PayPal™

Adaptive Payments Developer Guide

Last updated: October 30, 2012

Adaptive Payments Developer Guide

Document Number: 10097.en_US-20121030

© 2010-2012 PayPal, Inc. All rights reserved. PayPal is a registered trademark of PayPal, Inc. The PayPal logo is a trademark of PayPal, Inc. Other trademarks and brands are the property of their respective owners.
The information in this document belongs to PayPal, Inc. It may not be used, reproduced or disclosed without the written approval of PayPal, Inc.
Copyright © PayPal. All rights reserved. PayPal S.à r.l. et Cie, S.C.A., Société en Commandite par Actions. Registered office: 22-24 Boulevard Royal, L-2449, Luxembourg, R.C.S. Luxembourg B 118 349
Consumer advisory: The PayPal™ payment service is regarded as a stored value facility under Singapore law. As such, it does not require the approval of the Monetary Authority of Singapore. You are advised to read the terms and conditions carefully.

Notice of non-liability:
PayPal, Inc. is providing the information in this document to you "AS-IS" with all faults. PayPal, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. PayPal, Inc. assumes no liability for damages (whether direct or indirect), caused by errors or omissions, or resulting from the use of this document or the information contained in this document or resulting from the application or use of the product or service described herein. PayPal, Inc. reserves the right to make changes to any information herein without further notice.

Contents

What's New? 15

1.8.2 Features 15

Chapter 1 Introducing Adaptive Payments 17

Adaptive Payments Actors and Objects 17
Simple, Parallel, and Chained Payments 19
Simple Payments 20
Parallel Payments 20
Chained Payments. 21
Payment Approval 22
Adaptive Payments Service Permissions 23
Explicit Approval Payment Flow 23
Preapproved Payments Flow 25
Implicit Approval Payments Flow 27
Embedded Payments 27
Embedded Payment Flow Presentations 28
Kinds of Embedded Payments. 30
Embedded Payments Implementation Basics 31
Embedded Payment Experience 31
Preapprove Future Payments Checkbox 37
Shipping Address Information 39
Embedded Payment Experience 40
Setting Up Web Pages to Invoke the Embedded Payment Flow Using a Lightbox . . . 55
Setting Up Web Pages to Invoke the Embedded Payment Flow Using a Minibrowser . 57
Displaying and Collecting Shipping Addresses 58
Guest Payments 59
Fee Payment Configuration 60
Sender Pays the Fee 61
Receiver Pays the Fee in a Parallel Payment 62
Each Receiver Pays the Fee in a Chained Payment 62
Primary Receiver Pays the Fee in a Chained Payment 63

Chapter 2 Getting Started 65

Adaptive Payments API Operations 65
API Operations Related to Payments 65
API Operations Related to Preapprovals 65
Other API Operations 66
Adaptive Payments Endpoints 66
HTTP Headers 66
Authentication 67
Specifying JSON, NVP, or XML Data Formats 67
SOAP Messages 68
Specifying Application and Device Information 68
Making a Simple Payment (JSON) 69
Pay Request for Simple Payment 69
Pay Response for Simple Payment 69
Making a Parallel Payment (NVP) 70
Pay Request for Parallel Payment 70
Pay Response for Parallel Payment 70
Making a Chained Payment (XML) 71
Pay Request for Chained Payment 71
Pay Response for Chained Payment 72

Chapter 3 Pay API Operation 73

Pay Summary 73
Common Fields for All Payments 73
Parallel Payments 74
Chained Payments 74
Implicit Payments 75
Preapproved Payments 75
Payments for Digital Goods 75
Payment Notifications 75
Additional Notes About the Pay API Operation 76
PayRequest Message 77
PayRequest Fields 79
ClientDetails Fields 82
FundingConstraint Fields 83
FundingTypeList Fields 83
FundingTypeInfo Fields 84
ReceiverList Fields 84
Receiver Fields 84

PhoneNumberType Fields . . . 85
SenderIdentifier Fields . . . 86
AccountIdentifier Fields . . . 86
RequestEnvelope Fields . . . 86
PayResponse Message . . . 86
PayResponse Fields . . . 90
PayErrorList Fields . . . 91
FundingPlan Fields . . . 91
CurrencyType Fields . . . 91
FundingSource Fields . . . 92
CurrencyConversion Fields . . . 93
FundingPlanCharge Fields . . . 93
WarningDataLists Fields . . . 94
PayError Fields . . . 94
ErrorData Fields . . . 94
Receiver Fields . . . 95
PhoneNumberType Response Fields . . . 96
ResponseEnvelope Fields . . . 96
PPFault Message . . . 96
FaultMessage Fields . . . 97
ErrorData Fields . . . 98
ResponseEnvelope Fields . . . 98
Pay Examples Using NVP and CURL . . . 99
Pay Errors . . . 104

Chapter 4 PaymentDetails API Operation . . . 109

PaymentDetailsRequest Message . . . 109
PaymentDetailsRequest Fields . . . 109
RequestEnvelope Fields . . . 110
PaymentDetailsResponse Message . . . 110
PaymentDetailsResponse Fields . . . 113
FundingTypeList Fields . . . 116
FundingTypeInfo Fields . . . 116
PaymentInfoList Fields . . . 117
PaymentInfo Fields . . . 117
Receiver Fields . . . 118
SenderIdentifier Fields . . . 119
AccountIdentifier Fields . . . 120
PhoneNumberType Response Fields . . . 120

ResponseEnvelope Fields 120
PPFault Message 121
FaultMessage Fields 122
ErrorData Fields 122
ResponseEnvelope Fields 123
PaymentDetails Examples Using NVP and CURL 123
Payment Details Errors 125

Chapter 5 ExecutePayment API Operation 127

ExecutePaymentRequest Message 127
ExecutePaymentRequest Fields 127
RequestEnvelope Fields 128
ExecutePaymentResponse Message 128
ExecutePaymentResponse Fields 130
PayErrorList Fields 130
PayError Fields 130
ErrorData Fields 131
Receiver Fields 131
ResponseEnvelope Fields 132
PPFault Message 133
FaultMessage Fields 134
ErrorData Fields 134
ResponseEnvelope Fields 135
ExecutePayment Errors 135

Chapter 6 GetPaymentOptions API Operation 139

GetPaymentOptionsRequest Message 139
GetPaymentOptionsRequest Fields 139
RequestEnvelope Fields 139
GetPaymentOptionsResponse Message 140
GetPaymentOptionsResponse Fields 141
DisplayOptions Fields 142
InitiatingEntity Fields 143
Institution Customer Fields 143
SenderOptions Fields 144
ReceiverOptions Fields 144
InvoiceData Fields 144
InvoiceItem Fields 145

ReceiverIdentifier Fields 145
PhoneNumberType Response Fields 145
ResponseEnvelope Fields 146
PPFault Message 146
FaultMessage Fields 147
ErrorData Fields 148
ResponseEnvelope Fields 148
GetPaymentOptions Errors 149

Chapter 7 SetPaymentOptions API Operation 151

SetPaymentsOptionsRequest Message 151
SetPaymentOptionsRequest Fields 152
DisplayOptions Fields 153
InitiatingEntity Fields 153
Institution Customer Fields 154
SenderOptions Fields 154
ReceiverOptions Fields 155
InvoiceData Fields 155
InvoiceItem Fields 155
ReceiverIdentifier Fields 156
PhoneNumberType Fields 156
RequestEnvelope Fields 156
SetPaymentOptionsResponse Message 157
ResponseEnvelope Fields 157
PPFault Message 158
FaultMessage Fields 158
ErrorData Fields 159
ResponseEnvelope Fields 159
SetPaymentOptions Examples Using NVP and CURL 160
SetPaymentOptions Errors 161

Chapter 8 Preapproval API Operation 163

Preapproval Overview 163
Preapproval Notifications 163
Additional Notes About the PreApproval API Operation 164
PreapprovalRequest Message 164
PreapprovalRequest Fields 165
ClientDetails Fields 169

RequestEnvelope Fields 170
PreapprovalResponse Message 170
PreapprovalResponse Fields 171
ResponseEnvelope Fields 171
PPFault Message 171
FaultMessage Fields 172
ErrorData Fields 173
ResponseEnvelope Fields 173
Preapproval Examples Using NVP and CURL 174
Preapproval Errors 175

Chapter 9 PreapprovalDetails API Operation 177

PreapprovalDetailsRequest Message 177
PreapprovalDetailsRequest Fields 177
RequestEnvelope Fields 178
PreapprovalDetailsResponse Message 178
PreapprovalDetailsResponse Fields 180
AddressList Fields 183
Address Fields 183
BaseAddress Fields 184
ResponseEnvelope Fields 184
PPFault Message 185
FaultMessage Fields 186
ErrorData Fields 186
ResponseEnvelope Fields 187
PreapprovalDetails Examples Using NVP and CURL 187
Preapproval Details Errors 189

Chapter 10 CancelPreapproval API Operation 191

CancelPreapprovalRequest Message 191
CancelPreapprovalRequest Fields 191
RequestEnvelope Fields 192
CancelPreapprovalResponse Message 192
CancelPreapprovalResponse Fields 192
ResponseEnvelope Fields 193
PPFault Message 193
FaultMessage Fields 194
ErrorData Fields 195

ResponseEnvelope Fields 195
CancelPreapproval Errors 196

Chapter 11 ConvertCurrency API Operation 197

ConvertCurrencyRequest Message 197
ConvertCurrencyRequest Fields 197
CurrencyType Fields 198
CurrencyList Fields 200
CurrencyCodeList Fields 201
RequestEnvelope Fields 202
ConversionType Fields 202
ConvertCurrencyResponse Message 202
ConversionCurrencyResponse Fields 203
CurrencyConversionTable Fields 203
CurrencyConversionList Fields 204
CurrencyType Fields 204
CurrencyList Fields 205
ResponseEnvelope Fields 206
PPFault Message 206
FaultMessage Fields 207
ErrorData Fields 208
ResponseEnvelope Fields 208
ConvertCurrency Errors 209

Chapter 12 Refund API Operation 211

Refund API Overview 211
Refund Notifications 213
Additional Notes About the Refund API Operation 213
RefundRequest Message 214
RefundRequest Fields 215
ReceiverList Fields 217
Receiver Fields 217
PhoneNumberType Fields 218
RequestEnvelope Fields 218
Refund Response Message 219
RefundResponse Fields 221
RefundInfoList Fields 222
RefundInfo Fields 222

ErrorData Fields 224
Receiver Fields 224
PhoneNumberType Response Fields 225
ResponseEnvelope Fields 225
PPFault Message 226
FaultMessage Fields 227
ErrorData Fields 227
ResponseEnvelope Fields 228
Refund Examples Using NVP and CURL 228
Refund Errors 230

Chapter 13 GetFundingPlans API Operation 233

GetFundingPlans Overview 233
Additional Notes About the Pay API Operation 233
GetFundingPlansRequest Message 233
GetFundingPlansRequest Fields 234
RequestEnvelope Fields 234
GetFundingPlansResponse Message 234
GetFundingPlansResponse Fields 236
FundingPlan Fields 236
CurrencyType Fields 237
FundingSource Fields 238
CurrencyConversion Fields 239
ResponseEnvelope Fields 239
PPFault Message 240
FaultMessage Fields 241
ErrorData Fields 241
ResponseEnvelope Fields 242
GetFundingPlan Errors 242

Chapter 14 GetShippingAddresses API Operation 245

GetShippingAddressesRequest Message 245
GetShippingAddressesRequest Fields 245
RequestEnvelope Fields 246
GetShippingAddressesResponse Message 246
GetShippingAddressesResponse Fields 247
Address Fields 247
BaseAddress Fields 247

ResponseEnvelope Fields 248
PPFault Message 248
FaultMessage Fields 249
ErrorData Fields 250
ResponseEnvelope Fields 250
GetShippingAddresses Errors 251

Chapter 15 Adaptive Payment Commands and Redirects 253

Adaptive Payment Commands 253
_ap-payment Command 253
_ap-preapproval Command 254
Embedded Payment Flow Redirect 254
JavaScript Functions for Embedded Payments 255

Chapter 16 Instant Payment Notifications 257

Pay Message Variables 257
Preapproval Message Variables 260

Older Versions of the Adaptive Payments API 263

1.8.1 Features 263
SenderIdentifier Fields 263
TaxIdDetails Fields 263
1.8.0 Features 264
Changes to SenderOptions 264
Changes to ConvertCurrencyRequest Fields for Version 1.8.0 264
Changes to JavaScript Functions for Embedded Payments 264
New JavaScript Function for Version 1.8.0: AutoRedirectOnDone 265
1.7.0 Features 266
New API Operations for Version 1.7.0 266
Changes to ConfirmPreapprovalRequest Fields for Version 1.7.0 266
Changes to JavaScript Functions for Embedded Payments 267
1.6.0 Features 267
New API Operations for Version 1.6.0 267
Changes to PayRequest Fields for Version 1.6.0 267
Changes to PayResponse Fields for Version 1.6.0 268
Changes to ExecutePaymentRequest Fields for Version 1.6.0 268
Changes to GetPaymentOptionsResponse Fields for Version 1.6.0 268
Changes to SetPaymentOptionsRequest Fields for Version 1.6.0 268

Changes to PreapprovalRequest Fields for Version 1.6.0 268
Changes to Address Structure for Version 1.6.0 269
Changes to DisplayOptions Structure for Version 1.6.0 269
New CurrencyConversion Structure for Version 1.6.0 269
New InvoiceData Structure for Version 1.6.0 270
New InvoiceItem Structure for Version 1.6.0 270
New SenderOptions Structure for Version 1.6.0 270
New SenderIdentifier Structure for Version 1.6.0 270
New AccountIdentifier Structure for Version 1.6.0 271
New ReceiverOptions Structure for Version 1.6.0 271
New ReceiverIdentifier Structure for Version 1.6.0 271
Additional Error Messages for Version 1.6.0 271
1.5.0 Features 274
Ability to Delay a Chained Payment Feature in Version 1.5.0 274
PaymentDetails Pending Reason Feature in Version 1.5.0 274
1.4.0 Features 275
New API Operations for Version 1.4.0 275
Changes to the Pay API Operation 276
Changes to the PaymentDetails API Operation 276
Changes to the PreapprovalDetails API Operation 278
Changes to the PreapprovalDetails API Operation 278
Changes to the Refund API Operation 279
1.3.0 Features 279
Overview of Changes for Version 1.3.0 279
HTTP Header Changes 280
Changes to the Pay API Operation for Version 1.3.0 281
Changes to the PaymentDetails API Operation for Version 1.3.0 283
1.2.0 Features 284
Changes to the Pay API Operation for Version 1.2.0 285
Changes to the Payment Details API Operation for Version 1.2.0 285
Changes to Preapproval API Operation for Version 1.2.0 286
Changes to Preapproval Details API Operation for Version 1.2.0 286
1.1.0 Features 286
Changes to the Pay API Operation for Version 1.1.0 287
Changes to PaymentDetails API Operation for Version 1.1.0 287
Changes to Preapproval API Operation for Version 1.1.0 289
Changes to PreapprovalDetails API Operation for Version 1.1.0 289
Changes to Refund API Operation for Version 1.1.0 289

Revision History 291

Index **293**

Contents

What's New?

Adaptive Payments is intended for developers implementing payment solutions with PayPal's Adaptive Payments API. Check out what's new in the current release.

1.8.2 Features

Version 1.8.2 of the Adaptive Payments API adds a new `WarningData` field in the `Pay` response related to payments to are in the pending state. The `PreApproval` API also contains a new `requireInstantFundingSource` request field that can require an instant funding source for preapprovals.

NOTE: Changes to API operations are backward-compatible.

1 Introducing Adaptive Payments

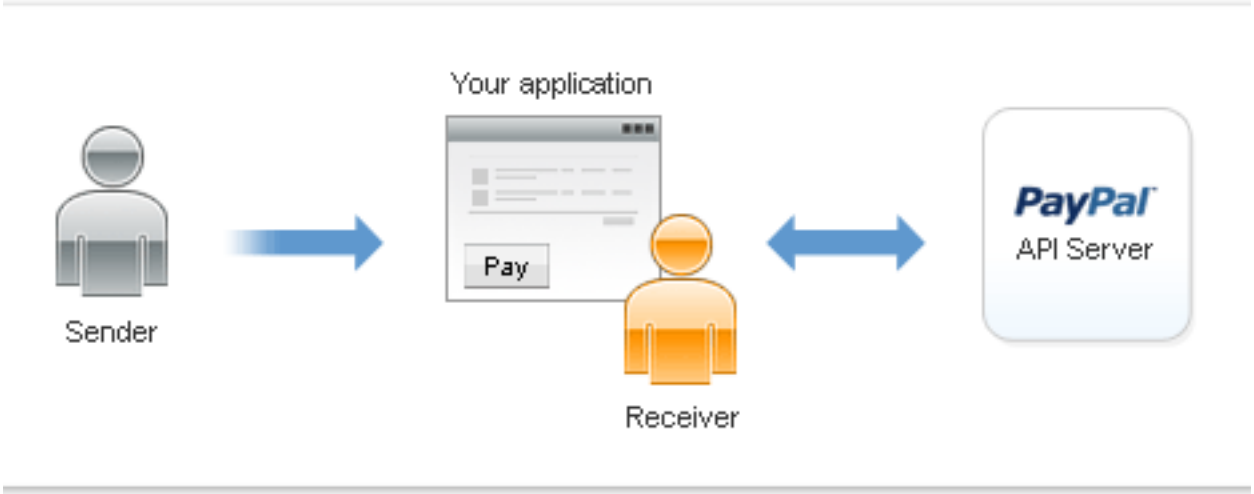
The Adaptive Payments API enables you to send money in many different scenarios, from simple to complex. For example, you might build a small send money application for a social networking site or a robust payroll system.

Adaptive Payments Actors and Objects

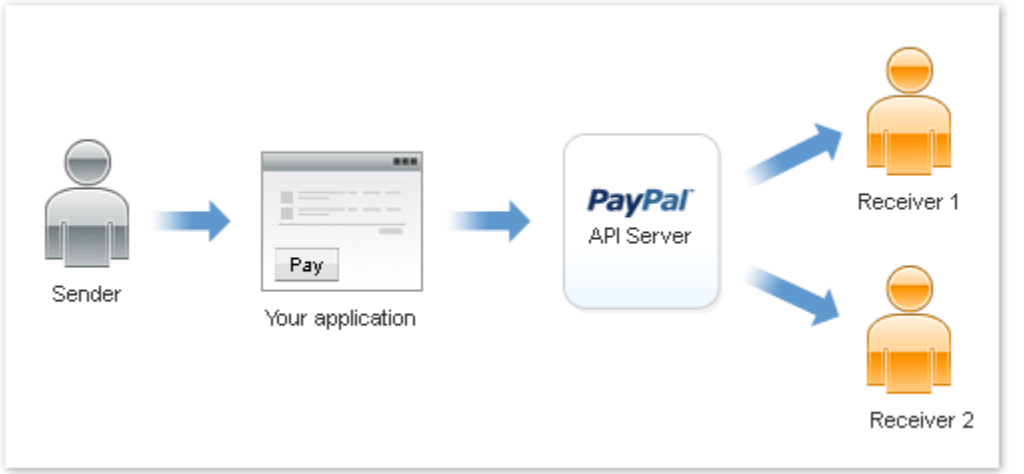
Adaptive payments handles payments between a *sender* of a payment and one or more *receivers* of the payment. You are an *application owner*, such as a merchant that owns a website, the owner of a widget on a social networking site, the provider of a payment application on mobile phones, and so on. Your application is the caller of Adaptive Payments API operations.

NOTE: The application owner must have a PayPal Business account. Senders and receivers can have any PayPal account type. Senders and receivers are not required to have PayPal accounts initially. PayPal prompts a sender to create an account before a payment can be completed. A receiver must create an account to receive the funds after the payment completes.

With many applications, you may be both the application owner and a receiver. For example, as the owner of a website, you are the receiver of payments from the senders who are your customers. The following diagram shows the relationship between a sender, you as a receiver, and PayPal:

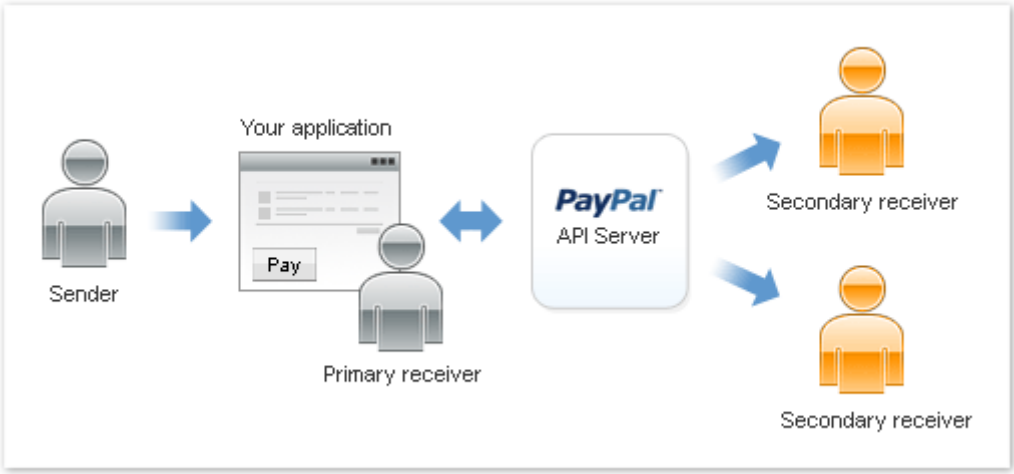


You are not required to be a receiver. For example, if you own a shopping cart, you are not required to receive payments directly. You can facilitate payments between the sender and receivers that provide the actual goods. The following diagram shows the relationship between a sender, you as an application owner that directs payments to receivers, and PayPal:



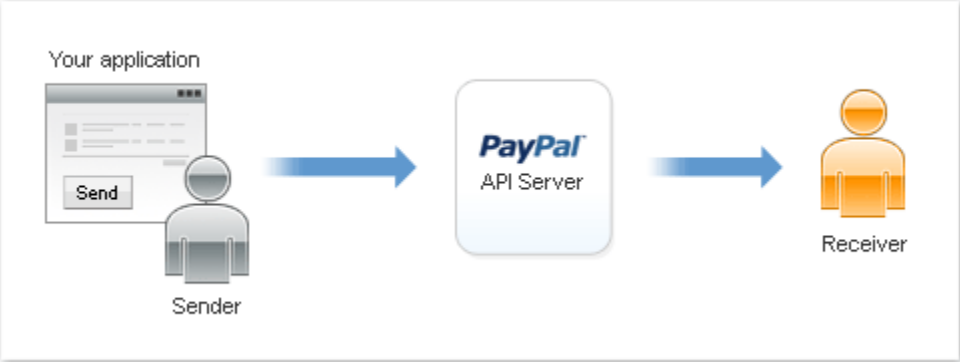
This diagram shows a payment in which the sender pays multiple receivers in a parallel payment. With parallel payments, the sender can see the transaction to each receiver.

The following diagram shows the relationship between a sender, you as an application owner that directs payments to receivers, and PayPal in a chained payment:



In a chained payment, the sender pays the primary receiver an amount, from which the primary receiver pays secondary receivers. The sender only knows about the primary receiver, not the secondary receivers. The secondary receivers only know about the primary receiver, not the sender.

The following diagram shows the relationship between you as both the sender and the application owner that directs payments to receivers:



For example, you might use this configuration in a sales commission application that transfers funds owed for commissions from your account to the accounts of your sales force.

Simple, Parallel, and Chained Payments

Adaptive Payments provides several kinds of payment: simple, parallel, and chained payments. You create each kind of payment with the `Pay` API.

- Simple payments enable a sender to send a single payment to a single receiver.

  For example, your website can use an Adaptive Payments payment flow to transfer money resulting from a sale from your customer's PayPal account to your own account. This is the traditional kind of payment.

- Parallel payments enable a sender to send a single payment to multiple receivers.

  For example, your application might be a shopping cart that enables a buyer to pay for items from several merchants with one payment. Your shopping cart allocates the payment to merchants that actually provided the items. PayPal then deducts money from the sender's account and deposits it in the receivers' accounts.

- Chained payments enable a sender to send a single payment to a primary receiver. The primary receiver keeps part of the payment and pays secondary receivers the remainder.

  For example, your application could be an online travel agency that handles bookings for airfare, hotel reservations, and car rentals. The sender sees only you as the primary receiver. You allocate the payment for your commission and the actual cost of services provided by other receivers. PayPal then deducts money from the sender's account and deposits it in both your account and the secondary receivers' accounts.

  NOTE: Chained payments also include delayed chained payments, in which payments to secondary receivers can be delayed for up to 90 days.

Simple Payments

Simple payments enable a sender to send a single payment to a single receiver. This is sometimes considered a traditional payment, such as a payment from a buyer to a seller.

Scenarios involving simple payments might include the following scenarios:

- Buyer makes a payment on a merchant's website.
- Buyer makes a single payment for a cart of items from the same merchant.
- Person on a social networking site makes a payment for a purchase to the receiver. For example, a sender sends money to pay for lunch at a restaurant.

In these cases, the sender sends a payment to a single receiver. The following example shows a sender making a simple payment:

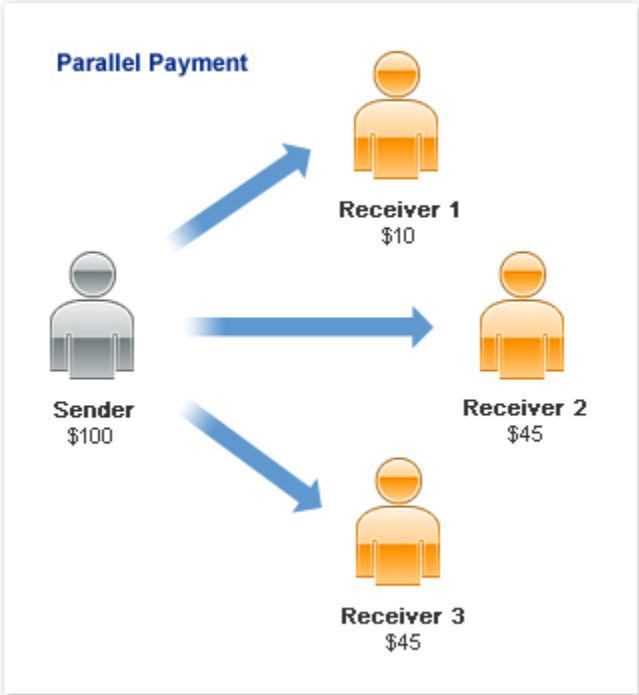
Parallel Payments

A parallel payment is a payment from a sender that is split directly among 2-6 receivers. Technically, a parallel payment is a set of multiple payments made in a single `Pay` request.

Parallel payments are useful in cases when a buyer intends to make a single payment for items from multiple sellers. Examples include the following scenarios:

- a single payment for multiple items from different merchants, such as a combination of items in your inventory and items that partners drop ship for you.
- purchases of items related to an event, such as a trip that requires airfare, car rental, and a hotel booking.

