Amazon Flexible Payments Service

API Reference

API Version 2010-08-28

Amazon Web Services

Amazon Flexible Payments Service: API Reference

Amazon Web Services

Welcome

This is the *Amazon Flexible Payments Service API Reference*. This is a detailed reference guide that describes all the API operations for Amazon FPS in detail. In addition, it provides sample requests, responses, and errors for the supported web services protocols.

Amazon Flexible Payments Service is a web service that enables developers to accept payments on their website. The payments can be for selling goods or services, raise donations, execute recurring payments, and send payments.

How Do I...?

How do I?	Relevant Sections
Decide whether Amazon FPS is right for my needs:	Amazon FPS detail page
Get started with Amazon FPS quickly	*Amazon Flexible Payments Service Getting Started Guide*
Learn about actions, common requests, common responses, and errors	API Overview (p. 3)
Find a comprehensive reference to the Amazon FPS APIs	Amazon FPS API Actions (p. 15)
Find a comprehensive reference to the Amazon FPS data types	Amazon FPS Data Types (p. 111)
Find a complete listing of example email messages sent by Amazon Payments in response to Amazon FPS API calls	Example Email Messages (p. 130)
Learn about the common response and request elements for Amazon FPS API calls	Common Request Parameters (p. 5) Common Response Elements (p. 6)
Find a complete listing for Amazon FPS instant payment notification (IPN) status codes and return elements	Amazon FPS Instant Payment Notification Field Reference (p. 125)

How do I?	Relevant Sections
Find a complete listing of errors?	Errors (p. 6)

API Overview

Topics

Amazon FPS provides 22 Query APIs. This section discusses the operations available in the Amazon FPS APIs, their semantics, and their required parameters. Examples of requests and responses are also provided.

API Actions

Actions encapsulate the possible interactions with Amazon FPS. These can be viewed as remote procedure calls and consist of a request and response message pair. Requests must be signed, so that Amazon FPS can authenticate them. For clarity, the sample requests and responses illustrating each operation described in this reference are not signed.

Values provided as parameters to the various actions must be of the indicated type. Standard XSD types (such as `string, boolean, int`) are prefixed with `xs:`. Complex types are prefixed with `xs:complexType`.

Amazon FPS WSDL Location

The current version of the Amazon FPS API is 2010-08-28.

The WSDL is located at https://fps.amazonaws.com/doc/2010-08-28/AmazonFPS.wsdl.

The schema is located at https://fps.amazonaws.com/doc/2010-08-28/AmazonFPS.xsd.

API Versioning

Because features and changes can introduce incompatible API changes, all Amazon FPS API updates are versioned. By including a version in the request, clients receive responses they can process.

Each API revision is assigned a version in date form (the current API version is 2010-08-28). This version is included in the request as part of the document namespace when making a request. The response that Amazon FPS returns honors the version included in the request.

Note
The WSDL should be treated as a moving target because it will always map to the latest release of the Amazon FPS API. If your software depends on retrieving the WSDL at runtime, we strongly recommend that you reference the specific version of the WSDL you are developing against.

Available Libraries

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific APIs instead of SOAP and Query. These libraries provide basic functions (not included in the APIs), such as request authentication, request retries, and error handling so that it is easier to get started. Libraries and resources are available for the following languages:

- Java
- Mobile
- PHP
- Ruby
- Windows and .NET

For FPS libraries and sample code in these and many other programming languages (many contributed by the AWS developer community), go to Sample Code & Libraries.

Amazon FPS Endpoints

Amazon FPS has four endpoints where you send requests, listed in the following table. Two are for sandbox testing of CBUI and API requests, and two are for production Co-Branded User Interface (CBUI) and API requests.

Endpoint	Purpose
`https://authorize.payments-sandbox.amazon.com/cobranded-ui/actions/start`	Sandbox endpoint for Co-Branded service requests.
`https://authorize.payments.amazon.com/cobranded-ui/actions/start`	Production endpoint for Co-Branded service requests.
`https://fps.sandbox.amazonaws.com`	Sandbox endpoint for Amazon FPS actions.
`https://fps.amazonaws.com`	Production endpoint for Amazon FPS actions.

Common Request Parameters

Each action in the API has its own specific set of parameters, but there is also a set of parameters that all actions use. This section describes those input parameters.

You only need to add these parameters in REST requests. SOAP requests include them by default.

The following table describes parameters that can be used in all requests.

Parameter	Description	Required
Action	The API operation, for example, `Settle` or `Refund`. Type: String: Default: None Constraint: Must be a valid operation such as Cancel, Refund, and so on.	Yes
AWSAccessKeyId	A string, distributed by Amazon FPS when you sign up to be a developer, that uniquely identifies the caller. Type: String Default: None	Yes
Signature	A value calculated using the request parameters and a SHA256 (preferred) or SHA1 HMAC encryption algorithm. Type: String Default: None	Yes
SignatureVersion	A value that specifies the *Signature* format. Type: Integer Default: None Valid Value: 2	Yes
SignatureMethod	A value that specifies the signing method. Type: String Default: None Valid Values: HmacSHA256 (preferred) and HmacSHA1.	Yes
Timestamp	A date-time value that marks the day and time the request was sent. Requests expire after a certain length of time to prevent malicious users from capturing requests and resubmitting them at a later time. Type: dateTime, for example, 2008-09-18T13:00:01Z Default: None	Yes
Version	The version number of the WSDL to use in processing the request. Version numbers are dates, such as 2008-09-17. For a list of version numbers, go to the Amazon Resource Center at http://aws.amazon.com/resources. Type: String Default: None	Yes

Common Response Elements

Each action in the API has its own set of response elements it uses. There are, however, a set of response elements that all actions use. The following table describes those common elements.

Element	Description
`ResponseMetadata`	Container element.
`RequestId`	Amazon FPS returns a *`RequestId`* element for every API call accepted for processing. The request ID is a reference to your API request that Amazon FPS can use to troubleshoot any issues related to the request. We recommend you store the request ID value for future reference. Because responses and requests can return asynchronously, you can use the request ID to sync responses with requests. Type: String Max Size: 64 Bytes
`signatureVersion`	A value that specifies the *`Signature`* format. Type: Integer Valid Values: 2
`signatureMethod`	A value that specifies the signing method. Type: String Valid Values: HmacSHA256 (preferred) and HmacSHA1.

Errors

Error	Description
`AccessFailure`	Account cannot be accessed. You can display the following message to your customers: Your account cannot be accessed. Retriable: Yes
`AccountClosed`	Account is not active. You can display the following message to your customers: *Your account is closed.* Retriable: Yes

Error	Description
AccountLimitsExceeded	The spending or receiving limit on the account is exceeded. This error can also occur when the specified bank account has not yet been verified. You can display the following message to your customers: *You have exceeded your spending or receiving limits. You can view your current limits at http://payments.amazon.com/sdui/sdui/viewlimits. You can upgrade these limits by adding and verifying a bank account as a payment method. Please visit Adding and Verifying Bank Accounts to learn how to add and instantly verify a bank account.* Retriable: Yes
AmountOutOfRange	The transaction amount is more than the allowed range. Ensure that you pass an amount within the allowed range. The transaction amount in a Pay operation using credit card or bank account must be greater than $0.01. Retriable: No
AuthFailure	AWS was not able to validate the provided access credentials. Please make sure that your AWS developer account is signed up for FPS. Retriable: Yes
ConcurrentModification	A retriable error can happen when two processes try to modify the same data at the same time. The developer should retry the request if this error is encountered. Retriable: Yes
DuplicateRequest	A different request associated with this caller reference already exists. You have used the same caller reference in an earlier request. Ensure that you use unique caller references for every new request. Even if your earlier request resulted in an error, you should still use a unique caller reference with every request and avoid this error. Retriable: No
InactiveInstrument	Payment instrument is inactive. The payment instrument is inactive, for example, a credit card has expired. Retriable: No

Error	Description
IncompatibleTokens	The transaction could not be completed because the tokens have incompatible payment instructions. If any assertion in one of the payment instructions fails, this error is displayed. As such, it may be caused by a number of reasons, for example: • One or more tokens has expired. • The recipient specified in the token is different from the actual recipient in the transaction. • There is violation on the amount restriction. • This token cannot be used with your application as another application has installed it.
InstrumentAccessDenied	The external calling application is not the recipient for this postpaid or prepaid instrument. The caller should be the liability holder. You are trying to access an instrument that you do not own.
InstrumentExpired	The prepaid or the postpaid instrument has expired. You must ask your customers to set up a new prepaid or postpaid agreement
InsufficientBalance	The sender, caller, or recipient's account balance has insufficient funds to complete the transaction. You must ask your customers to fund their accounts. You can then retry this request. Funding an account can take up to three to four business days using a bank account transfer. This error is also displayed if the party paying the Amazon FPS fees does not have a sufficient account balance. Retriable: Yes
InternalError	A retriable error that happens due to some transient problem in the system. The caller should retry the API call if this error is encountered. Retriable: Yes
InvalidAccountState	The account is either suspended or closed. Payment instructions cannot be installed on this account. You must ask your customer to set up a new account if the account is closed. Retriable: Yes

Error	Description
InvalidAccountState_Caller	The developer account cannot participate in the transaction. Your account is not active. Contact your AWS Representative for more information. Retriable: Yes
InvalidAccountState_Recipient	Recipient account cannot participate in the transaction. You can display the following message to your customer (sender): *Your Amazon Payments account is not active. Please visit http:// payments.amazon.com for more details.* Retriable: Yes
InvalidAccountState_Sender	Sender account cannot participate in the transaction. You can display the following message to your customer (sender): *Your Amazon Payments account is not active. Please visit http://payments.amazon.com for more details.* Retriable: Yes
InvalidCallerReference	The Caller Reference does not have a token associated with it. Use the caller reference value that was passed to the InstallPaymentInstruction operation or the Amazon FPS Co-Branded UI pipeline.
InvalidClientTokenId	The AWS Access Key Id you provided does not exist in our records. Please check that the AWS Access Key Id used to make the request is valid. Retriable: No
InvalidDateRange	The end date specified is before the start date or the start date is in the future. Specify the correct end date.
InvalidParams	One or more parameters in the request is invalid. For more information, see the parameter descriptions for the action in the API Reference. Parameters are case sensitive. Retriable: No
InvalidPaymentInstrument	The payment method used in the transaction is invalid. Specify a valid payment method

Error	Description
InvalidPaymentMethod	The cause for this error is dependent on the calling action: • For InstallPaymentInstruction, payment method specified in the GK construct is invalid. Specify the correct payment method. • For FundPrepaid and SettleDebt, the payment method specified in the token is invalid Use a token with payment method specified as any of ABT, ACH, and CC. For Quick accounts, only CC is acceptable.
InvalidRecipientForCCTransaction	This account cannot receive credit card payments. You can display the following message to your customers: You cannot receive credit card payment. Please visit http://payments.amazon.com to update your account to receive credit card payments."
InvalidSenderRoleFor AccountType	This token cannot be used for this operation. Ensure that the account used in this transaction is the same account used in the original transaction. In a refund transaction, the recipient making the refund payment must the be same recipient as in the original transaction. Retriable: No
InvalidTokenId	You did not install the token that you are trying to cancel. You do not have permission to cancel this token. You can cancel only the tokens that you own. Retriable: No
InvalidTokenId_Recipient	The recipient token specified is either invalid or canceled. You must install a new token if you are the recipient. If you are not the recipient, get a new payment authorization from the recipient. Retriable: No
InvalidTokenId_Sender	The send token specified is either invalid or canceled or the token is not active. You must ask your customer to set up a new payment authorization. Retriable: No
InvalidTokenType	An invalid operation was performed on the token, for example, getting the token usage information on a single use token. Retriable: No

Error	Description
InvalidTransactionId	The specified transaction could not be found or the caller did not execute the transaction or this is not a Pay or Reserve call. Specify the correct the transaction ID. Retriable: No
InvalidTransactionState	The transaction is not complete, or it has temporarily failed. Specify a duration of more than one hour. Retriable: No
NotMarketplaceApp	This is not an marketplace application or the caller does not match either the sender or the recipient. Please check that you are specifying the correct tokens. Retriable: Yes
OriginalTransactionFailed	The original transaction has failed. You cannot refund a transaction that has originally failed. Retriable: No
OriginalTransactionIncomplete	The original transaction is still in progress. Retry after the original transaction has completed. Retriable: Yes
PaymentInstrumentNotCC	The payment method specified in the transaction is not a credit card. You can only use a credit card for this transaction. Use only a credit card for this transaction.
PaymentMethodNotDefined	Payment method is not defined in the transaction. Specify the payment method in the sender token.
PrepaidFundingLimitExceeded	An attempt has been made to fund the prepaid instrument at a level greater than its recharge limit. Retriable: Yes, after adjusting the amount of funds to a valid level.
RefundAmountExceeded	The refund amount is more than the refundable amount. You are not allowed to refund more than the original transaction amount. Retriable: No
SameSenderAndRecipient	The sender and receiver are identical, which is not allowed. Retriable: No
SameTokenIdUsedMultipleTimes	This token is already used in earlier transactions. The tokens used in a transaction should be unique.

Error	Description
SenderNotOriginalRecipient	The sender in the refund transaction is not the recipient of the original transaction. The token you passed as the refund sender token does not belong to the recipient of the original transaction. Pass the correct refund sender token. Retriable: No
SettleAmountGreaterThanDebt	The amount being settled or written off is greater than the current debt. You cannot settle an amount greater than what is owed. Retriable: No
SettleAmountGreaterThan ReserveAmount	The amount being settled is greater than the reserved amount. You cannot settle an amount greater than what is reserved. Retriable: No
SignatureDoesNotMatch	The request signature calculated by Amazon does not match the signature you provided. Check your AWS Secret Access Key and signing method. For more information, see "Working with Signatures" in the *Amazon Flexible Payments Service Getting Started Guide*. Retriable: No
TokenAccessDenied	Permission is denied to cancel the token. You are not allowed to cancel this token. Retriable: No
TokenNotActive	The token is canceled. A new token needs to be created. Retriable: No
TokenNotActive_Recipient	The recipient token is canceled. If you are the recipient, set up a new recipient token using the `InstallPaymentInstruction` operation or direct your customers to the Recipient Token Installation Pipeline to set up recipient token. Retriable: No
TokenNotActive_Sender	The sender token is canceled. You must ask your customer to set up a new payment authorization because the current authorization is not active. Retriable: No

Error	Description
TokenUsageError	The token usage limit is exceeded. If the usage has exceeded for this period, then wait for the next period before making another transaction. If the usage has exceeded for the entire authorization period, then ask your customer to set up a new payment authorization.
TransactionDenied	This transaction is not allowed. You are not allowed to do this transaction. Check your credentials. Retriable: No
TransactionFullyRefundedAlready	This transaction has already been completely refunded. You are not allowed to refund more than the original transaction amount. Retriable: No
TransactionTypeNotRefundable	You cannot refund this transaction. Refund is allowed only on the `Pay` operation. Retriable: No
UnverifiedAccount_Recipient	The recipient's account must have a verified bank account or a credit card before this transaction can be initiated. You can display the following message to your customer (recipient): *Your Amazon Payments account is not active. Please visit http://payments.amazon.com for more details.* Retriable: No
UnverifiedAccount_Sender	The sender's account must have a verified U.S. credit card or a verified U.S bank account before this transaction can be initiated. You can display the following message to your customers: *Please add a U.S. credit card or U.S. bank account and verify your bank account before making this payment.* Retriable: No
UnverifiedBankAccount	A verified bank account should be used for this transaction. Visit the http://payments.amazon.com web site to verify your bank account. Retriable: No
UnverifiedEmailAddress_Caller	The caller account must have a verified email address. You cannot make a web service API call without verifying your email address. Go to http://payments.amazon.com web site and make payments. Retriable: No

Error	Description
UnverifiedEmailAddress_Recipient	The recipient account must have a verified email address for receiving payments. You can display the following message to your customers: *You cannot receive payments. Please verify your email address. Go to http://payments.amazon.com to verify your account and receive payments.* Retriable: No
UnverifiedEmailAddress_Sender	The sender account must have a verified email address for this payment You can display the following message to your customers: *You cannot receive payments. Please verify your email address. Go to http://payments.amazon.com to verify your account and receive payments.* Retriable: No

Amazon FPS API Actions

Topics

Amazon FPS has five parts, each providing a different slice of Amazon FPS functionality:

- **Amazon FPS Basic Quick Start.** Facilitates a one-time payment between a buyer and a developer (you) who is also the merchant for e-commerce, digital content, donations, or services.
- **Amazon FPS Marketplace Quick Start.** Facilitates a one-time payment between a buyer and a merchant, where you are a third-party developer (also known as a *caller*) who hosts the merchant's product pages and order pipeline. With this unique three-party transaction model, you can charge a fee to process transactions in which you are neither the buyer nor the merchant.
- **Amazon FPS Advanced Quick Start.** Facilitates multiple or recurring payments between a buyer and a seller for e-commerce, digital content, donations, or services.
- **Amazon FPS Aggregated Payments Quick Start.** Facilitates aggregated micro-transactions into a single, larger transaction using prepaid and postpaid capabilities.
- **Amazon FPS Account Management Quick Start.** Access buyer and developer account activity programmatically. Alternatively, you can view account activity and balances on the Amazon Payments web site.

This reference provides one comprehensive alphabetical listing, and five quick start listings. DataTypes, IPN values, and example email messages are provided in a comprehesive fashion only, without division by quick start.

You can use these parts separately or in combination, as they share a common WSDL and schema. This guide, the API Reference, contains reference information from each of the quick starts, as well an information common to them all, including

- API reference pages, containing parameter definitions, error codes specific to the api, example requests, example responses, and example IPN responses.
- A listing of the request and reponse parameters common to most Amazon FPS APIs

- Comprehensive listing of data types used by Amazon FPS APIs
- Comprehensive listing of instant payment notification status codes and return values used in response to an Amazon FPS API call
- Comprehensive listing of error codes used by Amazon FPS APIs
- A full listing of email messages commonly sent by Amazon Payments in addition to return codes and IPN responses.

Basic Quick Start Actions

The following APIs are specific to the Amazon FPS Basic Quick Start Developer Guide:

- Cancel (p. 18)
- GetTokensByCaller (p. 55)
- GetTransactionStatus (p. 68)
- Pay (p. 72)
- Refund (p. 80)
- Reserve (p. 86)
- Settle (p. 94)
- VerifySignature (p. 103)

Advanced Quick Start Actions

The following APIs are specific to the Amazon FPS Advanced Quick Start Developer Guide:

- Cancel (p. 18)
- CancelToken (p. 21)
- GetRecipientVerificationStatus (p. 49)
- GetTokensByCaller (p. 55)
- GetTransactionStatus (p. 68)
- Pay (p. 72)
- Refund (p. 80)
- Reserve (p. 86)
- Settle (p. 94)
- VerifySignature (p. 103)

Marketplace Quick Start Actions

The following APIs are specific to the Amazon Flexible Payments Service Marketplace Quick Start Developer Guide:

- Cancel (p. 18)
- GetRecipientVerificationStatus (p. 49)
- GetTokensByCaller (p. 55)
- GetTransactionStatus (p. 68)
- Pay (p. 72)
- Refund (p. 80)

- Reserve (p. 86)
- Settle (p. 94)
- VerifySignature (p. 103)

Aggregated Payments Quick Start Actions

The following APIs are specific to the Amazon Flexible Payments Service Aggregated Payments Quick Start Developer Guide:

- CancelToken (p. 21)
- FundPrepaid (p. 24)
- GetDebtBalance (p. 41)
- GetOutstandingDebtBalance (p. 44)
- GetPrepaidBalance (p. 47)
- GetTokensByCaller (p. 55)
- GetTotalPrepaidLiability (p. 61)
- GetTransactionStatus (p. 68)
- Pay (p. 72)
- Refund (p. 80)
- SettleDebt (p. 98)
- VerifySignature (p. 103)
- WriteOffDebt (p. 106)

Account Management Quick Start Actions

The following APIs are specific to the Amazon Flexible Payments Service Account Management Quick Start Developer Guide:

- GetAccountActivity (p. 30)
- GetAccountBalance (p. 38)
- GetTokens (p. 52)
- GetTokenUsage (p. 58)
- GetTransaction (p. 64)
- GetTransactionStatus (p. 68)
- VerifySignature (p. 103)

All Actions

Amazon FPS provides the following APIs. Each is shown with the associated quick start guides that use it.

- Cancel (p. 18) (Basic, Advanced and Marketplace)
- CancelToken (p. 21) (Advanced and Aggregated)
- FundPrepaid (p. 24) (Aggregated)
- GetAccountActivity (p. 30) (Account Management)
- GetAccountBalance (p. 38) (Account Management)

- GetDebtBalance (p. 41) (Aggregated)
- GetOutstandingDebtBalance (p. 44) (Aggregated)
- GetPrepaidBalance (p. 47) (Aggregated)
- GetRecipientVerificationStatus (p. 49) (Advanced and Marketplace)
- GetTokens (p. 52) (Account Management)
- GetTokensByCaller (p. 55) (Basic, Advanced, Aggregated and Marketplace)
- GetTokenUsage (p. 58) (Account Management)
- GetTotalPrepaidLiability (p. 61) (Aggregated)
- GetTransaction (p. 64) (Account Management)
- GetTransactionStatus (p. 68) (Basic, Advanced, Aggregated and Marketplace)
- Pay (p. 72) (Basic, Advanced, Aggregated and Marketplace)
- Refund (p. 80) (Basic, Advanced, Aggregated and Marketplace)
- Reserve (p. 86) (Basic, Advanced and Marketplace)
- Settle (p. 94) (Basic, Advanced and Marketplace)
- SettleDebt (p. 98) (Aggregated)
- VerifySignature (p. 103) (Basic, Advanced, Aggregated and Marketplace)
- WriteOffDebt (p. 106) (Aggregated)

Cancel

Description

The `Cancel` action cancels a *reserved* or *pending* transaction. Once the transaction is canceled, you can't then settle it. You also can't use `Cancel` on a *completed* transaction. After a transaction is completed, you can do a refund if you want to reverse the order.

If the sender's credit card was in a reserved state, it is not part of this action to make sure the reserved status is removed. card.

This action appears in the Basic, Advanced, and Marketplace Quick Starts, as well as in the Amazon Simple Pay Advanced Guide.

