	Amazon FPS has removed the Aggregated Payments option. References to the AWS Developer Resource Center and AWS Support Center have been removed.	2013-08-23
Editorial Update	Added language to clarify that the Amazon Payments service has been designed and developed for use within a web browser only. Our service cannot be used within a native application (including, without limitation, iOS, Android, RIM and Windows operating systems).	2013-10-18