In these cases, the sender knows the receivers and the amount paid to each one. The following example shows a sender paying 3 receivers in a single parallel payment:



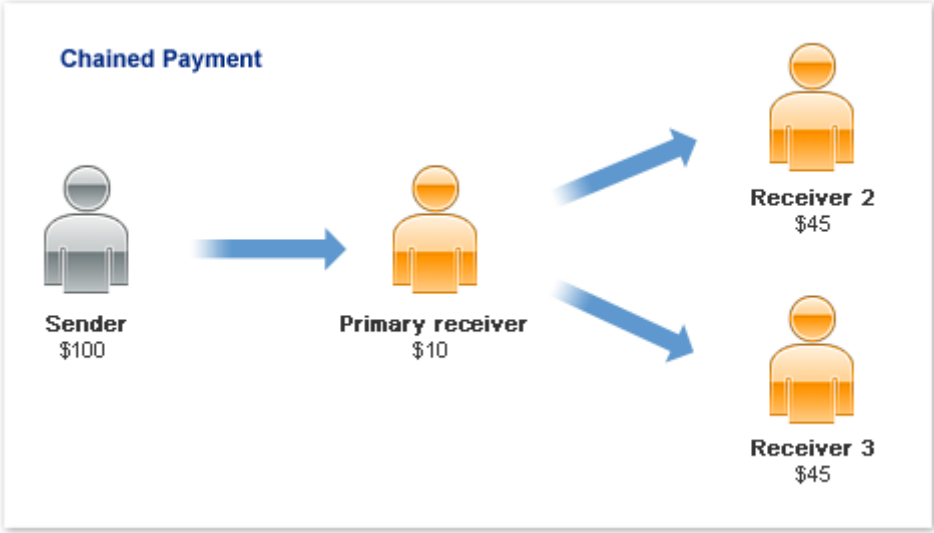
NOTE: This scenario is an example only and does not take PayPal fees into account.

Chained Payments

A chained payment is a payment from a sender that is indirectly split among multiple receivers. It is an extension of a typical payment from a sender to a receiver, in which a receiver, known as the *primary receiver*, passes part of the payment to other receivers, who are called *secondary receivers*.

NOTE: The API caller must get permission from PayPal to use chained payments.

You can have at most one primary receiver and 1-5 secondary receivers. Chained payments are useful in cases when the primary receiver acts as an agent for other receivers. The sender deals only with the primary receiver and does not know about the secondary receivers, including how a payment is split among receivers. The following example shows a sender making a payment of $100:



In this example, the primary receiver receives $100 from the sender's perspective. The primary receiver actually receives only $10 and passes a total of $90 to secondary receivers, Receiver 2 and Receiver 3.

NOTE: This scenario is an example only and does not take PayPal fees into account.

Delayed Chained Payments

By default, payments to all receivers in a chained payment are immediate. However, you can choose to delay a payment to a secondary receiver. For example, as primary receiver, you may require secondary receivers to perform some action, such as shipping goods or waiting for expiration of a return period, before making payment. To complete the payment, you must explicitly execute a payment to secondary receivers after the sender pays you. The payment must occur within 90 days, after which you cannot complete the payment as part of the original chained payment.

Payment Approval

The sender of a payment must approve the transfer. The sender can log in to paypal.com to approve each payment, preapprove payments, or when the sender is your application, be implicitly approved to make payments.

NOTE: Preapproval requests require permission from PayPal.

There are 3 kinds of payment approvals:

- Explicit approval payments, in which the sender logs in to paypal.com to approve each payment.

  Explicitly approving payments is the traditional way to pay with PayPal. This method is the only option unless the sender has set up a preapproval agreement or you, the API caller, are also the sender.

- Preapproved payments, in which a sender logs in to PayPal and sets up preapprovals that approve future payments or set up a preapproval during the embedded payment flow.

  The sender logs in to paypal.com once to set up the preapproval. After the sender agrees to the preapproval, specific approval is unnecessary.
- Implicit approval payments, in which your application is both the sender of a payment and the caller of the Adaptive Payments `Pay` API.

  In this case, PayPal makes the payment from your own account, which eliminates the need for approval.

Adaptive Payments Service Permissions

Adaptive Payments services are divided into 2 categories: *standard services* that do not require specific permission to use and *advanced services* that require permission from PayPal to use. You obtain permission to use a service when you submit an application to PayPal.

You can use the following standard services without requesting specific permission:

- Making simple or parallel payments that require explicit approval of the sender
- Getting payment details
- Making refunds
- Performing currency conversions

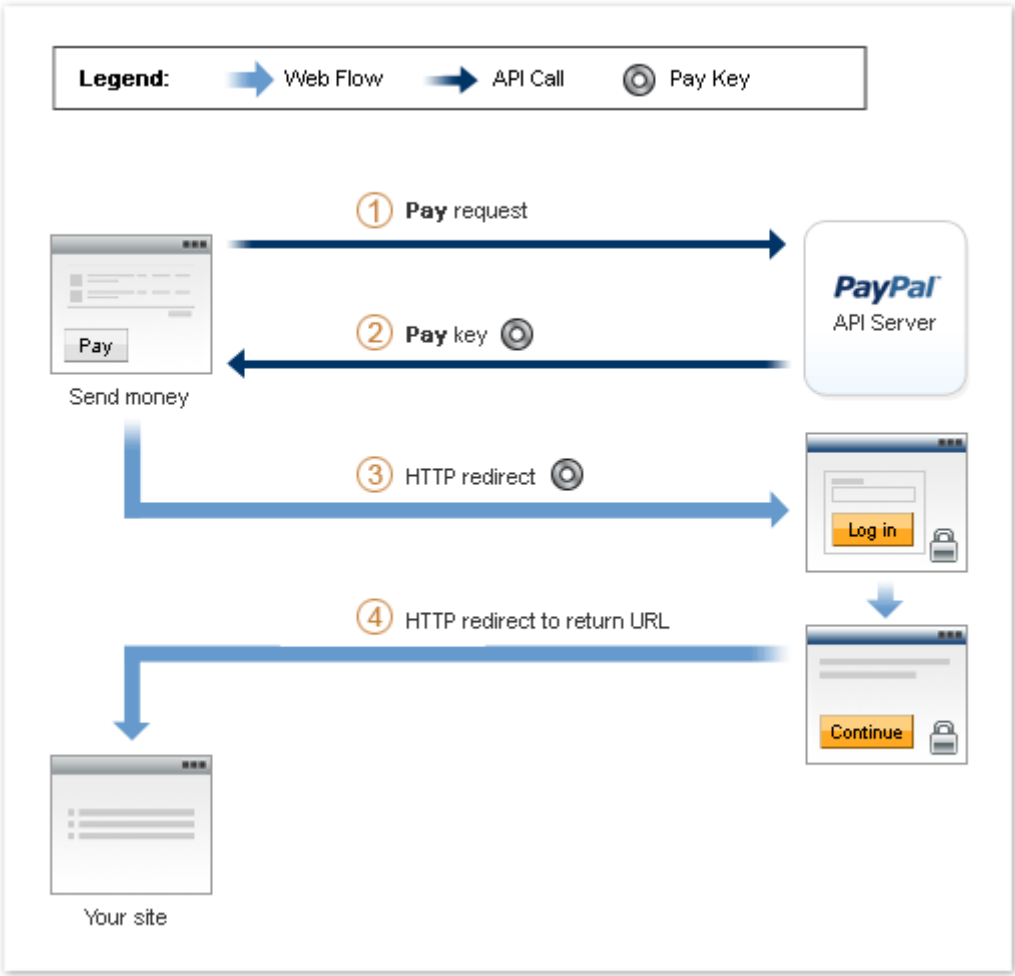
To use any other service, you must receive permission from PayPal to use the service when you submit your application. For example, if your application makes chained payments, which is an advanced service, you request permission when you submit your application. Later, if you modify your application to support preapprovals, which is also an advanced service, you must resubmit your application to obtain permission from PayPal to use the service.

IMPORTANT: If you allow a third party to PayPal to execute an application on your behalf, the third party becomes the API caller because the party is now calling the Adaptive Payments API. The third party must also have permission from PayPal to use the advanced service. For example, if an application supports chained payments, both you and the third party must have permission to use the service.

Explicit Approval Payment Flow

Explicit approval payments require the sender to log in to paypal.com to approve the payment. You control the interaction between your application and PayPal by specifying URLs for redirection in various situations.

The following diagram shows the basic flow of control for a payment (such as payment for items in a shopping cart at a web store) that requires the sender to log in to paypal.com to approve the payment:



The following items correspond to the circled numbers in the diagram:

1. Your site or device sends a `Pay` request to PayPal on behalf of a sender.
2. PayPal responds with a key that you use when you direct the sender to PayPal.
3. You redirect your sender's browser to paypal.com.
4. After the sender authorizes the transfer of funds, PayPal redirects your sender's browser to the location you specify.

NOTE: The cancel operation is not shown in the diagram. Cancellation is handled by a separate cancellation URL to which PayPal redirects the sender's browser any time the sender cancels while on paypal.com.

In addition to these steps, PayPal notifies you and the sender of the payment.

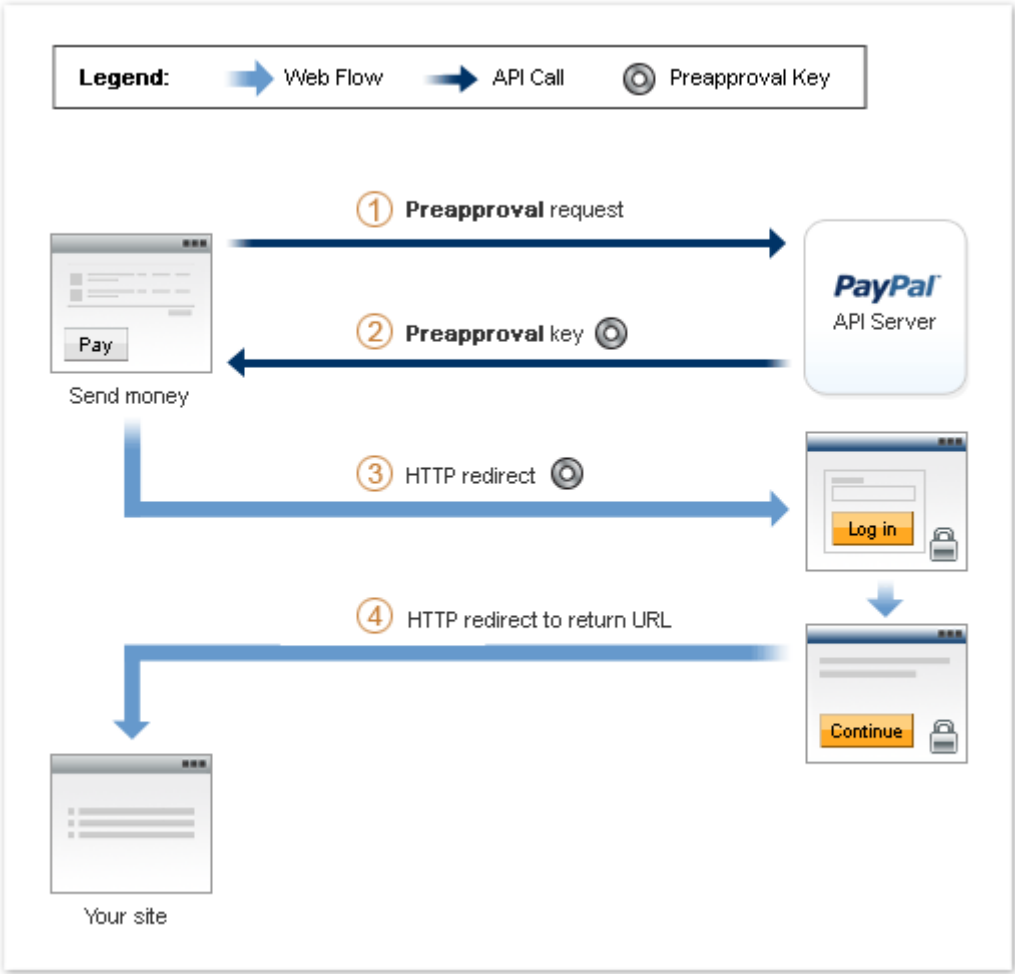
Preapproved Payments Flow

Preapproved payments require the sender to log in to paypal.com to set up the payment agreement with a particular vendor. You control the interaction between your application and PayPal by specifying URLs for redirection in various situations.

The sender logs in to paypal.com and sets up the preapproval, which includes setting the following information:

- duration of the preapproval, from the start date to the end date, inclusive
- the maximum amount being preapproved
- the maximum number of payments

The following diagram shows the basic flow of control during a preapproval operation:



The following items correspond to the circled numbers in the diagram:

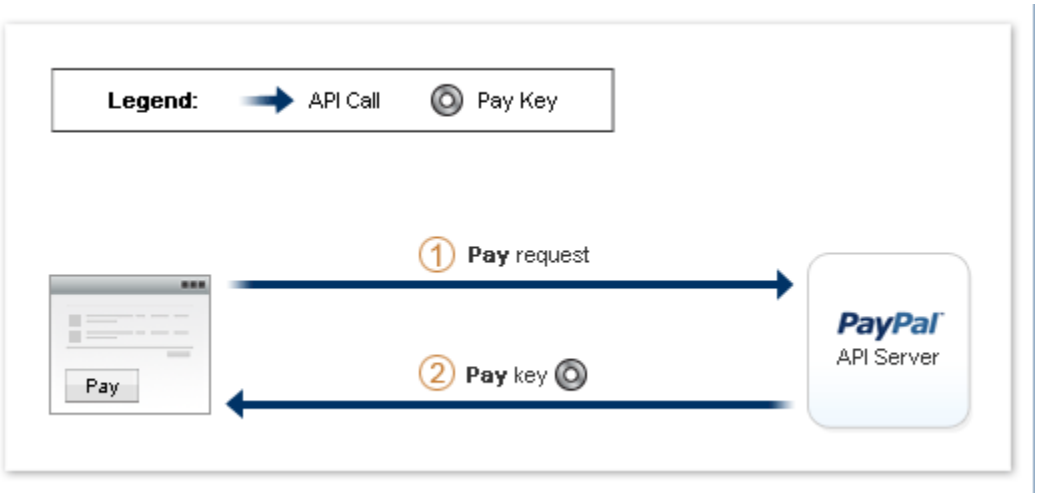
1. Your site or device sends a `Preapproval` request to PayPal on behalf of a sender.

2. PayPal responds with a key, called a *preapproval key*, that you use when you direct the sender to PayPal, and once the preapproval has been established, whenever you automatically complete a payment on behalf of the sender.
3. You redirect your sender's browser to PayPal.
4. After the sender logs in to paypal.com and sets up the preapproval, PayPal redirects the sender's browser to a location you specify.

NOTE: The cancel operation is not shown in the above diagram. Cancellation is handled by a separate cancellation URL to which PayPal redirects the sender's browser any time the sender cancels while on paypal.com.

In addition to the steps shown above, PayPal sends an email notifying you and the sender that the preapproval has been created.

After the sender sets up the approval, you can make payments on the sender's behalf directly. The sender is no longer required to log in to PayPal to complete the payment. The following diagram shows the basic flow of control during a `Pay` operation:



The following items correspond to the circled numbers in the diagram:

1. Your site or device sends a `Pay` request to PayPal on behalf of a sender. You can require the sender to provide a personal identification number (PIN); however, logging in to paypal.com is no longer required.

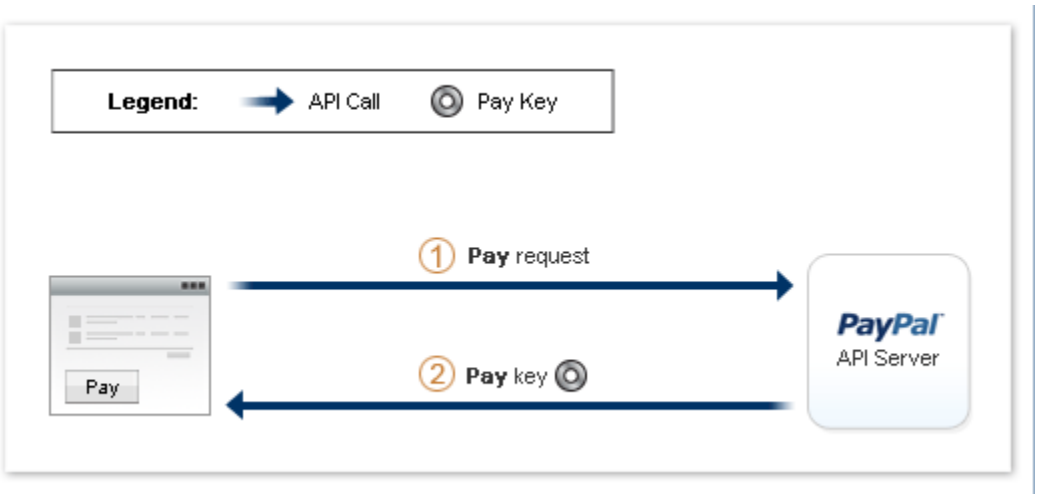
   NOTE: You must provide a preapproval key that identifies the agreement.

2. PayPal still responds with a payment key that you can use for other API operations, such as for obtaining details of the payment or for issuing a refund.

Implicit Approval Payments Flow

Implicit approval payments are payments where the sender and the API caller are using the same account. Because PayPal draws the funds for the payment from your own account, there is no approval necessary, and as such there is no visible flow for implicit approval payments.

The following diagram shows the basic flow of control during an implicitly approved payment operation:



The following items correspond to the circled numbers in the diagram:

1. Your site or device sends a `Pay` request to PayPal.

   NOTE: A web flow is not required.

2. PayPal responds with a key that you use for other API operations.

Embedded Payments

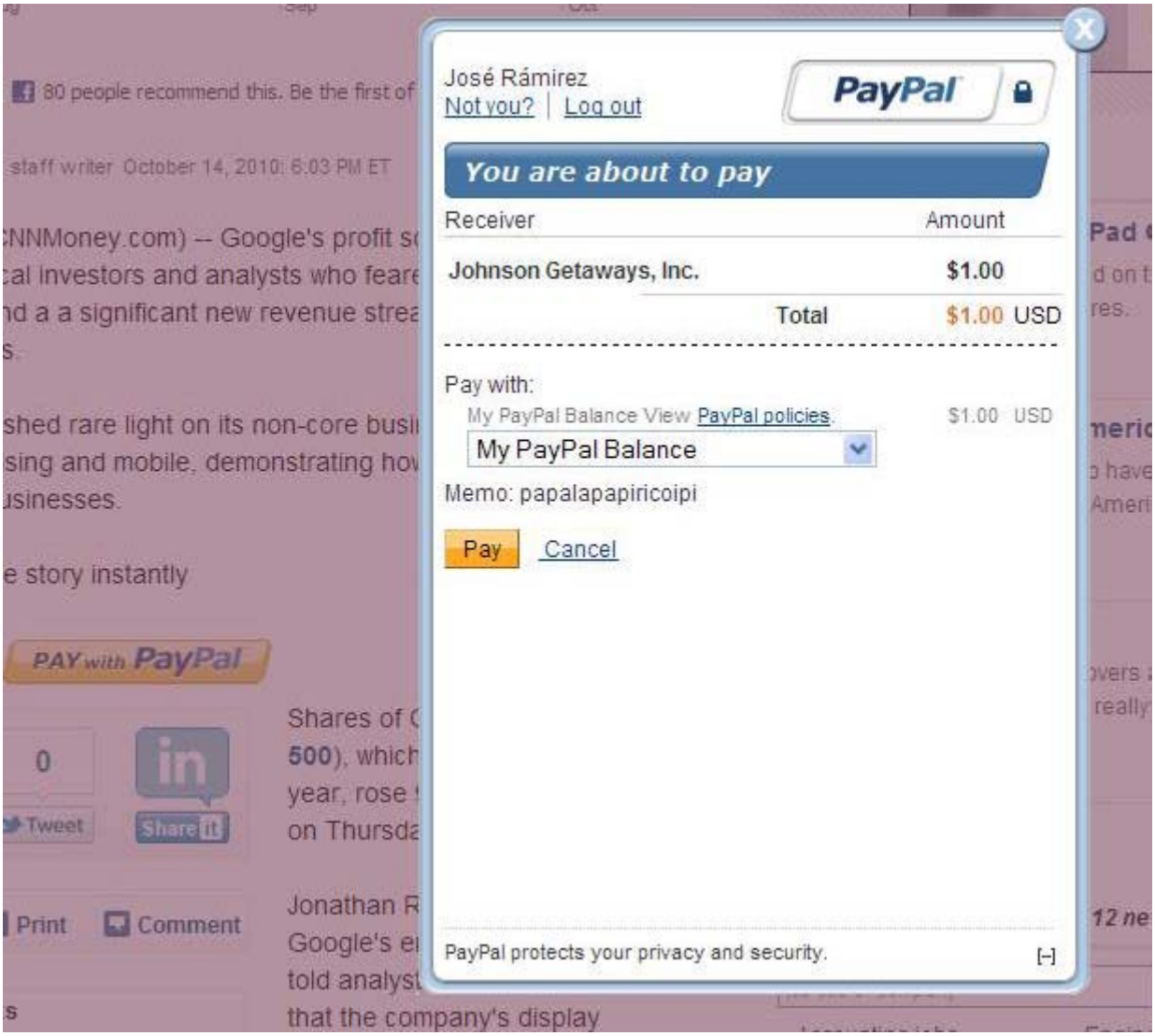
An *embedded payment* is a payment that initiates a visual presentation of the Adaptive Payments payment flow in which the sender appears to never leave your checkout or payment page. Embedded payments make it easier for a sender to make a payment because PayPal may allow the sender to bypass the PayPal login step.

The ability to bypass the login relies on a *remember me* cookie, which is stored in the sender's browser when the sender chooses to take advantage of being remembered. Opting in reduces the number of steps to purchase goods or services without significantly increasing the risk that the sender's account might be misused. After the initial login, PayPal bypasses the login step if subsequent payments meet specific criteria, such as a subsequent payment for a small amount.

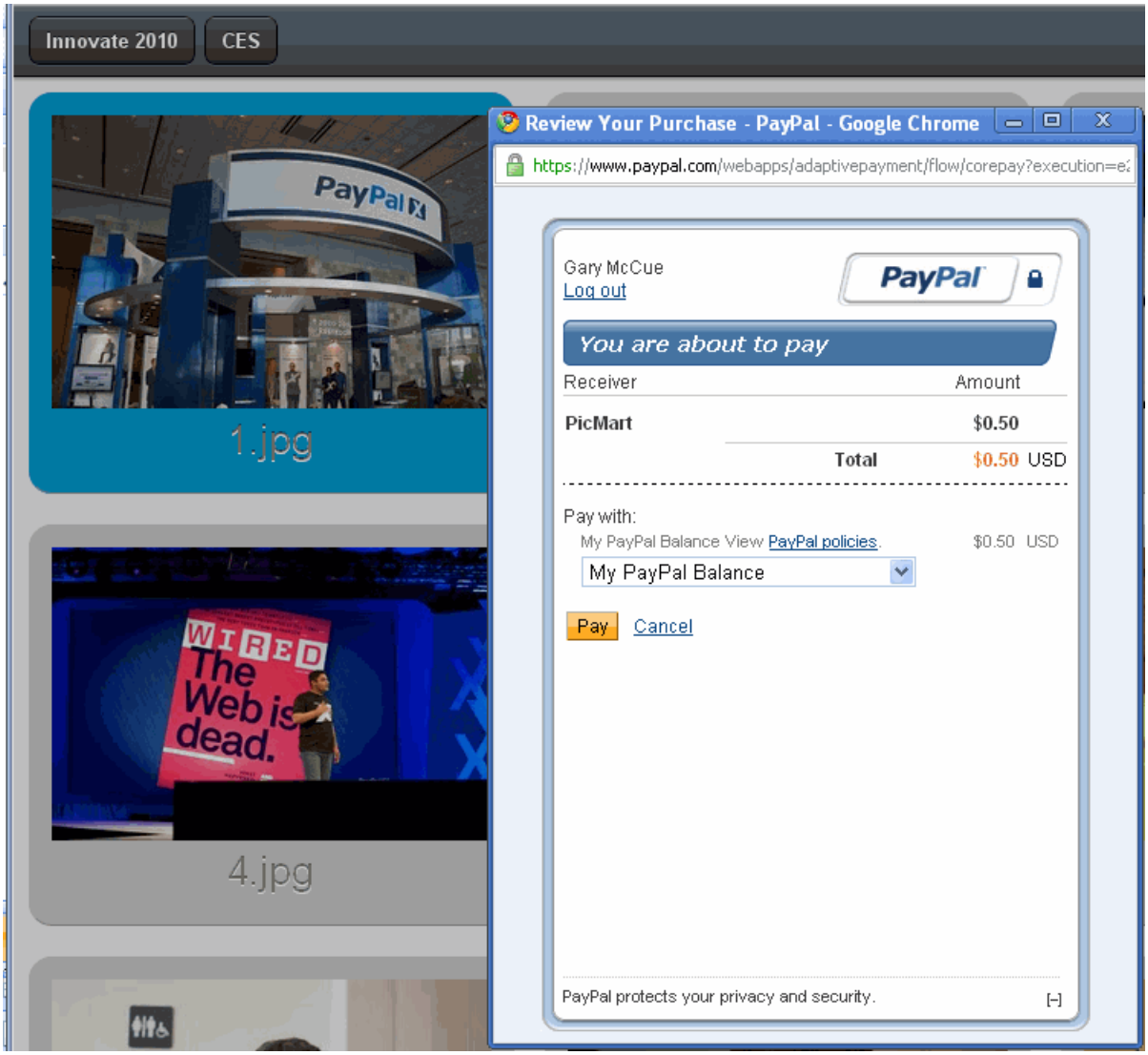
Embedded Payment Flow Presentations

PayPal provides the following kinds of visual presentations for the embedded payment flow:

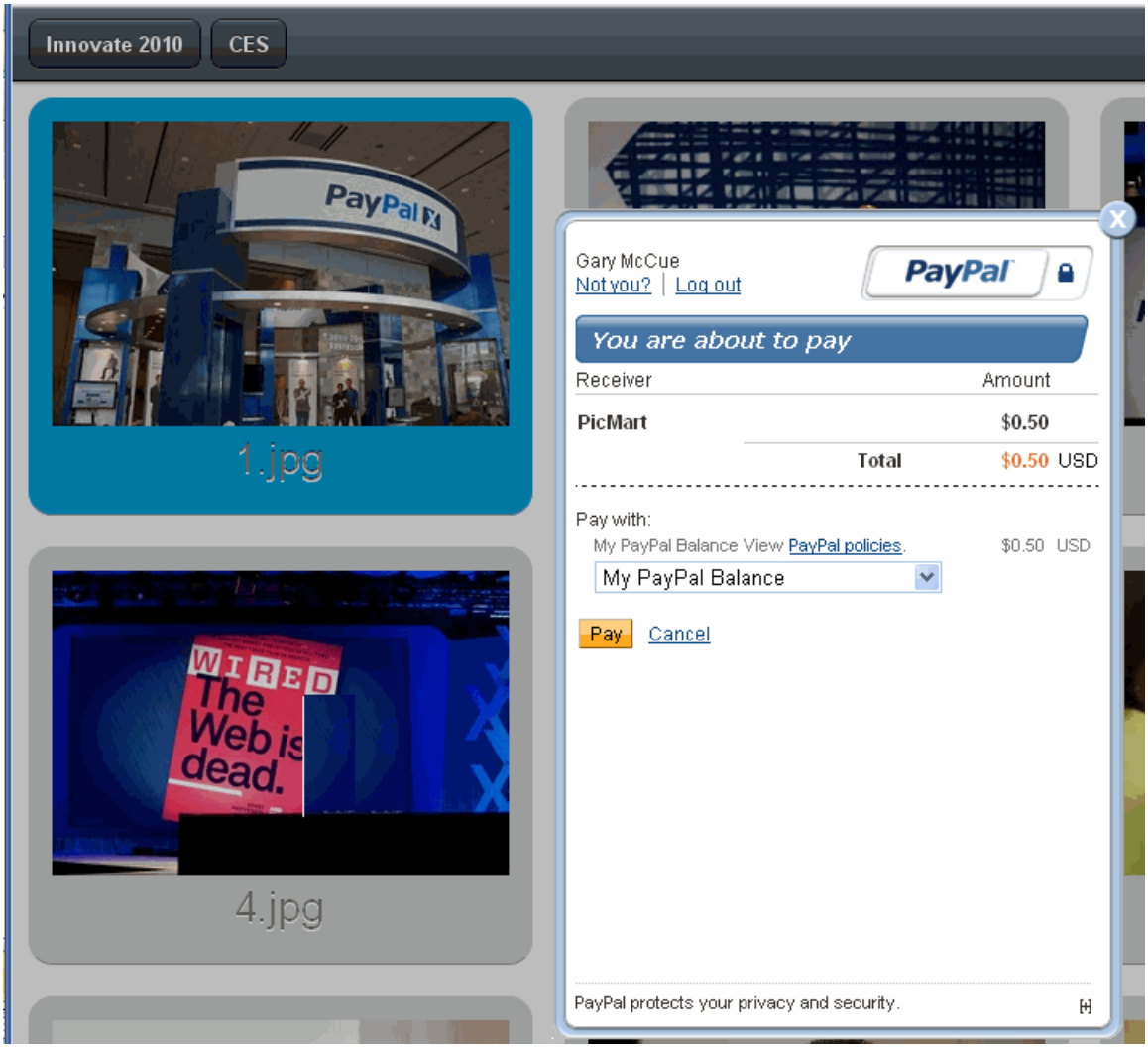
- The payment flow can be embedded as a lightbox on your web page, which gives the impression that the payment is handled completely within your website:



- The payment flow can appear in a minibrowser window that appears in front of your web page, which does not require an IFRAME:



- The payment flow can be embedded as a lightbox in an IFRAME on your web page, which gives the impression that the payment is handled completely within your website:



You choose your preferred visual presentation when you invoke the embedded payment flow. In some cases, PayPal may override your choice to use a lightbox; for example, when the sender is required to log into PayPal for the initial payment.