Request Parameters

Parameter	Description	Required
`Description`	Describes the reason for cancellation. Type: String Default: None	No
`OverrideIPNURL`	Specifies the URL to receive the Instant Payment Notification (IPN) for the request. The IPN URL set with this parameter takes precedence over the IPN URL set in your account. Type: String Default: None	No

Parameter	Description	Required
TransactionId	Specifies the transaction that needs to be canceled. This ID should have been returned by Amazon in a prior Pay or Reserve call. Type: String Default: None Constraint: Max size = 35 characters	Yes

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
TransactionId	The ID of the completed transaction. It is the same as the TransactionID provided in the request. Type: String Size: 35 Bytes
TransactionStatus	The status of the cancellation request. Type: TransactionStatus (p. 121) Size: 20 Bytes

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountClosed (p. 6)
- AuthFailure (p. 7)
- ConcurrentModification (p. 7)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTransactionState (p. 11)
- SignatureDoesNotMatch (p. 12)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=Cancel
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
```

```
&Description=MyWish
&Signature=yOedrTuiMoMrKt8SwugDDnfd0nydyoX9uPq1H1SUCl4%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-06T09%3A14%3A58.796Z
&TransactionId=14GKI1SKSR1V6DO1RCCB32RBR6KLODMGQUD
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body
    wsu:id="body" xmlns:wsu=
    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
    <ns2:Cancel xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
        <ns2:TokenId>
            B1HA64FUMCEB43QAJBTP3TVMJZGFLAX2DJJ3ZPAFHH1VNPGR74I83ZZI4HJ5NGEK
        </ns2:TokenId>
        <ns2:ReasonText>Buyer closed the transaction.</ns2:ReasonText>
    </ns2:Cancel>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<CancelResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <CancelResult>
    <TransactionId>14GKI1SKSR1V6DO1RCCB32RBR6KLODMGQUD</TransactionId>
    <TransactionStatus>Cancelled</TransactionStatus>
  </CancelResult>
  <ResponseMetadata>
    <RequestId>6fe4b755-a328-419d-8967-e1d3b43779fc:0</RequestId>
  </ResponseMetadata>
</CancelResponse>
```

Sample Response to SOAP Request

```
<CancelResponse
  xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <CancelResult>
    <TransactionId>
      13N91G4R7478CJLIGROQH2VQJSM
    </TransactionId>
    <TransactionStatus>
      Success
    </TransactionStatus>
  </CancelResult>
</CancelResponse>
```

Sample IPN Success Notification to Rest Request

```
--------------------------
transactionId: 14GKI1SKSR1V6DO1RCCB32RBR6KLODMGQUD
statusMessage: The transaction was explicitly cancelled by the caller.
transactionDate: 1254820475
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference08
transactionAmount: USD 1.00
transactionStatus: CANCELLED
operation: RESERVE
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: jWDbBxtEhw2rQEyMeEXcpWCgoZvm8rjLEnmg38oYoPPR7NbMGgmMA9/5CDjt9Q/FMktKM
bARXnZF
YTzHj3YOKiAM3vxI0zT1oTiSdBx1KBRFzK7mauxxlQv5BYxjFX+R5cl+keCaT2nQyrp3agdrIIp5
MZ5Oy9dBuYMwMFWXoZZor90EidD23hBdZSOOzQRUdzKaKJsF14RQVrKcf5pDCs1HaB6LBKbATaNT
RSxxrviIXy9JcWRQhJwzcc1H6cFOJDpNFSJ03b0Z94eL/XNu9BU7bT4KRWb+OHF0Pn53yf4zyBT9
jTD+94WeujCxwE2rF0j5+brmXp/+Sn/RccDG7w==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: Reserve
statusCode: Cancelled
--------------------------
```

CancelToken

Description

The `CancelToken` action cancels multi-use, recurring, prepaid, and postpaid payment tokens. You can use this action at any time during the life of the token. After this request completes successfully, Amazon FPS stops all further payments that use the specified token. There is no way to reactivate a canceled token.

You can only cancel tokens that you have created.

This action appears in the Advanced and Aggregated Quick Starts.

Request Parameters

Parameter	Description	Required
`OverrideIPNURL`	Specifies the URL to receive the Instant Payment Notification (IPN) for the request. The IPN URL set with this parameter takes precedence over the IPN URL set in your account. Type: String Default: None	No

Parameter	Description	Required
ReasonText	Reason for canceling the payment token. Type: String Default: None	No
TokenId	Specifies the token to cancel. You should have stored this value when it was returned as part of the response to the Co-Branded service request. Type: String Default: None	Yes

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

The response for this API includes only elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountClosed (p. 6)
- AuthFailure (p. 7)
- ConcurrentModification (p. 7) (Advanced)
- DuplicateRequest (p. 7)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTokenId (p. 10)
- SignatureDoesNotMatch (p. 12)
- TokenAccessDenied (p. 12)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=CancelToken
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&ReasonText=MyWish
&Signature=IZD9O%2FWGqhkzO%2FdLTQ7Tn8KUAmtZXqIEg6gypwkGeWQ%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T08%3A46%3A37.156Z
&TokenId=D739IT9TMC4FK9KB56PDKJWAQGXDZ3B8X3SJNGVH3UEF5GQ7XAQZMEIL4OGEZKGX
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body
wsu:id="body" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:CancelToken xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
     <ns2:TokenId>
      B1HA64FUMCEB43QAJBTP3TVMJZGFLAX2DJJ3ZPAFHH1VNPGR74I83ZZI4HJ5NGEK
     </ns2:TokenId>
     <ns2:ReasonText>Buyer left the system.</ns2:ReasonText>
   </ns2:CancelToken>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<CancelTokenResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <ResponseMetadata>
    <RequestId>a10e0ad6-148f-4afe-8bcd-e80a2680793d:0</RequestId>
  </ResponseMetadata>
</CancelTokenResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
    MESSAGE123
   </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
    CancelToken:Response
   </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
    urn:uuid:5fb6a949-a481-4f6f-bdfb-bcdde0daea6b
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <CancelTokenResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
     <ResponseMetadata>
       <RequestId>5fb6a949-a481-4f6f-bdfb-bcdde0daea6b:0</RequestId>
     </ResponseMetadata>
   </CancelTokenResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample IPN Notification to Rest Request

```
-------------------------
signatureVersion: 2
signatureMethod: RSA-SHA1
customerEmail: test-caller@amazon.com
tokenId: D739ATGTM94QK9NBU6P4KDWACGXDZ8BVX3TJHGVP3XEFMGE7XVQTMEIL4OGFZMGP
callerReference: CallerReference19
notificationType: TokenCancellation
signature: flxZtuxk3jb0Ww4g4duMjx1s8EQnIC7kPHqKKu0t4trp1/8ZU6ohtm9V1xB1mdxDnJ37lpy
fL7rp
wE5tiKjJ8agm1OzPjp9rwEVOEMcdscopTVhh9AG2HTNGyWyyaRlIPlXiV3mpPyMrttLiOkrYB8ak
YZ9fMbXUB9gKzMVzNhh58auyD/weMV/WIX3DDSJslsp0kg6frHv5F5CYrprwv4S+cXQxXdgJlRC3
UJO8bH68bwlFnyyzPz4+TnbB5xMDatpwkBOFCWO5+tmwlwJHyAUa7z6XJgwj27YIIjFSJolWLKwK
iZHqPNYNjKHE190sQMQBLHcnkZeexig6wYHK5w==
tokenType: SingleUse
dateInstalled: Oct 8, 2009
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
customerName: Test Business
-------------------------
```

FundPrepaid

Description

The `FundPrepaid` action transfers money from the sender's payment instrument specified by the *`SenderTokenId`* to the prepaid balance on the instrument specified by *`PrepaidInstrumentId`*. The prepaid balance can be funded using a credit card or Amazon Payments account balance.

> **Note**
> The value you provide for the *`SenderTokenId`* is the *`fundingTokenId`* value you received in the Prepaid Token API response, not the value of the *`prepaidSenderTokenID`* parameter.

This action appears in the Advanced and Aggregated Quick Starts.

Request Parameters

Parameter	Description	Required
`CallerDescription`	Description of this transaction for the caller. Type: String Default: None Constraint: Max size = 160 characters	No
`CallerReference`	A value you provide that uniquely identifies the request. . Type: String Default: None Constraint: Max size= 128 characters	Yes

Parameter	Description	Required
DescriptorPolicy	Specifies the entity whose name and contact details would be displayed in the sender's credit card or bank account statement. Type: DescriptorPolicy (p. 114) Default: None	No
FundingAmount	Amount to fund the prepaid instrument. To learn how to specify the amount correctly in a REST request, see the example request at the end of this topic. Type: Amount (p. 113) Default: None	Yes
OverrideIPNURL	Specifies the URL to receive the Instant Payment Notification (IPN) for the request. The IPN URL set with this parameter takes precedence over the IPN URL set in your account. Type: String Default: None	No
PrepaidInstrumentId	Specifies the prepaid instrument to fund. Use the value of the *prepaidInstrumentID* parameter from the Prepaid Token API response. Type: String Default: None Max Size: 128 Bytes	Yes
SenderDescription	Description of this transaction for the sender. Type: String Default: None Constraint: Max size = 128 bytes	No
SenderTokenId	The value of the *fundingTokenID* parameter you received in the Prepaid Token API response. Type: String Default: None Constraint: Max size = 65 characters	Yes
TransactionTimeoutInMins	Specifies the number of minutes before the request times out. Use this parameter to specify a timeout value that is acceptable for your business. If Amazon FPS cannot complete the transaction in the time allotted, the transaction is marked as failed and you receive an IPN notification (if you are using IPN). Type: Integer (number of minutes) Default: 10080	No

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
`TransactionId`	Unique ID generated by Amazon FPS for this transaction. This element is returned if the transaction was accepted by Amazon FPS. Type: String Constraint: Max size = 35 characters
`TransactionStatus`	Provides the status of the transaction. Type: TransactionStatus (p. 121)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountLimitsExceeded (p. 7)
- AmountOutOfRange (p. 7)
- AuthFailure (p. 7)
- DuplicateRequest (p. 7)
- InactiveInstrument (p. 7)
- IncompatibleTokens (p. 8)
- InstrumentAccessDenied (p. 8)
- InstrumentExpired (p. 8)
- InsufficientBalance (p. 8)
- InternalError (p. 8)
- InvalidAccountState_Caller (p. 9)
- InvalidAccountState_Sender (p. 9)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidPaymentInstrument (p. 9)
- InvalidPaymentMethod (p. 10)
- InvalidTokenId_Sender (p. 10)
- SignatureDoesNotMatch (p. 12)
- TokenNotActive_Sender (p. 12)
- TransactionDenied (p. 13)
- UnverifiedAccount_Recipient (p. 13)
- UnverifiedAccount_Sender (p. 13)
- UnverifiedBankAccount (p. 13)
- UnverifiedEmailAddress_Caller (p. 13)
- UnverifiedEmailAddress_Recipient (p. 14)
- UnverifiedEmailAddress_Sender (p. 14)

Examples

The following sections show a sample request and response.

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=FundPrepaid
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=MyWish
&CallerReference=CallerReference10
&FundingAmount.CurrencyCode=USD
&FundingAmount.Value=1
```

```
&PrepaidInstrumentId=573IEMDCGRZHJ4P1D7BC3GMNBFPQU1VLNT7R4CT7A7DJIX96LKZNKSXUG
PIGTTI8
&SenderTokenId=563INMLCG3ZJJ4L1I7BB31MN2FBQUCVXNTDRTCT5A2DJDXG6LNZ7KSNUJPI7TVIF
&Signature=2a%2BpPIcPmNSQE7kUtd1S0WsTjx5vPyzPlw%2B95teoQyQ%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T08%3A37%3A54.765Z
&Version=2008-09-17
```

Sample Query Request

```
GET\n
fps.payments.amazon.com\n
Action=FundPrepaid
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=MyWish
&CallerReference=CallerReference10
&FundingAmount.CurrencyCode=USD
&FundingAmount.Value=1
&PrepaidInstrumentId=573IEMDCGRZHJ4P1D7BC3GMNBFPQU1VLNT7R4CT7A7DJIX96LKZNKSXUG
PIGTTI8
&SenderTokenId=563INMLCG3ZJJ4L1I7BB31MN2FBQUCVXNTDRTCT5A2DJDXG6LNZ7KSNUJPI7TVIF
&Signature=2a%2BpPIcPmNSQE7kUtd1S0WsTjx5vPyzPlw%2B95teoQyQ%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T08%3A37%3A54.765Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body wsu:Id="body" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:FundPrepaid xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
     <ns2:SenderTokenId>
      R74QEB8KRTSE4PN2S8KI7PLD2LXZGGI6UMF8HXUHJ9PVKLI2EISA5MHPIZJ4V4NF
     </ns2:SenderTokenId>
     <ns2:PrepaidInstrumentId>
      RA4QLBMKRVSK4PZ2J8KN72LDDL4ZGTI4UM581XUPJJPV6LZ2EGS55MBP5ZJ5V7NA
     </ns2:PrepaidInstrumentId>
     <ns2:FundingAmount>
       <ns2:CurrencyCode>USD</ns2:CurrencyCode>
       <ns2:Value>50</ns2:Value>
     </ns2:FundingAmount>
     <ns2:CallerReference>
      ReferenceString????f2VW12290544025471
     </ns2:CallerReference>
     <ns2:CallerDescription>
      DescriptionString-????j38y12290544025481
     </ns2:CallerDescription>
   </ns2:FundPrepaid>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<FundPrepaidResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <FundPrepaidResult>
    <TransactionId>14GN2BUHUAV4KG5S8USHN79PQH1NGN5ADK4</TransactionId>
    <TransactionStatus>Pending</TransactionStatus>
  </FundPrepaidResult>
  <ResponseMetadata>
    <RequestId>325c3342-a972-4d73-9e28-c56442ad56de:0</RequestId>
  </ResponseMetadata>
</FundPrepaidResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
  MESSAGE123</wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
  http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
  FundPrepaid:Response
   </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
  urn:uuid:dd5439cf-ee48-4531-9736-0176e847b823
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <FundPrepaidResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
     <FundPrepaidResult>
       <TransactionId>13OKLJO453E6VCQ6NDHD8CG9RUG2DZLZC3E</TransactionId>
       <TransactionStatus>Pending</TransactionStatus>
     </FundPrepaidResult>
     <ResponseMetadata>
       <RequestId>dd5439cf-ee48-4531-9736-0176e847b823:0</RequestId>
     </ResponseMetadata>
   </FundPrepaidResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample IPN Pending Notification to Rest Request

```
--------------------------
transactionId: 14GN2BUHUAV4KG5S8USHN79PQH1NGN5ADK4
statusMessage: The transaction is awaiting a response from the backend payment
processor.
transactionDate: 1254904691
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference10
```

```
transactionAmount: USD 1.00
transactionStatus: PENDING
operation: FUND_PREPAID
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: iySHTqfzaF7bwypiE6BwBYiRCsB1SOy/RhDuiwvPZNzNPhDQW9TqGg
FI2U1kjiXuTHKR42xBmocm
3GNdh3hI8foYqzGHFVn2NQKr8ewUxr16KlrDC7cB+3beFD8TemsH8eGQeTwikgheMxiZEkoLHFpo
ZfVSBawjVtkG72s+/4lXVVY2bf1D8rhb67cXSDMbyb4TxYoJRZCpLCyH956GB/pexoNKKFCcBLNm
9Vd50ObSnhHTC/0oyk+Lf8G6OdVLHLFEATQQ57kKBND7mlH0wRG2xjgtObDM8xncVFGWkNQbaHH3
GqG5wsB5t1mcfpMPRXjQ4GAEFnUi9VQgAltyWQ==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: PendingNetworkResponse
--------------------------
```

Sample IPN Success Notification to Rest Request