Kinds of Embedded Payments

Embedded payments can include

- simple payments
- parallel payments
- chained payments

You can also enable preapprovals for future payments or enable shipping addresses to be associated with embedded payments.

IMPORTANT: Payments for digital goods must use the embedded payment flow. You cannot associate a preapproval for future payments or enable shipping addresses in a payment for digital goods.

Embedded Payments Implementation Basics

To implement the embedded payment flow, you must

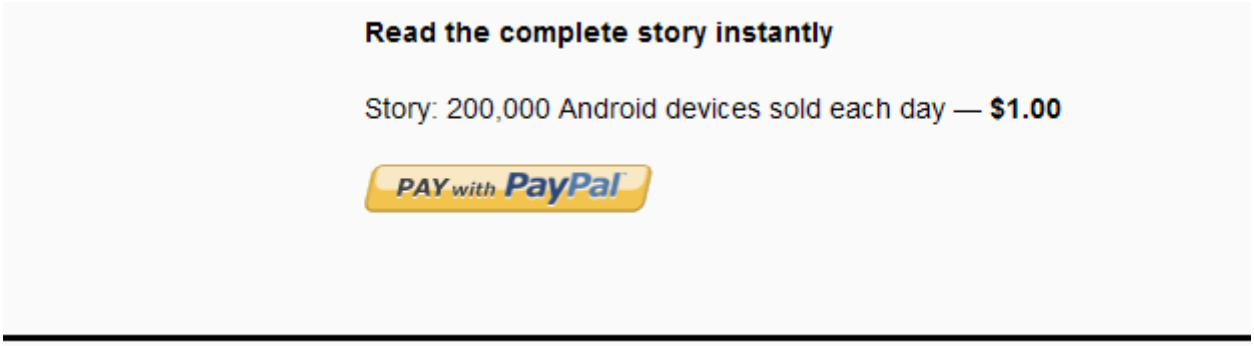
- include JavaScript code from PayPal on your checkout or payment web pages
- use the functions provided in the JavaScript to coordinate the PayPal flow with the appearance of your web pages
- launch your preferred embedded payment flow, which is either the lightbox or minibrowser, and redirect the sender's browser to the PayPal URL that supports embedded payments, which is `https://www.paypal.com/webapps/adaptivepayment/flow/pay?paykey=...`

You must call the `Pay` API operation to obtain a payment key before launching the embedded payment flow. If the payment is specifically for digital goods, modify your `Pay` API operation to specify that each receiver is receiving payment for digital goods.

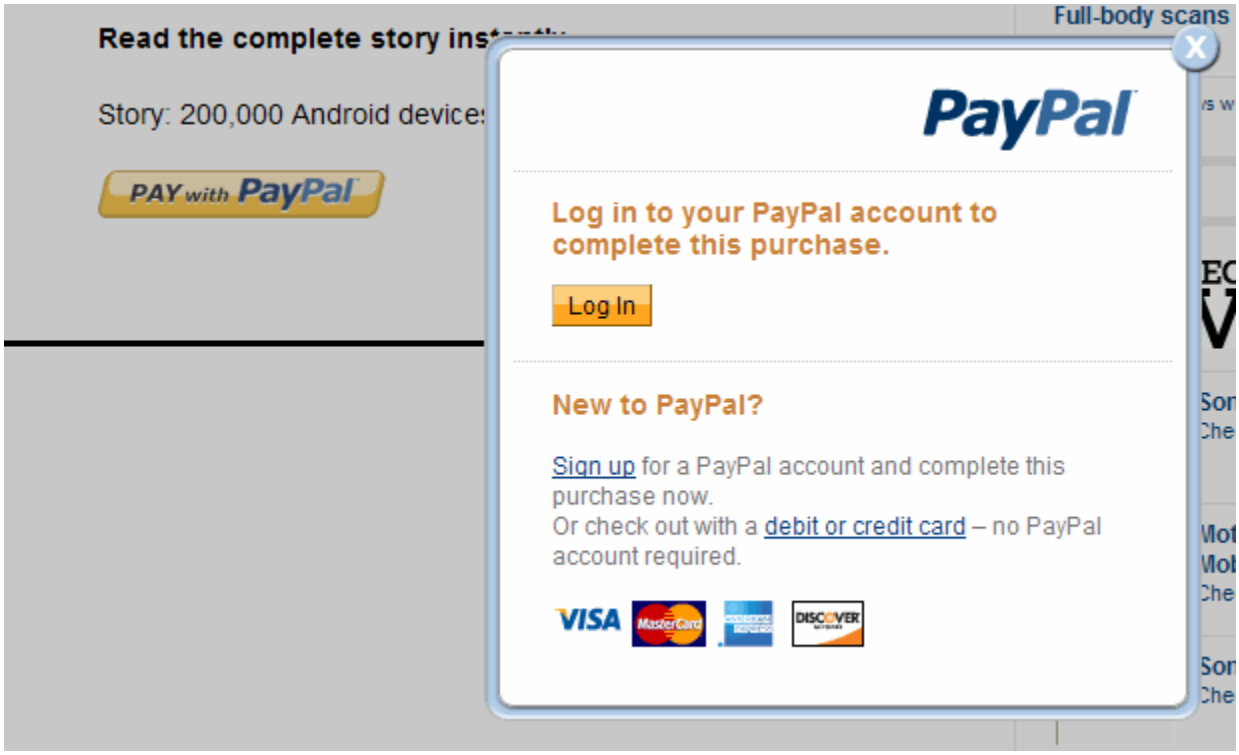
Embedded Payment Experience

To the sender of a payment, the embedded payment experience appears to be built into your website. The PayPal-supplied JavaScript provides all the code needed to set up the flow as an IFRAME within the sender's browser and as a pop-up mini-browser that appears in front of your website.

Typically, the sender initiates a payment by clicking a button:



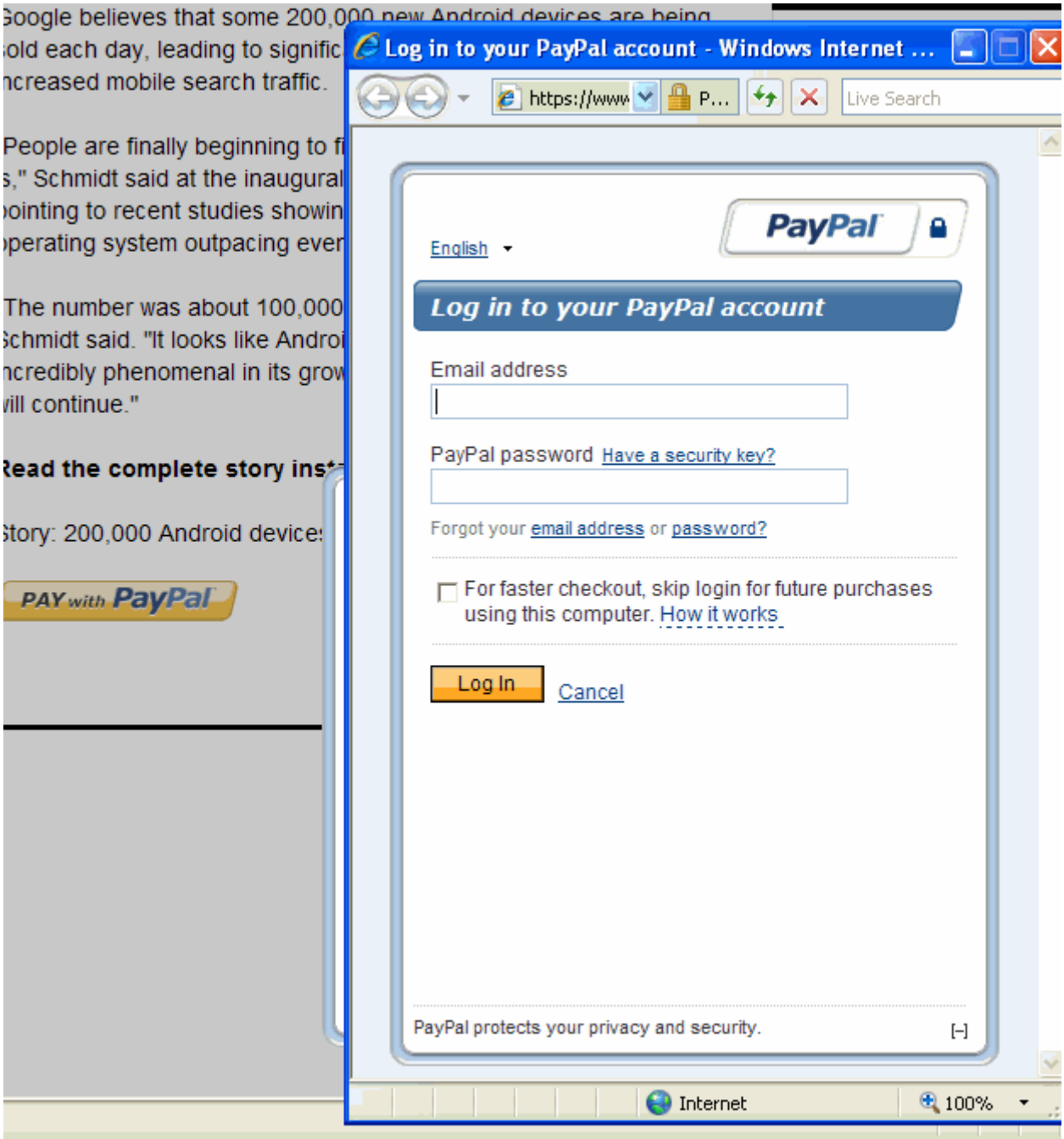
PayPal responds to the JavaScript that initiates the flow. If it is the first payment, or if PayPal determines that the payment requires the sender to log in, PayPal displays a Log In button in the IFRAME created by the JavaScript:



The IFRAME also allows the sender to sign up for a PayPal account or pay as a guest without logging in.

NOTE: Guest checkout only provides the visual benefits of an embedded payment. It does not reduce the payment steps.

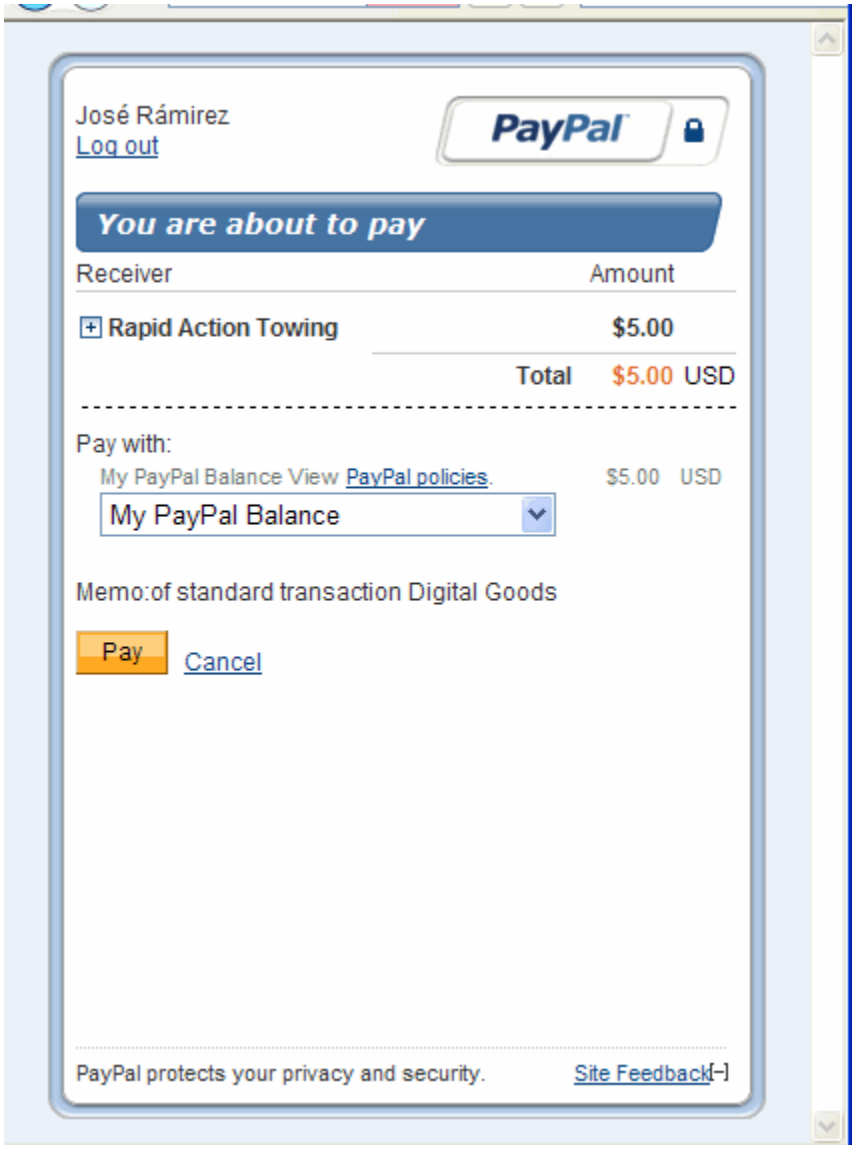
After clicking Log In, PayPal displays a **Log in to your PayPal account** page in the minibrowser. The sender enters an email address and password and can also check a box, which allows the sender to skip this step for subsequent purchases:



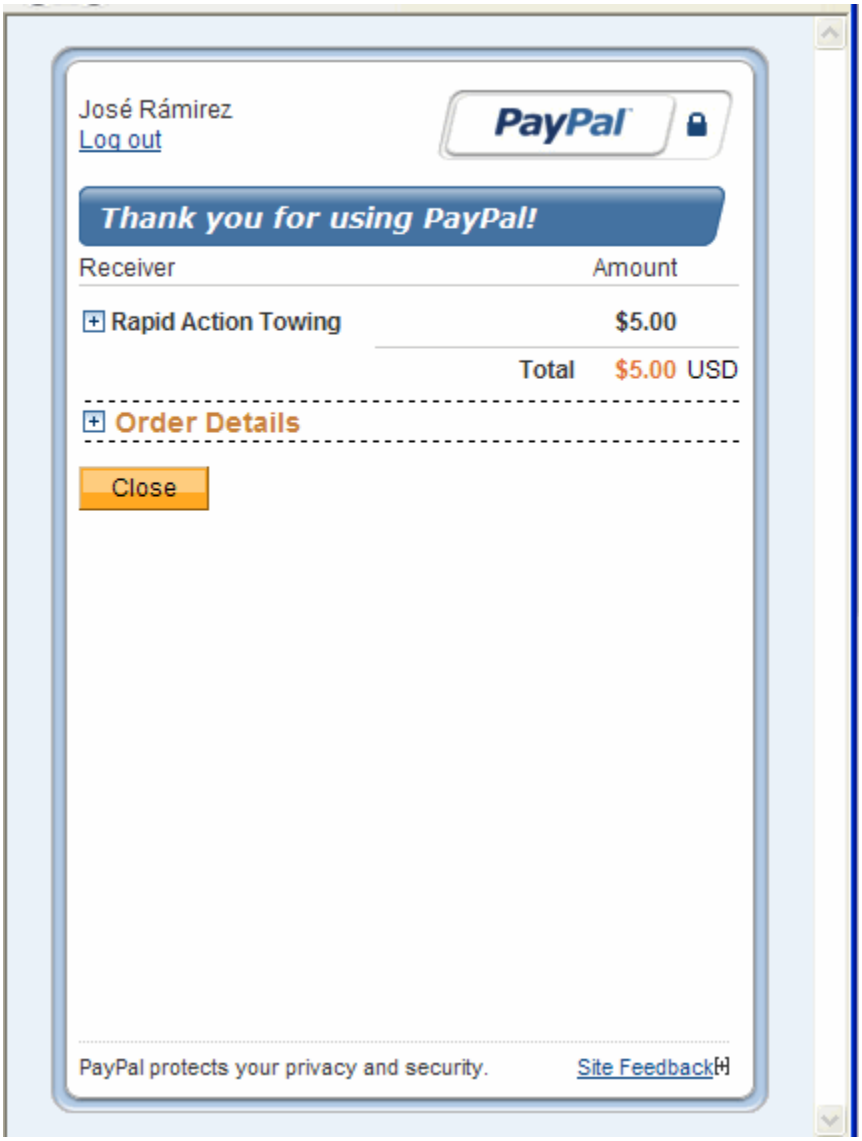
The checkbox controls the *remember me* behavior for log in. This behavior allows the sender to skip the log in step in cases where there is little risk of the account being misused.

IMPORTANT: Opting in to the *remember me* behavior does not guarantee that the sender will not be asked to provide log in credentials.

After logging in, PayPal displays the **You are about to pay** page, sometimes called the *review your payment* page in the minibrowser:



If the sender chooses **Cancel**, PayPal redirects the sender's browser to the cancel URL specified in the Pay API operation's request message. If the sender chooses **Pay**, the **Thank you for using PayPal** page appears in the minibrowser:

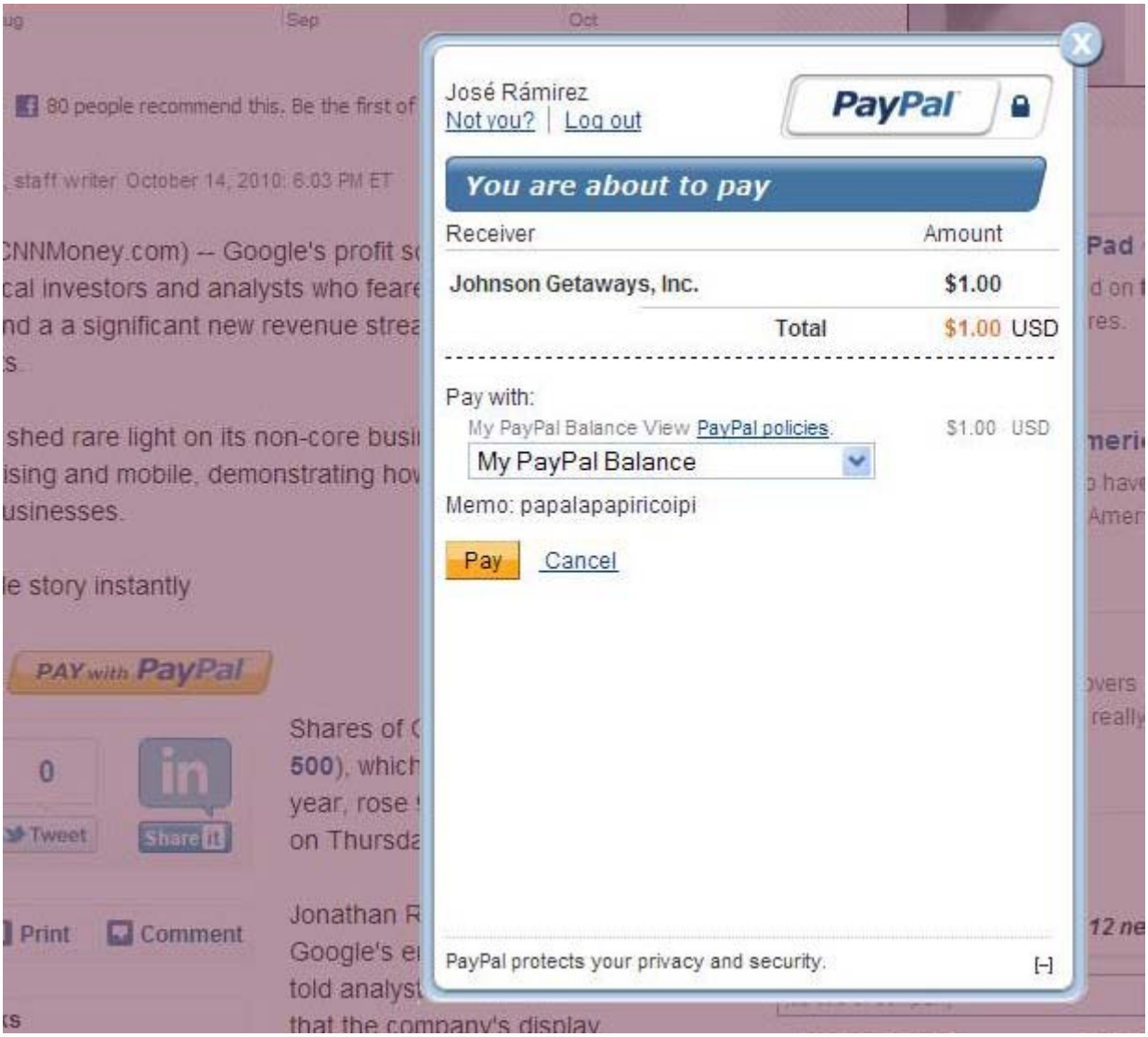


When the sender clicks **Close**, PayPal redirects the sender's browser to the return URL specified in the Pay API operation's request message.

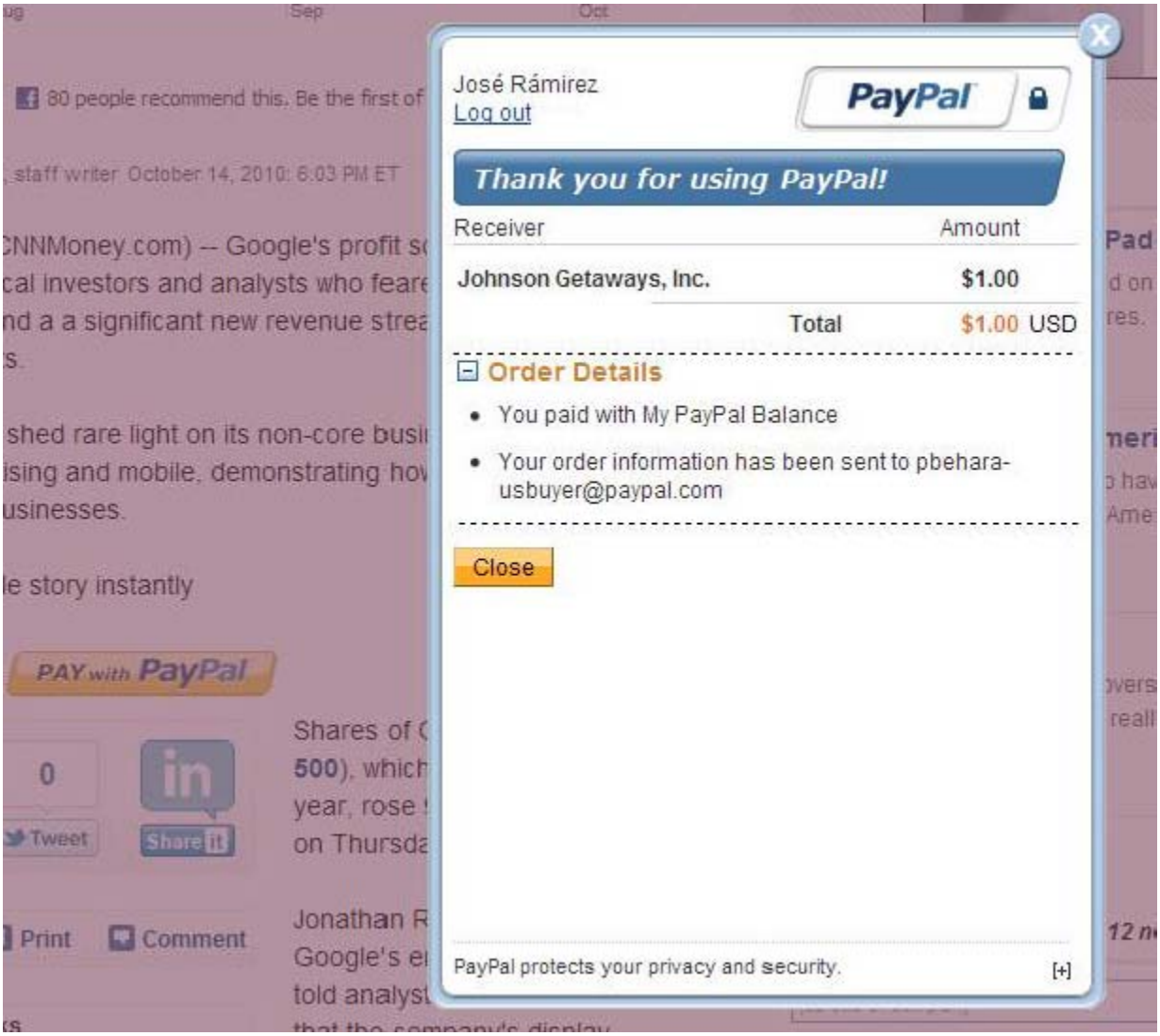
NOTE: You are responsible for closing the minibrowser after PayPal redirects to the page specified in either the return or cancel URL. PayPal provides a JavaScript function that you call to close a PayPal minibrowser or lightbox.

For subsequent payments in which PayPal does not require the account holder to log in again and the account holder has chosen to be remembered, the PayPal payment pages appear in a lightbox rather than in a minibrowser and PayPal bypasses the log in steps. For these payments, the actions you take to launch the flow and close the lightbox, remain the same.

For example, the lightbox for reviewing a payment would appear in your page as follows:

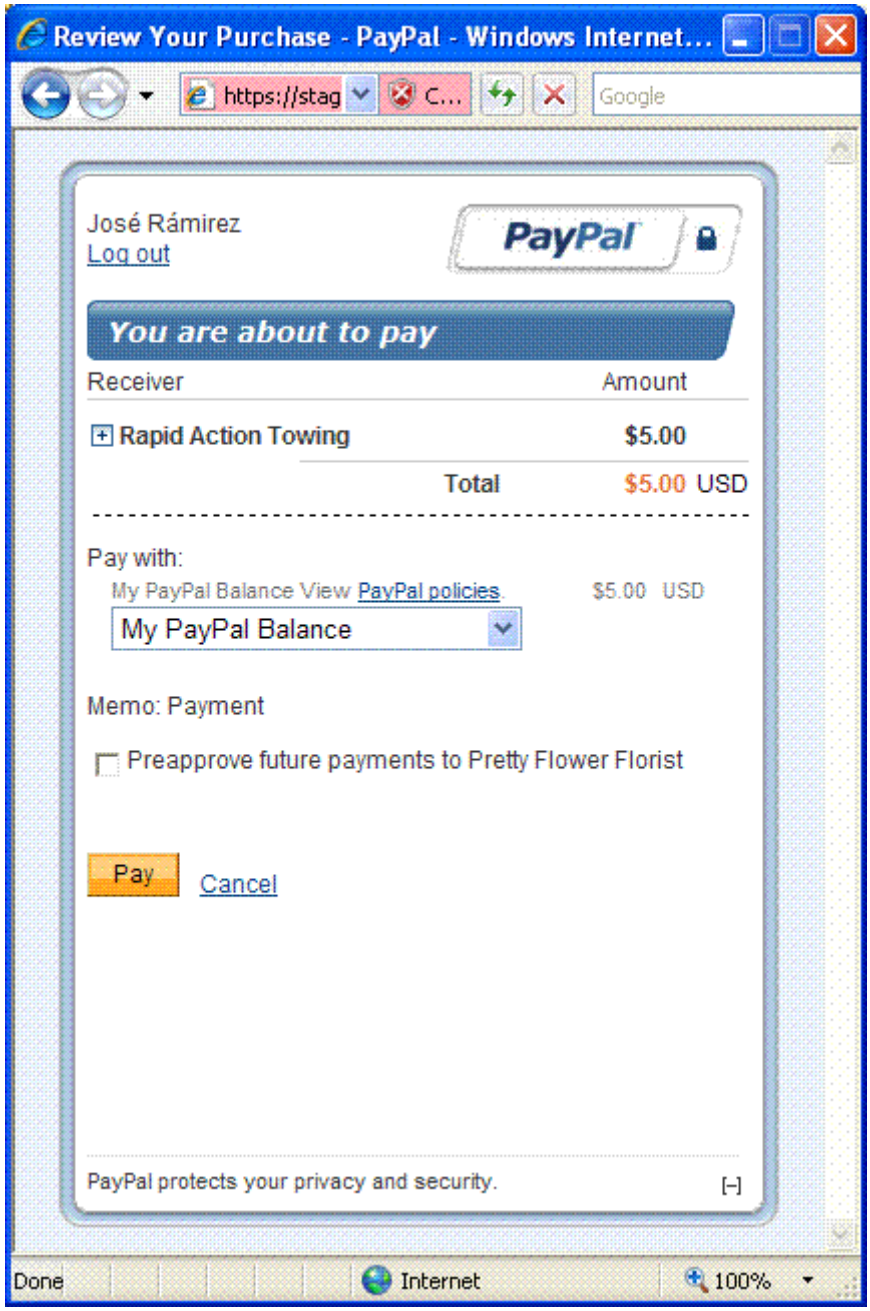


The lightbox containing the confirmation would appear in your page as follows:

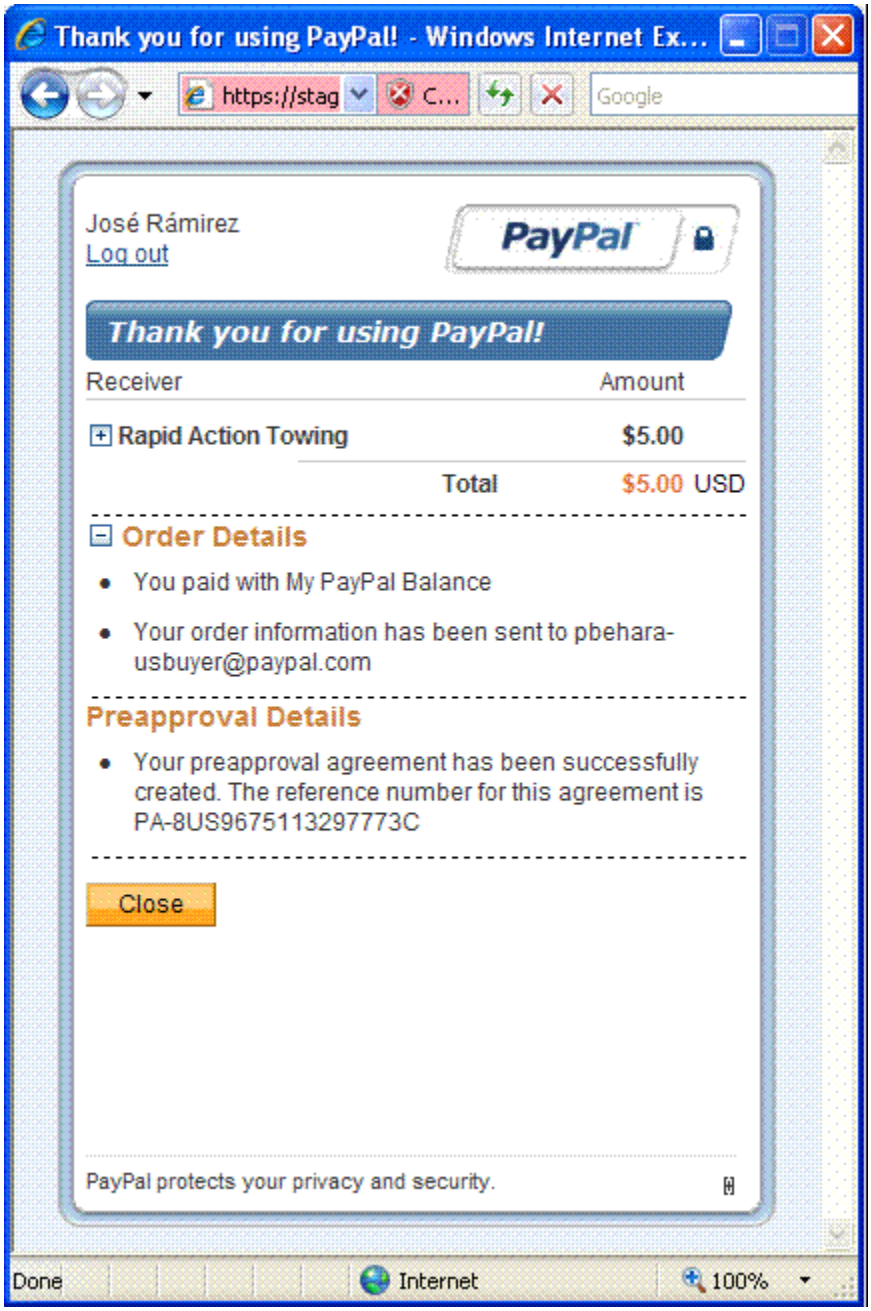


Preapprove Future Payments Checkbox

You can add a **Preapprove future payments** checkbox to the sender's embedded payment experience, which enables the sender to preapprove subsequent payments. If you specify both a payment key and a preapproval key when you call the `Pay` API operation, PayPal displays the checkbox on the payment page in the minibrowser:



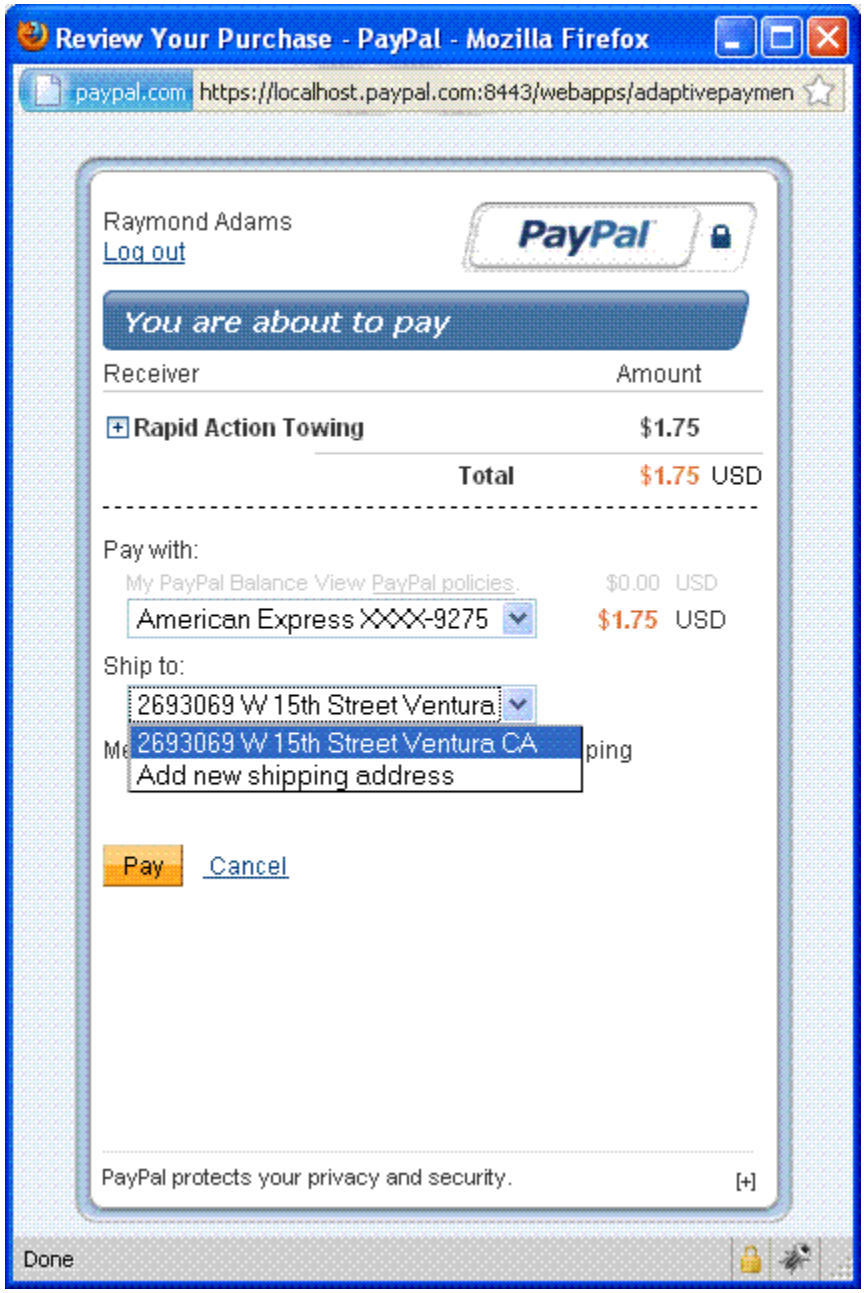
If the payment sender checks the preapproval box, the confirmation page provides details about the preapproval:



NOTE: Unless there is an error with the payment itself, PayPal transfers the money regardless of whether the preapproval for future payments was successful.

Shipping Address Information

You can display and collect shipping address information. PayPal displays the default shipping address when you set `senderOptions.requireShippingAddressSelection` to `true` in your request to `SetPaymentOptions`:



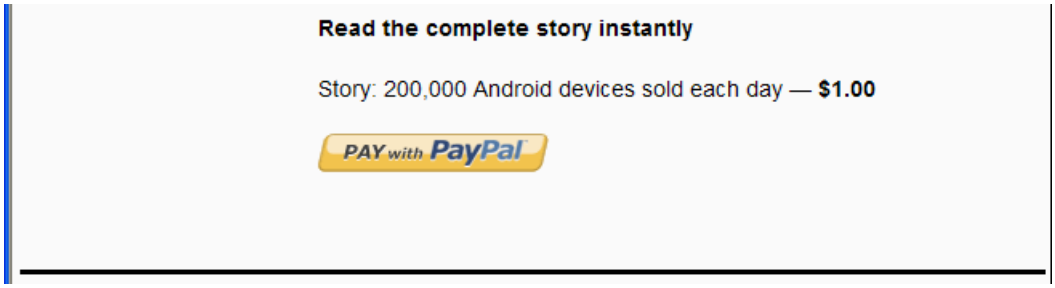
The sender of a payment can select one of the available shipping addresses or add a new shipping address by selecting **Add new shipping address** from the **Ship to:** drop-down menu:

Embedded Payment Experience

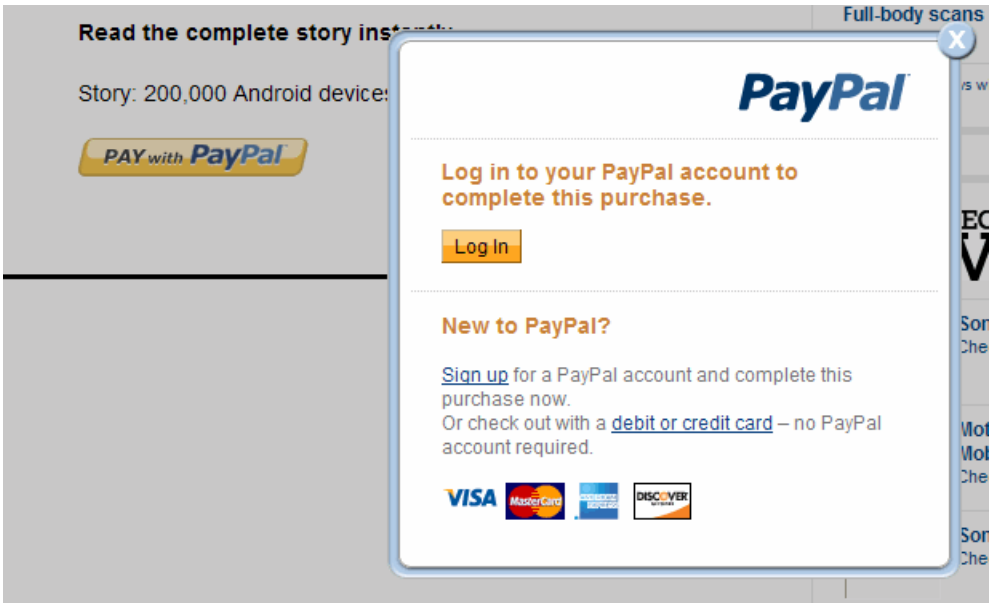
To the sender of a payment, the embedded payment experience appears to be built into your website. The PayPal-supplied JavaScript provides all the code needed to set up the flow as an

IFRAME within the sender's browser and as a pop-up mini-browser that appears in front of your website.

Typically, the sender initiates a payment by clicking a button:



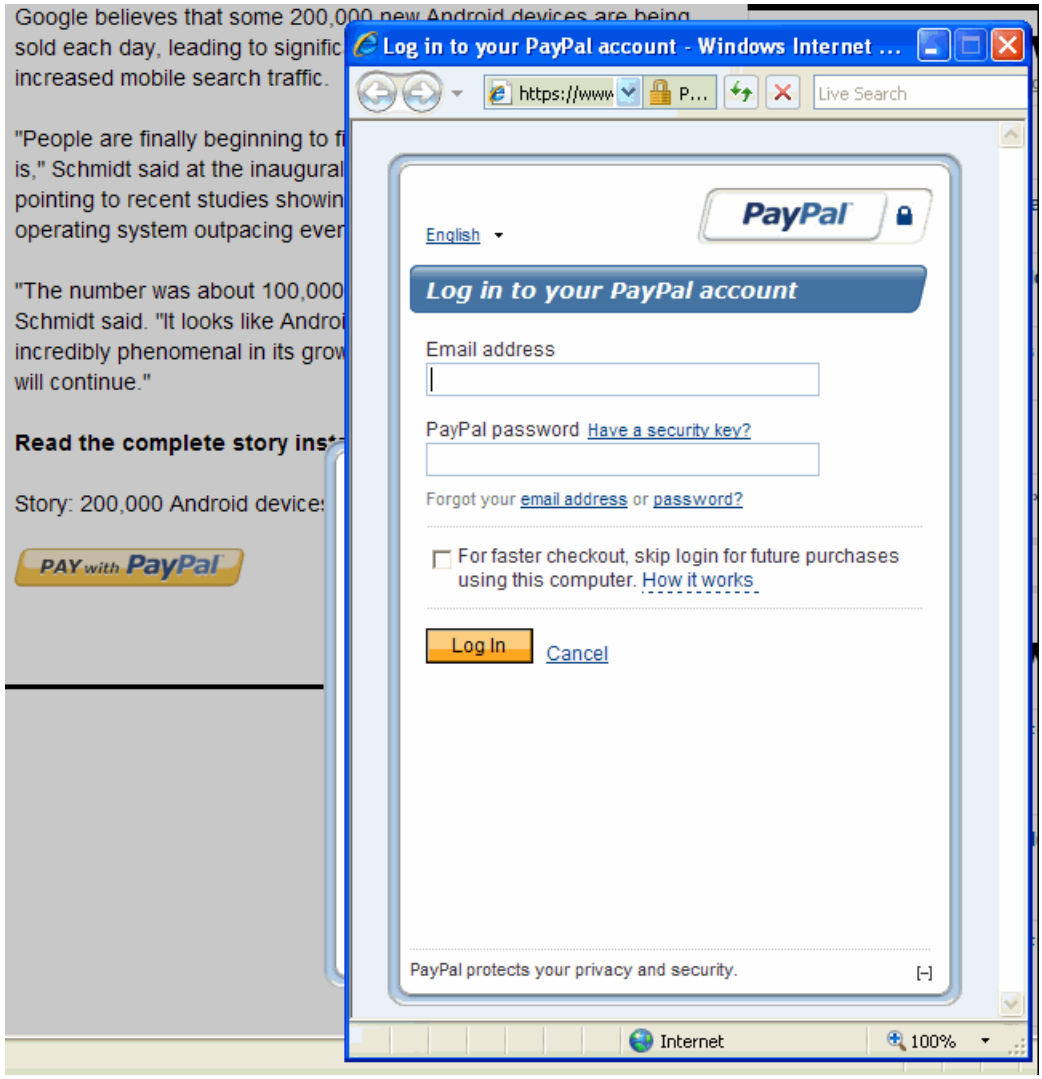
PayPal responds to the JavaScript that initiates the flow. If it is the first payment, or if PayPal determines that the payment requires the sender to log in, PayPal displays a **Log In** button in the IFRAME created by the JavaScript:



The IFRAME also allows the sender to sign up for a PayPal account or pay as a guest without logging in.

NOTE: Guest checkout only provides the visual benefits of an embedded payment. It does not reduce the payment steps.

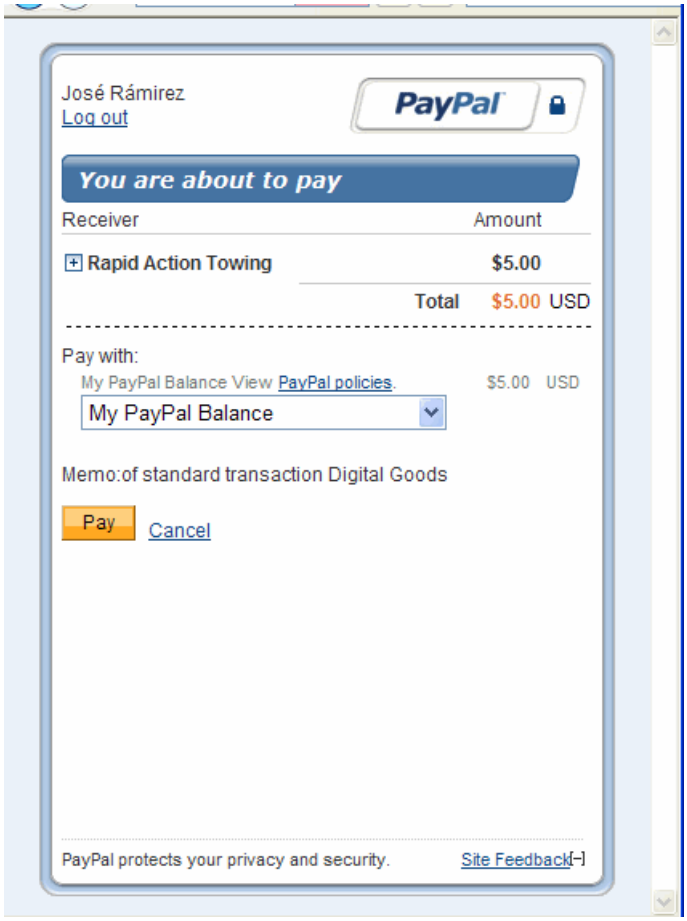
After clicking Log In, PayPal displays a **Log in to your PayPal account** page in the minibrowser. The sender enters an email address and password and can also check a box, which allows the sender to skip this step for subsequent purchases:



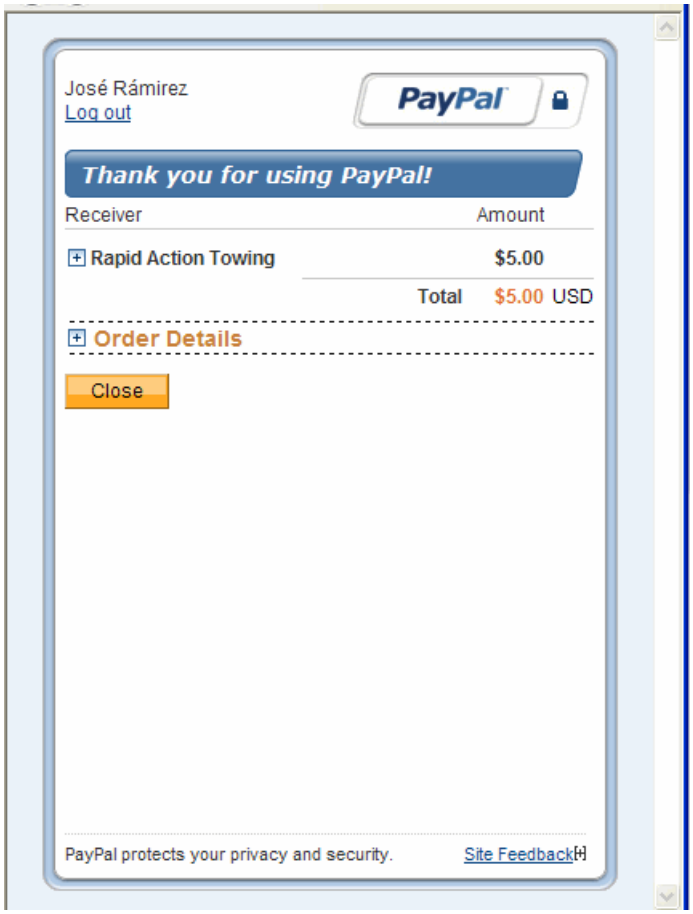
The checkbox controls the *remember me* behavior for log in. This behavior allows the sender to skip the log in step in cases where there is little risk of the account being misused.

IMPORTANT: Opting in to the remember me behavior does not guarantee that the sender will not be asked to provide log in credentials.

After logging in, PayPal displays the **You are about to pay** page, sometimes called the *review your payment* page in the minibrowser:



If the sender chooses **Cancel**, PayPal redirects the sender's browser to the cancel URL specified in the Pay API operation's request message. If the sender chooses **Pay**, the **Thank you for using PayPal** page appears in the minibrowser:

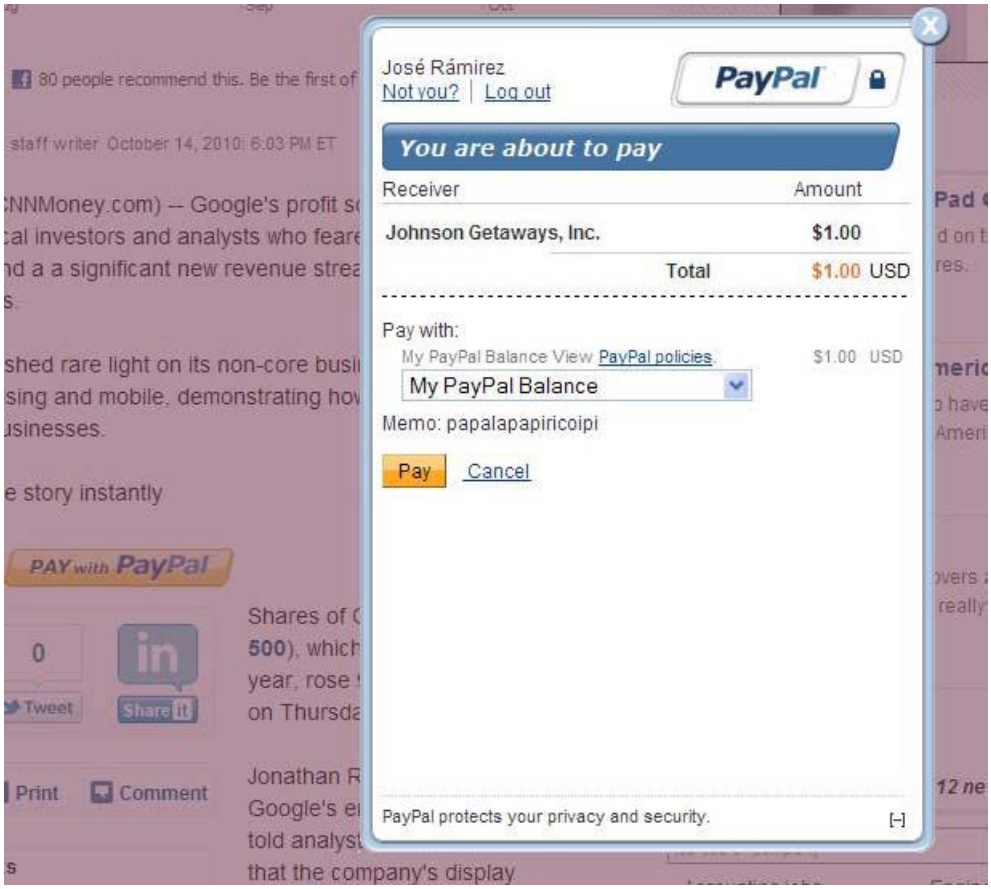


When the sender clicks **Close**, PayPal redirects the sender's browser to the return URL specified in the Pay API operation's request message.