```
--------------------------
transactionId: 14GN2BUHUAV4KG5S8USHN79PQH1NGN5ADK4
statusMessage: The transaction was successful and the payment instrument was
charged.
transactionDate: 1254904691
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference10
transactionAmount: USD 1.00
transactionStatus: SUCCESS
operation: FUND_PREPAID
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: we8dbtnHqzj9MViz2yYzreeDy00MVrbafEnspAhjZwnLGWKowAc9pMkHEnPxiwpHjiVf
HQqEmMJu
zDuuWZI8bp/vaVmF/nCVUxhpV/EEdhGjawtyNs1XPUT6lCdzEitT+oykk3pj/p1se6s0FFyWFcX7
a6ttWN06i0XgsQ9yl58opy683j4w0b2bOs9GySyH5uCIA7TpD7GsdnkkVpHlMSFnLy9+DFEKaNAo
Tx3WqyqOINvyOviRnB16IbvFNFsoULXTjodHjCDhY7kkX1IsfRkSDRxPX5Fu2eB78uGSo5jZXYJA
WAo4JGQjNVZAcSr18SvCRq12K7PpT4qrskEHoA==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: Success
--------------------------
```

Related Actions

- GetPrepaidBalance (p. 47)

GetAccountActivity

Description

The `GetAccountActivity` action returns transactions from an account for a given time period. You can further customize the results using the other request parameters.

This action appears in the Account Management Quick Start.

Request Parameters

Parameter	Description	Required
EndDate	Specifies the final date for the list of transactions to return. If no end date is not specified, Amazon FPS returns transactions to the current date. Type: dateTime Default: Current date	No
FPSOperation	Filters the results by Amazon FPS action. For example, the value `Pay` returns only transactions involving the `Pay` action. Type: FPSOperation (p. 114) Default: None	No
MaxBatchSize	Specifies the maximum number of transactions returned in the response. Type: Integer Default: 200 Constraints: Between 20 and 200	No
PaymentMethod	Specifies the payment method, such as CC or ABT. Type: PaymentMethod (p. 117) Default: None	No
ResponseGroup	Subheading that allows you to group the responses. Type: String Default: None	No
Role	List of roles arranged in sort order based on the role: sender, recipient, or caller. Type: TransactionalRole (p. 121) Default: None	No
SortOrderByDate	Specifies how to sort the results and in what order. The date used is the date the request was received by Amazon FPS. Type: SortOrderByDate (p. 118) Default: Descending Valid Values: Descending | Ascending	No

Parameter	Description	Required
StartDate	Specifies the first date of transactions to return. Type: dateTime Constraints: Present, past dates	Yes
Status	Filters the results based on the transaction status. Type: TransactionStatus (p. 121) Default: None	No

You must also use parameters that are common to all requests that are described in Common Request Parameters (p. 5). The common parameters are defaulted in SOAP calls but must be explicitly added in REST calls. Parameter names are case sensitive.

Response Elements

Element	Description
BatchSize	Specifies the total number of results returned. This element is always returned. Type: Integer
StartTimeForNextTransaction	Provides the start time for the next transaction. Amazon FPS returns a maximum of 200 results for one request. You can use the value returned by this parameter as the start time/date for the next request. For example, if you need the account activity for the period Jan-1-2010 to Dec-31-2010 and there are more than 200 transactions during that period, Amazon FPS returns a maximum of 200 transactions and sends the date, Apr-20-2010, which is the date of the next transaction to be returned. You can get the next 200 transactions using Apr-20-2010 as the new start date. This element is returned only if there are more than 200 transactions. Type: dateTime
Transaction	Specifies the list of transactions. This element is returned only if there are transactions. Type: Transaction (p. 120)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountClosed (p. 6)
- AuthFailure (p. 7)

- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidDateRange (p. 9)
- InvalidParams (p. 9)
- SignatureDoesNotMatch (p. 12)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetAccountActivity
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Signature=Bs3etBhuZ2Huf8gxLO0EaG4evxq%2BjzT2Bjg6YMAF3RE%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2&Version=2008-09-17
&StartDate=2009-10-07Z
&Timestamp=2009-10-07T11%3A14%3A56.406Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body
    wsu:Id="body"
    xmlns:wsu=
    "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
    <ns2:GetAccountActivity
      xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
      <ns2:StartDate>2005-01-10T07:39:32.000-08:00</StartDate>
      <ns2:EndDate>2009-01-10T07:39:37.000-08:00</EndDate>
    </ns2:GetAccountActivity>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetAccountActivityResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <GetAccountActivityResult>
    <BatchSize>5</BatchSize>
    <Transaction>
      <TransactionId>14GN2BUHUAV4KG5S8USHN79PQH1NGN5ADK4</TransactionId>
      <CallerTransactionDate>
        2009-10-07T01:37:54.765-07:00
      </CallerTransactionDate>
      <DateReceived>2009-10-07T01:38:11.262-07:00</DateReceived>
      <DateCompleted>2009-10-07T01:38:12.857-07:00</DateCompleted>
      <TransactionAmount>
        <CurrencyCode>USD</CurrencyCode>
        <Value>1.000000</Value>
```

```
      </TransactionAmount>
      <FPSOperation>FundPrepaid</FPSOperation>
      <TransactionStatus>Success</TransactionStatus>
      <StatusMessage>
        The transaction was successful and the payment instrument was charged.
      </StatusMessage>
      <StatusCode>Success</StatusCode>
      <TransactionPart>
        <Role>Recipient</Role>
        <Name>Test Business</Name>
        <FeesPaid>
          <CurrencyCode>USD</CurrencyCode>
          <Value>0.100000</Value>
        </FeesPaid>
      </TransactionPart>
      <TransactionPart>
        <Role>Caller</Role>
        <Name>Test Business</Name>
        <Reference>CallerReference10</Reference>
        <Description>MyWish</Description>
        <FeesPaid>
          <CurrencyCode>USD</CurrencyCode>
          <Value>0.000000</Value>
        </FeesPaid>
      </TransactionPart>
      <PaymentMethod>CC</PaymentMethod>
      <SenderName>Test Business</SenderName>
      <CallerName>Test Business</CallerName>
      <RecipientName>Test Business</RecipientName>
      <FPSFees>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.100000</Value>
      </FPSFees>
      <Balance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>7.400000</Value>
      </Balance>
      <SenderTokenId>
563INMLCG3ZJJ4L1I7BB31MN2FBQUCVXNTDRTCT5A2DJDXG6LNZ7KSNUJPI7TVIF
      </SenderTokenId>
    </Transaction>
    <Transaction>
      <TransactionId>14GN1O5992IEOB3ELM1SCUFTSOQ3C6S7NR2</TransactionId>
     <CallerTransactionDate>2009-10-07T01:27:21.469-07:00</CallerTransactionDate>

      <DateReceived>2009-10-07T01:27:22.793-07:00</DateReceived>
      <DateCompleted>2009-10-07T01:27:23.335-07:00</DateCompleted>
      <TransactionAmount>
        <CurrencyCode>USD</CurrencyCode>
        <Value>4.000000</Value>
      </TransactionAmount>
      <FPSOperation>Pay</FPSOperation>
      <TransactionStatus>Success</TransactionStatus>
      <StatusMessage>
      The transaction was successful and the payment instrument was charged.
      </StatusMessage>
      <StatusCode>Success</StatusCode>
      <TransactionPart>
```

```
      <Role>Recipient</Role>
      <Name>Test Business</Name>
      <Reference>Prepaid Digital Download - 1254904041469</Reference>
      <Description>Prepaid Digital Download</Description>
      <FeesPaid>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
      </FeesPaid>
    </TransactionPart>
    <TransactionPart>
      <Role>Caller</Role>
      <Name>Test Business</Name>
      <Reference>Prepaid Digital Download - 1254904034205</Reference>
      <Description>
        Prepaid Digital Download - payment for mp3 from digital.
      </Description>
      <FeesPaid>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
      </FeesPaid>
    </TransactionPart>
    <PaymentMethod>Prepaid</PaymentMethod>
    <SenderName>Test Business</SenderName>
    <CallerName>Test Business</CallerName>
    <RecipientName>Test Business</RecipientName>
    <FPSFees>
      <CurrencyCode>USD</CurrencyCode>
      <Value>0</Value>
    </FPSFees>
    <Balance>
      <CurrencyCode>USD</CurrencyCode>
      <Value>6.500000</Value>
    </Balance>
    <SenderTokenId>
      513I1MGCG6ZZJ49157BZ3EMNJFAQU6V9NTSRUCTEANDJ3X46LGZNKSJUVPIXTPID
    </SenderTokenId>
    <RecipientTokenId>
      D639FT4TMP4QK9UBH6PAK2WAXGHDZSBUX3UJSGVX3LEFVGU7XDQXMENL4OGVZEGB
    </RecipientTokenId>
  </Transaction>
  <Transaction>
    <TransactionId>14GN1NHHN489BFGH6D8BMGT8NLSR2DJ4PNK</TransactionId>
    <CallerTransactionDate>
      2009-10-07T01:26:58.190-07:00
    </CallerTransactionDate>
    <DateReceived>2009-10-07T01:27:02.583-07:00</DateReceived>
    <DateCompleted>2009-10-07T01:27:04.435-07:00</DateCompleted>
    <TransactionAmount>
      <CurrencyCode>USD</CurrencyCode>
      <Value>5.000000</Value>
    </TransactionAmount>
    <FPSOperation>FundPrepaid</FPSOperation>
    <TransactionStatus>Success</TransactionStatus>
    <StatusMessage>
      The transaction was successful and the payment instrument was charged.
    </StatusMessage>
    <StatusCode>Success</StatusCode>
    <TransactionPart>
```

```
      <Role>Caller</Role>
      <Name>Test Business</Name>
      <Reference>Prepaid Digital Download - 1254903995419</Reference>
      <FeesPaid>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
      </FeesPaid>
    </TransactionPart>
    <TransactionPart>
      <Role>Recipient</Role>
      <Name>Test Business</Name>
      <FeesPaid>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.300000</Value>
      </FeesPaid>
    </TransactionPart>
    <PaymentMethod>CC</PaymentMethod>
    <SenderName>Test Business</SenderName>
    <CallerName>Test Business</CallerName>
    <RecipientName>Test Business</RecipientName>
    <FPSFees>
      <CurrencyCode>USD</CurrencyCode>
      <Value>0.300000</Value>
    </FPSFees>
    <Balance>
      <CurrencyCode>USD</CurrencyCode>
      <Value>6.500000</Value>
    </Balance>
    <SenderTokenId>
      513ISM2CGDZPJ4S1D7BH3HMNIFCQUAVNNTQRXCTHAUDJLXV6LMZLKSTUKPITTXIV
    </SenderTokenId>
  </Transaction>
  <Transaction>
    <TransactionId>14GMNT2PDVUJA18L44TO4DIFJEJRF9LTV2T</TransactionId>
    <CallerTransactionDate>
      2009-10-06T22:35:02.031-07:00
    </CallerTransactionDate>
    <DateReceived>2009-10-06T22:35:18.317-07:00</DateReceived>
    <DateCompleted>2009-10-06T22:35:19.332-07:00</DateCompleted>
    <TransactionAmount>
      <CurrencyCode>USD</CurrencyCode>
      <Value>1.000000</Value>
    </TransactionAmount>
    <FPSOperation>Refund</FPSOperation>
    <TransactionStatus>Success</TransactionStatus>
    <StatusMessage>
      The transaction was successful and the payment instrument was charged.
    </StatusMessage>
    <StatusCode>Success</StatusCode>
    <TransactionPart>
      <Role>Caller</Role>
      <Name>Test Business</Name>
      <Reference>CallerReference09</Reference>
      <FeesPaid>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
      </FeesPaid>
    </TransactionPart>
```

```
    <TransactionPart>
      <Role>Sender</Role>
      <Name>Test Business</Name>
      <FeesPaid>
        <CurrencyCode>USD</CurrencyCode>
        <Value>-0.100000</Value>
      </FeesPaid>
    </TransactionPart>
    <PaymentMethod>CC</PaymentMethod>
    <SenderName>Test Business</SenderName>
    <CallerName>Test Business</CallerName>
    <RecipientName>Test Business</RecipientName>
    <FPSFees>
      <CurrencyCode>USD</CurrencyCode>
      <Value>-0.100000</Value>
    </FPSFees>
    <Balance>
      <CurrencyCode>USD</CurrencyCode>
      <Value>1.800000</Value>
    </Balance>
  </Transaction>
  <Transaction>
    <TransactionId>14GMNRDSJ6TJTNDUTOUA917PIFJDSGNB2JP</TransactionId>
    <CallerTransactionDate>
      2009-10-06T22:34:24.053-07:00
    </CallerTransactionDate>
    <DateReceived>2009-10-06T22:34:24.147-07:00</DateReceived>
    <DateCompleted>2009-10-06T22:34:25.223-07:00</DateCompleted>
    <TransactionAmount>
      <CurrencyCode>USD</CurrencyCode>
      <Value>1.000000</Value>
    </TransactionAmount>
    <FPSOperation>Pay</FPSOperation>
    <TransactionStatus>Success</TransactionStatus>
    <StatusMessage>
      The transaction was successful and the payment instrument was charged.
    </StatusMessage>
    <StatusCode>Success</StatusCode>
    <TransactionPart>
      <Role>Recipient</Role>
      <Name>Test Business</Name>
      <Description>SubscriptionTesting</Description>
      <FeesPaid>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.100000</Value>
      </FeesPaid>
    </TransactionPart>
    <TransactionPart>
      <Role>Caller</Role>
      <Name>Test Business</Name>
      <Reference>63314e32-d6b0-4abd-a0ab-7b89717ba5cb</Reference>
      <FeesPaid>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
      </FeesPaid>
    </TransactionPart>
    <PaymentMethod>CC</PaymentMethod>
    <SenderName>Test Business</SenderName>
```

```
      <CallerName>Test Business</CallerName>
      <RecipientName>Test Business</RecipientName>
      <FPSFees>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.100000</Value>
      </FPSFees>
      <Balance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>2.700000</Value>
      </Balance>
      <SenderTokenId>
        533I1M9CGUZ9J4M197BM3LMNKFVQUFVFNT5RRCT2ACDJBXV6LRZ6KSRUSPI6T3I3
      </SenderTokenId>
      <RecipientTokenId>
        D139MTVTMK4CK9QB26PKKLWA1GHDZGBGX3SJLGVU37EFUGJ7XVQIMETLSOGAZJGV
      </RecipientTokenId>
    </Transaction>
  </GetAccountActivityResult>
  <ResponseMetadata>
    <RequestId>87e1570a-ef8c-4846-8265-74d07a6a83fb:0</RequestId>
  </ResponseMetadata>
</GetAccountActivityResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
  <wsa:RelatesTo
      xmlns:wsa="http://www.w3.org/2005/08/addressing">
      MESSAGE123
    </wsa:RelatesTo>
  <wsa:To
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   http://www.w3.org/2005/08/addressing/anonymous
  </wsa:To>
  <wsa:Action
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   GetAccountActivity:Response
  </wsa:Action>
  <wsa:MessageID
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   urn:uuid:a9e1fc80-03f6-4e1b-a1c0-541df545afac
  </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
  <GetAccountActivityResponse
   xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
   <GetAccountActivityResult>
    <BatchSize>1</BatchSize>
    <Transaction>
     <TransactionId>13OKM6RHU2ZT12OFCICMZ326V8GFJ955UGM</TransactionId>
     <CallerTransactionDate>2008-02-11T20:10:28.000-08:00</
      CallerTransactionDate>
      <DateReceived>2008-02-11T20:10:28.798-08:00</DateReceived>
      <DateCompleted>2008-02-11T20:10:43.973-08:00</DateCompleted>
```

```
      <TransactionAmount>
       <CurrencyCode>USD</CurrencyCode>
       <Value>1.000000</Value>
      </TransactionAmount>
      <Operation>Pay</Operation>
      <Status>Success</Status>
      <ErrorCode/>
      <TransactionParts>
       <AccountId>BREJXDLYGGDQEN4ETUEGBHC11CJ1CKTEXAMPLE</AccountId>
       <Role>Caller</Role>
       <Name>BusinessNamehxUflVbJGOmzJLg</Name>
       <Reference>ReferenceStringuWZr8Cnt12316019683371</Reference>
       <Description>DescriptionString-uWZr7AYi12316019683371</
        Description>
        <FeePaid>
         <CurrencyCode>USD</CurrencyCode>
         <Amount>0.000000</Amount>
        </FeePaid>
      </TransactionParts>
      <PaymentMethod>ABT</PaymentMethod>
      <SenderName>John Smith</SenderName>
      <CallerName>BusinessNamekfimuuyvLLMaBLa</CallerName>
      <RecipientName>BusinessNamenvJsQqcbIvpAdcW</RecipientName>
      <FPSFees>
       <CurrencyCode>USD</CurrencyCode>
       <Value>0</Amount>
      </FPSFees>
      <Balance>
       <CurrencyCode>USD</CurrencyCode>
       <Value>970.000000</Amount>
      </Balance>
    </Transaction>
    <StartTimeForNextTransaction>2009-01-28T14:46:20.613-08:00</
     StartTimeForNextTransaction>
   </GetAccountActivityResult>
   <ResponseMetadata>
    <RequestId>9faeed71-9362-4eb8-9431-b99e92b441ee:0</RequestId>
   </ResponseMetadata>
  </GetAccountActivityResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- GetTokenUsage (p. 58)
- GetTokens (p. 52)

GetAccountBalance

Description

The `GetAccountBalance` action returns the current balance of your account.

This action appears in the Account Management Quick Start.

Request Parameters

This action requires only the parameters that are common to all requests. They are described in Common Request Parameters (p. 5). The common parameters are defaulted in SOAP calls but must be explicitly added in REST calls. Parameter names are case sensitive.

Response Elements

Element	Description
`AccountBalance`	Specifies the current balance. Type: String

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AuthFailure (p. 7)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- SignatureDoesNotMatch (p. 12)

Examples

The following sections show a sample request and response.

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetAccountBalance
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Signature=FyQVfGnvleChBRKrWY9XpyXTDfQ09oSdlnGBKw4527Y%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A15%3A46.546Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Envelope>
 <SOAP-ENV:Body wsu:Id="body"
   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
  <ns2:GetAccountBalance
      xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/"/>
```

```
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetAccountBalanceResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
   <GetAccountBalanceResult>
      <AccountBalance>
         <TotalBalance>
            <CurrencyCode>USD</CurrencyCode>
            <Value>7.400000</Value>
         </TotalBalance>
         <PendingInBalance>
            <CurrencyCode>USD</CurrencyCode>
            <Value>0.000000</Value>
         </PendingInBalance>
         <PendingOutBalance>
            <CurrencyCode>USD</CurrencyCode>
            <Value>0.000000</Value>
         </PendingOutBalance>
         <AvailableBalances>
            <DisburseBalance>
               <CurrencyCode>USD</CurrencyCode>
               <Value>7.400000</Value>
            </DisburseBalance>
            <RefundBalance>
               <CurrencyCode>USD</CurrencyCode>
               <Value>7.400000</Value>
            </RefundBalance>
         </AvailableBalances>
      </AccountBalance>
   </GetAccountBalanceResult>
   <ResponseMetadata>
      <RequestId>7b74a504-7517-4d81-8312-1427570d028c:0</RequestId>
   </ResponseMetadata>
</GetAccountBalanceResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
  <wsa:RelatesTo
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   MESSAGE123
  </wsa:RelatesTo>
  <wsa:To
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   http://www.w3.org/2005/08/addressing/anonymous
  </wsa:To>
  <wsa:Action
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   GetAccountBalance:Response
  </wsa:Action>
```

```
  <wsa:MessageID
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   urn:uuid:fce1f1e5-de4e-401b-b0bb-905e84ee933b
  </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
  <GetAccountBalanceResponse
   xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
   <GetAccountBalanceResult>
    <AccountBalance>
     <TotalBalance>
      <CurrencyCode>USD</CurrencyCode>
      <Value>10.750000</Value>
     </TotalBalance>
     <PendingInBalance>
      <CurrencyCode>USD</CurrencyCode>
      <Value>0.000000</Value>
     </PendingInBalance>
     <PendingOutBalance>
      <CurrencyCode>USD</CurrencyCode>
      <Value>0.000000</Value>
     </PendingOutBalance>
     <AvailableBalances>
      <DisburseBalance>
       <CurrencyCode>USD</CurrencyCode>
       <Value>10.750000</Value>
      </DisburseBalance>
      <RefundBalance>
       <CurrencyCode>USD</CurrencyCode>
       <Value>10.750000</Value>
      </RefundBalance>
     </AvailableBalances>
    </AccountBalance>
   </GetAccountBalanceResult>
   <ResponseMetadata>
    <RequestId>fce1f1e5-de4e-401b-b0bb-905e84ee933b:0</RequestId>
   </ResponseMetadata>
  </GetAccountBalanceResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- GetTransaction (p. 64)

GetDebtBalance

Description

The `GetDebtBalance` action retrieves the balance of a credit instrument. You can query for the debt balance only on the instruments for which you are the sender or the recipient.

This action appears in the Aggregated Quick Start.

Request Parameters

Parameter	Description	Required
CreditInstrumentId	ID for the postpaid credit instrument to query. Type: String Default: None Max Size: 128 Bytes	Yes

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
DebtBalance	Specifies the available balance on this account. Type: DebtBalance (p. 114)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AuthFailure (p. 7)
- InstrumentAccessDenied (p. 8)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidPaymentInstrument (p. 9)
- SignatureDoesNotMatch (p. 12)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetDebtBalance
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CreditInstrumentId=553IBMFCG7Z8J4D1H7BB3HMNPFQQUPVUNT8RHCT1AXDJNXE6LJZ4KSKU2PI1TTIG
&Signature=4nwBSR0wrlBcy%2BWZz08TXcnGth1Qbln15IxocPDNukQ%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A20%3A33.609Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body
  wsu:Id="body"
xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:GetDebtBalance xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
     <ns2:CreditInstrumentId>
      J398G1HA5XJRX4843JGZSTMYEZVPJ7YEARXBPCFIGJ9M4UDUMXXGXJCEBKCGI9N6
     </ns2:CreditInstrumentId>
   </ns2:GetDebtBalance>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetDebtBalanceResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <GetDebtBalanceResult>
    <DebtBalance>
      <AvailableBalance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>5.000000</Value>
      </AvailableBalance>
      <PendingOutBalance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
      </PendingOutBalance>
    </DebtBalance>
  </GetDebtBalanceResult>
  <ResponseMetadata>
    <RequestId>73f8efcd-0ea3-4015-b7da-5da1b1111b82:0</RequestId>
  </ResponseMetadata>
</GetDebtBalanceResponse>
```

Sample Response to SOAP Request

```
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
    MESSAGE123
    </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
    GetDebtBalance:Response
   </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
    urn:uuid:d7d45a93-8699-4042-a014-0ae0b8102b07
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <GetDebtBalanceResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
```

```
    <GetDebtBalanceResult>
      <DebtBalance>
        <AvailableBalance>
          <CurrencyCode>USD</CurrencyCode>
          <Value>0.000000</Value>
        </AvailableBalance>
        <PendingOutBalance>
          <CurrencyCode>USD</CurrencyCode>
          <Value>0.000000</Value>
        </PendingOutBalance>
      </DebtBalance>
    </GetDebtBalanceResult>
    <ResponseMetadata>
      <RequestId>d7d45a93-8699-4042-a014-0ae0b8102b07:0</RequestId>
    </ResponseMetadata>
  </GetDebtBalanceResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- WriteOffDebt (p. 106)
- SettleDebt (p. 98)

GetOutstandingDebtBalance

Description

`GetOutstandingDebtBalance` retrieves the total accumulated debt on all credit instruments issued by the caller. The caller acts as creditor on all of these credit instruments.

This action appears in the Aggregated Quick Start.

Request Parameters

`GetOutstandingDebtBalance` only uses parameters that are common to all requests (for a list, see Common Request Parameters (p. 5)). Amazon FPS uses your *`AWSAccessKeyId`* to retrieve the correct results.

Response Elements

Element	Description
`OutstandingDebt`	Specifies the total outstanding debt on all the credit instruments. Type: OutstandingDebtBalance (p. 116)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountClosed (p. 6)
- AuthFailure (p. 7)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- SignatureDoesNotMatch (p. 12)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetOutstandingDebtBalance
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Signature=hy5DUE2o85mI6cEIeT%2BcicKn19OdFpjHfZKICS8IM5Y%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A21%3A21.750Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body
  wsu:Id="body"
  xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:GetOutstandingDebtBalance
   xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/"/>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetOutstandingDebtBalanceResponse xmlns="http://fps.amazonaws.com/doc/2008-09-
17/">
  <GetOutstandingDebtBalanceResult>
    <OutstandingDebt>
      <OutstandingBalance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>5.000000</Value>
      </OutstandingBalance>
      <PendingOutBalance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
```

```
      </PendingOutBalance>
    </OutstandingDebt>
  </GetOutstandingDebtBalanceResult>
  <ResponseMetadata>
    <RequestId>ae85ac11-a38c-4c42-94fc-b71c9598c76f:0</RequestId>
  </ResponseMetadata>
</GetOutstandingDebtBalanceResponse>
```

Sample Response to SOAP Request

```
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
  MESSAGE123</wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
  http://www.w3.org/2005/08/addressing/anonymous
    </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
  GetOutstandingDebtBalance:Response
    </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
  urn:uuid:6661b766-61db-46c0-9334-5276354c4d80
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <GetOutstandingDebtBalanceResponse
    xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
     <GetOutstandingDebtBalanceResult>
       <OutstandingDebt>
         <OutstandingBalance>
           <CurrencyCode>USD</CurrencyCode>
           <Value>0.000000</Value>
         </OutstandingBalance>
         <PendingOutBalance>
           <CurrencyCode>USD</CurrencyCode>
           <Value>0.000000</Value>
         </PendingOutBalance>
       </OutstandingDebt>
     </GetOutstandingDebtBalanceResult>
     <ResponseMetadata>
       <RequestId>6661b766-61db-46c0-9334-5276354c4d80:0</RequestId>
     </ResponseMetadata>
   </GetOutstandingDebtBalanceResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- FundPrepaid (p. 24)

GetPrepaidBalance

Description

The `GetPrepaidBalance` action retrieves the balance of a prepaid payment token.

Note
You can query for the prepaid balance only if you are the sender or the recipient for the prepaid instrument.

This action appears in the Aggregated Quick Start.

Request Parameters

Parameter	Description	Required
PrepaidInstrumentId	ID for the prepaid instrument that holds the prepaid balance. Type: String Default: None Max Size: 128 Bytes	Yes

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
`PrepaidBalance`	Specifies the balance on the prepaid instrument. Type: PrepaidBalance (p. 117)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AuthFailure (p. 7)
- InstrumentAccessDenied (p. 8)
- InvalidClientTokenId (p. 9)
- InternalError (p. 8)
- InvalidParams (p. 9)
- InvalidPaymentInstrument (p. 9)
- SignatureDoesNotMatch (p. 12)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetPrepaidBalance
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&PrepaidInstrumentId=573IEMDCGRZHJ4P1D7BC3GMNBFPQU1VLNT7R4CT7A7DJIX96LKZNKSXUG
PIGTTI8
&Signature=5NVm1DjMZ8SllhFxfwkewKRmF%2FXb3iUPiAWT4AqksJg%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A26%3A29.171Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body
  wsu:Id="body"
  xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:GetPrepaidBalance
   xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
     <ns2:PrepaidInstrumentId>
      G5H24VEMGPLG2QKS5MGN7MYCKW2WRZKRBQ4IFDT5IXNS8X2V94BZPXCY8CCZZHHQ
     </ns2:PrepaidInstrumentId>
   </ns2:GetPrepaidBalance>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetPrepaidBalanceResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <GetPrepaidBalanceResult>
    <PrepaidBalance>
      <AvailableBalance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>2.000000</Value>
      </AvailableBalance>
      <PendingInBalance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
      </PendingInBalance>
    </PrepaidBalance>
  </GetPrepaidBalanceResult>
  <ResponseMetadata>
    <RequestId>f04c9ea5-13a2-4379-9129-39cbbad7bcfe:0</RequestId>
  </ResponseMetadata>
</GetPrepaidBalanceResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
    MESSAGE123
   </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
    GetPrepaidBalance:Response
   </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
    urn:uuid:46e3f4d8-6ab7-489a-a332-5adbd12de23b
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <ns3:GetPrepaidBalanceResponse
     xmlns:ns2="http://fps.amazonaws.com/private/doc/2007-01-08/"
     xmlns:ns3="http://fps.amazonaws.com/doc/2007-01-08/"
     xmlns:ns4="http://fps.amazonaws.com/internal/doc/2007-01-08/">
     <PrepaidBalance>
       <AvailableBalance>
         <CurrencyCode>USD</CurrencyCode>
         <Amount>0.000000</Amount>
       </AvailableBalance>
       <PendingInBalance>
         <CurrencyCode>USD</CurrencyCode>
         <Amount>0.000000</Amount>
       </PendingInBalance>
     </PrepaidBalance>
     <Status>Success</Status>
     <RequestId>46e3f4d8-6ab7-489a-a332-5adbd12de23b:0</RequestId>
   </ns3:GetPrepaidBalanceResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- FundPrepaid (p. 24)

GetRecipientVerificationStatus

Description

`GetRecipientVerificationStatus` enables you to test that the intended recipient has a verified Amazon Payments account before you present the payment option for that seller or recipient on your website. The *`RecipientVerificationStatus`* return parameter enables you to determine whether the account is unlimited in the amount of money it can receive.

This action appears in the Advanced and Marketplace Quick Starts.

Request Parameters

Parameter	Description	Required
RecipientTokenID	The recipient token returned by the Co-branded user interface. Type: String	Yes

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
RecipientVerificationStatus	Status of the verification. Type: RecipientVerificationStatus (p. 117)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors are returned only for REST. For SOAP, an error results in a SOAP fault. If the response status is failure, the `Errors` element includes an error code that identifies the source of the failure. If the response status is success, the elements listed in the preceding table are returned.

Errors

This action can return the following errors:

- InternalError (p. 8)
- InvalidAccountState (p. 8)
- InvalidParams (p. 9)
- InvalidTokenId (p. 10)
- TokenNotActive (p. 12)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetRecipientVerificationStatus
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&RecipientTokenId=09DG234OGD
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2008-08-06T13%3A00%3A01Z
&TokenId=254656Example83987
&Version=2008-09-17
&Signature=[URL-encoded signature value]
```

Sample SOAP Request

```
<SOAP-ENV:Envelope>
 <SOAP-ENV:Body wsu:Id="body"
  xmlns:wsu=
    "http://docs.oasis-open.org/wss/2004/01/&#x200b;oasis-200401-wss-wssecurity-
utility-1.0.xsd">
   <ns2:GetRecipientVerificationStatus
    xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
      <ns2:RecipientTokenId>
       M2ABVSN2BH3KRUV9261D59HI2C2TBZDURKFB9MCU88BTHS876ZATXRIHEFZBBKIM
      </ns2:RecipientTokenId>
   </ns2:GetRecipientVerificationStatus>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetRecipientVerificationResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">

  <GetRecipientVerificationResult>
    <RecipientVerificationStatus>
     VerificationComplete
    </RecipientVerificationStatus>
  </GetRecipientVerificationResult>
  <ResponseMetadata>
    <RequestId>197e2085-1ed7-47a2-93d8-d76b452acc74:0</RequestId>
  </ResponseMetadata>
</GetRecipientVerificationResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
      MESSAGE123
    </wsa:RelatesTo>
    <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
      http://www.w3.org/2005/08/addressing/anonymous
    </wsa:To>
    <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
      GetRecipientVerificationStatus:Response
    </wsa:Action>
    <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
      urn:uuid:95f01489-c0de-451d-8f0d-63b44b9c5bd5
    </wsa:MessageID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <GetRecipientVerificationStatusResponse
      xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
      <GetRecipientVerificationStatusResult>
        <RecipientVerificationStatus>
          VerificationComplete
```

```
        </RecipientVerificationStatus>
      </GetRecipientVerificationStatusResult>
      <ResponseMetadata>
        <RequestId>95f01489-c0de-451d-8f0d-63b44b9c5bd5:0</RequestId>
      </ResponseMetadata>
    </GetRecipientVerificationStatusResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetTokens

Description

The `GetTokens` action returns all or a subset of the tokens that you installed on your account.

This action appears in the Account Management Quick Start.

Request Parameters

Parameter	Description	Required
CallerReference	A value you provide that uniquely identifies the request. Type: String Default: None Constraint: Max size = 128 characters	No
TokenStatus	Filters the results based on the status of the token. Type: TokenStatus (p. 118) Default: None	No
TokenType	Filters the result based on the token type. Type: TokenType (p. 119) Default: None	No

You must also use parameters that are common to all requests that are described in Common Request Parameters (p. 5). The common parameters are defaulted in SOAP calls but must be explicitly added in REST calls. Parameter names are case sensitive.

Response Elements

Element	Description
Token	The list of the caller's tokens. Type: Token (p. 119)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountClosed (p. 6)
- AuthFailure (p. 7)
- InvalidClientTokenId (p. 9)
- InternalError (p. 8)
- InvalidParams (p. 9)
- SignatureDoesNotMatch (p. 12)

Examples

The following sections show a sample request and response.

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetTokens
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerReference=CallerReference12
&Signature=Dzp4usKpQujx9x74WFxl5BO2C3ID65PlEb2MXwkyV8M%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A38%3A11.796Z
&Version=2008-09-17
```

Sample SOAP Request

```
</SOAP-ENV:Envelope>
  <SOAP-ENV:Body
    wsu:Id="body"
   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">
    <ns2:GetTokens
      xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
      <ns2:CallerReference>293847SKDJRH29387</ns2:CallerReference>
    </ns2:GetTokens>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetTokensResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
 <GetTokensResult>
  <Token>
   <TokenId>
    D439DTSTMP4FK9NBL6PEKZWAPGRDZ2BDX3MJNGVX37EF3GA7XRQHMEELQOGFZ9GK
   </TokenId>
```

```
   <TokenStatus>Active</TokenStatus>
   <DateInstalled>2009-10-07T04:37:57.375-07:00</DateInstalled>
   <CallerReference>CallerReference12</CallerReference>
   <TokenType>SingleUse</TokenType>
   <OldTokenId>
    D439DTSTMP4FK9NBL6PEKZWAPGRDZ2BDX3MJNGVX37EF3GA7XRQHMEELQOGFZ9GK
   </OldTokenId>
  </Token>
 </GetTokensResult>
 <ResponseMetadata>
  <RequestId>c9db3c80-ff03-4a32-b6b6-ee071cd118c8:0</RequestId>
 </ResponseMetadata>
</GetTokensResponse>
```