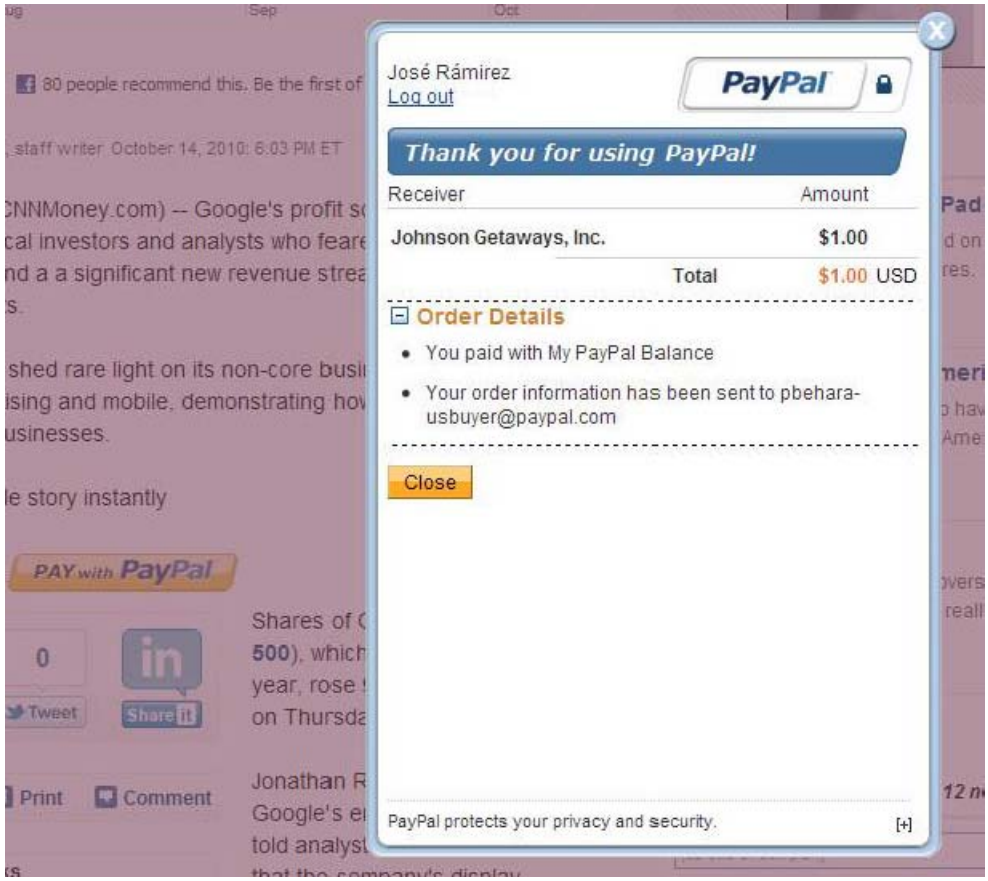
NOTE: You are responsible for closing the minibrowser after PayPal redirects to the page specified in either the return or cancel URL. PayPal provides a JavaScript function that you call to close a PayPal minibrowser or lightbox.

For subsequent payments in which PayPal does not require the account holder to log in again and the account holder has chosen to be remembered, PayPal bypasses the log in steps.

For example, the lightbox for reviewing a payment would appear in your page as follows:



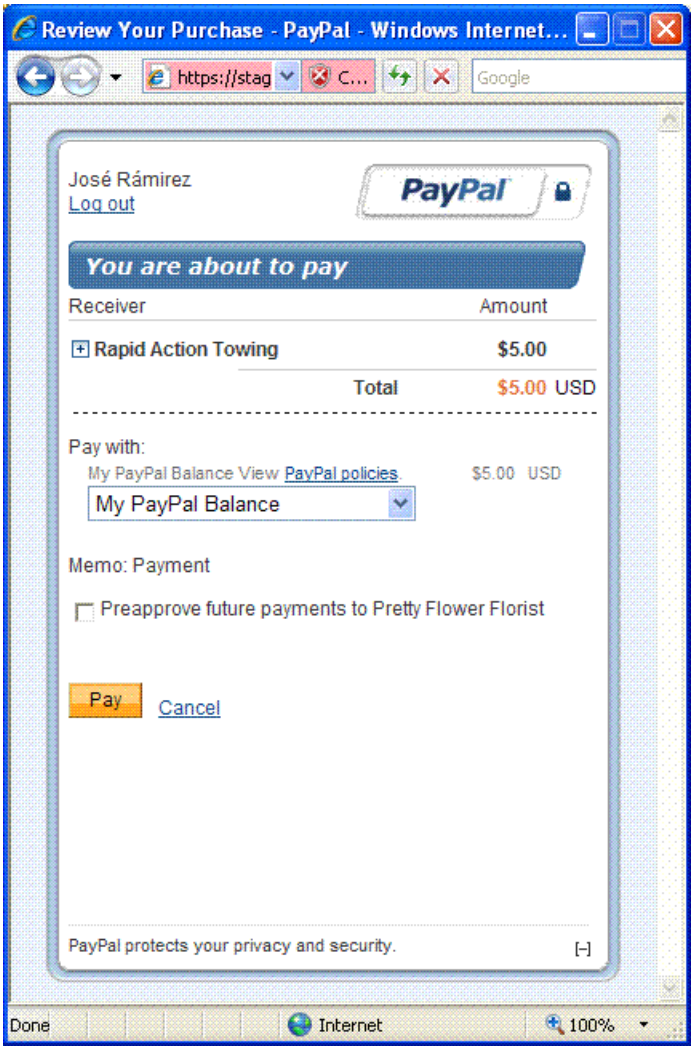
The lightbox containing the confirmation would appear in your page as follows:



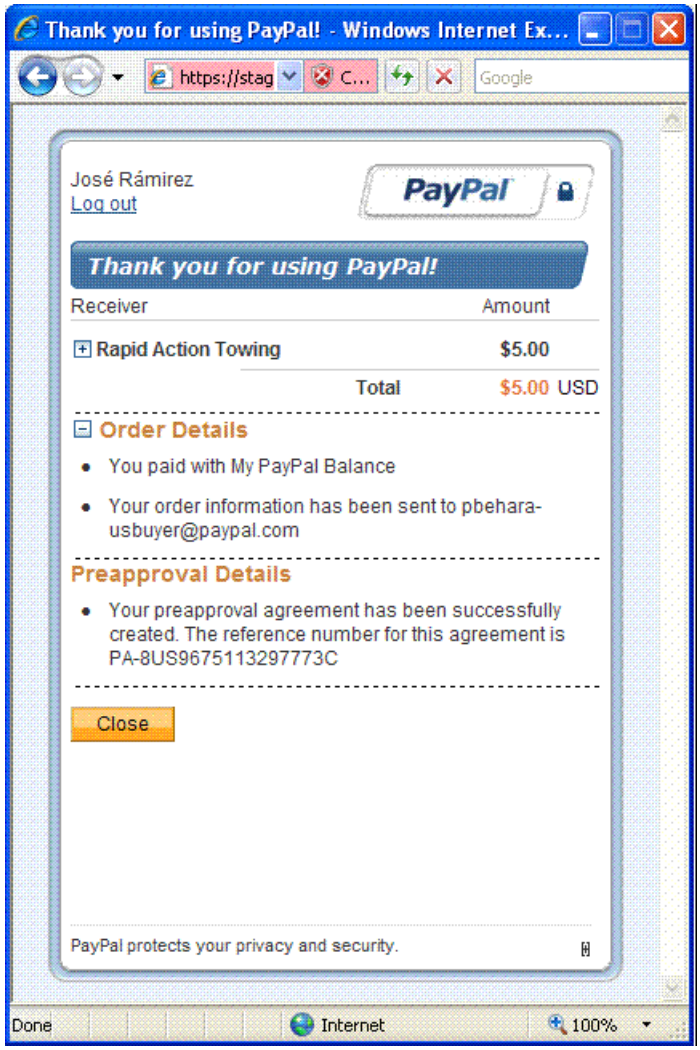
The actions you take to launch the flow and close the lightbox are the same steps you take for the minibrowser.

Preapprove Future Payments Checkbox

You can add a **Preapprove future payments** checkbox to the sender's embedded payment experience, which enables the sender to preapprove subsequent payments. If you invoke the embedded payment flow, passing both a payment key obtained by calling `Pay` and a preapproval key obtained by calling `Preapproval`, PayPal displays the checkbox on the payment page:



If the payment sender checks the preapproval box, the confirmation page provides details about the preapproval:



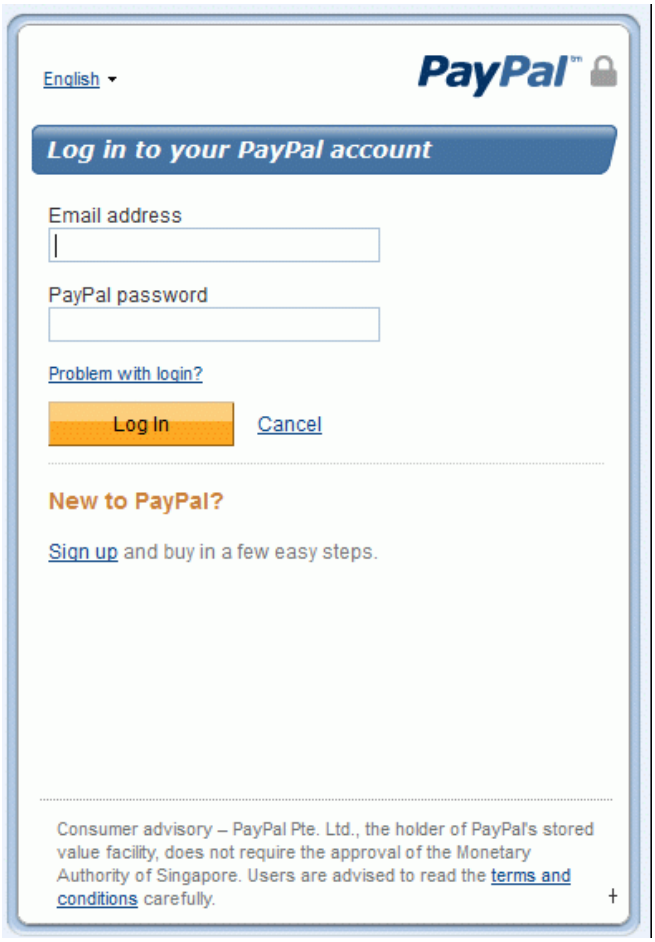
NOTE: Unless there is an error with the payment itself, PayPal transfers the money regardless of whether the preapproval for future payments was successful.

Embedded Preapproval Experience

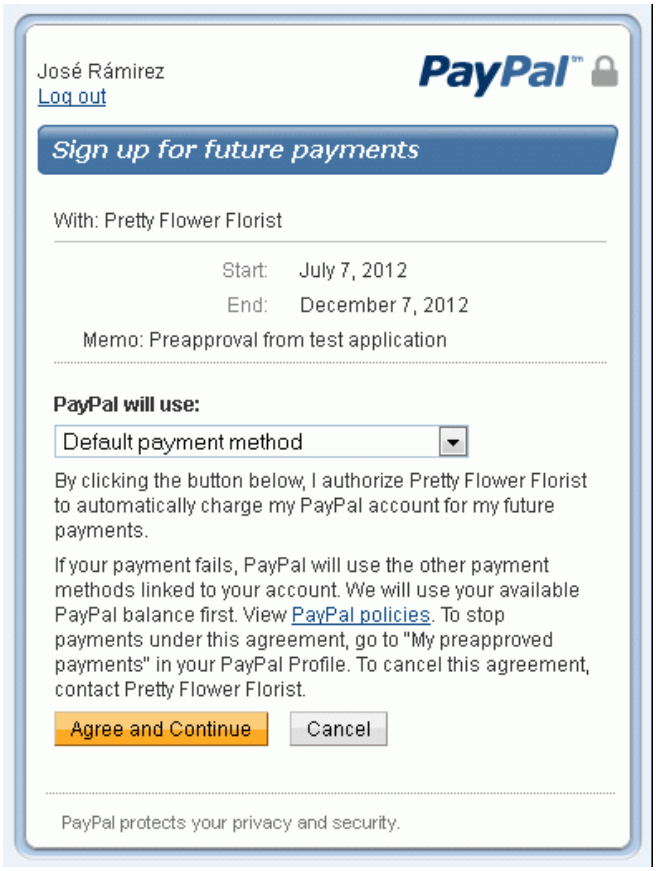
Adaptive Payments provides a preapproval experience using a mini-browser.You can invoke the embedded preapproval flow by passing the preapproval key. Here is an example :

```
https://www.paypal.com/webapps/adaptivepayment/flow/preapproval?preapprovalKey=PA-xxxxxxxxxxxx&expType=mini
```

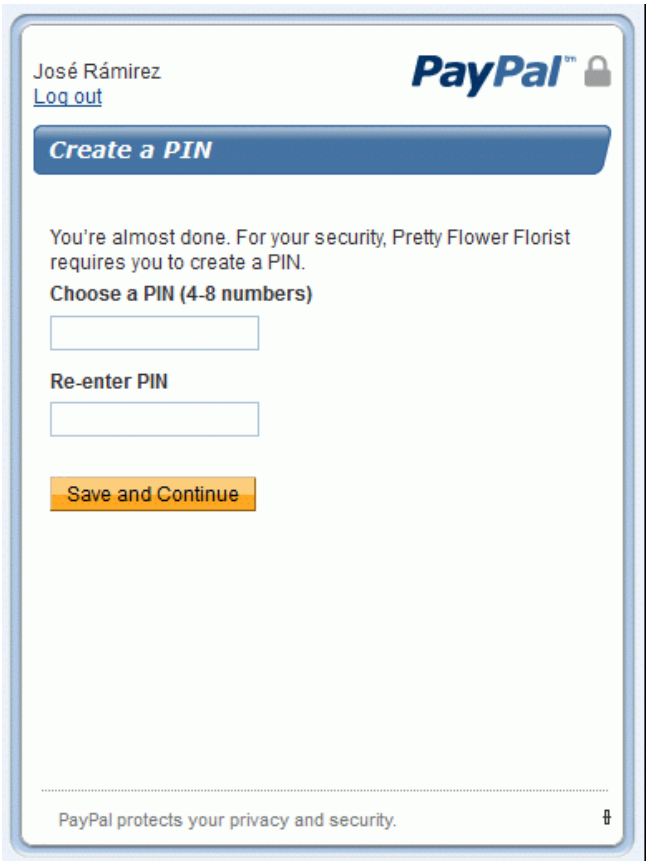
PayPal first asks the user to log into their PayPal account:



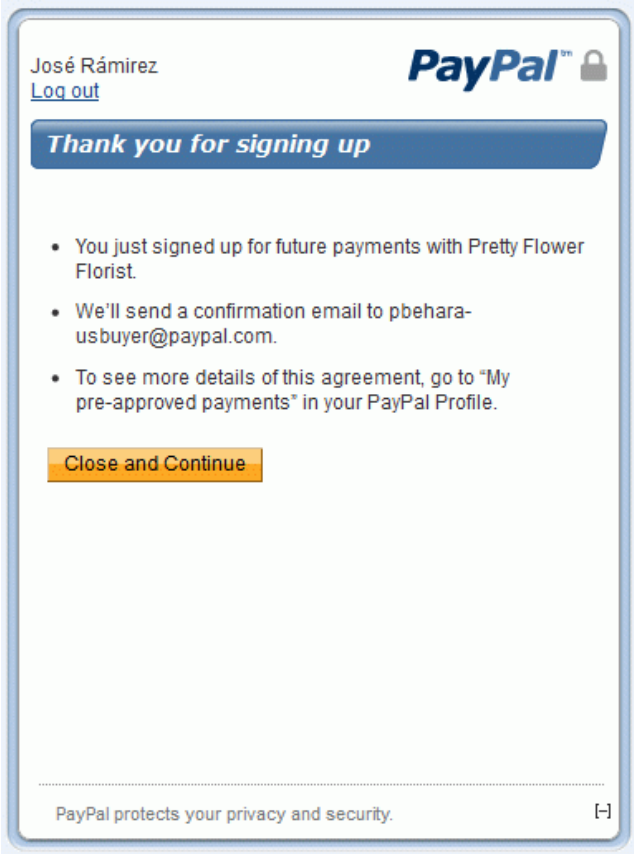
After signing in, the user is presented with a consent form that the user must agree to:



If the seller has enabled PIN code entry, after consenting to the agreement, the user is prompted to enter a PIN code by the seller:

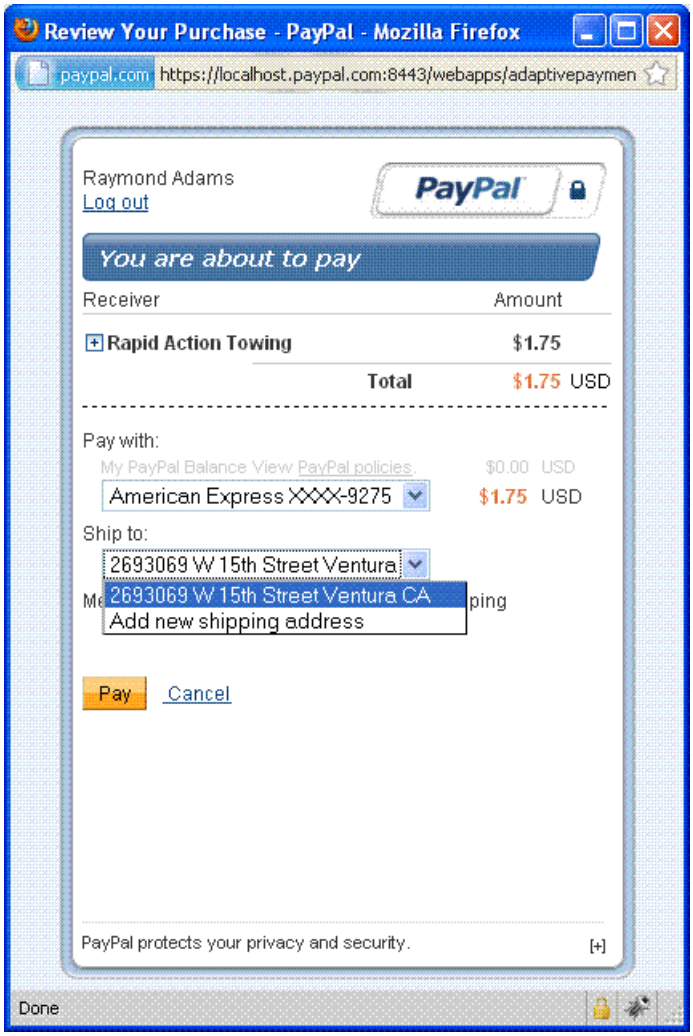


Finally, the user is presented with a confirmation screen:

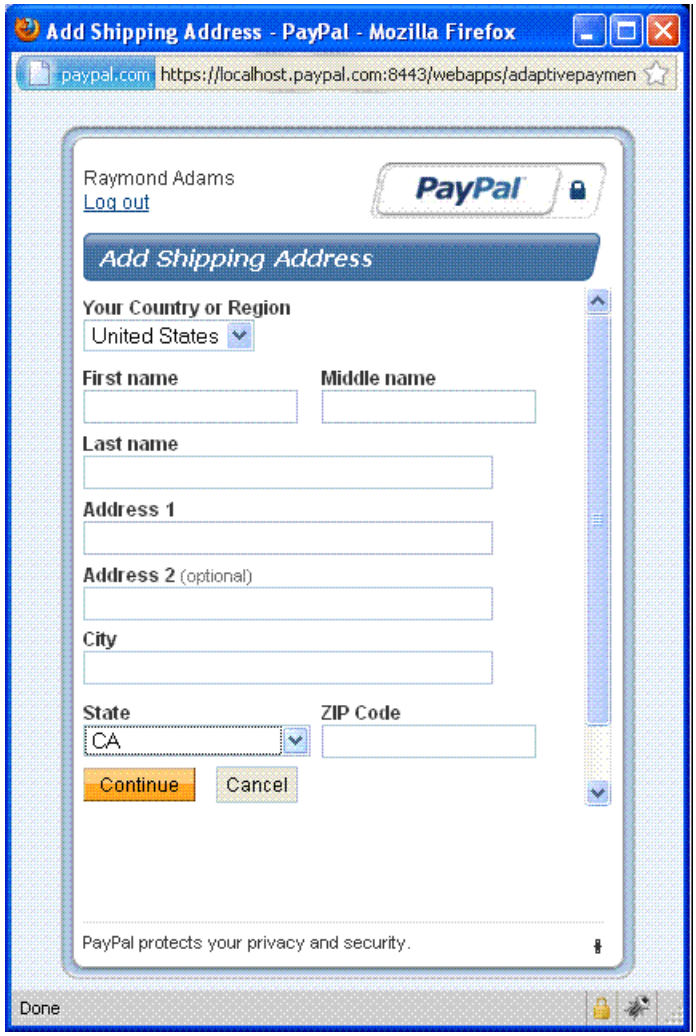


Shipping Address Selection

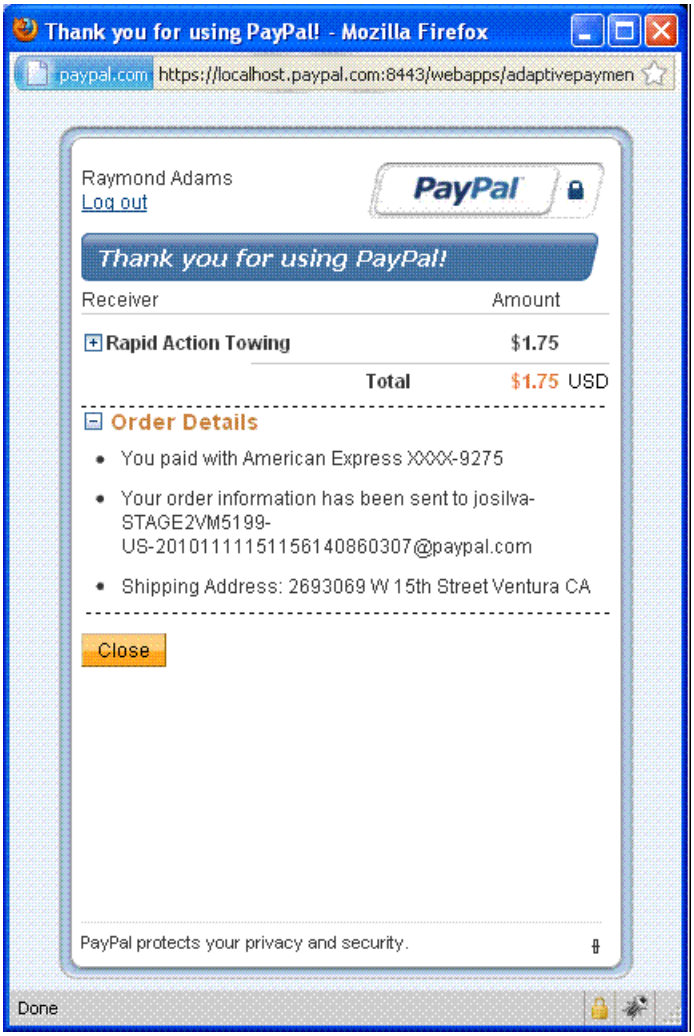
You can display and collect shipping address information for a transaction with the embedded payment flow. PayPal displays the default shipping address when you set `senderOptions.requireShippingAddressSelection` to `true` in your request to `SetPaymentOptions`:



The sender of a payment can select one of the available shipping addresses or add a new shipping address by selecting **Add new shipping address** from the **Ship to:** drop-down menu:



After the sender of the payment clicks **Pay**, PayPal displays the selected shipping address on the **Thank you for using PayPal** page:



You can call the `GetShippingAddresses` API operation to obtain the selected shipping address for the transaction using the key assoicated with the payment.

Setting Up Web Pages to Invoke the Embedded Payment Flow Using a Lightbox

Use the JavaScript functions in `https://www.paypalobjects.com/js/external/dg.js` to set up and control the embedded payment flow. This example shows how to initiate the embedded payment flow after obtaining a payment key.

This example assumes that you obtain a payment key before initiating the flow and that the key does not change or expire before the sender completes the flow.

To set up a web page to invoke the embedded payment flow:

1. Call the `Pay` API operation to obtain a valid pay key.
2. Create your form or button.
   - You must include the pay key and redirect to `https://www.paypal.com/webapps/adaptivepayment/flow/pay`.
   - Optionally, include a preapproval key if you want to enable the selection of **Preapproval for future payments**
   - Specify that a lightbox opens in the PayPal-created `IFRAME`, `PPDGFrame`.
   - Set the `expType` parameter to indicate your preference for the context in which the PayPal payment flow is displayed. You must specify `light` for lightbox.

```
<form action=
   "https://www.paypal.com/webapps/adaptivepayment/flow/pay"
   target="PPDGFrame">
   <input id="type" type="hidden" name="expType" value="light">
   <input id="paykey" type="hidden" name="paykey" value="AP-...">
<input id="preapprovalkey" type="hidden" name="preapprovalkey" value="PA-...">
<input type="submit" id="submitBtn" value="Pay with PayPal">
</form>
```

NOTE: To modify an existing application to use the embedded payment flow, change the redirect from

`https://www.paypal.com/webscr?cmd=_ap-payment&paykey=...` to `https://www.paypal.com/webapps/adaptivepayment/flow/pay?paykey=...` after obtaining the pay key.

3. Include the PayPal JavaScript functions from `dg.js`.

```
<script src="https://www.paypalobjects.com/js/external/dg.js">
</script>
```

4. Create an embedded flow object and associate it with your payment form or button.

```
<script>
var dgFlow = new PAYPAL.apps.DGFlow({ trigger: 'submitBtn' });
</script>
```

After Completing This Task:

On the pages you identify as the return and cancel URLs in the `Pay` API operation, you must include the PayPal JavaScript functions from `dg.js` and close the PayPal window, as in the following example:

```
dgFlow = top.dgFlow || top.opener.top.dgFlow;
     dgFlow.closeFlow();
     top.close();
```

Setting Up Web Pages to Invoke the Embedded Payment Flow Using a Minibrowser

Use the JavaScript functions in `https://www.paypalobjects.com/js/external/apdg.js` to set up and control the embedded payment flow. This example shows how to initiate the embedded payment flow after obtaining a payment key.

This example assumes that you obtain a payment key before initiating the flow and that the key does not change or expire before the sender completes the flow.

To set up a web page to invoke the embedded payment flow:

1. Call the `Pay` API operation to obtain a valid pay key.
2. Create your form or button.
   - You must include the pay key and redirect to `https://www.paypal.com/webapps/adaptivepayment/flow/pay`.
   - Optionally, include a preapproval key if you want to enable the selection of **Preapproval for future payments**
   - Set the `expType` parameter to `mini` to indicate your preference for the context in which the PayPal payment flow is displayed.

```
<form action=
   "https://www.paypal.com/webapps/adaptivepayment/flow/pay"
   target="PPDGFrame">
   <input id="type" type="hidden" name="expType" value="mini">
   <input id="paykey" type="hidden" name="paykey" value="AP-...">
<input id="preapprovalkey" type="hidden" name="preapprovalkey" value="PA-...">
<input type="submit" id="submitBtn" value="Pay with PayPal">
</form>
```

NOTE: To modify an existing application to use the embedded payment flow, change the redirect from `https://www.paypal.com/webscr?cmd=_ap-payment&paykey=...` to `https://www.paypal.com/webapps/adaptivepayment/flow/pay?paykey=...` after obtaining the pay key.

3. Include the PayPal JavaScript functions from `apdg.js`.

```
<script src="https://www.paypalobjects.com/js/external/apdg.js">
</script>
```

4. Create an embedded flow object and associate it with your payment form or button.

```
     <script>
        function <<returnFunctionName>>() {
           <<Your code goes here.>>
}
     </script>
```

After Completing This Task:

The following minibrowser return script can be used to determine whether the payment successfully completed:

```
PAYMENTDETAILS=$(wget --no-check-certificate \
    --output-document=- \
    --quiet \
    --header="X-PAYPAL-SERVICE-VERSION: 1.0.0" \
    --header="X-PAYPAL-SECURITY-USERID: $APIUSER" \
    --header="X-PAYPAL-SECURITY-PASSWORD: $APIPASS" \
    --header="X-PAYPAL-SECURITY-SIGNATURE: $APISIG" \
    --header="X-PAYPAL-REQUEST-DATA-FORMAT: NV" \
    --header="X-PAYPAL-RESPONSE-DATA-FORMAT: NV" \
    --header="X-PAYPAL-APPLICATION-ID:  $APPLICATIONID" \
    --post-data="payKey=$PAYKEY&requestEnvelope.errorLanguage=en_US" \
  https://www.paypal.com/AdaptivePayments/PaymentDetails)
if echo $PAYMENTDETAILS | grep -q "\&status=COMPLETED"
then  echo "Thank you for approving pay key $PAYKEY"
else  echo "Sorry, you were unable to approve pay key $PAYKEY.  Please try
another transaction!"
fi
```

Displaying and Collecting Shipping Addresses

PayPal displays the default shipping address and allows the payment sender to change the address when you set `senderOptions.requireShippingAddressSelection` to `true` in your request to the `SetPaymentOptions` API operation. You call the `GetShippingAddresses` API operation to obtain the selected shipping address after invoking the embedded payment flow.

The steps in this example assume that you have implemented the JavaScript for invoking the embedded payment flow, that you have set up your button or form to invoke the flow, and that you have included the code to close the window associated with the flow.

To display and collect the selected shipping address

1. Call the `Pay` API operation with `actionType` set to `CREATE` to obtain a payment key.
2. Set `senderOptions.requireShippingAddressSelection` to `true` in your request to `SetPaymentOptions` and call the API operation.

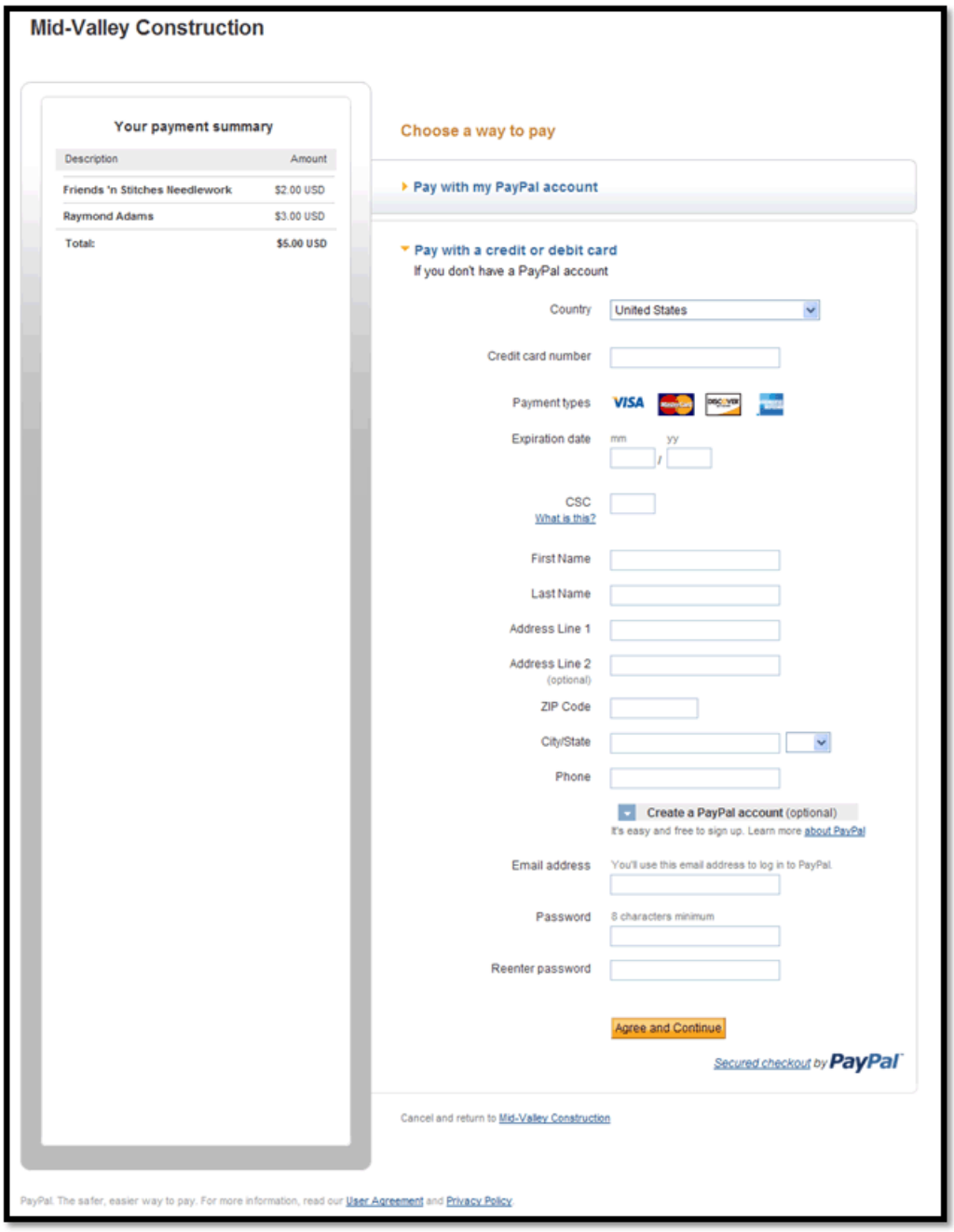
3. Redirect the payment sender's browser to the embedded payment flow at `https://www.paypal.com/webapps/adaptivepayment/flow/pay?paykey=...` after obtaining the pay key.
4. After returning from the flow, call the `GetShippingAddresses` API operation to obtain the selected shipping address.

Guest Payments

Adaptive payments supports *guest payments*, which are payments using a sender's credit card without logging into PayPal to complete the transaction.The sender is not explicitly identified as a PayPal account holder during the transaction and is not required to have a PayPal account.

Each receiver of a guest payment must be a verified PayPal Business Verified or Premier Verified account holder.

PayPal handles guest payments in the same way that it handles explicitly approved payments. Instead of logging into PayPal to complete transaction, the sender provides credit card information on the PayPal payment screen:



NOTE: For European Union countries, only 10 guest payments are allowed per card.

Fee Payment Configuration

You can set up a payment transaction so that either the sender of a payment pays the fee or the receivers of a payment pay the fee. If receivers pay the fee, you can specify whether the

primary receiver in a chained payment pays the entire fee or whether all receivers pay a portion of the fee.

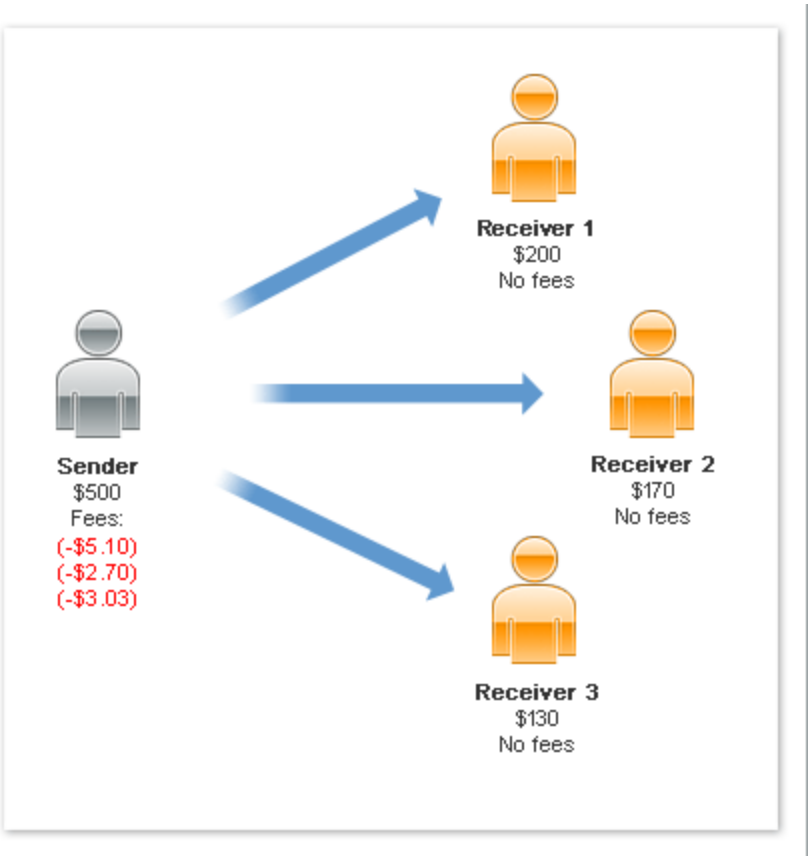
You can specify who pays these fees. Fee payment configurations include:

- Sender Pays the Fee
- Receiver Pays the Fee in a Parallel Payment
- Each Receiver Pays the Fee in a Chained Payment
- Primary Receiver Pays the Fee in a Chained Payment

NOTE: Fees are determined by PayPal and are based on criteria, such as the transaction volume of the receiver. In the examples that follow, the fees shown are representative only and not actual fees.

Sender Pays the Fee

The sender can pay a fee for a simple payment, parallel payment, or a chained payment. The following example shows fees being paid by the sender of a parallel payment, based on the assessment for each receiver:

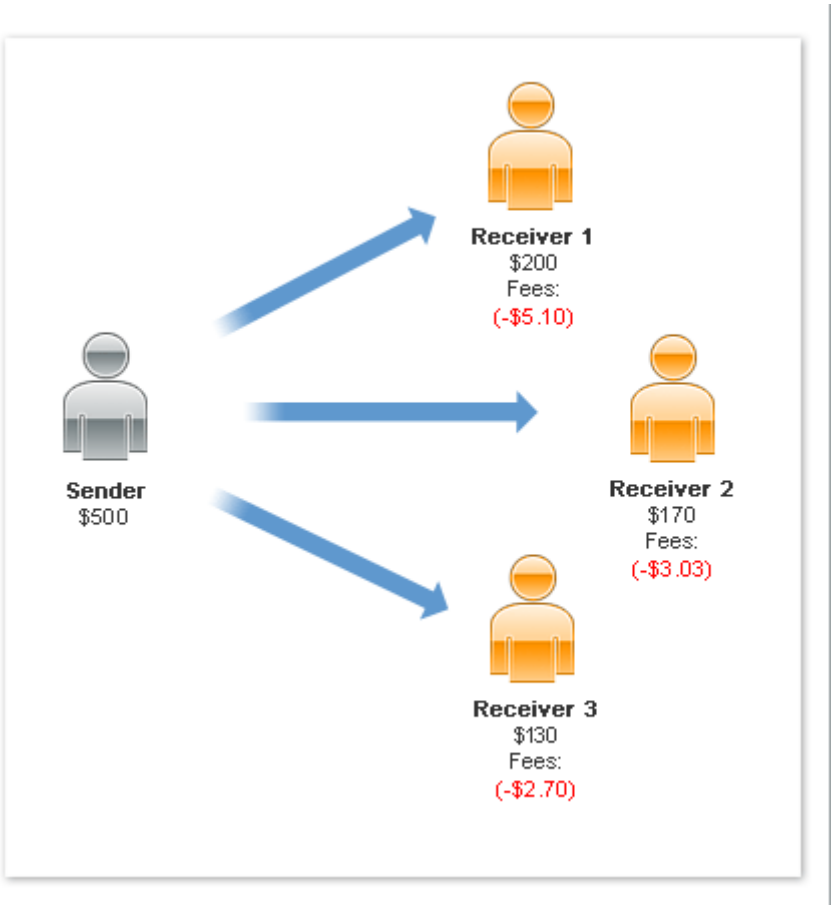


In this example, the sender pays a total of $510.83, which includes all fees.

NOTE: The scenario above is an example only and is not representative of actual PayPal fees.

Receiver Pays the Fee in a Parallel Payment

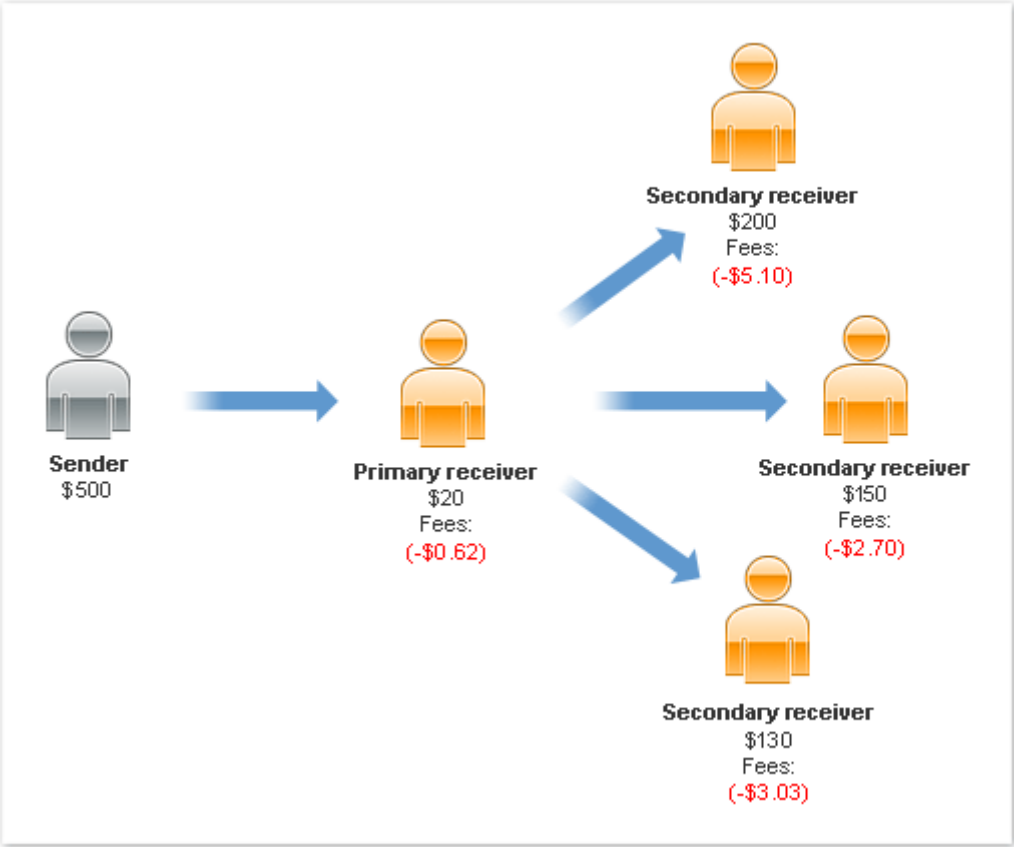
If the receivers pay the fee in a parallel payment, each receiver pays a portion of the fee, based on their assessment. The following example shows the receivers paying the fees:



NOTE: The scenario above is an example only and is not representative of actual PayPal fees.

Each Receiver Pays the Fee in a Chained Payment

If the receivers pay the fee in a chained payment, each receiver pays a portion of the fee, based on their assessment. The following example shows the receivers paying the fees:

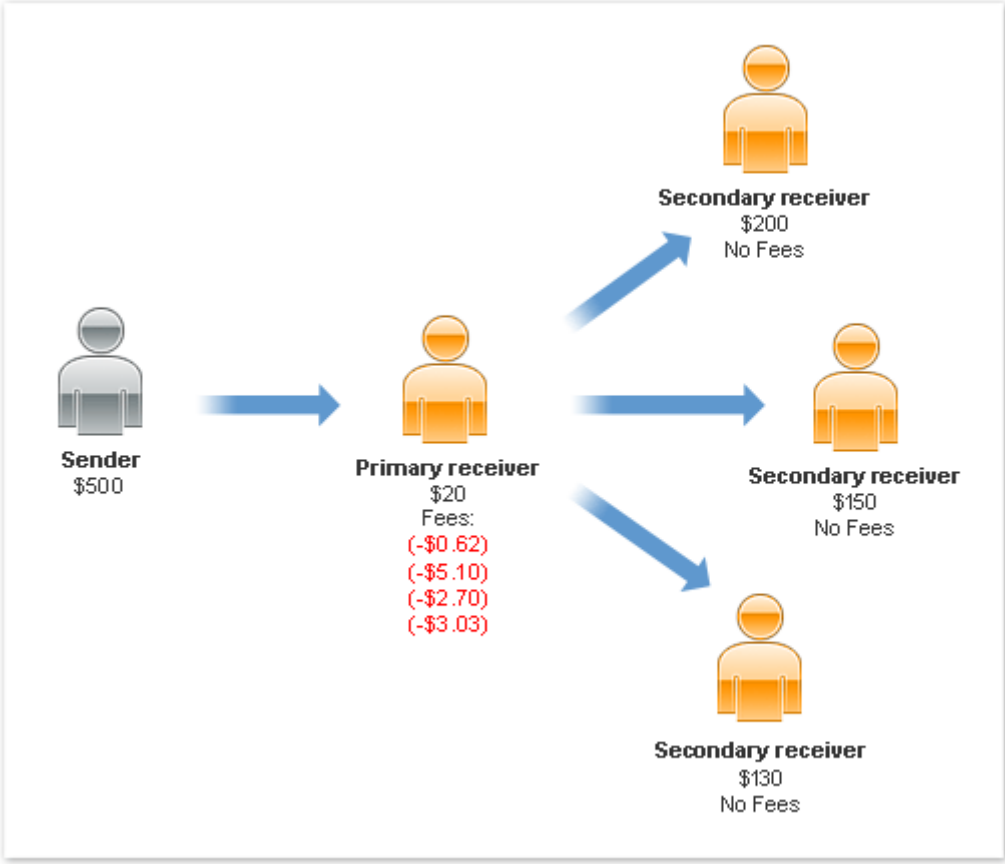


In this example, the primary receiver, identified as the merchant, pays a fee for $20 received. Each of the other receivers also pay a fee on the amount each receives.

NOTE: The scenario above is an example only and is not representative of actual PayPal fees.

Primary Receiver Pays the Fee in a Chained Payment

If only the primary receiver pays the fee in a chained payment, other receivers pay no fees. The fees paid by the primary receiver, however, are based upon the total fees assigned to all receivers. The following example shows only the primary receiver, identified as the merchant, paying all fees:



NOTE: The scenario above is an example only and is not representative of actual PayPal fees.

2 Getting Started

These basic scenarios get you up and running quickly with the Adaptive Payments API. The sample code shows different combinations of requests with different bindings in various programming languages.

Adaptive Payments API Operations

Adaptive Payments provides API operations, enabling you to build an application that handles payments, preapprovals for payments, refunds, and additional operations related to payments. Some kinds of payments and operations require specific permission to use.

API Operations Related to Payments

API Operation	Description
`Pay`	Transfers funds from a sender's PayPal account to one or more receivers' PayPal accounts (up to 6 receivers)
`PaymentDetails`	Obtains information about a payment created with the `Pay` API operation
`ExecutePayment`	Executes a payment
`GetPaymentOptions`	Obtain the settings specified with the `SetPaymentOptions` API operation
`SetPaymentOptions`	Sets payment options

API Operations Related to Preapprovals

API Operation	Description
`Preapproval`	Sets up preapprovals, which is an approval to make future payments on the sender's behalf
`PreapprovalDetails`	Obtains information about a preapproval
`CancelPreapproval`	Cancels a preapproval
`ConfirmPreapproval`	Confirms that you can use the specified preapproval to make payments

Other API Operations

API Operation	Description
`Refund`	Refunds all or part of a payment
`ConvertCurrency`	Obtains the current foreign exchange (FX) rate for a specific amount and currency
`GetFundingPlans`	Determines the funding sources that are available for a specified payment
`GetAllowedFundingSources`	Obtains the funding sources associated with a preapproval.
`GetShippingAddresses`	Obtains the selected shipping address
`GetAvailableShippingAddresses`	Obtains available shipping addresses

Adaptive Payments Endpoints

The endpoint is determined by the API operation and the environment in which you want to execute the API operation. For example, if you want to send a `Pay` request to the sandbox endpoint, specify the following URL:

`https://svcs.sandbox.paypal.com/AdaptivePayments/Pay`

You can specify the following endpoints:

Environment	Endpoint
Production	`https://svcs.paypal.com/AdaptivePayments/`*API_operation*
Sandbox	`https://svcs.sandbox.paypal.com/AdaptivePayments/`*API_operation*
Beta sandbox	`https://svcs.beta-sandbox.paypal.com/AdaptivePayments/`*API_operation*

HTTP Headers

Each request message includes HTTP headers specifying authentication, the application ID, the device ID or IP address, and the payload format or protocol (SOAP).

Adaptive Payments supports request bodies with JSON, NVP, and XML data formats for REST implementations. You can specify different formats for the request and response, such as sending the request in JSON and requesting an XML response.

For SOAP, you must also include a specific SOAP protocol header (see the SOAP messages section).

The following is an example of HTTP headers for NVP in Java for a web implementation:

```
headers.put("X-PAYPAL-SECURITY-USERID", "tok261_biz_api.abc.com");
headers.put("X-PAYPAL-SECURITY-PASSWORD","1244612379");
headers.put("X-PAYPAL-SECURITY-SIGNATURE","lkfg9groingghb4uw5"
headers.put("X-PAYPAL-DEVICE-IPADDRESS", "168.212.226.204");
headers.put("X-PAYPAL-REQUEST-DATA-FORMAT", "NV");
headers.put("X-PAYPAL-RESPONSE-DATA-FORMAT", "NV");
headers.put("X-PAYPAL-APPLICATION-ID", "APP-80W284485P519543T");
```

NOTE: HTTP headers are case sensitive.

Authentication

Use your PayPal account API credentials to authenticate your application. Your API credentials include an API username and API password. If you are using 3-token authentication, you must also specify an API signature. If you are using a certificate, the certificate is used with the username and password; the signature is not used. To specify API credentials, include the following HTTP headers in your request message (observing case sensitivity):

HTTP Headers for Authentication

Header	Description
X-PAYPAL-SECURITY-USERID	Your API username
X-PAYPAL-SECURITY-PASSWORD	Your API password
X-PAYPAL-SECURITY-SIGNATURE	Your API signature, which is required only if you use 3-token authorization; a certificate does not use a signature
X-PAYPAL-SECURITY-SUBJECT	Third-party permission specification, which specifies the email address or phone number (for mobile) of the party on whose behalf you are calling the API operation. The subject must grant you third-party access in their PayPal profile. **NOTE:** Resources specified by the API operation, such as a payment or preapproval identified by a key, must be owned by the subject granting the third-party permission.

Specifying JSON, NVP, or XML Data Formats

Use the HTTP header X-PAYPAL-REQUEST-DATA-FORMAT to specify the data format the request body. You can send messages using JSON, NVP or straight XML.

Use the and X-PAYPAL-RESPONSE-DATA-FORMAT headers to specify the data format for the response.

For SOAP messages, refer to the next section.

HTTP Headers for JSON, NVP, and XML Data Formats

Header	Description
X-PAYPAL-REQUEST-DATA-FORMAT	The payload format for the request. Allowable values are: • NV – Name-value pairs • XML – Extensible markup language • JSON – JavaScript object notation
X-PAYPAL-RESPONSE-DATA-FORMAT	The payload format for the response. Allowable values are: • NV – Name-value pairs • XML – Extensible markup language • JSON – JavaScript object notation

SOAP Messages

To use Adaptive Payments with SOAP, include the HTTP headers for authentication as described in the section Authentication and the application ID as described in the next section.

In addition, include the X-PAYPAL-MESSAGE-PROTOCOL header with a SOAP11 value.

The following is a header example for an Adaptive Payments API call for a SOAP message:

```
headers.put("X-PAYPAL-SECURITY-USERID", "tok261_biz_api.abc.com");
headers.put("X-PAYPAL-SECURITY-PASSWORD","1244612379");
headers.put("X-PAYPAL-SECURITY-SIGNATURE","lkfg9groingghb4uw5"
headers.put("X-PAYPAL-DEVICE-IPADDRESS", "168.212.226.204");
headers.put("X-PAYPAL-MESSAGE-PROTOCOL", "SOAP11");
headers.put("X-PAYPAL-APPLICATION-ID","APP-80W284485P519543T");
```

Below are the service name, port type, binding and location for SOAP as defined in the Adaptive Payments WSDL.

```
<wsdl:service name="AdaptivePayments">
     <wsdl:port name="AdaptivePaymentsSOAP11_http"
     <binding="services:AdaptivePaymentsSOAP11Binding">
     <soap:address location="https://svcs.paypal.com/AdaptivePayments" />
```

Specifying Application and Device Information

You also must identify the application. You can optionally identify other information associated with the client and the API version:

HTTP Headers for Application and Device identification

Header	Description
X-PAYPAL-APPLICATION-ID	*(Required)* Your application's identification, which is issued by PayPal. **NOTE:** Check X.com for which application ID must be defined for working in the sandbox.
X-PAYPAL-DEVICE-ID	*(Optional)* Client's device ID, such as a mobile device's IMEI number or a web browser cookie.
X-PAYPAL-DEVICE-IPADDRESS	*(Required)* Client's IP address.
X-PAYPAL-SERVICE-VERSION	*(Optional)* The version of an API operation to use. By default, PayPal executes a request with the current version of an API operation. **NOTE:** PayPal recommends not specifying a version unless it is absolutely required.

Making a Simple Payment (JSON)

A *simple* payment is when a sender (whose account is debited) sends a payment (amount and currency) to a single receiver (whose account is credited).

- You send a `PayRequest` message to PayPal
- You receive a response with a pay key.
- You must redirect the sender's browser to PayPal to approve the payment.

Pay Request for Simple Payment

```
{"returnUrl":"http://example.com/returnURL.htm", \
"requestEnvelope":{"errorLanguage":"en_US"},"currencyCode":"USD", \
"receiverList":{"receiver":[{"email":"david@example.com", \
     "amount":"10.00",}]},"cancelUrl":"http://example.com/cancelURL.htm",\
"actionType":"PAY"}
```

Pay Response for Simple Payment

```
{"responseEnvelope":\
{"timestamp":"2009-10-06T14:30:39.383-07:00","ack":"Success",\
"correlationId":"cfe8f8783f1d3","build":"DEV"},\
"payKey":"AP-17266198048308436","paymentExecStatus":"CREATED"}
```

The response includes a **pay key**, which is a token you use in subsequent calls to Adaptive Payments APIs to identify this particular payment.

In this particular scenario, the `paymentExecStatus` variable is set to `CREATED` instead of `COMPLETED`, which indicates that the payment has been created, but has not yet been executed.

Making a Parallel Payment (NVP)

A *parallel* payment is when a sender (whose account is debited) sends a single payment (amount and currency) up to 6 receivers. The sender can see each payment to a receiver.

- You send a `PayRequest`, specifying an amount to be paid for each receiver.
- You receive a response with a pay key.
- You must redirect the sender's browser to PayPal to approve the payment.