Sample Response to SOAP Request

This section shows a sample SOAP response.

```
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
  <wsa:RelatesTo
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   MESSAGE123
  </wsa:RelatesTo>
  <wsa:To
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   http://www.w3.org/2005/08/addressing/anonymous
  </wsa:To>
  <wsa:Action
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   GetTokens:Response
  </wsa:Action>
  <wsa:MessageID
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   urn:uuid:edf2a735-2b50-4c71-9a1a-943d946d4a89
  </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
  <GetTokensResponse
   xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
   <GetTokensResult>
    <Token>
     <TokenId>
            M157V8ZXS4AF9C132FNT5HXNNX8ZK7K1C5IHVRE4VZJTCEBLP8X1CTBCSFTFHKCK
          </TokenId>
      <FriendlyName>FriendlyName-????pHhq12275863261891</
       FriendlyName>
       <TokenStatus>Active</TokenStatus>
       <DateInstalled>2008-04-24T20:12:06.200-08:00</DateInstalled>
       <CallerReference>293847SKDJRH29387</CallerReference>
       <TokenType>Unrestricted</TokenType>
       <OldTokenId>
              M157V8ZXS4AF9C132FNT5HXNNX8ZK7K1C5IHVRE4VZJTCEBLP8X1CTBCSFTFHKCK
              </OldTokenId>
       <PaymentReason>Testing</PaymentReason>
```

```
    </Token>
   </GetTokensResult>
   <ResponseMetadata>
    <RequestId>edf2a735-2b50-4c71-9a1a-943d946d4a89:0</RequestId>
   </ResponseMetadata>
  </GetTokensResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- GetTokenUsage (p. 58)

GetTokenByCaller

Description

The `GetTokenByCaller` action returns the details about the token specified by a *`tokenId`* or *`CallerReference`*. The *`CallerReference`* is the value you passed in the Co-Branded service request, whereas the *`tokenId`* is the value you received in the Co-Branded service response.

This action appears in the Basic, Advanced, Aggregated and Marketplace Quick Starts.

Request Parameters

Parameter	Description	Required
`CallerReference`	A value you provide that uniquely identifies the request. . Type: String Default: None Constraint: Max size = 128 characters Condition: Required if *`TokenId`* is not specified.	Conditional
`TokenId`	The sender token ID that the Co-Branded service returned. Type: String Default: None Constraint: Max size = 65 characters Condition: Required if *`CallerReference`* is not specified.	Conditional

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
`Token`	Details of the specified token. Type: Token (p. 119)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountClosed (p. 6)
- AuthFailure (p. 7)
- InternalError (p. 8)
- InvalidCallerReference (p. 9)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTokenId (p. 10)
- SignatureDoesNotMatch (p. 12)

Examples

The following sections show a sample request and response.

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetTokenByCaller
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerReference=callerReferenceSingleUse10
&Signature=7E43HRAge3s57KDtEW3%2Fv0CE3Rh4TkVuOpk%2FIU%2FJIEY%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A29%3A03.281Z
&TokenId=543IJMECGZZ3J4K1F7BJ3TMNXFBQU9VXNT7RRCTNAJDJ8X36L1ZRKSUUPPIBTTIK
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body wsu:Id="body"
xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:GetTokenByCaller xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
     <ns2:CallerReference>
      ReferenceString????AxaM12275863261891
     </ns2:CallerReference>
   </ns2:GetTokenByCaller>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetTokenByCallerResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <GetTokenByCallerResult>
    <Token>
      <TokenId>
        543IJMECGZZ3J4K1F7BJ3TMNXFBQU9VXNT7RRCTNAJDJ8X36L1ZRKSUUPPIBTTIK
      </TokenId>
      <FriendlyName>Friendly1339359778</FriendlyName>
      <TokenStatus>Active</TokenStatus>
      <DateInstalled>2009-10-07T04:29:05.054-07:00</DateInstalled>
      <CallerReference>callerReferenceSingleUse10</CallerReference>
      <TokenType>SingleUse</TokenType>
      <OldTokenId>
        543IJMECGZZ3J4K1F7BJ3TMNXFBQU9VXNT7RRCTNAJDJ8X36L1ZRKSUUPPIBTTIK
      </OldTokenId>
      <PaymentReason>PaymentReason</PaymentReason>
    </Token>
  </GetTokenByCallerResult>
  <ResponseMetadata>
    <RequestId>45b6c560-8aa9-463c-84be-80eeefb21034:0</RequestId>
  </ResponseMetadata>
</GetTokenByCallerResponse>
```

Sample Response to SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
    MESSAGE123
   </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
 GetTokenByCaller:Response
 </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
 urn:uuid:4802991b-76dd-4f7b-8bd7-a1428cfbb9f2
 </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <GetTokenByCallerResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
     <GetTokenByCallerResult>
       <Token>
         <TokenId>
          M157V8ZXS4AF9C132FNT5HXNNX8ZK7K1C5IHVRE4VZJTCEBLP8X1CTBCSFTFHKCK
         </TokenId>
         <FriendlyName>
          FriendlyName-pHhq12275863261891
         </FriendlyName>
         <TokenStatus>Active</TokenStatus>
         <DateInstalled>2008-04-24T20:12:06.200-08:00</DateInstalled>
```

```
        <CallerReference>ReferenceStringAxaM12275863261891</CallerReference>
        <TokenType>Unrestricted</TokenType>
        <OldTokenId>
         M157V8ZXS4AF9C132FNT5HXNNX8ZK7K1C5IHVRE4VZJTCEBLP8X1CTBCSFTFHKCK
        </OldTokenId>
        <PaymentReason>Testing</PaymentReason>
      </Token>
    </GetTokenByCallerResult>
    <ResponseMetadata>
      <RequestId>4802991b-76dd-4f7b-8bd7-a1428cfbb9f2:0</RequestId>
    </ResponseMetadata>
  </GetTokenByCallerResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GetTokenUsage

Description

The `GetTokenUsage` action returns the usage of the given token ID over the last two time periods for the limits defined on the token. You define the limits against which the usage is measured before installing the token with the Amazon FPS Co-Branded service.

> **Note**
> This action works only with multi-use and recurring-use tokens. It does not return token usage for single-use tokens.

This action appears in the Account Management Quick Start.

Request Parameters

Parameter	Description	Required
TokenId	The token ID for the token you want usage data for. Type: String Default: None Constraint: Max size = 64 characters	Yes

You must also use parameters that are common to all requests that are described in Common Request Parameters (p. 5). The common parameters are defaulted in SOAP calls but must be explicitly added in REST calls. Parameter names are case sensitive.

Response Elements

Element	Description
`TokenUsageLimits`	A list containing the details of this token's usage for each limit defined while installing the token. Type: TokenUsageLimit (p. 120)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AuthFailure (p. 7)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTokenId (p. 10)
- InvalidTokenType (p. 10)
- SignatureDoesNotMatch (p. 12)
- TokenAccessDenied (p. 12)

Examples

The following sections show a sample request and response.

Sample REST Request

```
https://fps.amazonaws.com/?
Action=GetTokenUsage
&accessKey=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2008-08-06T13%3A00%3A01Z
&TokenId=254656Example83987
&Signature=[URL-encoded signature value]
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Envelope>
 <SOAP-ENV:Body
  wsu:Id="body"
  xmlns:wsu="http://docs.oasis-open.org/ wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd"
  >
  <ns2:GetTokenUsage
   xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/"
   >
   <ns2:TokenId>
    23KMPG91JJBN37MI23IE4MXK7J8AC3JB7ESI5T1E3URQXA2TFXRK7XTRJZTDXLWK
   </ns2:TokenId>
  </ns2:GetTokenUsage>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetTokenUsageResponse
  xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
   <GetTokenUsageResult>
       <TokenUsageLimits>
           <Amount>
              <CurrencyCode>USD</CurrencyCode>
              <Value>10.000000</Value>
           </Amount>
           <LastResetAmount>
               <CurrencyCode>USD</CurrencyCode>
               <Value>0.000000</Value>
            </LastResetAmount>
            <LastResetTimestamp>
              2008-01-01T02:00:00.000-08:00
            </LastResetTimestamp>
       </TokenUsageLimits>
       <TokenUsageLimits>
           <Count>1</Count>
           <LastResetCount>0</LastResetCount>
           <LastResetTimestamp>
            2008-01-01T02:00:00.000-08:00
           </LastResetTimestamp>
       </TokenUsageLimits>
   </GetTokenUsageResult>
   <ResponseMetadata>
       <RequestId>9faeed71-9362-4eb8-9431-b99e92b441ee:0</RequestId>
   </ResponseMetadata>
</GetTokenUsageResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
  <wsa:RelatesTo
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   MESSAGE123
  </wsa:RelatesTo>
  <wsa:To
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   http://www.w3.org/2005/08/addressing/anonymous
  </wsa:To>
  <wsa:Action
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   GetTokenUsage:Response
  </wsa:Action>
  <wsa:MessageID
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   urn:uuid:9faeed71-9362-4eb8-9431-b99e92b441ee
  </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
  <GetTokenUsageResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
```

```
   <GetTokenUsageResult>
    <TokenUsageLimits>
     <Amount>
      <CurrencyCode>USD</CurrencyCode>
      <Value>10.000000</Value>
     </Amount>
     <LastResetAmount>
      <CurrencyCode>USD</CurrencyCode>
      <Value>0.000000</Value>
     </LastResetAmount>
     <LastResetTimestamp>2008-01-01T02:00:00.000-08:00</
      LastResetTimestamp>
    </TokenUsageLimits>
    <TokenUsageLimits>
     <Count>1</Count>
     <LastResetCount>0</LastResetCount>
     <LastResetTimestamp>2008-01-01T02:00:00.000-08:00</
      LastResetTimestamp>
    </TokenUsageLimits>
   </GetTokenUsageResult>
   <ResponseMetadata>
    <RequestId>9faeed71-9362-4eb8-9431-b99e92b441ee:0</RequestId>
   </ResponseMetadata>
  </GetTokenUsageResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- GetTokens (p. 52)

GetTotalPrepaidLiability

Description

The `GetTotalPrepaidLiability` action returns the recipient's total liability resulting from all of his or her prepaid instruments.

This action appears in the Aggregated Quick Start.

Request Parameters

`GetTotalPrepaidLiability` only uses parameters that are common to all requests (for a list, see Common Request Parameters (p. 5)). Amazon FPS uses your *`AWSAccessKeyId`* to retrieve the correct results.

Response Elements

Element	Description
`OutstandingPrepaidLiability`	Specifies the sum of all the balances of all the prepaid instruments where you hold the prepaid liability. Type: OutstandingPrepaidLiability (p. 116)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountClosed (p. 6)
- AuthFailure (p. 7)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- SignatureDoesNotMatch (p. 12)

Examples

The following sections show a sample request and response.

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetTotalPrepaidLiability
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Signature=O%2FVVBU%2FzHk7Rp%2FSjFsE932PGpaJU777klujOjx%2FxL8I%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A40%3A12.109Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body
  wsu:Id="body"
xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:GetTotalPrepaidLiability
    xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/"/>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetTotalPrepaidLiabilityResponse
  xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <GetTotalPrepaidLiabilityResult>
    <OutstandingPrepaidLiability>
      <OutstandingBalance>
        <CurrencyCode>USD</CurrencyCode>
```

```
        <Value>2.000000</Value>
      </OutstandingBalance>
      <PendingInBalance>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.000000</Value>
      </PendingInBalance>
    </OutstandingPrepaidLiability>
  </GetTotalPrepaidLiabilityResult>
  <ResponseMetadata>
    <RequestId>ac7b28ea-c340-4638-be46-d4a8fa54a801:0</RequestId>
  </ResponseMetadata>
</GetTotalPrepaidLiabilityResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
    MESSAGE123
   </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
    GetTotalPrepaidLiability:Response
   </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
    urn:uuid:16d69143-2542-4b8f-8642-8d96d8a3a8fb
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <GetTotalPrepaidLiabilityResponse
   xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
     <GetTotalPrepaidLiabilityResult>
       <OutstandingPrepaidLiability>
         <OutstandingBalance>
           <CurrencyCode>USD</CurrencyCode>
           <Value>50.000000</Value>
         </OutstandingBalance>
         <PendingInBalance>
           <CurrencyCode>USD</CurrencyCode>
           <Value>0.000000</Value>
         </PendingInBalance>
       </OutstandingPrepaidLiability>
     </GetTotalPrepaidLiabilityResult>
     <ResponseMetadata>
       <RequestId>16d69143-2542-4b8f-8642-8d96d8a3a8fb:0</RequestId>
     </ResponseMetadata>
   </GetTotalPrepaidLiabilityResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- FundPrepaid (p. 24)
- GetPrepaidBalance (p. 47)

GetTransaction

Description

The `GetTransaction` action returns details of the transaction specified in *`transactionId`*. You can use this action only for transactions within your own account.

This action appears in the Account Management Quick Start.

Request Parameters

Parameter	Description	Required
`TransactionId`	Transaction ID of the transaction you want to get. Type: String Default: None Constraint: Max size = 35 characters	Yes

You must also use parameters that are common to all requests that are described in Common Request Parameters (p. 5). Parameter names are case sensitive.

Response Elements

Element	Description
`Transaction`	Contains the transaction details. Type: TransactionDetail (p. 123)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following synchronous errors, which occur within the status for this action.

- AccessFailure (p. 6)
- AuthFailure (p. 7)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTransactionId (p. 11)
- SignatureDoesNotMatch (p. 12)

Examples

The following sections show a sample request and response.

Sample REST Request

```
https://fps.sandbox.amazonaws.com/?
Action=GetTransactionsForSubscription
&SubscriptionId=SubscriptionId
&Version=2008-09-17
&AWSAccessKeyId=AccessKey
&Timestamp=2011-03-11T06%3A59%3A40Z
&SignatureVersion=2
&Signature=SignatureCalculated
&SignatureMethod=HmacSHA256
```

Sample SOAP Request

```
<SOAP-ENV:Envelope>
 <SOAP-ENV:Body
  wsu:Id="body"
  xmlns:wsu=
  "http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
  <ns2:GetTransaction
   xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
   <ns2:TransactionId>
    13OKM6RHU2ZT12OFCICMZ326V8GFJ955UGM
   </ns2:TransactionId>
  </ns2:GetTransaction>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetTransactionResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <GetTransactionResult>
    <Transaction>
      <TransactionId>14GK6BGKA7U6OU6SUTNLBI5SBBV9PGDJ6UL</TransactionId>
      <CallerReference>CallerReference02</CallerReference>
      <CallerDescription>MyWish</CallerDescription>
      <DateReceived>2009-10-05T22:50:08.010-07:00</DateReceived>
      <DateCompleted>2009-10-05T22:50:09.086-07:00</DateCompleted>
      <TransactionAmount>
        <CurrencyCode>USD</CurrencyCode>
        <Value>1.000000</Value>
      </TransactionAmount>
      <FPSFees>
        <CurrencyCode>USD</CurrencyCode>
        <Value>0.100000</Value>
      </FPSFees>
      <FPSFeesPaidBy>Recipient</FPSFeesPaidBy>
      <SenderTokenId>
```

```
        553ILMLCG6Z8J431H7BX3UMN3FFQU8VSNTSRNCTAASDJNX66LNZLKSZU3PI7TXIH
      </SenderTokenId>
      <FPSOperation>Pay</FPSOperation>
      <PaymentMethod>CC</PaymentMethod>
      <TransactionStatus>Success</TransactionStatus>
      <StatusCode>Success</StatusCode>
      <StatusMessage>
        The transaction was successful and the payment instrument was charged.
      </StatusMessage>
      <SenderName>Test Business</SenderName>
      <SenderEmail>new_premium@amazon.com</SenderEmail>
      <CallerName>Test Business</CallerName>
      <RecipientName>Test Business</RecipientName>
      <RecipientEmail>test-caller@amazon.com</RecipientEmail>
      <StatusHistory>
        <Date>2009-10-05T22:50:08.092-07:00</Date>
        <TransactionStatus>Pending</TransactionStatus>
        <StatusCode>PendingNetworkResponse</StatusCode>
      </StatusHistory>
      <StatusHistory>
        <Date>2009-10-05T22:50:09.086-07:00</Date>
        <TransactionStatus>Success</TransactionStatus>
        <StatusCode>Success</StatusCode>
      </StatusHistory>
    </Transaction>
  </GetTransactionResult>
  <ResponseMetadata>
    <RequestId>0702960e-8221-4e04-9413-ca7d010d3b06:0</RequestId>
  </ResponseMetadata>
</GetTransactionResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope
 xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
  <wsa:RelatesTo
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   MESSAGE123
  </wsa:RelatesTo>
  <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
   http://www.w3.org/2005/08/addressing/anonymous
  </wsa:To>
  <wsa:Action
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   GetTransaction:Response
  </wsa:Action>
  <wsa:MessageID
   xmlns:wsa="http://www.w3.org/2005/08/addressing">
   urn:uuid:d1a8fbe9-8ebf-487d-8cec-a274808b016e
  </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
  <GetTransactionResponse
   xmlns="http://fps.amazonaws.com/doc/2008-09-17/"
   >
```

```
<GetTransactionResult>
 <Transaction>
  <TransactionId>
   13OKM6RHU2ZT12OFCICMZ326V8GFJ955UGM
  </TransactionId>
  <CallerReference>
   ReferenceString????6pQ412290550286101
  </CallerReference>
  <CallerDescription>
   DescriptionStringRDIOmldco9838298
  </CallerDescription>
  <SenderDescription>
   DescriptionStringsErHkl0kOnrlaa1j1
  </SenderDescription>
  <DateReceived>2008-12-11T20:10:28.798-08:00</DateReceived>
  <DateCompleted>2008-12-11T20:10:43.973-08:00</DateCompleted>
  <TransactionAmount>
   <CurrencyCode>USD</CurrencyCode>
   <Value>1.000000</Value>
  </TransactionAmount>
  <FPSFees>
   <CurrencyCode>USD</CurrencyCode>
   <Value>0.100000</Value>
  </FPSFees>
  <MarketplaceFees>
   <CurrencyCode>USD</CurrencyCode>
   <Value>0.150000</Value>
  </MarketplaceFees>
  <FPSFeesPaidBy>Recipient</FPSFeesPaidBy>
  <SenderTokenId>
   53FZS3LSLEPIRG2I4R3F2RGCCZMJXBURF4QQPA5XLC3QD14LMXJ5EG2K4LGLGJJN
  </SenderTokenId>
  <RecipientTokenId>
   F3FTBFRVBM1TLPGJIIHV43NHMD6IX6EZAUNVNHUFZ6MHN15ZIEUFGIHMCPALQDYR
  </RecipientTokenId>
  <FPSOperation>Pay</FPSOperation>
  <PaymentMethod>CC</PaymentMethod>
  <TransactionStatus>Success</TransactionStatus>
  <StatusCode>Success</StatusCode>
  <StatusMessage>
   The transaction was successful and the payment
   instrument was charged.
  </StatusMessage>
  <SenderName>John Smith</SenderName>
  <SenderEmail>johnsmith@example.com</SenderEmail>
  <CallerName>BusinessNamekfimuuyvLLMaBLa</CallerName>
  <RecipientName>BusinessNamenvJsQqcbIvpAdcW</RecipientName>
  <RecipientEmail>sandbox-lralw@example.com</RecipientEmail>
  <RelatedTransaction>
   <TransactionId>
    13OKM7E767ZR76D1FK1JOJM5T5OODHLMIRQ
   </TransactionId>
  </RelatedTransaction>
  <StatusHistory>
   <Date>2008-12-11T20:10:29.047-08:00</Date>
   <Status>Pending</Status>
   <Amount>1.000000</Amount>
  </StatusHistory>
```

```
    <StatusHistory>
     <Date>2008-12-11T20:10:43.973-08:00</Date>
     <Status>Success</Status>
     <Amount>1.000000</Amount>
    </StatusHistory>
   </Transaction>
  </GetTransactionResult>
  <ResponseMetadata>
   <RequestId>d1a8fbe9-8ebf-487d-8cec-a274808b016e:0</RequestId>
  </ResponseMetadata>
 </GetTransactionResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Related Actions

- GetTokens (p. 52)

GetTransactionStatus

Description

The `GetTransactionStatus` action returns the status of the transaction specified by the *`TransactionId`*. .

This action appears in all Amazon FPS Quick Starts as well as in the Amazon Simple Pay Advanced Guide.

Request Parameters

Parameter	Definition	Required
`TransactionId`	The transaction's ID. Type: String Constraint: Max size = 35 characters Default: None	Yes

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
`CallerReference`	A value you provide that uniquely identifies the request. Type: String Size: 128 bytes

Element	Description
StatusCode	Shorthand code that specifies the status of the transaction. Expands on the information in the *TransactionStatus* field. For example, if *TransactionStatus* is PENDING, this field might be *PendingVerification*, or *PendingNetworkResponse*. Type: String Size: 64 bytes Valid Values: See Status Codes (p. 69)
StatusMessage	A description of the transaction status. Type: String (LOB, Large Object)
TransactionId	Unique ID generated by Amazon FPS for this transaction. This element is returned if the transaction was accepted by Amazon FPS. Type: String Size: 35 Bytes
TransactionStatus	The status of the transaction. Provides a short code on the status of the transaction, for example "PENDING". Type: TransactionStatus (p. 121) Size: 20 bytes

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Status Codes

This action can return the following values for StatusCode.

Status Code	Message
Canceled	The transaction was explicitly canceled by the caller.
Expired	This reserved amount on the payment instrument was not settled within the timeout period OR The transaction could not be completed within the specified timeout.
PendingNetworkResponse	This transaction is awaiting a response from the backend payment processor OR (Message returned by backend payment processor)
PendingVerification	The transaction has been flagged for manual investigation
Success	The requested amount was reserved successfully against the given payment instrument. OR The transaction was successful and the payment instrument was charged.

Status Code	Message
`TransactionDenied`	(Message returned by backend payment processor). OR The transaction was denied after investigation.

Errors

This action can return the following synchronous errors, which occur within the status for this action.

- AccessFailure (p. 6)
- AuthFailure (p. 7)
- InternalError (p. 8)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTransactionId (p. 11)
- SignatureDoesNotMatch (p. 12)

Examples

The following sections show a sample request and response.

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=GetTransactionStatus
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Signature=2l60qD6%2BDIfVEN7ZiHM0AcUKACZt0GYKFtIryqkCb6g%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-06T09%3A12%3A06.921Z
&TransactionId=14GKE3B85HCMF1BTSH5C4PD2IHZL95RJ2LM
&Version=2008-09-17
```

Sample Query Request

```
GET\n
fps.sandbox.amazonaws.com\n
Action=GetTransactionStatus
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&Signature=2l60qD6%2BDIfVEN7ZiHM0AcUKACZt0GYKFtIryqkCb6g%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-06T09%3A12%3A06.921Z
&TransactionId=14GKE3B85HCMF1BTSH5C4PD2IHZL95RJ2LM
&Version=2008-09-17
```

Sample Soap Request

```
<SOAP-ENV:Body
  wsu:Id="body"
  xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:GetTransactionStatus
    xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
     <ns2:TransactionId>
      13N9ZL42F2SJLLIGH7RB6Q8IO8BTM62LGI3
     </ns2:TransactionId>
   </ns2:GetTransactionStatus>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<GetTransactionStatusResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <GetTransactionStatusResult>
    <TransactionId>14GKE3B85HCMF1BTSH5C4PD2IHZL95RJ2LM</TransactionId>
    <TransactionStatus>Success</TransactionStatus>
    <CallerReference>CallerReference07</CallerReference>
    <StatusCode>Success</StatusCode>
    <StatusMessage>
      The transaction was successful and the payment instrument was charged.
    </StatusMessage>
  </GetTransactionStatusResult>
  <ResponseMetadata>
    <RequestId>13279842-6f84-41ef-ae36-c1ededaf278d:0</RequestId>
  </ResponseMetadata>
</GetTransactionStatusResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
    MESSAGE123
  </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
  </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
    GetTransactionStatus:Response
  </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
    urn:uuid:10d09e74-ba0a-4b3b-9eea-f873e589f496
  </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <GetTransactionStatusResponse
    xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
```

```
    <GetTransactionStatusResult>
      <TransactionId>13N9ZL42F2SJLLIGH7RB6Q8IO8BTM62LGI3</TransactionId>
      <TransactionStatus>Success</TransactionStatus>
      <CallerReference>ReferenceString????6geW12275895867941</CallerReference>
      <StatusCode>Success</StatusCode>
      <StatusMessage>message</StatusMessage>
    </GetTransactionStatusResult>
    <ResponseMetadata>
      <RequestId>10d09e74-ba0a-4b3b-9eea-f873e589f496:0</RequestId>
    </ResponseMetadata>
  </GetTransactionStatusResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Pay

Description

Non-aggregated applications The `Pay` action initiates a transaction to move funds from a sender to a recipient. The *`SenderTokenId`*, obtained from a Co-Branded service request, specifies the payment instrument the sender chose to execute the transaction. If the payment method specified is Amazon account balance transfer (ABT), the transaction completes synchronously. If the payment method is a bank account (ACH) or a credit card (CC), the transaction completes asynchronously.

Aggregated applications The `Pay` action initiates a transaction to move funds from a sender to a recipient. The *`SenderTokenId`*, which you obtained earlier from the Co-Branded service response, represents the prepaid payment token that references the prepaid instrument ID, or the postpaid payment token that references the postpaid credit instrument.

Marketplace applications The marketplace implementation of `Pay` includes the recipient token ID, which identifies the recipient. You get this in the response from a marketplace Co-Branded service request (which you make when the recipient signs up on your website for your marketplace services). The recipient token ID returned identifies the recipient and is required when you later move money from the sender to the recipient.

In addition, for marketplace applications, the `Pay` parameters also specify the marketplace fee and who is charged (the caller or recipient). The marketplace fee is typically the fee you charge the recipient for the service of hosting the recipient's e-commerce store. The fee can be charged on a per-transaction basis and consist of a flat fee, a percentage of the transaction, or a combination of the two.

This action appears in the Basic, Advanced, Aggregated and Marketplace Quick Starts.

Request Parameters

Parameter	Description	Required
`CallerDescription`	Description of this transaction for the caller. Type: String Default: None Constraint: Max size = 160 characters	No

Parameter	Description	Required
CallerReference	A value you provide that uniquely identifies the request. Type: String Default: None Constraint: Max size = 128 characters	Yes
ChargeFeeTo	**Advanced and Marketplace Quick Starts** Specifies the participant paying the Amazon FPS fee in the transaction. The participant can only be a recipient or a caller. The following rules apply for specifying this parameter. • If you are playing the role of a recipient and a caller, then set the value of this parameter to *Recipient*. • If you are playing the role of caller and facilitating the transaction between a sender and a recipient, where the recipient pays the fee, then the fee is collected from the funds that are received from the sender. • If you (caller) are paying the fees, then the fee is collected from your account balance. Ensure that you have a sufficient account balance to cover for the fees. If your account has an insufficient account balance, Amazon FPS rejects the transaction. • *ChargeFeeTo* must be set to be consistent with the value for the Recipient Token API *recipientPaysFee* parameter set when the recipient signed up for your marketplace services. Otherwise, you will get an error message. Type: String Default: Recipient Valid values: `Recipient` | `Caller`	No
DescriptorPolicy	**Advanced, Basic, and Marketplace Quick Starts** Specifies the entity whose name and contact details would be displayed in the sender's credit card or bank account statement. Type: DescriptorPolicy (p. 114) Default: None	No

Parameter	Description	Required
MarketplaceFixedFee	**Advanced and Marketplace Quick Starts** Specifies the fee charged by the marketplace developer as a fixed amount of the transaction. The *MarketplaceFixedFee* is a separate fee from the Amazon Payments fee, which is paid by the caller or recipient. You can express the fixed fee as an amount, such as 10 to mean $10. If you charge a variable fee per transaction, use the *MarketplaceVariableFee* parameter. **Important** **Advanced and Marketplace Quick Starts** The value for *MarketPlaceFixedFee* must be less than or equal to the amount specified for the recipient token. If not, an *InvalidParams* error is returned with the following messages: "The MarketPlaceFixedFee (*$amount-specified*) specified is greater than the maximum fixed fee (*$amount-agreed*) agreed by the recipient. Type: Amount (p. 113) Default: If both the *MarketplaceFixedFee* and the *MarketplaceVariableFee* are unspecified, then the corresponding maximum values, if any, from the recipient token are used.	No
MarketplaceVariableFee	**Advanced and Marketplace Quick Starts** Specifies the fee charged by the marketplace developer as a percentage of the transaction. The *MarketplaceVariableFee* is a separate fee from the Amazon Payments fee and is paid by the recipient. You can express the variable fee as a decimal, such as 5 to mean 5%. If you charge a fixed amount per transaction, use the *MarketplaceFixedFee* parameter. **Important** **Advanced and Marketplace Quick Starts** The value for *MarketPlaceVariableFee* must be less than or equal to the amount specified for the recipient token. If not, an *InvalidParams* error is returned with the following messages: "The MarketPlaceVariableFee (*$amount-specified*) specified is greater than the maximum variable fee (*$amount-agreed*) agreed by the recipient. Type: Decimal Default: None	No

Parameter	Description	Required
OverrideIPNURL	Specifies the URL to receive the Instant Payment Notification (IPN) for the request. The IPN URL set with this parameter takes precedence over the IPN URL set in your account. Type: String Default: None	No
RecipientTokenId	**Advanced and Marketplace Quick Starts** Specifies the recipient token used in the transaction. You obtain this value in response from the Co-Branded service Recipient Token API). Type: String Default: None Condition: Required for marketplace transactions	Conditional
SenderDescription	Description of this transaction for the sender. If you use dynamic soft descriptors, you must specify a value for the sender description. Type: String Default: None Constraint: Max size = 160 characters Condition: If you use dynamic soft descriptors, you must specify a value for the sender description.	Conditional
SenderTokenId	Specifies the sender token used in the transaction. You obtain this value from the response to the Co-Branded service request. Type: String Default: None	Yes
TransactionAmount	Transaction amount charged to the sender for the purchase of an item or service. To understand how to correctly specify the amount in a REST request, see the example request at the end of this topic. Type: Amount (p. 113) Default: None	Yes
TransactionTimeoutInMins	Specifies the number of minutes before the request times out. Use this parameter to specify a timeout value that is acceptable for your business. If Amazon FPS cannot complete the transaction in the time allotted, the transaction is marked as failed and you receive an IPN notification (if you are using IPN). Type: Integer (number of minutes) Default: 10080 (seven days)	No

You must also include parameters that are common to all requests. The common parameters are defaulted in SOAP calls but must be explicitly added in REST calls. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
`TransactionId`	Unique ID generated by Amazon FPS for this transaction. This element is returned if the transaction was accepted by Amazon FPS. If the transaction is a Refund request, this parameter will contain the id of the Refund transaction only. Type: String Size: 35 Bytes
`TransactionStatus`	Provides the status of the transaction. Use this to determine if the transaction has completed, failed, or has not completed yet. Type: TransactionStatus (p. 121)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Pay careful attention to all of the response elements listed in the preceding table, especially the response status element which indicates success or failure for the `Pay` operation. Errors are returned only for REST. For SOAP, an error results in a SOAP fault. If the response status is failure, the `Errors` element includes an error code that identifies the source of the failure. If the response status is success, the elements listed in the preceding table are returned.

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountLimitsExceeded (p. 7)
- AmountOutOfRange (p. 7)
- AuthFailure (p. 7)
- BadRule (Advanced Quick Start)
- DuplicateRequest (p. 7)
- IncompatibleTokens (p. 8)
- InstrumentAccessDenied (p. 8) (Aggregated Quick Start)
- InstrumentExpired (p. 8) (Aggregated Quick Start)
- InsufficientBalance (p. 8)
- InternalError (p. 8)
- InvalidAccountState_Caller (p. 9)
- InvalidAccountState_Recipient (p. 9)
- InvalidAccountState_Sender (p. 9)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTokenId_Recipient (p. 10) (Advanced and Marketplace Quick Starts)
- InvalidTokenId_Sender (p. 10)
- NotMarketplaceApp (p. 11) (Advanced and Marketplace Quick Starts)
- PaymentMethodNotDefined (p. 11) (Advanced Quick Starts)
- SameSenderAndRecipient (p. 11)
- SameTokenIdUsedMultipleTimes (p. 11) (Advanced and Marketplace Quick Starts)
- SignatureDoesNotMatch (p. 12)
- TokenNotActive_Recipient (p. 12) (Advanced and Marketplace Quick Starts)
- TokenNotActive_Sender (p. 12)
- TokenUsageError (p. 13) (Advanced and Aggregated Quick Starts)
- TransactionDenied (p. 13)
- UnverifiedAccount_Recipient (p. 13)
- UnverifiedAccount_Sender (p. 13)
- UnverifiedBankAccount (p. 13)
- UnverifiedEmailAddress_Caller (p. 13)
- UnverifiedEmailAddress_Recipient (p. 14)
- UnverifiedEmailAddress_Sender (p. 14)

Examples

Sample REST Request for non-Marketplace Applications

```
https://fps.sandbox.amazonaws.com?
Action=Pay
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=MyWish
&CallerReference=CallerReference02
&SenderTokenId=553ILMLCG6Z8J431H7BX3UMN3FFQU8VSNTSRNCTAASDJNX66LNZLKSZU3PI7TXIH
&Signature=0AgvXMwJmLxwdMaiE7lMHZxc6384h%2FjBkiTserQFpBQ%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-06T05%3A49%3A52.843Z
&TransactionAmount.CurrencyCode=USD
&TransactionAmount.Value=1
&Version=2008-09-17
```

Sample REST Request for Marketplace Applications

```
https://fps.sandbox.amazonaws.com?
Action=Pay
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=MyWish
&CallerReference=CallerReference02
&RecipientTokenId=254656Example83987
&SenderTokenId=553ILMLCG6Z8J431H7BX3UMN3FFQU8VSNTSRNCTAASDJNX66LNZLKSZU3PI7TXIH
&Signature=0AgvXMwJmLxwdMaiE7lMHZxc6384h%2FjBkiTserQFpBQ%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-06T05%3A49%3A52.843Z
&TransactionAmount.CurrencyCode=USD
&TransactionAmount.Value=1
&Version=2008-09-17
```

Sample SOAP Request for non-Marketplace Applications

```
<SOAP-ENV:Body wsu:Id="body"
  xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:Pay xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
   <ns2:SenderTokenId>76PSX31MM77T81ExampleQVDNQPG5GFAK</ns2:SenderTokenId>
     <ns2:TransactionAmount>
       <ns2:CurrencyCode>USD</ns2:CurrencyCode>
       <ns2:Amount>1.1</ns2:Amount>
     </ns2:TransactionAmount>
     <ns2:CallerReference>
      ReferenceString????rpXe12275876325471
     </ns2:CallerReference>
     <ns2:CallerDescription>
      DescriptionString-????i86x12275876325471
     </ns2:CallerDescription>
```

```
     <ns2:SenderDescription>
      DescriptionString-????0m6112275876325471
     </ns2:SenderDescription>
   </ns2:Pay>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample SOAP Request for Marketplace Applications

```
<SOAP-ENV:Body wsu:Id="body"
  xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:Pay xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
   <ns2:SenderTokenId>76PSX31MM77T81ExampleQVDNQPG5GFAK</ns2:SenderTokenId>
   <ns2:RecipientTokenId>6SC9UJ1VJEExampleBTBNUNEYUBJM1K</ns2:RecipientTokenId>
     <ns2:TransactionAmount>
       <ns2:CurrencyCode>USD</ns2:CurrencyCode>
       <ns2:Amount>1.1</ns2:Amount>
     </ns2:TransactionAmount>
     <ns2:CallerReference>
      ReferenceString????rpXe12275876325471
     </ns2:CallerReference>
     <ns2:CallerDescription>
      DescriptionString-????i86x12275876325471
     </ns2:CallerDescription>
     <ns2:SenderDescription>
      DescriptionString-????0m6112275876325471
     </ns2:SenderDescription>
   </ns2:Pay>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<PayResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
   <PayResult>
      <TransactionId>14GK6BGKA7U6OU6SUTNLBI5SBBV9PGDJ6UL</TransactionId>
      <TransactionStatus>Pending</TransactionStatus>
   </PayResult>
   <ResponseMetadata>
      <RequestId>c21e7735-9c08-4cd8-99bf-535a848c79b4:0</RequestId>
   </ResponseMetadata>
</PayResponse>
```

Sample Response to SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
```

```
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
    MESSAGE123
   </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
    Pay:Response
   </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
    urn:uuid:b415f09d-5924-4315-b31a-21c977c85c39
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <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>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample IPN Pending Notification to Rest Request

```
--------------------------
transactionId: 14GK6BGKA7U6OU6SUTNLBI5SBBV9PGDJ6UL
statusMessage: The transaction is awaiting a response from the backend payment
processor.
transactionDate: 1254808208
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference02
transactionAmount: USD 1.00
transactionStatus: PENDING
operation: PAY
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: uhP7uiCAvF/wTpRg6U279KTGPU2QHt23WiwNIB43i4ni1AEZOmBCTa3tUh1ugwxvIMSRASB
hiG0u
rUl22IAXbt1iXfYprM2VrS0W0/W23BpkxInuNeAQWKu4W5/uuOJ1gVqyXsmxdFqJM7KKOh3IuUdC
wSfvPooR2qDQ2r5H/HjcOHfWQZk+BknX1w+aYpBRTa/mTYVxI6yq39mRyYPyMmh8r+tIPDevfnV1
B7sRljhXkJZh6rHJEi7CHq4oqbf8HZ38xaaqyggWy310SmMOuY3YcxNng0TOdbkgNAozMIQgfOsL
4yxiyVIZZJEKFPgT/OdebCZkR/raY1JeuBdYOg==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: PendingNetworkResponse
--------------------------
```

Sample IPN Success Notification to Rest Request

```
--------------------------
transactionId: 14GK6BGKA7U6OU6SUTNLBI5SBBV9PGDJ6UL
statusMessage: The transaction was successful and the payment instrument was
charged.
transactionDate: 1254808208
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference02
transactionAmount: USD 1.00
transactionStatus: SUCCESS
operation: PAY
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: yuYUR4IkONbOfrerafrzC6raA90suk+jKXCgaV1LY0DxieYCAG2tAf9S7Rt231kzr0mhM
MOIH0oe
ocHId3zdXp+2VaUbE4qGjPGfImpaBVxtxVwcdQP6cSFnvnKAbPbmQMdeIHMlgDeqVdtu5BO5skwj
e6bkDs+b8TQ3pHBYmXDc69aHceGqWAjMujs6m4HH3Othlb5Rj54s1IedwTi63HyQo+IAyRWvGPTn
nT6YlV0ajG38GCPoS9Wqa+UKcIr0sLoPY0y2StCDyjYHz7iVx+6lzG1eeCmZ++rAKU8swwhBiWGZ
56ajlKTzhoIJnK5yk7jFYreRt+Ff0W2fEnvEyQ==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: Success
--------------------------
```

Related Actions

- Refund (p. 80)
- Reserve (p. 86) **Basic and Marketplace Quick Starts**
- Settle (p. 94) **Basic and Marketplace Quick Starts**

Refund

Description

You use `Refund` to refund a successfully completed payment transaction. You can refund less than the amount paid. The default, however, is to refund the full amount to the sender.

Only the caller of the original transaction can perform a refund.

This action appears in the Basic, Advanced, Aggregated and Marketplace Quick Starts, as well as in the Amazon Simple Pay Advanced Guide.

Request Parameters

Parameter	Description	Required
CallerDescription	Description of this transaction for the caller. Type: String Default: None Constraint: Max size = 160 characters	No
CallerReference	A value you provide that uniquely identifies the request. Type: String Default: None Constraint: Max size = 128 characters	Yes
OverrideIPNURL	Specifies the URL to receive the Instant Payment Notification (IPN) for the request. The IPN URL set with this parameter takes precedence over the IPN URL set in your account. Type: String Default: None	No
RefundAmount	Specifies the amount to be refunded. To understand how to correctly specify the amount in a REST request, see the example request at the end of this topic. Type: Amount (p. 113) Default: Original transaction amount or any amount remaining Constraint: The total refund amount cannot exceed the original transaction amount.	No
TransactionId	Transaction ID of the transaction to be refunded. Type: String Default: None Constraint: Max size = 35 characters	Yes
MarketplaceRefundPolicy	**Basic and Marketplace Quick Starts** Specifies the refund choice from the `MarketplaceRefundPolicy` enumeration: • MasterTxnOnly • MarketplaceTxnOnly • MasterAndMarketplaceTxn The marketplace developer can refund the master transaction, the marketplace fee, or both. The Marketplace Fee is a separate fee from the Amazon Payments fee and is paid by the recipient. Type: Enumeration Default: MasterTxnOnly	No

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
`TransactionId`	This is the ID (max size = 35 characters) of the transaction named in the request. Type: String Size: 35 Bytes
`TransactionStatus`	Provides the status of the transaction. Type: TransactionStatus (p. 121)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AmountOutOfRange (p. 7)
- AuthFailure (p. 7)
- ConcurrentModification (p. 7)
- DuplicateRequest (p. 7)
- InternalError (p. 8)
- InvalidAccountState_Caller (p. 9)
- InvalidAccountState_Recipient (p. 9)
- InvalidAccountState_Sender (p. 9)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTransactionId (p. 11)
- OriginalTransactionFailed (p. 11)
- OriginalTransactionIncomplete (p. 11)
- RefundAmountExceeded (p. 11)
- SignatureDoesNotMatch (p. 12)
- TransactionDenied (p. 13)
- TransactionFullyRefundedAlready (p. 13)
- TransactionTypeNotRefundable (p. 13)
- UnverifiedEmailAddress_Caller (p. 13)
- UnverifiedEmailAddress_Sender (p. 14)

Examples

Sample REST Request (Basic Quick Start)

```
https://fps.sandbox.amazonaws.com?
Action=Refund
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=MyWish
&CallerReference=CallerReference03
&RefundAmount.CurrencyCode=USD
&RefundAmount.Value=1
&Signature=V6pU3PvDPkPhR9Eu7yZXnFZHuEFafLE5sBPgqqCELEU%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
```

```
&Timestamp=2009-10-06T05%3A51%3A49.578Z
&TransactionId=14GK4TNCAQ84NK9VITEHKAS94RAD9ZE2AQD
&Version=2008-09-17
```

Sample REST Request (Advanced, Aggregated, and Marketplace Quick Starts)

```
https://fps.sandbox.amazonaws.com?
Action=Refund
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=MyWish
&CallerReference=CallerReference03
&RefundAmount.CurrencyCode=USD
&RefundAmount.Value=1
&Signature=V6pU3PvDPkPhR9Eu7yZXnFZHuEFafLE5sBPgqqCELEU%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-06T05%3A51%3A49.578Z
&TransactionId=14GK4TNCAQ84NK9VITEHKAS94RAD9ZE2AQD
&Version=2008-09-17
```

Sample SOAP Request (Basic Quick Start)

```
<SOAP-ENV:Body wsu:Id="body"
  xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:Pay xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
   <ns2:SenderTokenId>76PSX31MM77T81ExampleQVDNQPG5GFAK</ns2:SenderTokenId>
     <ns2:TransactionAmount>
       <ns2:CurrencyCode>USD</ns2:CurrencyCode>
       <ns2:Amount>1.1</ns2:Amount>
     </ns2:TransactionAmount>
     <ns2:CallerReference>
      ReferenceString????rpXe12275876325471
     </ns2:CallerReference>
     <ns2:CallerDescription>
      DescriptionString-????i86x12275876325471
     </ns2:CallerDescription>
     <ns2:SenderDescription>
      DescriptionString-????0m6112275876325471
     </ns2:SenderDescription>
   </ns2:Pay>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample SOAP Request (Advanced, Aggregated, and Marketplace Quick Starts)

```
<SOAP-ENV:Body wsu:Id="body"
  xmlns:wsu=
```

```
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:Pay xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
   <ns2:SenderTokenId>76PSX31MM77T81ExampleQVDNQPG5GFAK</ns2:SenderTokenId>
   <ns2:RecipientTokenId>6SC9UJ1VJEExampleBTBNUNEYUBJM1K</ns2:RecipientTokenId>
     <ns2:TransactionAmount>
       <ns2:CurrencyCode>USD</ns2:CurrencyCode>
       <ns2:Amount>1.1</ns2:Amount>
     </ns2:TransactionAmount>
     <ns2:CallerReference>
      ReferenceString????rpXe12275876325471
     </ns2:CallerReference>
     <ns2:CallerDescription>
      DescriptionString-????i86x12275876325471
     </ns2:CallerDescription>
     <ns2:SenderDescription>
      DescriptionString-????0m6112275876325471
     </ns2:SenderDescription>
   </ns2:Pay>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<RefundResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <RefundResult>
    <TransactionId>14GK6F2QU755ODS27SGHEURLKPG72Z54KMF</TransactionId>
    <TransactionStatus>Pending</TransactionStatus>
  </RefundResult>
  <ResponseMetadata>
    <RequestId>1a146b9a-b37b-4f5f-bda6-012a5b9e45c3:0</RequestId>
  </ResponseMetadata>
</RefundResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
      MESSAGE123
    </wsa:RelatesTo>
    <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
      http://www.w3.org/2005/08/addressing/anonymous
    </wsa:To>
    <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
      Refund:Response
    </wsa:Action>
    <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
      urn:uuid:6d63adbb-611e-40ee-9262-a29c30e8ecaa
    </wsa:MessageID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <RefundResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
```

```
      <RefundResult>
        <TransactionId>13N91G4R7478C8ZLHEF93JLIGROQH2VQJSM</TransactionId>
        <TransactionStatus>Success</TransactionStatus>
      </RefundResult>
      <ResponseMetadata>
        <RequestId>6d63adbb-611e-40ee-9262-a29c30e8ecaa:0</RequestId>
      </ResponseMetadata>
    </RefundResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample IPN Pending Notification to Rest Request

```
--------------------------
transactionId: 14GK6F2QU755ODS27SGHEURLKPG72Z54KMF
statusMessage: The transaction is awaiting a response from the backend payment
processor.
transactionDate: 1254808324
signatureVersion: 2
signatureMethod: RSA-SHA1
parentTransactionId: 14GK4TNCAQ84NK9VITEHKAS94RAD9ZE2AQD
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference03
transactionAmount: USD 1.00
transactionStatus: PENDING
operation: REFUND
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: mzis1HbeiiLx5j8nrUR3UeIVz3bcxVDG82JOW0gIEXO1FXxBVZHwPPBFCEVcyBMu8wtNT
Mph/ylu
okjBi8w9Q6shMswBteq9bwNQA9qbDRT256ckoqdwfCf0910lYVj+wNSKkezF6Clptjgsn0wMjMQO
D9QBuOAAA9qV6VnUorRumPZ1psY/17FUvDwKVUMPEkZNO1mn7lcLFZJJp1aMkIj+RmraafTUUM62
U0VMYKSR5pDEp0ifThn0Za4DogV0ZoGJrB/+gPhA07FdtnkM4uG5jgwqOCVyOA4ayP7uJpb7oImj
8Jhi60+EWUUbbUShTEsjTxqQtM8UKvsM6XAjdA==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: PendingNetworkResponse
--------------------------
```

Sample IPN Success Notification to Rest Request

```
--------------------------
transactionId: 14GK6F2QU755ODS27SGHEURLKPG72Z54KMF
statusMessage: The transaction was successful and the payment instrument was
charged.
transactionDate: 1254808324
signatureVersion: 2
signatureMethod: RSA-SHA1
parentTransactionId: 14GK4TNCAQ84NK9VITEHKAS94RAD9ZE2AQD
buyerEmail: new_premium@amazon.com
```