In the example below, Paul makes a single payment of $14, which is split into a $9 payment to Andrea and a $5 payment to Linda. The following event sequence takes place:

Pay Request for Parallel Payment

```
&actionType=PAY
&cancelUrl=http:\\example.com\cancel.htm
&currencyCode=USD
&receiverList.receiver(0).amount=9.00
&receiverList.receiver(0).email=andrea@example.com
&receiverList.receiver(1).amount=5.00
&receiverList.receiver(1).email=linda@example.com
&requestEnvelope.errorLanguage=en_US
&returnUrl=http:\\example.com\return.htm
```

Pay Response for Parallel Payment

```
responseEnvelope.timestamp=2009-11-03T08%3A12.937-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=b1cc3eabfa4c1
&responseEnvelope.build=942345
&payKey=AP-688241038Y786593D
&paymentExecStatus=CREATED
```

The response includes a **pay key**, which is a token you use in subsequent calls to Adaptive Payments APIs to identify this particular payment.

In this particular scenario, the `paymentExecStatus` variable is set to `CREATED` instead of `COMPLETED`, which indicates that the payment has been created, but has not yet been executed.

Making a Chained Payment (XML)

A *chained* payment is when a sender sends a payment to a PayPal-registered receiver who is the primary receiver.

- You send a `PayRequest`, enabling the primary receiver.
- You receive a response with a pay key.
- You must redirect the sender's browser to PayPal to approve the payment.

With chained payments, the sender only sees the transaction to the primary API caller. The receiver only sees the transaction from the primary API caller. The transactions from and to the receivers are hidden from the sender and receivers.

In the example below, Tim makes a single payment of $100 to Frank, who is the primary receiver. Of this amount, Frank keeps $25 and pays Yvonne $75. The following event sequence takes place:

Pay Request for Chained Payment

```
<PayRequest>
   <requestEnvelope>
     <errorLanguage>en_US</errorLanguage> </requestEnvelope>
   <cancelUrl xmlns="">http://exammple.com/cancelURL.htm</cancelUrl>
   <actionType>PAY</actionType>
   <currencyCode xmlns="">USD</currencyCode>
     <receiverList xmlns="">
        <receiver>
           <amount>100</amount>
           <email>frank@example.com</email>
           <primary>true</primary>
        </receiver>
        <receiver>
           <amount>75</amount>
           <email>yvonne@example.com</email>
           <primary>false</primary>
        </receiver>
   </receiverList>
   <returnUrl xmlns="">http://example.com/returnURL.htm</returnUrl>
  </PayRequest>
```

Pay Response for Chained Payment

```
<?xml version='1.0' encoding='UTF-8'?>
<ns2:PayResponse xmlns:ns2="http://svcs.paypal.com/types/ap">
<responseEnvelope>
<timestamp>2009-10-06T17:24:03.874-07:00</timestamp>
<ack>Success</ack><correlationId>eca3a204200f4</correlationId>
<build>1044393</build></responseEnvelope>
<payKey>AP-688241038Y786593D</payKey>
<paymentExecStatus>CREATED</paymentExecStatus></ns2:PayResponse>
```

The response includes a **pay key**, which is a token you use in subsequent calls to Adaptive Payments APIs to identify this particular payment.

In this particular scenario, the `paymentExecStatus` variable is set to `CREATED` instead of `COMPLETED`, which indicates that the payment has been created, but has not yet been executed.

3 Pay API Operation

Use the Pay API operation to transfer funds from a sender's PayPal account to one or more receivers' PayPal accounts. You can use the Pay API operation to make simple payments, chained payments, or parallel payments; these payments can be explicitly approved, preapproved, or implicitly approved.

Pay Summary

Create your PayRequest message by setting the common fields. If you want more than a simple payment, add fields for the specific kind of request, which include parallel payments, chained payments, implicit payments, and preapproved payments.

Common Fields for All Payments

For each kind of payment, you must specify values for the following fields:

Field	Description
actionType	The action for this request. Possible values are: • PAY – Use this option if you are not using the Pay request in combination with ExecutePayment. • CREATE – Use this option to set up the payment instructions with SetPaymentOptions and then execute the payment at a later time with the ExecutePayment. • PAY_PRIMARY – For chained payments only, specify this value to delay payments to the secondary receivers; only the payment to the primary receiver is processed.
receiverList.receiver(0).email	A receiver's email address
receiverList.receiver(0).amount	Amount to be credited to the receiver's account
currencyCode	The code for the currency in which the payment is made; you can specify only one currency, regardless of the number of receivers
cancelUrl	URL to redirect the sender's browser to after canceling the approval for a payment; it is always required but only used for payments that require approval (explicit payments)

Field	Description
`returnUrl`	URL to redirect the sender's browser to after the sender has logged into PayPal and approved a payment; it is always required but only used if a payment requires explicit approval
`requestEnvelope.errorLanguage`	The code for the language in which errors are returned, which must be `en_US`.

Parallel Payments

For a simple payment, you must specify values for the fields above. For a parallel payment, you must specify both the above values and the following fields:

Field	Description
`receiverList.receiver(`*n*`).email`	The receiver's email address for each receiver, where *n* is between 0 and 5 for a maximum of 6 receivers
`receiverList.receiver(`*n*`).amount`	Amount to send to each corresponding receiver

Chained Payments

For a chained payment, you must specify all required fields for a parallel payment and you must also specify enable the primary receiver:

Field	Description
`receiverList.receiver(`*n*`).email`	The receivers' email addresses, where *n* is between 0 and 5 for a total of one primary receiver and between one and 5 secondary receivers
`receiverList.receiver(`*n*`).amount`	Amount to send to each corresponding receiver
`receiverList.receiver(`*n*`).primary`	Set to `true` to indicate a chained payment; only one receiver can be a primary receiver. Omit this field, or set it to `false` for simple and parallel payments.

By default, a payment requires explicit approval in which the sender must log in to PayPal. In that case, you are not required to specify the sender's email address. To initiate the approval process, you must redirect the sender to PayPal as follows:

`https://www.paypal.com/webscr?cmd=_ap-payment&paykey=`*value*

NOTE: The command is `_ap-payment`. You obtain the value of the `paykey` parameter from the `payKey` field in the `Pay` API operation's response message.

Implicit Payments

If you are the API caller and you specify your email address in the `senderEmail` field, PayPal implicitly approves the payment without redirecting to PayPal:

Field	Description
`senderEmail`	Sender's email address

Preapproved Payments

If the sender has set up a preapproval, you can use the preapproval to avoid explicit approval. In that case, you must specify values for the fields below.

Field	Description
`preapprovalKey`	Preapproval key for the approval set up between you and the sender
`pin`	Sender's personal identification number, if one was specified when the sender agreed to the approval

Payments for Digital Goods

You handle payments for digital goods in the same way you handle payments for other goods and services, with the following exceptions:

- To specify a payment for digital goods, you must specify `DIGITALGOODS` for each receiver in your receiver list; specify `receiverList.receiver(`*n*`).paymentType=DIGITALGOODS` for each receiver, where *n* identifies the receiver, starting with 0.
- If you specify a payment for digital goods, you cannot specify a `senderEmail` address or include a funding constraint.
- You must redirect the sender to the following PayPal URL to complete the payment for digital goods:
  `https://www.paypal.com/webapps/adaptivepayment/flow/pay?paykey=...`

Payment Notifications

Notifications are sent after payment is executed, which follows approval of the payment by the sender if required:

- PayPal sends an email to the sender and all receivers associated with a payment when the transfer is complete; you receive an email only if you are also the payment sender or receiver.

- When the payment is complete, PayPal sends an IPN message to the URL specified in the `ipnNotificationUrl` field of the `Pay` request.

Additional Notes About the Pay API Operation

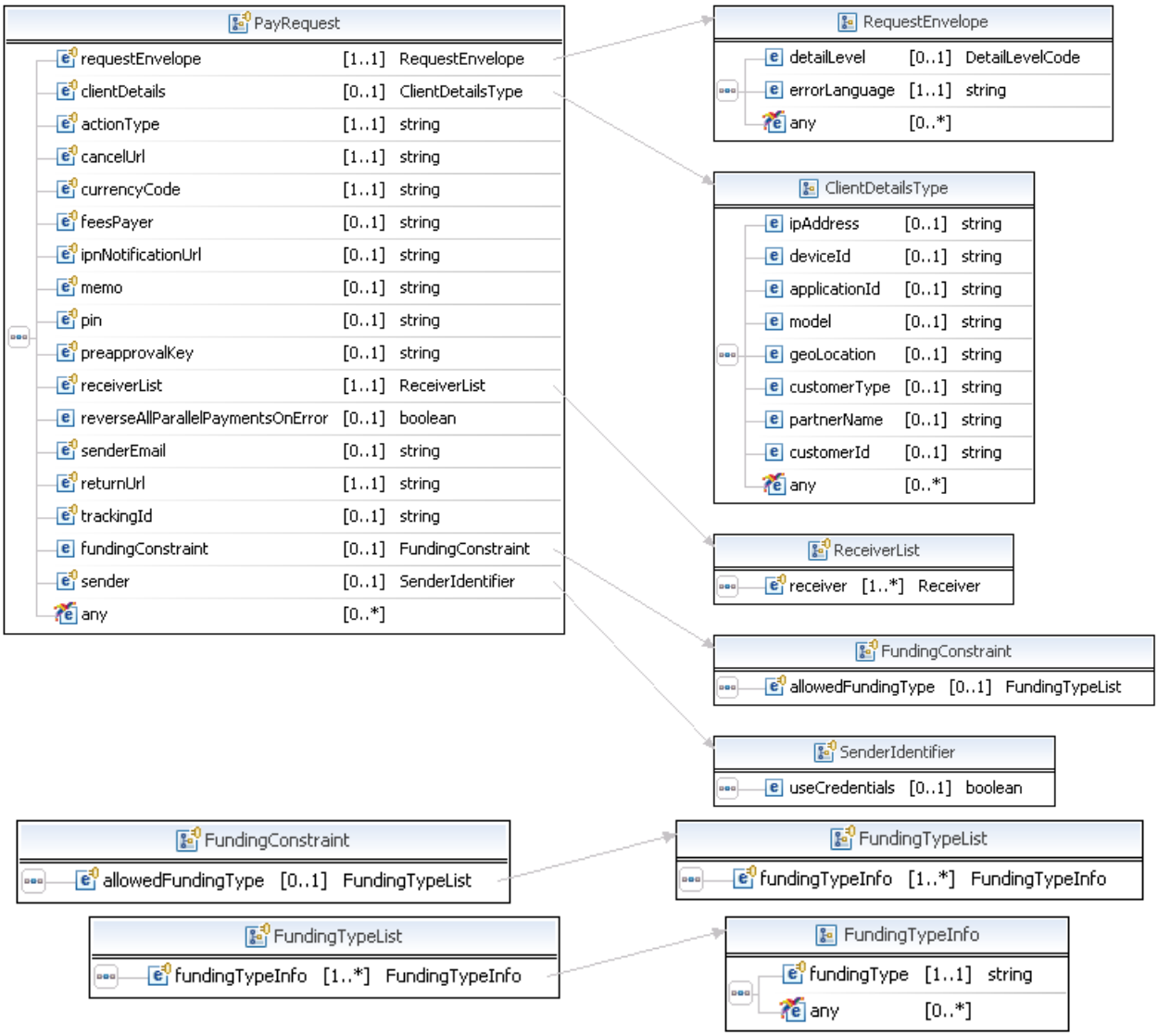
1. With parallel payments, fund transfers to some receivers may occur before others. Therefore, set `reverseAllParallelPaymentsOnError` to `true`. In the event of an error, this setting guarantees that transfers to all receivers are reversed and all funds are returned to the sender. If `reverseAllParallelPaymentsonError` is disabled, completed transfers are not reversed and funds that have already been transferred are no longer available to the sender.
2. You can specify your own unique tracking ID in the `trackingID` field and use this value to obtain information about a payment or to request a refund. The tracking ID is provided as a convenience in cases when you already maintain an ID that you want to associate with a payment. You can also track payments using the `payKey`.
3. You can use the `Pay` API operation to make *unilateral* payments under limited circumstances. A unilateral payment is a payment that is made to a receiver who does not have a PayPal account. Unilateral payments can be used with simple or parallel payments that are implicit or preapproved. Unilateral payments are not designed to be used with chained payments or payments that require manual approval through the web flow.

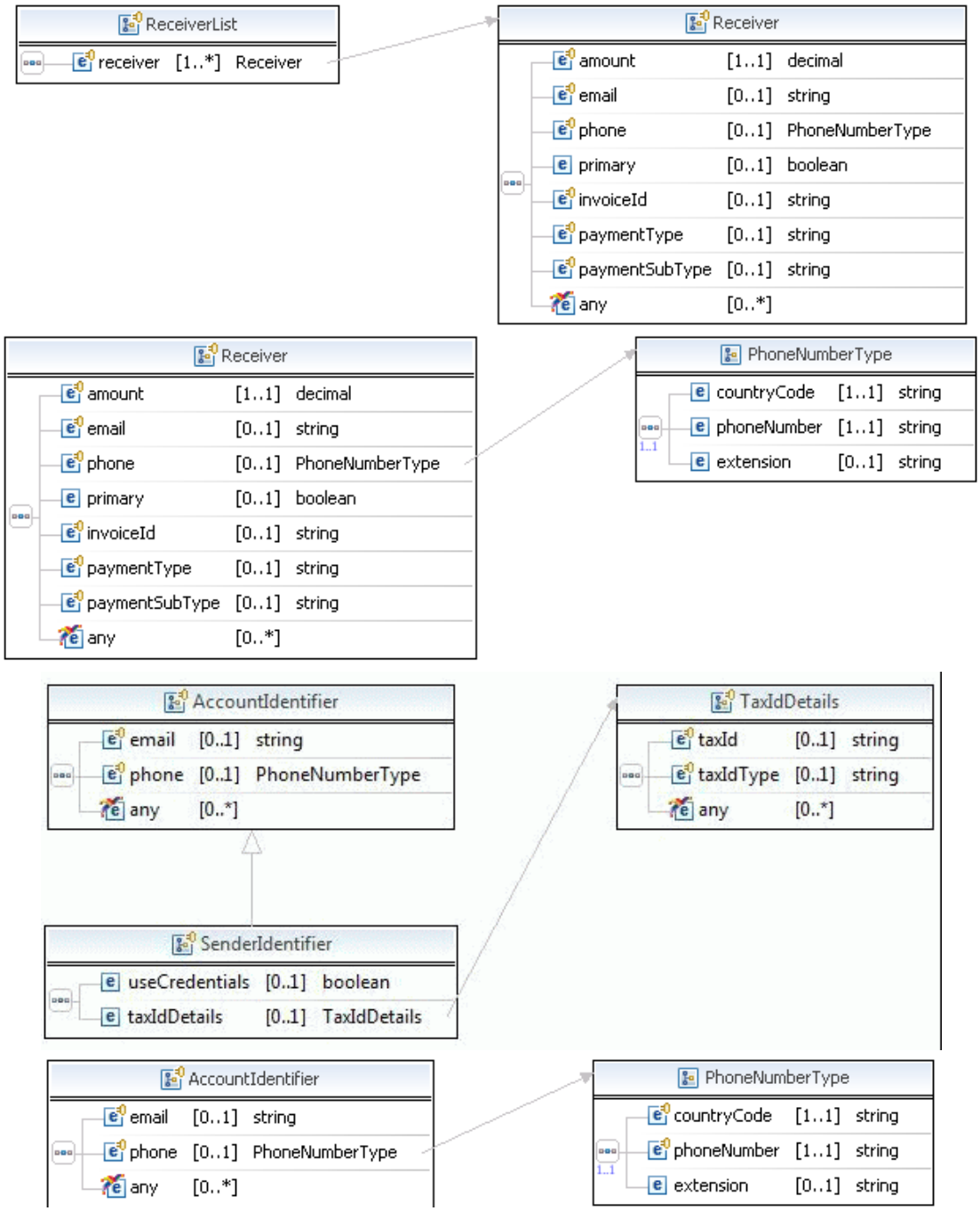
   When you send a unilateral payment, you send a payment request that includes an email address for a receiver, and this email address is not linked to a registered PayPal account. The receiver receives an email notifying the receiver to create an account and claim the payment.

   PayPal holds a payment to a receiver whose email address is not yet registered or confirmed until the receiver creates a PayPal account and confirms the email address. If a refund specifies a receiver whose email address is not yet registered or confirmed, the payment to the receiver is canceled.
4. For a guest payment, do not set the sender's email address or the preapproval key. Specifying the sender's email address prevents the sender from choosing the guest payment option. Implicit approval and preapproval options are not allowed. Each receiver of a guest payment must be a verified PayPal business or premier account holder.
5. You can specify a value for either `senderEmail` or one of the `SenderIdentifier` fields. If you use `SenderIdentifier` to identify the sender, you can only specify `sender.email` or set `sender.useCredentials` to `true`, but not both; `sender.phone` is reserved for future use.
6. Specify `senderEmail` or `sender.email` to set up a payment where the sender and the API caller are the same (implicit approval) or for a preapproved payment.
7. If `senderEmail` or `sender.email` is specified in the request, the email address must be registered with paypal.com. If you do not include one of these fields in the payment request, the sender can either log in using an existing account that is not associated with a receiver or create a new account during the payment flow.

PayRequest Message

The `PayRequest` message contains the instructions required to make a payment.




ReceiverList
receiver [1..*] Receiver
Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]
Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]
PhoneNumberType
countryCode [1..1] string
phoneNumber [1..1] string
extension [0..1] string
AccountIdentifier
email [0..1] string
phone [0..1] PhoneNumberType
any [0..*]
TaxIdDetails
taxId [0..1] string
taxIdType [0..1] string
any [0..*]
SenderIdentifier
useCredentials [0..1] boolean
taxIdDetails [0..1] TaxIdDetails
AccountIdentifier
email [0..1] string
phone [0..1] PhoneNumberType
any [0..*]
PhoneNumberType
countryCode [1..1] string
phoneNumber [1..1] string
extension [0..1] string

PayRequest Fields

Field	Description
`actionType`	`xs:string` *(Required)* Whether the `Pay` request pays the receiver or whether the `Pay` request is set up to create a payment request, but not fulfill the payment until the `ExecutePayment` is called. Allowable values are: • `PAY` – Use this option if you are not using the `Pay` request in combination with `ExecutePayment`. • `CREATE` – Use this option to set up the payment instructions with `SetPaymentOptions` and then execute the payment at a later time with the `ExecutePayment`. • `PAY_PRIMARY` – For chained payments only, specify this value to delay payments to the secondary receivers; only the payment to the primary receiver is processed.
`cancelUrl`	`xs:string` *(Required)* The URL to which the sender's browser is redirected if the sender cancels the approval for the payment after logging in to paypal.com to approve the payment. Specify the URL with the HTTP or HTTPS. Maximum length: 1024 characters
`clientDetails`	`common:ClientDetailsType` *(Optional)* Information about the sender.

Field	Description
currencyCode	xs:string *(Required)* The currency code. Allowable values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD
feesPayer	xs:string *(Optional)* The payer of PayPal fees. Allowable values are: • SENDER – Sender pays all fees (for personal, implicit simple/parallel payments; do not use for chained or unilateral payments) • PRIMARYRECEIVER – Primary receiver pays all fees (chained payments only) • EACHRECEIVER – Each receiver pays their own fee (default, personal and unilateral payments) • SECONDARYONLY – Secondary receivers pay all fees (use only for chained payments with one secondary receiver)

Field	Description
fundingConstraint	ap:FundingConstraint *(Optional)* Specifies a list of allowed funding types for the payment. This is a list of funding selections that can be combined in any order to allow payments to use the indicated funding type. If this Parameter is omitted, the payment can be funded by any funding type that is supported for Adaptive Payments. **NOTE:** FundingConstraint is unavailable to API callers with standard permission levels; for more information, refer to the section Adaptive Payments Permission Levels.
ipnNotificationUrl	xs:string *(Optional)* The URL to which you want all IPN messages for this payment to be sent. Maximum length: 1024 characters
memo	xs:string *(Optional)* A note associated with the payment (text, not HTML). Maximum length: 1000 characters, including newline characters
pin	xs:string *(Optional)* The sender's personal identification number, which was specified when the sender signed up for a preapproval.
preapprovalKey	xs:string *(Optional)* The key associated with a preapproval for this payment. The preapproval key is required if this is a preapproved payment. **NOTE:** The Preapproval API is unavailable to API callers with Standard permission levels.
receiverList	ap:ReceiverList *(Required)* Information about the receivers of the payment.
requestenvelope	common:RequestEnvelope *(Required)* Information common to each Method, such as the language in which an error message is returned.
returnUrl	xs:string *(Required)* The URL to which the sender's browser is redirected after approving a payment on paypal.com. Specify the URL with the HTTP or HTTPS designator. Maximum length: 1024 characters
reverseAllParallelPaymentsOnError	xs:boolean *(Optional)* Whether to reverse parallel payments if an error occurs with a payment. Allowable values are: • true – Each parallel payment is reversed if an error occurs • false – Only incomplete payments are reversed (default)

Field	Description
`senderEmail`	`xs:string` *(Optional)* Sender's email address. Maximum length: 127 characters
`sender`	`ap:SenderIdentifier` *(Optional)* Sender's identifying information.
`trackingId`	`xs:string` *(Optional)* A unique ID that you specify to track the payment. **NOTE:** You are responsible for ensuring that the ID is unique. Maximum length: 127 characters

ClientDetails Fields

Field	Description
`applicationId`	`xs:string` *(Optional)* Your application's identification, such as the name of your application
`customerId`	`xs:string` *(Optional)* Your ID for this sender Maximum length: 127 characters
`customerType`	`xs:string` *(Optional)* Your identification of the type of customer Maximum length: 127 characters
`deviceId`	`xs:string` *(Optional)* Sender's device ID, such as a mobile device's IMEI number or a web browser cookie. If a device ID was passed with the `PayRequest`, use the same ID here. Maximum length: 127 characters
`geoLocation`	`xs:string` *(Optional)* Sender's geographic location Maximum length: 127 characters
`ipAddress`	`xs:string` *(Optional)* Sender's IP address. If an IP addressed was passed with the `PayRequest`, use the same ID here.
`model`	`xs:string` *(Optional)* A sub-identification of the application Maximum length: 127 characters

Field	Description
partnerName	xs:string *(Optional)* Your organization's name or ID Maximum length: 127 characters

FundingConstraint Fields

Field	Description
allowedFundingType	ap:FundingTypeList *(Optional)* Specifies a list of allowed funding selections for the payment. This is a list of funding selections that can be combined in any order to allow payments to use the indicated funding type. If this field is omitted, the payment can be funded by any funding type that is supported for Adaptive Payments. **NOTE:** FundingConstraint is unavailable to API callers with standard permission levels; for more information, refer to the section Adaptive Payments Permission Levels. **NOTE:** To use iACH, omit this field and do not specify a funding source for the payment.

FundingTypeList Fields

Field	Description
fundingTypeInfo	ap:FundingTypeInfo *(Optional)* Specifies a list of allowed funding selections for the payment. This is a list of funding selections that can be combined in any order to allow payments to use the indicated funding type. If this field is omitted, the payment can be funded by any funding type that is supported for Adaptive Payments. **NOTE:** FundingConstraint is unavailable to API callers with standard permission levels; for more information, refer to the section Adaptive Payments Permission Levels.

FundingTypeInfo Fields

Field	Description
`fundingType`	`xs:string` *(Required)* Specifies a list of allowed funding selections for the payment. This is a list of funding selections that can be combined in any order to allow payments to use the indicated funding type. If this field is omitted, the payment can be funded by any funding type that is supported for Adaptive Payments. Allowable values are: • `ECHECK` – Electronic check • `BALANCE` – PayPal account balance • `CREDITCARD` – Credit card **NOTE:** `ECHECK` and `CREDITCARD` include `BALANCE` implicitly. **NOTE:** `FundingConstraint` is unavailable to API callers with standard permission levels; for more information, refer to the section Adaptive Payments Permission Levels.

ReceiverList Fields

Field	Description
`receiver`	`ap:Receiver` *(Required)* Receiver is the party whose account is credited.

Receiver Fields

Field	Description
`amount`	`xs:decimal` *(Required)* Amount to be paid to the receiver.
`email`	`xs:string` Receiver's email address. This address can be unregistered with paypal.com. If so, a receiver cannot claim the payment until a PayPal account is linked to the email address. The `PayRequest` must pass either an email address or a phone number. Maximum length: 127 characters
`invoiceId`	`xs:string` *(Optional)* The invoice number for the payment. This data in this field shows on the Transaction Details report. Maximum length: 127 characters

Field	Description
paymentType	xs:string *(Optional)* The transaction type for the payment. Allowable values are: • GOODS – This is a payment for non-digital goods • SERVICE – This is a payment for services (default) • PERSONAL – This is a person-to-person payment • CASHADVANCE – This is a person-to-person payment for a cash advance • DIGITALGOODS – This is a payment for digital goods • BANK_MANAGED_WITHDRAWAL – This is a person-to-person payment for bank withdrawals, available only with special permission. **NOTE:** Person-to-person payments are valid only for parallel payments that have the feesPayer field set to EACHRECEIVER or SENDER.
paymentSubType	xs:string *(Optional)* The transaction subtype for the payment.
phone	common:PhoneNumberType A type to specify the receiver's phone number. The PayRequest must pass either an email address or a phone number as the payment receiver.
primary	xs:boolean *(Optional)* Whether this receiver is the primary receiver, which makes the payment a chained payment. You can specify at most one primary receiver. Omit this field for simple and parallel payments. Allowable values are: • true – Primary receiver • false – Secondary receiver (default)

PhoneNumberType Fields

Field	Description
countryCode	xs:string *(Required)* Telephone country code.
phoneNumber	xs:string *(Required)* Telephone number.
extension	xs:string *(Optional)* Telephone extension.

SenderIdentifier Fields

Field	Description
`useCredentials`	`xs:boolean` *(Optional)* If `true`, use credentials to identify the sender; default is `false`.