```
notificationType: TransactionStatus
callerReference: CallerReference03
transactionAmount: USD 1.00
transactionStatus: SUCCESS
operation: REFUND
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: sDq9YvW7L29W2NSIC/wjC5yLyR4QJSQyt/7iHhNiEdwFoGVkrLjJHiBloPfJxzznHnmMtCR
sUQ+A
d3tZ0NdemMxf0qYM9NX93PyG0KBKXShKeM0Da39cvnC05tZmtxpfCuZT5ECRydr+BqRo/DOlx1Yg
93gihZ83qHWR8bpqQcBwsu7vD4c4m4mTZ4I75gw+NXKRDD+vCPFDNEKRnh5kQz+Tjjg4bnNYEEcG
Rf6UZfS2lvMzdj0c37RUY6t4gQ3W3Z9G/REGjC98JBuTimk/kc1HoSc+xe6WtAH/siNurisyqgoB
HWnQM8iRqLEHj/m9y6vx5EBHBokD1BJMIiiZNg==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: Success
--------------------------
```

Related Actions

- Pay (p. 72)
- Reserve (p. 86) **Advanced, Basic and Marketplace Quick Starts**
- Settle (p. 94) **Advanced, Basic and Marketplace Quick Starts**

Reserve

Description

The `Reserve` operation reserves the total price of a purchase against the sender's payment instrument. To charge the payment instrument, you must subsequently issue a `Settle` request. A reserve authorization is only valid for 7 days. After that, Amazon FPS automatically cancels the transaction and notifies you.

Note
You can settle a reserved transaction only once.

The marketplace implementation of `Reserve` includes the recipient token ID, which identifies the recipient. You get this in the response from a marketplace Co-Branded service request (which you make when the recipient signs up on your website for your marketplace services). The recipient token ID returned identifies the recipient and is required when you later move money from the sender to the recipient.

The `Reserve` parameters also specify the marketplace fee and who is charged for it (the caller or recipient). The marketplace fee is typically the fee charged by the caller to the recipient for the service of hosting the recipient's e-commerce store. The fee can be charged on a per-transaction basis and consists of a flat fee, a percentage of the transaction, or a combination of the two.

To cancel a reserved payment, send a `Cancel` request.

This action appears in the Basic, Advanced, and Aggregated Quick Starts.

Request Parameters

Parameter	Description	Required
CallerDescription	Description of this transaction for the caller. Type: String Default: None Constraint: Max size = 160 characters Condition: If you use dynamic, soft descriptors, you must supply a caller description. For more information, see *DescriptorPolicy*.	No
CallerReference	A value you provide that uniquely identifies the request. Type: String Default: None Constraint: Max size = 128 characters	Yes
ChargeFeeTo	**Marketplace Quick Start** Specifies the participant paying the Amazon FPS fee in the transaction. The participant can only be a recipient or a caller. The following rules apply for specifying this parameter. • If you are playing the role of a recipient and a caller, then set the value of this parameter to *recipient*. • If you are playing the role of caller and facilitating the transaction between a sender and a recipient, where the recipient pays the fee, then the fee is collected from the funds that are received from the sender. • If you (caller) are paying the fees, then the fee is collected from your account balance. Ensure that you have a sufficient account balance to cover for the fees. If your account has an insufficient account balance, Amazon FPS rejects the transaction. Type: String Default: None Valid values: `Recipient` | `Caller`	No
DescriptorPolicy	Specifies the entity whose name and contact details would be displayed in the sender's credit card or bank account statement. Type: Descriptor Policy (p. 114) Default: None	No

Parameter	Description	Required
MarketplaceFixedFee	**Marketplace Quick Start** Specifies the fee charged by the marketplace developer as a fixed amount of the transaction. The *MarketplaceFixedFee* is a separate fee from the Amazon Payments fee, which is paid by the caller or recipient. You can express the fixed fee as an amount, such as 10 to mean $10. If you charge a variable fee per transaction, use the *MarketplaceVariableFee* parameter. Type: Amount (p. 113) Default: If both the *MarketplaceFixedFee* and the *MarketplaceVariableFee* are unspecified, then the corresponding maximum values, if any, from the recipient token are used.	No
MarketplaceVariableFee	**Marketplace Quick Start** Specifies the fee charged by the marketplace developer as a percentage of the transaction. The *MarketplaceVariableFee* is a separate fee from the Amazon Payments fee and is paid by the recipient. You can express the variable fee as a decimal, such as 5 to mean 5%. If you charge a fixed amount per transaction, use the *MarketplaceFixedFee* parameter. Type: Decimal Default: None	No
OverrideIPNURL	Specifies the URL to receive the Instant Payment Notification (IPN) for the request. The IPN URL set with this parameter takes precedence over the IPN URL set in your account. Type: String Default: None	No
RecipientTokenId	**Marketplace Quick Start** Specifies the recipient token used in the transaction. You obtain this value in response from the Co-Branded service Recipient Token API. Type: String Default: None	Yes
SenderDescription	Description of this transaction for the sender. If you use dynamic soft descriptors, you must specify a value for the sender description. Type: String Default: None Constraint: Max size = 160 characters Condition: If you use dynamic soft descriptors, you must specify a value for the sender description. For more information, see *DescriptorPolicy*.	Conditional

Parameter	Description	Required
SenderTokenId	Specifies the sender token to be used for this transaction. You obtain this value in a Co-Branded service response. Type: String Default: None	Yes
TransactionAmount	Transaction amount charged to the sender. To understand how to correctly specify the amount in a REST request, see the example request at the end of this topic. Type: Amount (p. 113) Default: None	Yes
TransactionTimeoutInMins	**Basic, Advanced, and Marketplace Quick Starts**Specifies the number of minutes before the request times out. Use this parameter to specify a timeout value that is acceptable for your business. If Amazon FPS cannot complete the transaction in the time allotted, the transaction is marked as failed and you receive an IPN notification (if you are using IPN). Type: Integer (number of minutes) Default: 10080 (seven days)	No

You must also include parameters that are common to all requests. The common parameters are defaulted in SOAP calls but must be explicitly added in REST calls. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
TransactionId	Unique ID generated by Amazon FPS for this transaction. This element is returned if the transaction was accepted by Amazon FPS. If the transaction is a Refund request, this parameter will contain the id of the Refund transaction only. Type: String Size: 35 Bytes
TransactionStatus	Provides the status of the transaction. Type: TransactionStatus (p. 121)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountLimitsExceeded (p. 7)
- AmountOutOfRange (p. 7)
- AuthFailure (p. 7)
- DuplicateRequest (p. 7)
- IncompatibleTokens (p. 8)
- InternalError (p. 8)
- InvalidAccountState_Caller (p. 9)
- InvalidAccountState_Recipient (p. 9)
- InvalidAccountState_Sender (p. 9)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidPaymentMethod (p. 10)
- InvalidRecipientForCCTransaction (p. 10)
- InvalidTokenId_Sender (p. 10)
- InvalidTokenId_Recipient (p. 10) (Basic Quick Start)
- NotMarketplaceApp (p. 11) (Marketplace Quick Start)
- PaymentInstrumentNotCC (p. 11)
- SameSenderAndRecipient (p. 11) (Marketplace Quick Start)
- SameTokenIdUsedMultipleTimes (p. 11) (Marketplace Quick Start)
- SignatureDoesNotMatch (p. 12)
- TokenNotActive_Recipient (p. 12)
- TokenNotActive_Sender (p. 12)
- TransactionDenied (p. 13)
- UnverifiedAccount_Recipient (p. 13)
- UnverifiedAccount_Sender (p. 13)
- UnverifiedEmailAddress_Caller (p. 13)
- UnverifiedEmailAddress_Recipient (p. 14)
- UnverifiedEmailAddress_Sender (p. 14)

Examples

Sample REST Request for a Marketplace Application

```
https://fps.sandbox.amazonaws.com?
Action=Reserve
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=Reserve
&CallerReference=CallerReference05
&RecipientTokenId=254656Example83987
&SenderTokenId=553IPMACGAZ2J4N1L7BJ3UMNRFTQU4V9NT4RJCTVADDJKXQ6L1ZAKSIUNPIRTTI1
&Signature=JZ0eeVTM5LwbvziLdA%2FSMve7mgrEoTvTGZJ%2BpsgZkM0%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-06T07%3A51%3A04.140Z
&TransactionAmount.CurrencyCode=USD
&TransactionAmount.Value=1
&Version=2008-09-17
```

Sample REST Request for non-Marketplace Applications

```
https://fps.sandbox.amazonaws.com?
Action=Reserve
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=Reserve
&CallerReference=CallerReference05
&SenderTokenId=553IPMACGAZ2J4N1L7BJ3UMNRFTQU4V9NT4RJCTVADDJKXQ6L1ZAKSIUNPIRTTI1
&Signature=JZ0eeVTM5LwbvziLdA%2FSMve7mgrEoTvTGZJ%2BpsgZkM0%3D
&SignatureMethod=HmacSHA256
```

```
&SignatureVersion=2
&Timestamp=2009-10-06T07%3A51%3A04.140Z
&TransactionAmount.CurrencyCode=USD
&TransactionAmount.Value=1
&Version=2008-09-17
```

Sample SOAP Request for a Marketplace Application

```
<SOAP-ENV:Body wsu:Id="body" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:Reserve xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
   <ns2:SenderTokenId>76PSX31MM77T81ExampleQVDNQPG5GFAK</ns2:SenderTokenId>
   <ns2:RecipientTokenId>6SC9UJ1VJEExampleBTBNUNEYUBJM1K</ns2:RecipientTokenId>
     <ns2:TransactionAmount>
       <ns2:CurrencyCode>USD</ns2:CurrencyCode>
       <ns2:Amount>1.10</ns2:Amount>
     </ns2:TransactionAmount>
     <ns2:CallerReference>
      ReferenceString????Qlrd12275864150791
     </ns2:CallerReference>
     <ns2:CallerDescription>
      DescriptionString-????UQAu12275864150791
     </ns2:CallerDescription>
     <ns2:SenderDescription>
      DescriptionString-????kbPT12275864150791
     </ns2:SenderDescription>
   </ns2:Reserve>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample SOAP Request for non-Marketplace Applications

```
<SOAP-ENV:Body wsu:Id="body" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:Reserve xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
   <ns2:SenderTokenId>76PSX31MM77T81ExampleQVDNQPG5GFAK</ns2:SenderTokenId>
     <ns2:TransactionAmount>
       <ns2:CurrencyCode>USD</ns2:CurrencyCode>
       <ns2:Amount>1.10</ns2:Amount>
     </ns2:TransactionAmount>
     <ns2:CallerReference>
      ReferenceString????Qlrd12275864150791
     </ns2:CallerReference>
     <ns2:CallerDescription>
      DescriptionString-????UQAu12275864150791
     </ns2:CallerDescription>
     <ns2:SenderDescription>
      DescriptionString-????kbPT12275864150791
     </ns2:SenderDescription>
   </ns2:Reserve>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<ReserveResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <ReserveResult>
    <TransactionId>14GKD9GE66FAA63E6O6B2JDPZKN53LZ7F22</TransactionId>
    <TransactionStatus>Pending</TransactionStatus>
  </ReserveResult>
  <ResponseMetadata>
    <RequestId>d13273fc-fca8-4963-8fbc-66d03e66055f:0</RequestId>
  </ResponseMetadata>
</ReserveResponse>
```

Sample Response to SOAP Request

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
    MESSAGE123
   </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
    Reserve:Response
   </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
    urn:uuid:a9e1fc80-03f6-4e1b-a1c0-541df545afac
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <ReserveResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
     <ReserveResult>
       <TransactionId>13N8TKAK15P3GOIPLP796OKSB66C6K2LBEK</TransactionId>
       <TransactionStatus>Pending</TransactionStatus>
     </ReserveResult>
     <ResponseMetadata>
       <RequestId>a9e1fc80-03f6-4e1b-a1c0-541df545afac:0</RequestId>
     </ResponseMetadata>
   </ReserveResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample IPN Pending Notification to Rest Request

```
--------------------------
transactionId: 14GKD9GE66FAA63E6O6B2JDPZKN53LZ7F22
statusMessage: The transaction is awaiting a response from the backend payment
processor.
transactionDate: 1254815482
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
```

```
notificationType: TransactionStatus
callerReference: CallerReference05
operation: RESERVE
transactionStatus: PENDING
transactionAmount: USD 1.00
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: NvFCZMralNEepynuIhhXJc+jpK1ZMdFLBMcXFv6Vq1jhpdLX/B9T0llu
OUv74I6xgO8L2UemgV4S
ZCejlQZ3glwKnEM75lKVlHx34IKp1RFm1DjQOO5KaYGQUNMu1ouYK1YmQUHCuktdLnTXjkxjn0lv
9U4EyzDe8l/tLp2nlAqRF4J7PIhdTkWvBYNYhZrEy5A895OMf9uFtwX8Eyg4lTDMVwEWJoG8CTxJ
qtcsKabmbF9Blwhfe3f+viTnv39YRDb+PZKnpl/XqkKYdNEXClRy3g6xpF/14FJ4hA+A1UP+A+No
17b6lZuKmd5dbdvqTQKOxEAfR6lL1gTzAYY/8w==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: Reserve
statusCode: PendingNetworkResponse
--------------------------
```

Sample IPN Success Notification to Rest Request

```
--------------------------
transactionId: 14GKD9GE66FAA63E6O6B2JDPZKN53LZ7F22
statusMessage: The requested amount was reserved successfully against the given
payment instrument.
transactionDate: 1254815482
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference05
transactionAmount: USD 1.00
transactionStatus: RESERVED
operation: RESERVE
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: RIVZQHF+NmGUEbZNXijRcSwmeBTcYg/GCZD/xeUpLLXMwDNrM1D0+ewFLiUqJvdbQueUilB
kJPoB
5j+ZYvvrXfldEofaMZ85pz2pA/DyUicWR4e/DgcZrk/B7FO6LL9ki6aE0qPzpRR/nzRcLiu1lH2a
zUPnMVf3dT+SfDhaKyKIfX40QYL6U3m3NTaGYSUbBwzZczg9qTpu4zZ2kCK3uidg7P78sXQEnDhm
8kDAJC4obYFVlZi/Bd8UalxIYf2ko8SkhQ4vbsipjNg++HJ7KlJAa41GTVCrJfeX0Y4r7ToONEaQ
iu/zn8X+q/jPqgGZN+Z2KNls6XVw4Waw3eXbug==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: Reserve
statusCode: Success
--------------------------
```

Related Actions

- Pay (p. 72)

- Refund (p. 80)
- Settle (p. 94)

Settle

Description

The `Settle` action charges the sender's payment instrument for the purchase that was transacted using `Reserve`. You settle a transaction when you fulfill the order, for example, when you ship the purchased items.

This action appears in the Basic, Advanced, and Marketplace Quick Starts, as well as in the Amazon Simple Pay Advanced Guide.

Request Parameters

Parameter	Description	Required
`OverrideIPNURL`	Specifies the URL to receive the Instant Payment Notification (IPN) for the request. The IPN URL set with this parameter takes precedence over the IPN URL set in your account. Type: String Default: None	No
`ReserveTransactionId`	An identifier returned by `Reserve` that identifies the reserved transaction to be settled. Type: String Default: None Constraint: Max size = 35 characters	Yes
`TransactionAmount`	Amount to be settled. To understand how to correctly specify the amount in a REST request, see the example request at the end of this topic. Type: Amount (p. 113) Default: The amount reserved in the `Reserve` request Constraint: The amount cannot exceed the reserved amount.	No

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
`TransactionId`	Identifies the transaction that was settled. Type: String Size: 35 Bytes
`TransactionStatus`	Provides the status of the transaction. Type: TransactionStatus (p. 121)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountClosed (p. 6)
- AmountOutOfRange (p. 7)
- AuthFailure (p. 7)
- ConcurrentModification (p. 7)
- InternalError (p. 8)
- InvalidAccountState_Caller (p. 9)
- InvalidAccountState_Recipient (p. 9)
- InvalidAccountState_Sender (p. 9)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidTransactionId (p. 11)
- InvalidTransactionState (p. 11)
- SettleAmountGreaterThanReserveAmount (p. 12)
- SignatureDoesNotMatch (p. 12)
- TransactionDenied (p. 13)
- UnverifiedAccount_Recipient (p. 13)
- UnverifiedEmailAddress_Caller (p. 13)
- UnverifiedEmailAddress_Recipient (p. 14)
- UnverifiedEmailAddress_Sender (p. 14)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=Settle
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&ReserveTransactionId=14GKD9GE66FAA63E6O6B2JDPZKN53LZ7F22
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Signature=SJJLsIBghi7VIycBjX7c3hnfgZ%2FBvZbzqLtAZXDL8ys%3D
&Timestamp=2009-10-06T07%3A53%3A11.750Z
&TransactionAmount.CurrencyCode=USD
&TransactionAmount.Value=1
&Version=2008-09-17
```

Sample SOAP Request

```
https://fps.amazonaws.com/?
Action=Settle
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&SignatureVersion=2
&SignatureMethod=HmacSHA256
&Timestamp=2008-08-06T13%3A00%3A01Z
&TransactionId=254656Example83987
&Version=2008-09-17
&Signature=<URL-encoded signature value>
```

Sample Response to REST Request

```
<SettleResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <SettleResult>
    <TransactionId>14GKD9GE66FAA63E6O6B2JDPZKN53LZ7F22</TransactionId>
    <TransactionStatus>Pending</TransactionStatus>
  </SettleResult>
  <ResponseMetadata>
    <RequestId>9ed2008b-b230-4ed0-9210-095f77fc2359:0</RequestId>
  </ResponseMetadata>
</SettleResponse>
```

Sample Response to SOAP Request

```
<SettleResponse
  xmlns="https://fps.amazonaws.com/doc/2008-09-17/">
  <SettleResult>
    <TransactionId>
      254656Example83987
    </TransactionId>
    <TransactionStatus>
      Pending
    </TransactionStatus>
  </SettleResult>
  <ResponseMetadata>
    <RequestId>
      a8d5e97c-6a7e-4fe1-b019-58a428a5a68b:0
    </RequestId>
  </ResponseMetadata>
</SettleResponse>
```

Sample IPN Pending Notification to Rest Request

```
--------------------------
transactionId: 14GKD9GE66FAA63E6O6B2JDPZKN53LZ7F22
statusMessage: The transaction is awaiting a response from the backend payment
processor.
transactionDate: 1254815482
signatureVersion: 2
```

```
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference05
transactionAmount: USD 1.00
transactionStatus: PENDING
operation: SETTLE
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: zxymWMlhu4o+2rp
drBXu08EACZ3Mi3Z16x5+8+1Hbqkh4DTr1A6ry4fijBYkl32z4fMF9xnoGriW
2jzij7Vmc/4Vc4dEWCpbOq+be4JLfOELw08jJQintuk3kIXOPca06NMWQhGiC3m7kRF95nM2TJs7
jqbkAMrKyiZArcURMo0YpRZPIF7DlDlNRAebH2+0v0BxaUtombrDFW4UlSscuebXDNdgjp7KjCnT
BJGDJks9/wLKKvFtISQWHuvN2MiPzt7UmFwMLPh8jtpgQ6JxS+ipTPxbr7Km3IXIJJgJHpxmdQmg
ghrl4IX0zCKaVUb7Rh3z85/9F0yPB8A92nquzQ==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: Reserve
statusCode: PendingNetworkResponse
--------------------------
```

Sample IPN Success Notification to Rest Request

```
--------------------------
transactionId: 14GKD9GE66FAA63E6O6B2JDPZKN53LZ7F22
statusMessage: The transaction was successful and the payment instrument was
charged.
transactionDate: 1254815482
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference05
operation: SETTLE
transactionStatus: SUCCESS
transactionAmount: USD 1.00
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: pwozZP+lYONFq39g13ux44vFFMRAt4eJ9kOUWMV2uPCrvBqzi4LFYDQY5UE3VW8OUiW+qp
bukqFz
YNvE+8mh7adhX/qee2U8ZUUNZi6LaM3sKtpPxus2ZJ3wDVPjuO02Obtu1G6Eo79iMi8viX7Dz1LL
8pFTdhspHZb0XDWkuOt2pK2aELa7TOZ/pXXUFLvGrn4MOd6INwbyM2fvnJpIDTcNdzedBO3Rw3vp
2f2GfpFAZJD6Imu57rsr9RsHVUqu2bIhJaAgTRFleVKzMHQJqft5jo6M9N4vKmPfccsuAvoF+rDn
+/6a9VEvTBrVcvAhJ5jrBp3FkXYkOPbHchqHfQ==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: Reserve
statusCode: Success
--------------------------
```

Related Actions

- Pay (p. 72)
- Refund (p. 80)
- Reserve (p. 86)

SettleDebt

Description

This action appears in the Aggregated Quick Start.

The `SettleDebt` action settles the accumulated debt on a postpaid credit instrument using a credit card, bank account, or Amazon Payments account balance. The action decrements the debt balance on the postpaid credit instrument by the settlement amount, and moves money from the sender's payment instrument specified in the postpaid payment token to the recipient's account balance. The transaction fee is deducted from the settlement amount and the remaining amount is deposited into the recipient's account balance. For example, If the debt balance is $100, settlement amount is $90, and the Amazon FPS fee is $2.50, then the `SettleDebt` action transfers $87.50 into the recipient's account balance and decrements the debt balance by $90. The debt balance is now $10.

The `SettleDebt` action can't be used for amounts greater than the actual debt balance. For example, if the debt balance is $100, you can't use `SettleDebt` action for $110.

Settlement tokens can be used multiple times to settle debt (up to its credit limit) and you are charged a fee each time. You specify the credit limit of the token in the Co-Branded service request.

Note
The `Refund` action does not support `SettleDebt`.

Note
The value you provide for the *`SenderTokenId`* is the *`settlementTokenID`* value you received in the Postpaid Token API response, not the value of the *`creditSenderTokenID`* parameter.

Request Parameters

Parameter	Description	Required
`CallerDescription`	Description of this transaction for the caller. Type: String Default: None Constraint: Max size = 160 characters	No
`CallerReference`	A value you provide that uniquely identifies the request. . Type: String Default: None Constraint: Max size = 128 characters	Yes

Parameter	Description	Required
CreditInstrumentId	Specifies the credit instrument to settle. Use the value of the *creditInstrumentID* parameter from the Postpaid Token API response. Type: String Default: None Max Size: 128 Bytes	Yes
DescriptorPolicy	Specifies the entity whose name and contact details would be displayed in the sender's credit card or bank account statement. Type: DescriptorPolicy (p. 114) Default: None	No
OverrideIPNURL	Specifies the URL to receive the Instant Payment Notification (IPN) for the request. The IPN URL set with this parameter takes precedence over the IPN URL set in your account. Type: String Default: None	No
SenderDescription	Description of this transaction for the sender. Type: String Default: None Constraint: Max size = 128 bytes	No
SenderTokenId	The value of the *settlementTokenID* parameter you received in the Prepaid Token API response. Type: String Default: None	Yes
SettlementAmount	Specifies the amount of the settlement. Type: Amount (p. 113) Default: None Valid Values: Equal to or less than the debt balance	Yes
TransactionTimeoutInMins	Specifies the number of minutes before the request times out. Use this parameter to specify a timeout value that is acceptable for your business. If Amazon FPS cannot complete the transaction in the time allotted, the transaction is marked as failed and you receive an IPN notification (if you are using IPN). Type: Integer (number of minutes) Default: 10080	No

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
TransactionId	Unique ID generated by Amazon FPS for this transaction. This element is returned if the transaction was accepted by Amazon FPS. Type: String Size: 35 Bytes
TransactionStatus	Provides the status of the transaction. Type: TransactionStatus (p. 121)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AccountLimitsExceeded (p. 7)
- AmountOutOfRange (p. 7)
- AuthFailure (p. 7)
- DuplicateRequest (p. 7)
- InactiveInstrument (p. 7)
- IncompatibleTokens (p. 8)
- InstrumentAccessDenied (p. 8)
- InsufficientBalance (p. 8)
- InternalError (p. 8)
- InvalidAccountState_Caller (p. 9)
- InvalidAccountState_Recipient (p. 9)
- InvalidAccountState_Sender (p. 9)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidPaymentInstrument (p. 9)
- InvalidPaymentMethod (p. 10)
- InvalidTokenId_Sender (p. 10)
- SettleAmountGreaterThanReserveAmount (p. 12)
- SignatureDoesNotMatch (p. 12)
- TokenNotActive_Sender (p. 12)
- TokenUsageError (p. 13)
- TransactionDenied (p. 13)
- UnverifiedAccount_Recipient (p. 13)
- UnverifiedAccount_Sender (p. 13)
- UnverifiedBankAccount (p. 13)
- UnverifiedEmailAddress_Caller (p. 13)
- UnverifiedEmailAddress_Sender (p. 14)
- UnverifiedEmailAddress_Recipient (p. 14)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=SettleDebt
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=MyWish
&CallerReference=CallerReference13
&CreditInstrumentId=553IBMFCG7Z8J4D1H7BB3HMNPFQQUPVUNT8RHCT1AXDJNXE6LJZ4KSKU2PI1TTIG
&SenderTokenId=543IJMECGZZ3J4K1F7BJ3TMNXFBQU9VXNT7RRCTNAJDJ8X36L1ZRKSUUPPIBTTIK
```

```
&SettlementAmount.CurrencyCode=USD
&SettlementAmount.Value=1
&Signature=pwHqzj8Y1aQ1TFcer9aXCFkawg8PmIRcmOelQWvsvaQ%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A44%3A48.203Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Body wsu:Id="body" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
   <ns2:SettleDebt xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
     <ns2:SenderTokenId>
      J4D62EZ3K8IMB3MR18R15RMKNR7VJVMA3I49EFTQ4Q1PN4XE4BBZ5RRQ7NMKSFV6
     </ns2:SenderTokenId>
     <ns2:CreditInstrumentId>
      J3D6VEG3KHILB3PR58R65ZMKIRQVJHMB3I996FT84P1PP4QE4BBX5RTQCNMGSUVA
     </ns2:CreditInstrumentId>
     <ns2:SettlementAmount>
       <ns2:CurrencyCode>USD</ns2:CurrencyCode>
       <ns2:Value>50</ns2:Value>
     </ns2:SettlementAmount>
     <ns2:CallerReference>
      ReferenceString????gsBq12290565996541
     </ns2:CallerReference>
     <ns2:CallerDescription>
      DescriptionString-????d5ik12290565996541
     </ns2:CallerDescription>
   </ns2:SettleDebt>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<SettleDebtResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <SettleDebtResult>
    <TransactionId>14GND25ZN3B7O49QVHNASTT98UOBN83NC92</TransactionId>
    <TransactionStatus>Pending</TransactionStatus>
  </SettleDebtResult>
  <ResponseMetadata>
    <RequestId>9ddfbffc-f909-4628-b247-36a5ef3fc7f3:0</RequestId>
  </ResponseMetadata>
</SettleDebtResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <SOAP-ENV:Header>
   <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
```

```
    MESSAGE123
   </wsa:RelatesTo>
   <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
    http://www.w3.org/2005/08/addressing/anonymous
   </wsa:To>
   <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
    SettleDebt:Response
   </wsa:Action>
   <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
    urn:uuid:5787db56-61b7-4636-9fd1-7179523ca194
   </wsa:MessageID>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
   <SettleDebtResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
     <SettleDebtResult>
       <TransactionId>13OKNMPO244IU5G38V98NFTP7LPAKPGR73T</TransactionId>
       <TransactionStatus>Pending</TransactionStatus>
     </SettleDebtResult>
     <ResponseMetadata>
       <RequestId>5787db56-61b7-4636-9fd1-7179523ca194:0</RequestId>
     </ResponseMetadata>
   </SettleDebtResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample IPN Pending Notification to Rest Request

```
-------------------------
transactionId: 14GND25ZN3B7O49QVHNASTT98UOBN83NC92
statusMessage: The transaction is awaiting a response from the backend payment
processor.
transactionDate: 1254915904
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference13
transactionAmount: USD 1.00
transactionStatus: PENDING
operation: SETTLE_DEBT
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: gxuJJK/McM6xnFJ6L1QtyNymnDnxk+hIOBIedsnhJcokWsS
WYABM/R9JEMtvXFv8n7wI5rE6wXhX
GIAheElEbirJr/uho40tEmuio9DOwb7CPCVZj+L29feaqhQ74BEZjWoy8+28swMhJ1zaKNjoYJhk
QXuDKyaHv/lQ/A0Oc/3oVwVsgnovn1VVpRO38gX6fNjgEdKpk7dATuURANtKJUoxCHhkX1aBI4uz
XsB/Oym34YPXcsNb0IuIb0tJH+dSwz2WQjx8Sqj+1dBElpPkIHL1ka+R8y+uIL1nZSv1scunUaDN
QjcKgTA4kNnoNdhiZ1B/bYiIONngbvDwbtXp/Q==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: PendingNetworkResponse
-------------------------
```

Sample IPN Success Notification to Rest Request

```
--------------------------
transactionId: 14GND25ZN3B7O49QVHNASTT98UOBN83NC92
statusMessage: The transaction was successful and the payment instrument was
charged.
transactionDate: 1254915904
signatureVersion: 2
signatureMethod: RSA-SHA1
buyerEmail: new_premium@amazon.com
notificationType: TransactionStatus
callerReference: CallerReference13
transactionAmount: USD 1.00
transactionStatus: SUCCESS
operation: SETTLE_DEBT
recipientEmail: test-caller@amazon.com
buyerName: Test Business
signature: IE5CpU+NcpC57JQCUvuw2tnN+3F5ggyekqjrQa1uDUxY71KgJBv7GZHnHvGrF
dh9TEc7BP89nltv
v5xyDxX5l29yLihG8JOk3tOLGmP5UZyyTX64sfRek2PcscImuaCg+h3joSjN17ZDPBr6tBjGozQA
UoimBJXgPODIHbJ53VWPPkh88rrnukYzPkpg9bcfTnSBxHkbNG5Q3mxoH9hRiTKmPNi0fvbpddVK
f8GEwqbU9WUuxHNvky0VfVvByZvlYkkMGUSCpeHQjQiye2uqqJRnmI9o1xJFWeF8FSSz4cdCGLUi
Cz2Mw9+HbucjqNiVjOlaquGbNmso/67aUX4zCQ==
recipientName: Test Business
paymentMethod: CC
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: Success
--------------------------
```

Related Actions

- GetDebtBalance (p. 41)
- WriteOffDebt (p. 106)

VerifySignature

Description

`VerifySignature` enables you to verify the signature included with outbound notifications. A correctly formatted call using VerifySignature returns a positive result when the signature is valid for the response that contained it.

This action is a component of signature version 2. Because of this, you may only use it with responses which have a *`SignatureVersion`* value of *2*. As of 10 February, 2011, Amazon Payments signs all outbound responses with signature version 2. Unsigned outbound responses are no longer supported.

> **Note**
> You sign `VerifySignature` as you would any other Amazon FPS action.

This action appears in all Amazon FPS Quick Starts as well as in the Amazon Simple Pay Advanced Guide.

Request Parameters

Parameter	Description	Required
UrlEndPoint	A required field that contains the appropriate originating endpoint (either the returnUrl or ipnUrl) that received the response. For example, if your web application resides at `http://my-app-website.biz/`, the returnUrl might be `http://my-app-website.biz/amazon/success.php`, and the IPNUrl might be `http://my-app-website.biz/amazon/ipnProcessor.php`. Type: String Default: None Constraint: Cannot be null or empty	Yes
HttpParameters	Concatenated string of all URL-Encoded parameters which were included in the response containing the signature you want to verify. This includes the *certificateUrl*, *signatureVersion*, *signatureMethod* and *signature* parameters. For example, a correctly formatted and URL-encoded string resembles the following: `First%20Name=Joe&Last%20Name=Smith&signatureVersion=2 &signatureMethod=HMACSHA256&certificateUrl=https%253A %252F%252Ffps.amazonaws.com%252Fcert%252Fkey.pem&signature=aoeuAOE123eAUdhf]` **Tip** For validating the returnUrl, you can extract the query string from the returnUrl (excluding the '?' character). For validating the IPNUrl, concatenate the POST parameters. Type: String Default: None Constraint: Cannot be null or empty. In addition, because *VerifySignature* is a component of signature version 2, the value for *signatureVersion* must be *2*.	Yes

You must also use the *Action* parameter as described in Common Request Parameters (p. 5). Parameter names are case sensitive.

Response Elements

Element	Description
VerificationStatus	The result of the verification, either *Success* or *Failure*. Type: VerificationStatus

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- InternalServerError
- InvalidParams (p. 9)

Examples

Sample REST Request

This section shows a sample request.

```
https://fps.sandbox.amazonaws.com/?Action=VerifySignature&UrlEndPoint=h
ttp%3A%2F%2Fexample.com%3A8080%2Fipn.jsp&HttpParameters=expiry
%3D08%252F2015%26signature%3DynDukZ9%252FG77uSJVb5YM0cadwHVwYKPMKO
O3PNvgADbv6VtymgBxeOWEhED6KGHsGSvSJnMWDN%252FZl639AkRe9Ry%252F7zmn9CmiM
%252FZkp1XtshERGTqi2YL10GwQpaH17MQqOX3u1cW4LlyFoLy4celUFBPq1WM2ZJnaNZRJ
IEY%252FvpeVnCVK8VIPdY3HMxPAkNi5zeF2BbqH%252BL2vAWef6vfHkNcJPlOuOl6jP4E
%252B58F24ni%252B9ek%252FQH18O4kw%252FUJ7ZfKwjCCI13%252BcFybpofcKqddq8C
uUJj5Ii7Pdw1fje7ktzHeeNhF0r9siWcYmd4JaxTP3NmLJdHFRq2T%252FgsF3vK9m3gw%2
53D%253D%26signatureVersion%3D2%26signatureMethod%3DRSA-SHA1%26certific
ateUrl%3Dhttps%253A%252F%252Ffps.sandbox.amazonaws.com%252Fcerts%252F09
0909%252FPKICert.pem%26tokenID%3DA5BB3HUNAZFJ5CRXIPH72LIODZUNAUZIVP7UB7
4QNFQDSQ9MN4HPIKISQZWPLJXF%26status%3DSC%26callerReference%3DcallerRefe
renceMultiUse1&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE&Timestamp=2010-02-26
T19%3A48%3A05.000Z&Version=2008-09-17&SignatureVersion=2&SignatureMetho
d=HmacSHA256&Signature=fKRGL42K7nduDA47g6bJCyUyF5ZvkBotXE5jVcgyHvE%3D
```

Sample Query Request

```
GET\n
fps.sandbox.amazonaws.com\n
Action=VerifySignature&UrlEndPoint=http%3A%2F%2Fexample.com
%3A8080%2Fipn.jsp&HttpParameters=expiry%3D08%252F2015%26signature%3Dy
nDukZ9%252FG77uSJVb5YM0cadwHVwYKPMKOO3PNvgADbv6VtymgBxeOWEhED6KGHsGSvSJ
nMWDN%252FZl639AkRe9Ry%252F7zmn9CmiM%252FZkp1XtshERGTqi2YL10GwQpaH17MQq
OX3u1cW4LlyFoLy4celUFBPq1WM2ZJnaNZRJIEY%252FvpeVnCVK8VIPdY3HMxPAkNi5zeF
2BbqH%252BL2vAWef6vfHkNcJPlOuOl6jP4E%252B58F24ni%252B9ek%252FQH18O4kw%2
52FUJ7ZfKwjCCI13%252BcFybpofcKqddq8CuUJj5Ii7Pdw1fje7ktzHeeNhF0r9siWcYmd
4JaxTP3NmLJdHFRq2T%252FgsF3vK9m3gw%253D%253D%26signatureVersion%3D2%26s
ignatureMethod%3DRSA-SHA1%26certificateUrl%3Dhttps%253A%252F%252Ffps.sa
ndbox.amazonaws.com%252Fcerts%252F090909%252FPKICert.pem%26tokenID%3DA5
BB3HUNAZFJ5CRXIPH72LIODZUNAUZIVP7UB74QNFQDSQ9MN4HPIKISQZWPLJXF%26status
%3DSC%26callerReference%3DcallerReferenceMultiUse1&AWSAccessKeyId=AKIAI
OSFODNN7EXAMPLE&Timestamp=2010-02-26T19%3A48%3A05.000Z&Version=2008-09-
17&SignatureVersion=2&SignatureMethod=HmacSHA256&Signature=fKRGL42K7ndu
DA47g6bJCyUyF5ZvkBotXE5jVcgyHvE%3D
```

Sample Response to REST Request

This section shows a sample REST response.

```
<VerifySignatureResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <VerifySignatureResult>
    <VerificationStatus>Success</VerificationStatus>
  </VerifySignatureResult>
  <ResponseMetadata>
    <RequestId>197e2085-1ed7-47a2-93d8-d76b452acc74:0</RequestId>
  </ResponseMetadata>
</VerifySignatureResponse>
```

WriteOffDebt

Description

The `WriteOffDebt` action dismisses some or all of the debt accumulated on a specified postpaid credit instrument. You typically use this action when you want to offer your customers a discount. When you use this action, you settle the debt partially and dismiss the rest. You can also use this action for accounting purposes if you cannot retrieve the credit from the customer.

This action appears in the Aggregated Quick Start.

Request Parameters

Parameter	Description	Required
AdjustmentAmount	The amount to be written off. You can get the maximum amount that can be written off using `GetDebtBalance` on the credit instrument. To learn how to specify the amount correctly in a REST request, see the example request at the end of this topic. Type: Amount (p. 113) Default: None Valid Values: Less than or equal to the current debt balance	Yes
CallerDescription	Description of this transaction for the caller. Type: String Default: None Constraint: Max size = 160 characters	No
CallerReference	A value you provide that uniquely identifies the request. . Type: String Default: None Constraint: Max size= 128 characters	Yes

Parameter	Description	Required
CreditInstrumentId	Specifies the postpaid credit instrument for which the debt balance will be written off. Use the value of the *creditInstrumentID* parameter from the Postpaid Token API response. Type: String Default: None Max Size: 128 Bytes	Yes

For REST requests, you must also include parameters that are common to all requests. These parameters are included by default in SOAP requests. For more information, see Common Request Parameters (p. 5).

Response Elements

Element	Description
TransactionId	Unique ID generated by Amazon FPS for this transaction. This element is returned if the transaction was accepted by Amazon FPS. Type: String Size: 35 Bytes
TransactionStatus	Provides the status of the transaction. Type: TransactionStatus (p. 121)

Responses also include elements common to all responses. For more information, see Common Response Elements (p. 6).

Errors

This action can return the following errors:

- AccessFailure (p. 6)
- AmountOutOfRange (p. 7)
- AuthFailure (p. 7)
- DuplicateRequest (p. 7)
- InactiveInstrument (p. 7)
- InstrumentAccessDenied (p. 8)
- InternalError (p. 8)
- InvalidAccountState_Caller (p. 9)
- InvalidAccountState_Recipient (p. 9)
- InvalidAccountState_Sender (p. 9)
- InvalidClientTokenId (p. 9)
- InvalidParams (p. 9)
- InvalidPaymentInstrument (p. 9)
- InvalidTokenId_Sender (p. 10)
- SettleAmountGreaterThanDebt (p. 12)
- TransactionDenied (p. 13)
- UnverifiedEmailAddress_Caller (p. 13)
- UnverifiedEmailAddress_Sender (p. 14)
- SignatureDoesNotMatch (p. 12)

Examples

Sample REST Request

```
https://fps.sandbox.amazonaws.com?
Action=WriteOffDebt
&AdjustmentAmount.CurrencyCode=USD
&AdjustmentAmount.Value=1
&AWSAccessKeyId=AKIAIOSFODNN7EXAMPLE
&CallerDescription=MyWish
&CallerReference=CallerReference14
&CreditInstrumentId=553IBMFCG7Z8J4D1H7BB3HMNPFQQUPVUNT8RHCT1AXDJNXE6LJZ4KSKU2PI1TTIG
&Signature=5hNBxQ3ozBLj3auHDFMg3Mf41w3C0GIf%2F3r6VCoJP%2B0%3D
&SignatureMethod=HmacSHA256
&SignatureVersion=2
&Timestamp=2009-10-07T11%3A47%3A30.296Z
&Version=2008-09-17
```

Sample SOAP Request

```
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body wsu:Id="body" xmlns:wsu=
"http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
    <ns2:WriteOffDebt xmlns:ns2="http://fps.amazonaws.com/doc/2008-09-17/">
      <ns2:CreditInstrumentId>
        J79851HA56JBX434EJG9SGMYUZDPJPYZAR5BXCF4G59MGUBUMHX5XJ9EJKCZIXN4
      </ns2:CreditInstrumentId>
      <ns2:AdjustmentAmount>
        <ns2:CurrencyCode>USD</ns2:CurrencyCode>
        <ns2:Value>50</ns2:Value>
      </ns2:AdjustmentAmount>
      <ns2:CallerReference>
        ReferenceString????c5ie12290575241881
      </ns2:CallerReference>
      <ns2:CallerDescription>
        DescriptionString-????75gZ12290575241881
      </ns2:CallerDescription>
    </ns2:WriteOffDebt>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample Response to REST Request

```
<WriteOffDebtResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
  <WriteOffDebtResult>
    <TransactionId>14GND73BD1S57OEBM5O1RDLGU65HN51V9JK</TransactionId>
    <TransactionStatus>Success</TransactionStatus>
  </WriteOffDebtResult>
  <ResponseMetadata>
    <RequestId>76c22576-aeda-4d55-859a-20d56efb6068:0</RequestId>
  </ResponseMetadata>
</WriteOffDebtResponse>
```

Sample Response to SOAP Request

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsa:RelatesTo xmlns:wsa="http://www.w3.org/2005/08/addressing">
      MESSAGE123
    </wsa:RelatesTo>
    <wsa:To xmlns:wsa="http://www.w3.org/2005/08/addressing">
      http://www.w3.org/2005/08/addressing/anonymous</wsa:To>
    <wsa:Action xmlns:wsa="http://www.w3.org/2005/08/addressing">
      WriteOffDebt:Response
    </wsa:Action>
    <wsa:MessageID xmlns:wsa="http://www.w3.org/2005/08/addressing">
      urn:uuid:1370e898-8d81-45f0-9b74-20a9682ba531
    </wsa:MessageID>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
  <WriteOffDebtResponse xmlns="http://fps.amazonaws.com/doc/2008-09-17/">
    <WriteOffDebtResult>
      <TransactionId>13OKOJZJ25NM479TDVU4LREENBT72A3PIUR</TransactionId>
      <TransactionStatus>Success</TransactionStatus>
    </WriteOffDebtResult>
    <ResponseMetadata>
      <RequestId>1370e898-8d81-45f0-9b74-20a9682ba531:0</RequestId>
    </ResponseMetadata>
  </WriteOffDebtResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Sample IPN Success Notification to Rest Request

```
--------------------------
transactionId: 14GND73BD1S57OEBM5O1RDLGU65HN51V9JK
statusMessage: The transaction was successful and the payment instrument was
charged.
transactionDate: 1254916066
signatureVersion: 2
signatureMethod: RSA-SHA1
callerReference: CallerReference14
notificationType: TransactionStatus
transactionAmount: USD 1.00
operation: WRITE_OFF_DEBT
transactionStatus: SUCCESS
signature: viN57tez98wxhzMF7catc26ag4tDi
OnA+PBk023J1fIHw5eua4G0t3XGqOOyC0NBqm7zVdtFST7s
9rwPxgaRU4spj2yLd8PX4YH+sknZJAx+RCo1wQa7Yr5R3s9gbbfP//qk5ZDAjF40ygUJmmEMV0o4
gdYOJAp15v5MBFJT1zexluxY9EL/iMYNvJ4n+xyAQvjZvyI5cvW3XCbQ2VGGkOAkodb5+S+rw8My
TE3nCbZKwlEK68VUCP8HRVDssu9lLcYPJNUChZS71Sj97h0S1AyxRCnb5Enunqa9B2SjvNz1cjna
L/tKgNyCArM4EaXQ+Qx8+Bal4CR2c/14dAHeuw==
paymentMethod: ABT
certificateUrl: https://fps.sandbox.amazonaws.com/certs/090909/PKICert.pem
paymentReason: MyWish
statusCode: Success
--------------------------
```

Related Actions

- SettleDebt (p. 98)
- GetDebtBalance (p. 41)

Amazon FPS Data Types

Topics

This section describes the data types used by Amazon FPS. There is a comprehensive listing, as well as listings for simple enumerations and complex datatypes.

Enumeration Data Types

Following are the enumerated data types Amazon FPS uses:

Complex Data Types

Following are the complex data types Amazon FPS uses:

Data Type Descriptions

This section lists all data types Amazon FPS uses.

- TransactionStatus (p. 121)
- TransactionDetail (p. 123)
- TransactionPart (p. 124)

AccountBalance

Name	Description	Type
AvailableBalances	The total amount of money that is transferred to your account from a bank account transfer or a refund.	AvailableBalances (p. 113)
PendingInBalance	The total amount that is yet to be credited to your account.	Amount (p. 113)
PendingOutBalance	The total amount that is yet to be debited from your account.	Amount (p. 113)
TotalBalance	The total balance that is currently available in your account.	Amount (p. 113)

Amount

Name	Description	Type
CurrencyCode	The currency code of the amount. Amazon FPS currently supports only USD.	CurrencyCode (p. 114)
Value	The numeric value of the amount in dollars. Two optional decimal places are allowed. For example, 25.01 is $25.01, and 2500 is $2500.	String

AvailableBalances

Name	Description	Type
DisburseBalance	The total balance that has been disbursed.	Amount (p. 113)
RefundBalance	The total amount that has been refunded.	Amount (p. 113)

ChargeFeeTo

Name	Description	Type
Caller	Caller shall pay the fees.	String
Recipient	Recipient shall pay the fees.	String

CurrencyCode

Name	Description	Type
USD	The transaction uses U.S. dollars.	String

DebtBalance

Name	Description	Type
AvailableBalance	Available debt balance accumulated between recipient and sender.	Amount (p. 113)
PendingOutBalance	Any balance that is pending because of an external instrument was used to settle the debt.	Amount (p. 113)

DescriptorPolicy

Name	Description	Type
CSOwner	The recipient or caller customer service number. If you specify *Caller*, the customer service number for the caller is passed to the payment processor, which is the entity that actually processes payments on the person's credit card or bank account. Otherwise, the default value of *CSOwner* is Recipient.	The entity whose CS Phone number should be used. Valid values are either Recipient or Caller. . Default: Recipient
SoftDescriptorType	The type of soft descriptor. Valid values are either Static or Dynamic. If you specify Static, or do not specify a type, the soft descriptor in your account level setting is sent to the payment processor. If you specify Dynamic, the first 15 characters of sender description is sent to the payment processor.	The type of soft descriptor. Valid values are either Static or Dynamic. Default: Static

FPSOperation

These values are returned for non-IPN operations.

Name	Description	Type
Pay	All pay transactions.	String
Refund	All refund transactions.	String
Settle	All settle transactions.	String
SettleDebt	All debt settlement transactions.	String

Name	Description	Type
WriteOffDebt	All debt write-off transactions.	String
FundPrepaid	All funding of prepaid transactions.	String
Reserve	All reserve transactions.	String

These values are returned only for IPN operations.

Name	Description	Type
PAY	All pay transactions.	String
REFUND	All refund transactions.	String
SETTLE	All settle transactions.	String
SETTLE_DEBT	All debt settlement transactions.	String
WRITE_OFF_DEBT	All debt write-off transactions.	String
FUND_PREPAID	All funding of prepaid transactions.	String
RESERVE	All reserve transactions.	String
MULTI_SETTLE	All multi-settle transactions.	String
REAUTH	All transactions that required reauthorization.	String
DEPOSIT_FUNDS	All fund deposit transactions.	String
WITHDRAW_FUNDS	All fund withdrawal transactions.	String
CANCEL_TRANSACTION	All non user cancelled transactions.	String
CANCEL	All user cancelled transactions.	String

InstrumentId

Name	Description	Type
InstrumentId	An alphanumeric value that represents the payment instrument.	String Max size = 64 characters

InstrumentStatus

Name	Description	Type
Active	All active instruments installed for your application.	String
All	All instruments installed for your application.	String
Cancelled	All canceled instruments.	String

MarketplaceRefundPolicy

Name	Description	Type
MarketplaceTxnOnly	Caller refunds his fee to the recipient.	String
MasterAndMarketplaceTxn	Caller and Amazon FPS refund their fees to the sender, and the recipient refunds his amount	String
MasterTxnOnly	Caller does not refund his fee. Amazon FPS refunds its fee and the recipient refunds his amount plus the caller's fee to the sender. Type: String	String

OutstandingDebtBalance

Name	Description	Type
OutstandingBalance	Available debt balance accumulated between recipient and sender.	Amount (p. 113)
PendingOutBalance	Any balance that is pending because an external instrument was used to settle the debt.	Amount (p. 113)

OutstandingPrepaidLiability

Name	Description	Type
OutstandingBalance	Outstanding prepaid liability owed by this account to all the senders who bought prepaid instruments.	Amount (p. 113)
PendingInBalance	Any transient balance that is pending and yet to be settled.	Amount (p. 113)

PaymentMethod

Name	Description	Type
ABT	Amazon Payments account balance transfer.	String
ACH	Bank account transaction.	String
CC	Credit card transaction.	String
Debt	Transactions using a credit instrument as payment method.	String
Prepaid	Transactions using a prepaid instrument as payment method.	String

PrepaidBalance

Name	Description	Type
AvailableBalance	Available prepaid balance funded by the sender to pay a particular recipient.	Amount (p. 113)
PendingInBalance	Any balance that is pending because an external instrument is used to fund the instrument.	Amount (p. 113)

RecipientVerificationStatus

Name	Description	Type
VerificationComplete	The account is verified to accept payments.	String
VerificationPending	The account is not verified. The customer needs to contact Amazon Payments to resolve the issue.	String
VerificationComplete NoLimits	The account is verified to receive funds from Amazon Payments and has no receiving limits.	String

RelatedTransaction

Name	Description	Type
RelationType	Relation type of the related transaction.	RelationType (p. 118)
TransactionId	The Transaction ID of the related transaction.	String Max size = 35 characters

RelationType

Name	Description	Type
MarketplaceFee	Marketplace fee transactions.	String
Parent	Parent transactions.	String
Refund	Refund transactions.	String
RefundReversal	RefundReversal transactions.	String
Reserve	Reserve transactions.	String
Settle	Settle transactions.	String

SortOrderByDate

Name	Description	Type
Ascending	Return results in ascending order by date.	String
Descending	Return results in descending order by date (default).	String

StatusHistory

Name	Description	Type
Amount	The changed amount.	Amount (p. 113)
Date	The date when the status changed.	dateTime
StatusCode	The current status of the transaction.	String
TransactionStatus	The current status of the transaction.	TransactionStatus (p. 121)

TokenStatus

Name	Description	String
Active	The token is in active state.	String
Inactive	The token was canceled by the user and is inactive.	String

Token

Name	Description	Type
CallerReference	Account ID of the caller who initiated the original request.	String Max size = 128 bytes
DateInstalled	The date and time when the payment token was created on the caller's account.	dateTime
FriendlyName	A name that references the token.	String Max size = 128 characters
OldTokenId	The token ID linked to this token. The token that was created in place of this token.	String Size: 65 Bytes
PaymentReason	Payment reason passed during token installation.	String
TokenId	The token ID representing the payment instruction.	String Max size = 64 characters
TokenStatus	Specifies whether or not the token is active.	TokenStatus (p. 118)
TokenType	The type of the token (e.g., single-use, multi-use, etc.).	TokenType (p. 119)

TokenType

Name	Description	Type
MultiUse	Token that can be used multiple times.	String
Recurring	Token which is specifically marked for recurring payments.	String
SingleUse	Token that can be used only once.	String
Unrestricted	Token with unrestricted usage. Sender tokens with unlimited usage cannot be installed by external applications. Only recipient tokens can be installed with unrestricted usage.	String

TokenUsageLimit

Name	Description	Type
Amount	Amount paid in the latest time window with this token.	Amount (p. 113)
Count	Number of times this token was used in the latest time window.	Integer
LastResetAmount	Amount paid in the previous time window with this token.	Amount (p. 113)
LastResetCount	Number of times this token was used in the previous time window.	Integer
LastResetTimeStamp	The exact time when the latest time window started for this limit.	dateTime

Transaction

Name	Description	Type
Balance	Balance in prepaid account.	Amount (p. 113)
CallerName	The value in this field is dependent on the account type. For a personal account, the contact name is displayed. For a business or developer account, the business name is displayed.	String Max size = 128 characters
CallerTransactionDate	Date the caller provided for the transaction.	dateTime
DateCompleted	Date the transaction was completed.	dateTime
DateReceived	Date the transaction was received by Amazon FPS.	dateTime
FPSFees	Amount of fees collected by Amazon FPS for performing the transaction.	Amount (p. 113)
FPSOperation	The operation type.	FPS Operation (p. 114)
OriginalTransactionId	In the case of a refund, the TransactionID that is being reversed.	String Max size = 35 characters
PaymentMethod	Payment method used in the transaction.	Payment Method (p. 117)
RecipientName	The value in this field is dependent on the account type. For a personal account, the contact name is displayed. For a business or developer account, the business name is displayed.	String Max size = 128 characters
RecipientTokenID	The recipient token used in the transaction. Recipient tokens are needed when the caller and recipient are different people.	String Size: 65 Bytes

Name	Description	Type
SenderName	The value in this field is dependent on the account type. For a personal account, the contact name is displayed. For a business or developer account, the business name is displayed.	String Max size = 128 characters
SenderTokenID	The sender token used in the transaction.	String Size: 65 Bytes
StatusCode	A code that represents the current status of the transaction. Expands on the information in the *TransactionStatus* field. For example, if *TransactionStatus* is PENDING, this field might be *PendingVerification*, or *PendingNetworkResponse*.	String
StatusMessage	A short description of the current status of the transaction.	String
TransactionAmount	Total amount of the transaction.	Amount (p. 113)
TransactionId	Unique Amazon FPS-generated ID for the transaction.	String Max size = 35 characters
TransactionPart	List of individual parts of the transaction, with each one dealing with your account's role in the transaction.	Transaction Part (p. 124)
TransactionStatus	Provides a short code on the status of the transaction, for example "PENDING".	Transaction Status (p. 121)

TransactionalRole

Name	Description	Type
Caller	Role is the caller.	String
Recipient	Role is the recipient.	String
Sender	Role is the sender.	String

TransactionStatus

These values are returned for non-IPN operations.

Name	Description	Type
Cancelled	The transaction was canceled.	String

Name	Description	Type
Failure	The transaction failed. The API operation failed and Amazon FPS did not receive or record a transaction. You can retry the transaction only if a retriable error was returned.	String
Pending	The transaction is pending.	String
Reserved	The reserve request on the transaction succeeded. Amazon FPS reserves the purchase price against the sender's payment instrument.	String
Success	The transaction succeeded. You can fulfill the order for the customer.	String

TransactionStatus (IPN)

These values are returned for IPN operations only.

Name	Description	Type
CANCELLED	The transaction was canceled.	String
FAILURE	The transaction failed. The API operation failed and Amazon FPS did not receive or record a transaction. You can retry the transaction only if a retriable error has been returned.	String
PENDING	The transaction is pending.	String
RESERVED	The reserve request on the transaction succeeded. Amazon FPS reserves the purchase price against the sender's payment instrument.	String
SUCCESS	The transaction succeeded. You can fulfill the order for the customer.	String

TransactionDetail

Name	Description	Type
CallerNamePDF	The value in this field is dependent on the account type. For a personal account, the contact name is displayed. For a business or developer account, the business name is displayed.	String Max size = 128 characters
CallerDescription	Caller description the caller provided for the transaction.	String Constraint: Max size = 160 characters
CallerReference	Caller reference the caller provided for the transaction.	String Max size = 128 characters
CreditInstrumentID	In the case of a postpaid transaction, this is the credit instrument ID.	String
DateReceived	Date Amazon FPS received the transaction.	dateTime
DateCompleted	Date the transaction was completed.	dateTime
FPSFees	Amount of fees collected by Amazon FPS for performing the transaction.	Amount (p. 113)
FPSFeesPaidBy	The party paying the FPS fees for this transaction.	TransactionalRole (p. 121)
FPSOperation	The operation type.	FPSOperation (p. 114)
MarketPlaceFees	In the case of a marketplace transaction, this is the amount of any marketplace fee the caller has charged.	Amount (p. 113)
PaymentMethod	The payment method used.	PaymentMethod (p. 117)
PrepaidInstrumentID	In the case of a prepaid transaction, this is the prepaid instrument ID.	String
RecipientEmail	The email ID of the recipient of this transaction.	String
RecipientName	The value in this field is dependent on the account type. For a personal account, the contact name is displayed. For a business or developer account, the business name is displayed.	String Max size = 128 characters
RecipientTokenId	Recipient token ID used in the transaction. Recipient tokens are needed when the caller and recipient are different people.	String Size: 65 Bytes
RelatedTransaction	All transactions related to this transaction.	RelatedTransaction (p. 117)
SenderDescription	Sender description the caller provided for the transaction.	String Constraint: Max size = 160 characters

Name	Description	Type
SenderEmail	The email ID of the sender of this transaction. This is returned only if the caller is also the recipient of this transaction.	String
SenderName	The value in this field is dependent on the account type. For a personal account, the contact name is displayed. For a business or developer account, the business name is displayed.	String Max size = 128 characters
SenderTokenId	Sender token ID used in the transaction.	String Size: 65 Bytes
StatusCode	A code that represents the current status of the transaction.	String
StatusHistory	A list of all the previous status entries for this transaction.	StatusHistory (p. 118)
StatusMessage	A short description of the current status of the transaction.	String
TransactionAmount	Total amount of the transaction.	Amount (p. 113)
TransactionId	Unique Amazon FPS-generated ID for the transaction.	String Max size = 35 characters
TransactionStatus	The transaction status.	TransactionStatus (p. 121)

TransactionPart

Name	Description	Type
Description	Description provided by the entity.	String
FeesPaid	Fees the caller or recipient paid.	Amount (p. 113)
InstrumentId	Payment instrument involved in this transaction part.	String
Name	Name used for the role specified in *Role*.	String
Reference	Reference data provided by this party.	String
Role	Role played by this party.	TransactionalRole (p. 121)

Amazon FPS Instant Payment Notification Field Reference

Topics

Common IPN Response Elements

These IPN response elements are common to most types of transactions. For a list of IPN response elements for marketplace transactions, see IPN Responses for Marketplace Transactions (p. 128).

Name	Description
addressFullName	Full name of the buyer/sender. Type: String
addressLine1	Sender's address (first line). For IPN, this element is returned only if the value has been updated with Amazon. Type: String
addressLine2	Sender's address (second line). For IPN, this element is returned only if the value has been updated with Amazon. Type: String
addressState	Sender's state. For IPN, this element is returned only if the value has been updated with Amazon. Type: String
addressZip	Sender's post code. For IPN, this element is returned only if the value has been updated with Amazon. Type: String
addressCountry	Sender's country. For IPN, this element is returned only if the value has been updated with Amazon. Type: String

Name	Description
addressPhone	Sender's phone number. For IPN, this element is returned only if the value has been updated with Amazon. Type: String
buyerEmail	Sender's email address. **Note** The *buyerEmail* element is not returned when the recipient is not the caller (i.e., marketplace transactions). Type: String Size: 65 bytes
buyerName	Sender's name. Type: String Size: 128 bytes
certificateUrl	A url specifying the location of the certificate used for signing the response. Type: String Max Size: 1024 bytes
customerEmail	Customer's email address. Type: String Size: 65 bytes
customerName	Buyer/Sender Full Name. Type: String Size: 128 bytes
dateInstalled	If the *notificationType* element (below) is *TokenCancellation*, this element contains the date the token was installed. Type: String Size: 30 bytes
isShippingAddressProvided	If the IPN results include address updates, this element contains TRUE. Otherwise this element is not present in the response. Type: String
operation	The name of the payment action, also called an operation, used for this transaction. For example, **PAY** for the action Pay (p. 72). Type: String Max Size: 20 bytes
notificationType	Notification type may be either *TokenCancellation* or *TransactionStatus* Type: String Size: 20 bytes

Name	Description
paymentMethod	The payment method used by the sender. For more information, see the IPN values in PaymentMethod (p. 117). Type: String Size: 20 bytes
paymentReason	Reason for payment. Type: String
recipientEmail	Recipient's email address. **Note** As a security precaution, you should always check that the recipient email is the same as the one in your original request. Type: String Size: 65 bytes
recipientName	Recipient's name. Type: String Size: 128 bytes
signature	The encoded string the caller uses to verify the IPN. Amazon Payments calculates the signature using the elements in the returnURL. The merchant must have manually signed the request.. Type: String Size: 512 bytes
signatureVersion	A value that specifies the *Signature* format. Type: Integer Valid Values: 2
signatureMethod	A value that specifies the signing method. Type: String Valid Values: HmacSHA256 (preferred) and HmacSHA1.
tokenId	If *notificationType* is *TokenCancellation*, this element contains the ID of the cancelled token. Type: String Size: 65 bytes
tokenType	If *notificationType* is *TokenCancellation*, this element contains the type of the canceled token. Type: String Size: 20 bytes
transactionAmount	Specifies the amount payable in this transaction; for example, USD 10.00. Type: String Size: 30 bytes

Name	Description
transactionDate	The date when this transaction occurred, specified in seconds since the start of the epoch. Type: Long Size: 40 bytes
transactionId	Unique ID generated by Amazon FPS for this transaction. This element is returned if the transaction was accepted by Amazon FPS. Type: String Size: 35 bytes
transactionStatus	Specifies the status of the transaction. For more information, see TransactionStatus (IPN) (p. 122). Type: String

IPN Responses for Marketplace Transactions

The following IPN response elements are returned only for marketplace transactions.

IPN Marketplace Transaction Elements

Name	Description
buyerName	Sender's name. Type: String
operation	The name of the payment action, also called an operation, used for this transaction. Type: String Max Size: 20 bytes
paymentMethod	The payment method used by the sender. For more information, see the IPN values in PaymentMethod (p. 117). Type: String
paymentReason	Reason for payment. Type: String
recipientEmail	Recipient's email address. Type: String
recipientName	Recipient's name. Type: String
referenceId	If you specified a referenceId in the button creation form, Amazon Payments returns the referenceId to you. Type: String

Name	Description
signature	The encoded string the caller uses to verify the IPN. Amazon Payments calculates the signature using the elements in the returnURL. The merchant must have manually signed the request. For more information, see Handling the Receipt of IPN Notifications. We recommend that you always verify the signature using the method in How to Verify the IPN Signature. Type: String
status	Specifies the status of the transaction. For more information, see TransactionStatus (IPN) (p. 122). Type: String
transactionAmount	Specifies the amount payable in this transaction; for example, USD 10.00. This element is not being returned in the current version. Type: Double
transactionDate	The date when this transaction occurred, specified in seconds since the beginning of the epoch. Type: Long
transactionId	Unique ID generated by Amazon FPS for this transaction. This element is returned if the transaction was accepted by Amazon FPS. Type: String

Example Email Messages

Topics

The following section lists examples of all the email messages sent by Amazon Payments to applications created using any of the five quick start guides. The navigation table identifies to which quick start guide each example applies.

Email Notification Templates

Many transactions generate email messages from Amazon Payments, sent to either the buyer, seller, or web site owner in the case of a marketplace transaction. Transaction details are listed in the body of the email message. The content of the email message sent out depends on the transaction and its status.

The table here defines the templates that are used, and provides a link to an example message for each.

Email Template Name	Description
BAVerifFailed	**Account Management Quickstart** Bank account verification failed
BAVerifStartedForPersonalBusiness	**Account Management Quickstart** Bank account verification started for personal and business account
BAVerifStartedForDeveloper	**Account Management Quickstart** Bank account verification started for developer account
BAVerifSuccess	**Account Management Quickstart** Bank account verification successful
CCVerifFailedBusiness	**Account Management Quickstart** Credit card verification failed for business account
ConfirmEmailPersonal	**Account Management Quickstart** Email confirmation sent to confirm the email address for the personal account.

Email Template Name	Description
ConfirmEmailDeveloper	**Account Management Quickstart** Email confirmation sent to confirm the email address for a developer account.
ConfirmEmailBusiness	**Account Management Quickstart** Email confirmation sent to confirm the email address for a business account.
DailySummary	**Account Management Quickstart** Daily summary of transactions
DepositFailure	**Account Management Quickstart** Deposit failed
DepositFundsInitiated	**Account Management Quickstart** Request to deposit funds initiated
DepositSuccess	**Account Management Quickstart** Deposit successful
DeveloperRegistrationCompleted	**Account Management Quickstart** Developer registration complete
FundPrepaidFailure	**Aggregated Quickstart** Prepaid balance purchase failure
FundPrepaidSuccess	**Aggregated Quickstart** Prepaid balance purchase success
MonthlyNotif	**Account Management Quickstart** Notification for monthly statement
MPFeeRegistrationCallerPaysFee	**Marketplace Quickstart** MarketPlace registration email, fee paid by caller
MPFeeRegistrationRecipientPaysFee	**Marketplace Quickstart** MarketPlace registration email, fee paid by recipient
MultiPaymentFailure	**Advanced Quickstart** Payment failure from a multi use token (payment authorizations)
MultiPaymentSuccess	**Advanced Quickstart** Payment success from a multi use token (payment authorizations)
MultiuseTokenCancel	**Advanced Quickstart** Multi-use token has been canceled
MultiuseTokenInstall	**Advanced Quickstart** Mulit-use token has been installed
OnetimePaymentACHInit	**Basic and Marketplace Quickstarts** One-time payment (ACH) initiated
OnetimePaymentFailure	**Marketplace Quickstart** One-time payment failed
OnetimePaymentSuccess	**Basic and Marketplace Quickstarts** One-time payment successful

Email Template Name	Description
OnetimePaymentSuccessACH	**Basic and Marketplace Quickstarts** One-time ACH payment successful
PostpaidTokenCancel	**Aggregated Quickstart** Postpaid token canceled
PostpaidTokenInstall	**Aggregated Quickstart** Postpaid token installed
RecurringPaymentFailure	**Advanced Quickstart** Recurring payment has failed
RecurringPaymentSuccess	**Advanced Quickstart** Recurring payment has succeeded
RecurringPaymentSuccessACH	**Advanced Quickstart** Recurring ACH payment has succeeded
RecurringTokenCancel	**Advanced Quickstart** Recurring token is canceled
RecurringTokenInstall	**Advanced Quickstart** Recurring token is installed
SettleDebtFailure	**Aggregated Quickstart** Settle debt transaction failed
SettleDebtSuccess	**Aggregated Quickstart** Settle debt transaction succeeded
UpgradePersonalToBusiness	**Account Management Quickstart** Upgrade to business account successful
VerifyEmailSuccessPersonal	**Account Management Quickstart** Email address verified for personal account
VerifyEmailSuccessBusiness	**Account Management Quickstart** Email address verified for business account
WithdrawFailure	**Account Management Quickstart** Withdraw failed, bank unable to accept electronic transaction
WithdrawFundsInitiated	**Account Management Quickstart** Withdraw funds has been initiated

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 Guide	Covers the marketplace functionality of Amazon FPS.
Amazon Flexible Payments Service Advanced Quick Start Guide	Covers the multiple-payment functionality of Amazon FPS.
Amazon Flexible Payments Service Aggregated Payments Quick Start Guide	Covers aggregated micro-transactions and the prepaid and postpaid functionality of Amazon FPS.
Amazon Flexible Payments Service Account Management Quick Start Guide	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.
AWS Developer Resource Center	A central starting point for AWS documentation, code samples, release notes, and other information to help you build innovative applications.

Resource	Description
AWS Support Center	The home page for AWS Technical Support, including access to our Developer Forums, Technical FAQs, Service Status page, and Premium Support (if you are subscribed to this program).
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.

Glossary

ABT, Amazon Payments Account Balance	One method of payment available with Amazon Payments.
access key rotation	For added security, you can switch between an active and inactive access key on your AWS security credentials page.
AWS Access Key ID	A string distributed by AWS that uniquely identifies your AWS developer account. You include this ID in every request.
ACH, Bank Account Debits	One method of payment available with Amazon Payments.
buyer	The buyer pays the seller for a product or service.
caller	A developer who facilitates payment between a sender and a recipient.
chargeback	a reversal of a payment issued by the bank when the buyer disputes the charge.
Co-Branded User Interface (CBUI)	A set of Amazon Payments web pages which lead a user through a secure login and payment authorization pipeline. Once the pipeline is complete, the user is redirected to your website.
endpoint	The URI that specifies the destination of an API request.
HMAC	The Hash Message Authentication Code used to authenticate a message. The HMAC is calculated using a standard, hash cryptographic algorithm, such as SHA-256. This algorithm uses a key value to perform the encryption. That key is your *Secret Key*. For that reason, your Secret Key must remain a shared secret between you and Amazon Payments.
inbound requests	Button click or other form request to Amazon Payments. Also inbound notification.
Instant Payment Notification	A notification that is sent whenever a payment, refund, or reserved payment completes successfully or fails. The caller must host this notification service and provide Amazon Payments with its URL.
marketplace, marketplace scenario	An environment in which the caller charges a fee for facilitating a transaction between a sender and a recipient.
order pipeline	The steps through which an order passes between the time a customer selects an item and the customer's pay instrument is charged.
outbound notifications	Response from Amazon Payments to your Amazon FPS application by way of Return URL or IPN.

payment instrument	The method of payment a customer chooses to use with Amazon Payments. These are credit cards, Amazon Payments account balance (ABT), and bank account debits (ACH)
one time payment	An Amazon FPS payment processed with a single-use payment token. When the payment is made, the token may no longer be used.
recipient	A seller who receives a payment from a buyer (sender) in exchange for a service or product.
Recipient Token	Payment token created when a seller authorizes a payment of marketplace fees to you for hosting services, often with a *Register Now* button.
recurring payment	An Amazon FPS payment processed with a recurring payment token. Payments are made periodically using the same payment token. The token is valid until it expires.
reserve	The amount that is put in reserve against a credit card but not charged. Later, the transaction is settled (typically when the product is actually shipped).
sandbox	A part of the Amazon Payments web service where you can test the functionality of your application without incurring charges or purchasing products.
Secret Key	A string distributed by AWS that uniquely identifies your AWS developer account. The Secret Key is a shared secret between the developer and AWS. The Secret Key is used as the key in the HMAC algorithm that encrypts the signature.
seller	The seller receives money from a buyer in exchange for a service or product.
sender	The sender (also known as the buyer) pays a recipient for a product or service.
settle	To complete a transaction that has been reserved. If you don't charge the sender immediately upon the initiation of the purchase (and instead reserve the amount against the sender's credit card), you settle the transaction later, typically after you ship the product to the sender. Settle actually makes the reserved amount move from the sender to the recipient.
SHA1, SHA256	Secure Hash Algorithms used for Amazon Web Services signatures. SHA1 is an earlier version of the algorithm, which is currently being deprecated for Amazon Web Services. SHA256 is its more secure replacement.
signature	A URL-encoded string composed of request parameters and their values encrypted using an HMAC algorithm. Signatures are used to authenticate and safeguard requests.
Sender Token	Payment token created when buyers authorize purchase on their own behalf, often with a *Pay Now* button.
string-to-sign	Prior to calculating the HMAC signature, you first assemble the components for the signature in a sorted order, and then URL encode them. The pre-encrypted string is the string-to-sign.
website owner	A developer who uses Amazon Flexible Payments Service.

Document History

This documentation is associated with the 2010-08-28 version of the *Amazon Flexible Payments Service API Reference*. This guide was last updated on 10-December-2012.

The following table describes the important changes since the last release of this guide.

Change	Description	Release Date
Document Update	Minor editorial updates.	In this release
Initial Release	First publication of this reference.	August 4, 2010