AccountIdentifier Fields

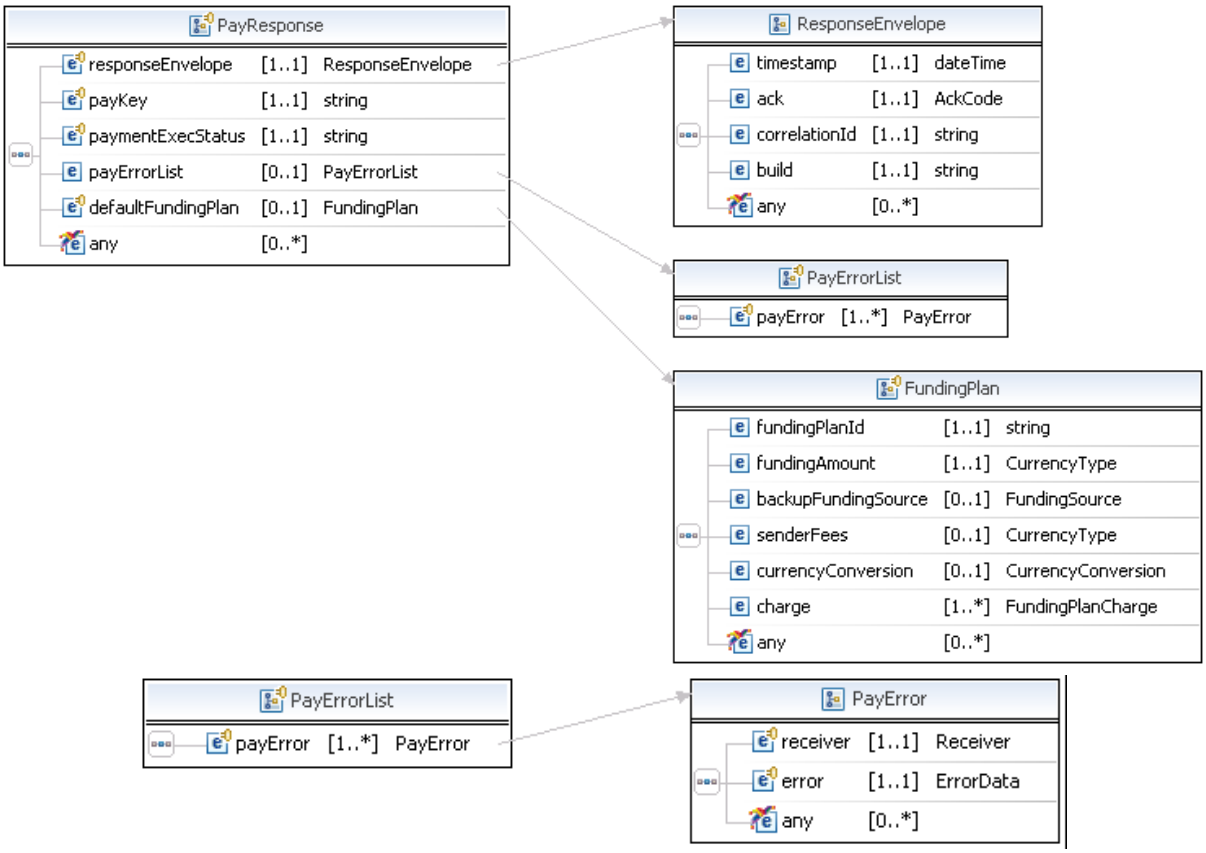
Field	Description
`email`	`xs:string` *(Optional)* Sender's email address. Maximum length: 127 characters
`phone`	`common:PhoneNumberType` *(Optional)* Sender's phone number.

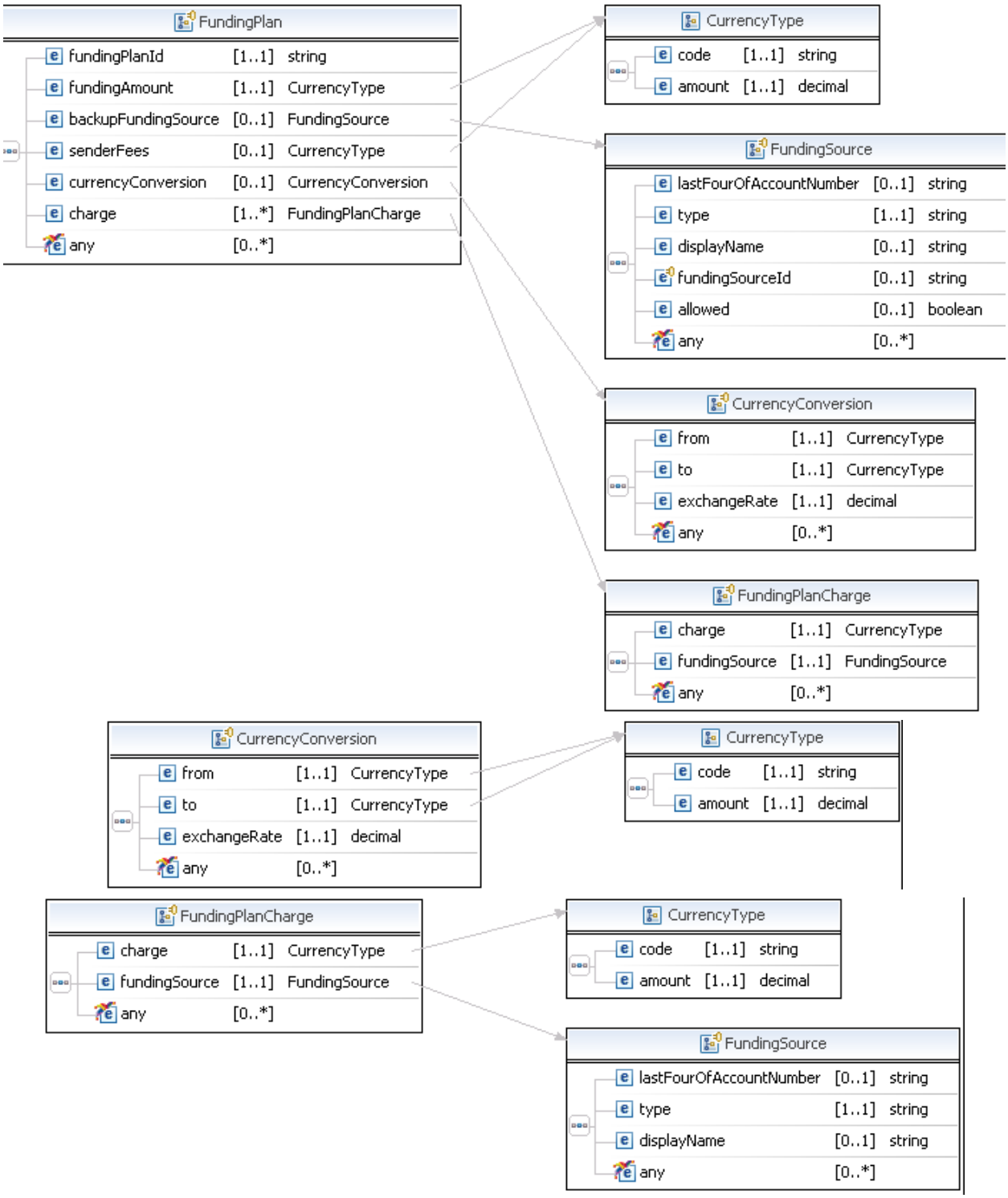
RequestEnvelope Fields

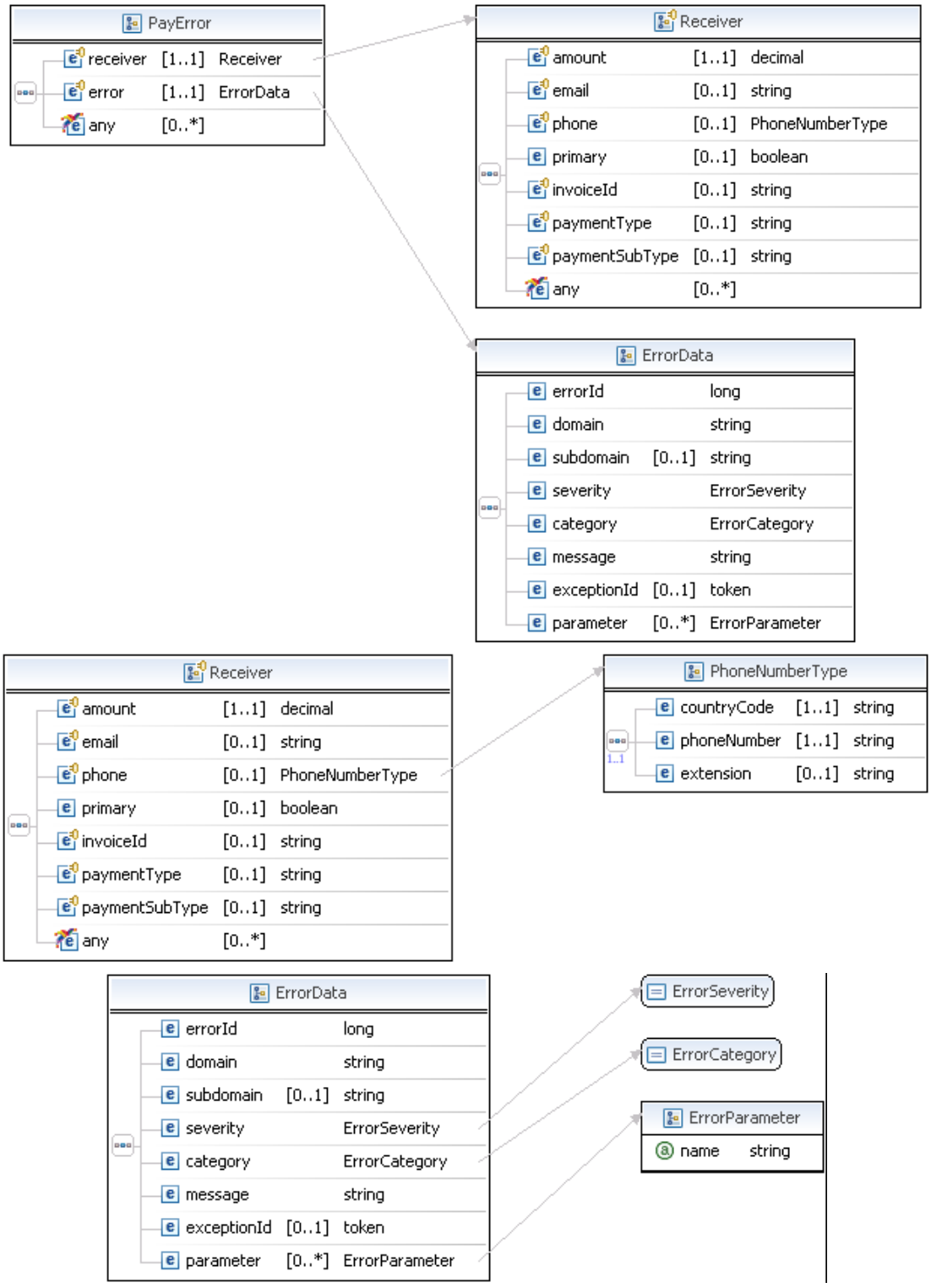
Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.
`errorLanguage`	`xs:string` *(Required)* RFC 3066 language in which error messages are returned; by default it is `en_US`, which is the only language currently supported.

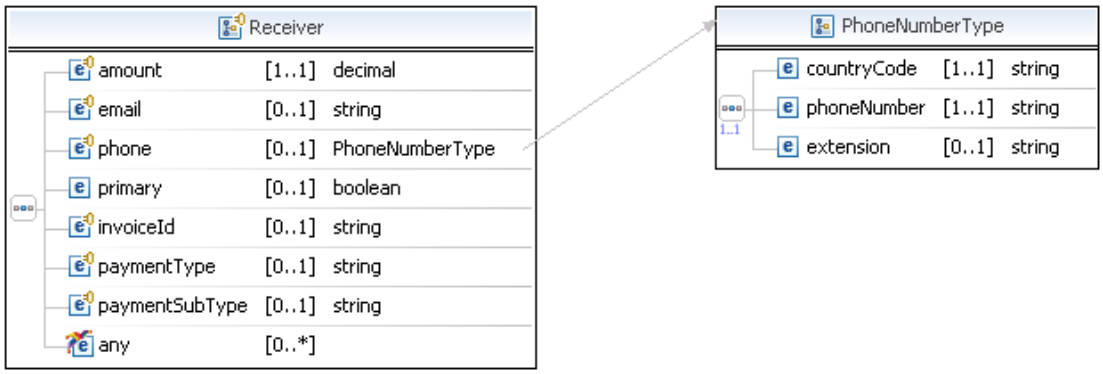
PayResponse Message

The `PayResponse` message contains a key that you can use to identify the payment and the payment's status.


PayResponse
responseEnvelope [1..1] ResponseEnvelope
payKey [1..1] string
paymentExecStatus [1..1] string
payErrorList [0..1] PayErrorList
defaultFundingPlan [0..1] FundingPlan
any [0..*]
ResponseEnvelope
timestamp [1..1] dateTime
ack [1..1] AckCode
correlationId [1..1] string
build [1..1] string
any [0..*]
PayErrorList
payError [1..*] PayError
FundingPlan
fundingPlanId [1..1] string
fundingAmount [1..1] CurrencyType
backupFundingSource [0..1] FundingSource
senderFees [0..1] CurrencyType
currencyConversion [0..1] CurrencyConversion
charge [1..*] FundingPlanCharge
any [0..*]
PayErrorList
payError [1..*] PayError
PayError
receiver [1..1] Receiver
error [1..1] ErrorData
any [0..*]


FundingPlan
fundingPlanId [1..1] string
fundingAmount [1..1] CurrencyType
backupFundingSource [0..1] FundingSource
senderFees [0..1] CurrencyType
currencyConversion [0..1] CurrencyConversion
charge [1..*] FundingPlanCharge
any [0..*]
CurrencyType
code [1..1] string
amount [1..1] decimal
FundingSource
lastFourOfAccountNumber [0..1] string
type [1..1] string
displayName [0..1] string
fundingSourceId [0..1] string
allowed [0..1] boolean
any [0..*]
CurrencyConversion
from [1..1] CurrencyType
to [1..1] CurrencyType
exchangeRate [1..1] decimal
any [0..*]
FundingPlanCharge
charge [1..1] CurrencyType
fundingSource [1..1] FundingSource
any [0..*]
CurrencyConversion
from [1..1] CurrencyType
to [1..1] CurrencyType
exchangeRate [1..1] decimal
any [0..*]
CurrencyType
code [1..1] string
amount [1..1] decimal
FundingPlanCharge
charge [1..1] CurrencyType
fundingSource [1..1] FundingSource
any [0..*]
CurrencyType
code [1..1] string
amount [1..1] decimal
FundingSource
lastFourOfAccountNumber [0..1] string
type [1..1] string
displayName [0..1] string
any [0..*]


PayError
receiver [1..1] Receiver
error [1..1] ErrorData
any [0..*]
Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]
ErrorData
errorId long
domain string
subdomain [0..1] string
severity ErrorSeverity
category ErrorCategory
message string
exceptionId [0..1] token
parameter [0..*] ErrorParameter
Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]
PhoneNumberType
countryCode [1..1] string
phoneNumber [1..1] string
extension [0..1] string
ErrorData
errorId long
domain string
subdomain [0..1] string
severity ErrorSeverity
category ErrorCategory
message string
exceptionId [0..1] token
parameter [0..*] ErrorParameter
ErrorSeverity
ErrorCategory
ErrorParameter
name string



PayResponse Fields

Field	Description
payKey	xs:string The pay key, which is a token you use in other Adaptive Payment APIs (such as the Refund Method) to identify this payment. The pay key is valid for **3 hours**; the payment must be approved while the pay key is valid.
payErrorList	xs:string Information about why a payment failed.
paymentExecStatus	xs:string The status of the payment. Possible values are: • CREATED – The payment request was received; funds will be transferred once the payment is approved • COMPLETED – The payment was successful • INCOMPLETE – Some transfers succeeded and some failed for a parallel payment or, for a delayed chained payment, secondary receivers have not been paid • ERROR – The payment failed and all attempted transfers failed or all completed transfers were successfully reversed • REVERSALERROR – One or more transfers failed when attempting to reverse a payment • PROCESSING – The payment is in progress • PENDING – The payment is awaiting processing
defaultFundingPlan	ap:FundingPlan Default funding plan.
responseEnvelope	common:responseEnvelope Common response information, including a timestamp and the response acknowledgement status.
warningDataList	ap:WarningDataList Warning ID and message related to payments that may be in the pending state.

PayErrorList Fields

Field	Description
payError	PayError indicates the error, if any, that resulted on an attempted payment to a receiver.

FundingPlan Fields

Field	Description
fundingPlanId	xs:string ID for this funding plan.
fundingAmount	common:CurrencyType Amount of available funding.
backupFundingSource	ap:FundingSource Backup funding source.
senderFees	common:CurrencyType Fees to be paid by the sender.
currencyConversion	ap:CurrencyConversion The currency conversion associated with the funding plan.
charge	ap:FundingPlanCharge The amount to be charged to this funding plan.

CurrencyType Fields

Field	Description
amount	xs:decimal The converted amount.

Field	Description
code	xs:string The currency code for the converted amount. Possible values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD

FundingSource Fields

Field	Description
lastFourOfAccountNumber	xs:string Last 4 digits or characters in account number.

Field	Description
type	xs:string Type of funding source, which is one of the following values:. • UNDEFINED • BALANCE • BANK_INSTANT • BANK_DELAYED • CREDITCARD • DEBITCARD • ACCOUNTS_RECEIVABLE
displayName	xs:string Display name of funding source.
fundingSourceId	xs:string Funding source ID.
allowed	xs:boolean Whether the funding source is allowed for this payment: • true – You can use this funding source for the payment • false – You cannot use this funding source (default)

CurrencyConversion Fields

Field	Description
from	ap:CurrencyType The currency to be converted.
to	ap:CurrencyType The currency resulting from the conversion.
exchangeRate	xs:decimal The exchange rate for the from currency to the to currency.

FundingPlanCharge Fields

Field	Description
charge	common:CurrencyType Amount charged to funding source.
fundingSource	ap:FundingSource Funding source being charged.

WarningDataLists Fields

Field	Description
`warningId`	`xs:long` ID corresponding to a service operation warning.
`message`	`xs:string` Message corresponding to a service operation warning.

PayError Fields

Field	Description
`error`	Detailed error information.
`receiver`	`ap:Receiver` Receiver is the party where funds are transferred to. A primary receiver receives a payment directly from the sender in a chained split payment. A primary receiver should not be specified when making a single or parallel split payment.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.

Field	Description
severity	common:ErrorSeverity The severity of the error encountered. Possible values are: • Error – Processing of the request was interrupted • Warning – Processing of the request was completed
subdomain	This field is not used.

Receiver Fields

Field	Description
amount	xs:decimal Amount to be paid to the receiver.
email	xs:string Receiver's email address. Maximum length: 127 characters
invoiceId	xs:string The invoice number for the payment. This data in this field shows on the Transaction Details report. Maximum length: 127 characters
paymentType	xs:string The transaction type for the payment. Possible values are: • GOODS – This is a payment for non-digital goods • SERVICE – This is a payment for services (default) • PERSONAL – This is a person-to-person payment • CASHADVANCE – This is a person-to-person payment for a cash advance • DIGITALGOODS – This is a payment for digital goods • BANK_MANAGED_WITHDRAWAL – This is a person-to-person payment for bank withdrawals, available only with special permission.
paymentSubType	xs:string The transaction subtype for the payment.
phone	common:PhoneNumberType The receiver's phone number.
primary	xs:boolean Whether this receiver is the primary receiver. If this field is set to true, this is a chained payment. If this field shows false, this is a simple or parallel payment. Possible values are: • true – Primary receiver (chained payment) • false – Secondary receiver (simple/parallel payment)

PhoneNumberType Response Fields

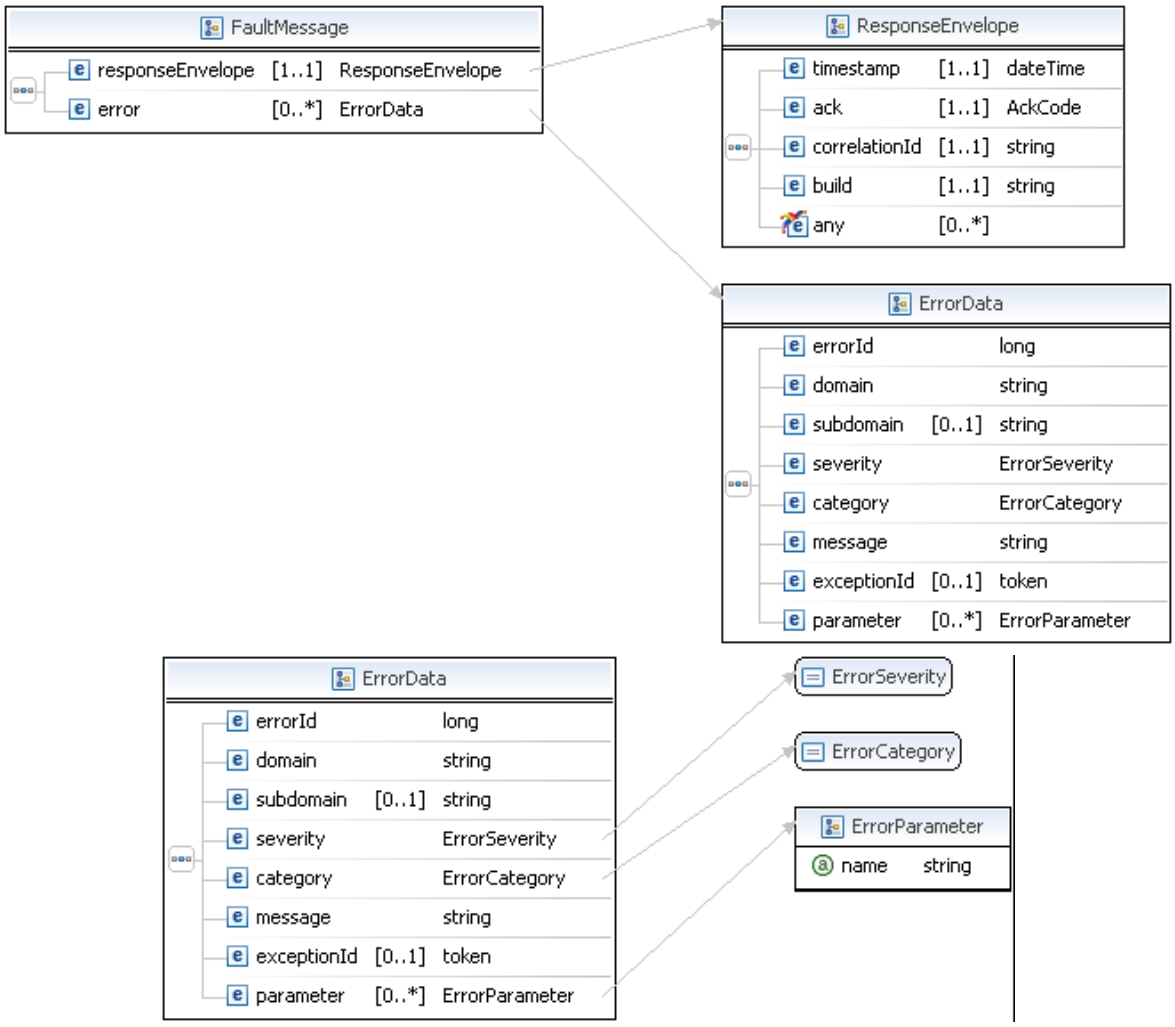
Field	Description
countryCode	xs:string Telephone country code.
phoneNumber	xs:string Telephone number.
extension	xs:string Telephone extension.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The PPFaultMessage returns ErrorData and the ResponseEnvelope information to your application if an error occurs.



FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.
`severity`	`common:ErrorSeverity` The severity of the error encountered. Possible values are: • `Error` – Processing of the request was interrupted • `Warning` – Processing of the request was completed
`subdomain`	This field is not used.

ResponseEnvelope Fields

Field	Description
`ack`	`common:AckCode` Acknowledgement code. It is one of the following values: • `Success` – The operation completed successfully. • `Failure` – The operation failed. • `SuccessWithWarning` – The operation completed successfully; however, there is a warning message. • `FailureWithWarning` – The operation failed with a warning message.
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

Pay Examples Using NVP and CURL

These examples all use NVP for the data binding and CURL to deliver the HTTP request to the PayPal sandbox endpoint. Line breaks are provided for ease of reading; each CURL command is a single line and each request and response is a string without line breaks or extra whitespace.

Simple payment example

In this example, the sender makes a payment of $100 to a PayPal-registered receiver. If you are the sender and the caller, the approval is implicit; otherwise, the sender must explicitly approve the payment.

The status of the request, which is identified in the paymentExecStatus field of the response, differs due to the kind of approval required. Implicit approval allows the request to be completed immediately. Explicit approval allows the request to be created; however, it is completed until the payment is approved.

NOTE: The sample code below uses the insecure setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/Pay -d
"requestEnvelope.errorLanguage=en_US
&actionType=PAY
&senderEmail=sender@domain
&receiverList.receiver(0).email=receiver@domain
&receiverList.receiver(0).amount=100.00
&currencyCode=USD
&feesPayer=EACHRECEIVER
&memo=Simple payment example.
&cancelUrl=http://your_cancel_url
&returnUrl=http://your_return_url
&ipnNotificationUrl=http://your_ipn_notification_url"
```

Response for an explicitly approved payment:

```
responseEnvelope.timestamp=2009-07-13T12%3A34%3A29.316-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=d615a365bed61
&responseEnvelope.build=DEV
&payKey=AP-3TY011106S4428730
&paymentExecStatus=CREATED
```

NOTE: You must redirect the sender to PayPal to complete the payment.

Response for an implicitly approved payment:

```
responseEnvelope.timestamp=2009-07-10T11%3A47%3A29.311-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=34e44c0bdbed6
&responseEnvelope.build=DEV
&payKey=AP-54224401WG093204T
&paymentExecStatus=COMPLETED
```

Parallel payment example

In this example, the sender makes a payment of $100 to a PayPal-registered receiver and $50 to another PayPal-registered receiver. If an error occurs, all funds are returned to the sender whether or not the funds were transferred to a receiver.

NOTE: The sample code below uses the `insecure` setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/Pay  -d
"actionType=PAY
&cancelUrl=http://your_cancel_url
&currencyCode=USD
&feesPayer=EACHRECEIVER
&memo=Parallel-payment-example
&receiverList.receiver(0).amount=100.00
&receiverList.receiver(0).email=receiver1@domain
&receiverList.receiver(0).primary=false
&receiverList.receiver(1).amount=50.00
&receiverList.receiver(1).email=receiver2@domain
&receiverList.receiver(1).primary=false
&requestEnvelope.errorLanguage=en_US
&returnUrl=http://your_return_url
&reverseAllParallelPaymentsOnError=true
&senderEmail=sender@domain
```

Response:

```
responseEnvelope.timestamp=2009-07-13T12%3A34%3A29.316-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=d615a365bed61
&responseEnvelope.build=DEV&payKey=AP-1CB41255YH2406602
&paymentExecStatus=CREATED
```

Chained payment example

In this example, the sender makes a payment of $100 to a PayPal-registered receiver who is identified as the primary receiver. This receiver sends $50 of the $100 to another PayPal-registered receiver.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/Pay  -d
"actionType=PAY
&cancelUrl=http://your_cancel_url
&clientDetails.applicationId=fitnesse
&clientDetails.ipAddress=127.0.0.1
&currencyCode=USD
&feesPayer=EACHRECEIVER
&memo=Parallel-payment-example
&receiverList.receiver(0).amount=100.00
&receiverList.receiver(0).email=receiver1@domain
&receiverList.receiver(0).primary=true
&receiverList.receiver(1).amount=50.00
&receiverList.receiver(1).email=receiver2@domain
&receiverList.receiver(1).primary=false
&requestEnvelope.errorLanguage=en_US
&returnUrl=http://your_return_url
&senderEmail=sender@domain
```

Response:

```
responseEnvelope.timestamp=2009-07-13T12%3A34%3A29.316-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=d615a365bed61
&responseEnvelope.build=DEV
&payKey=AP-7AF63289GN043650D
&paymentExecStatus=CREATED
```

Preapproved payment example

In this example, the sender has a valid preapproval agreement with you and makes a payment of $100 to a PayPal-registered receiver and $50 to another PayPal-registered receiver. The payment is completed without the sender logging in to paypal.com.

NOTE: The sample code below uses the `insecure` setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/Pay  -d
"actionType=PAY
&cancelUrl=http://your_cancel_url
&currencyCode=USD
&feesPayer=EACHRECEIVER
&memo=Preapproval-payment-example
&preapprovalKey=PA-9JR04288NR0519129
&receiverList.receiver(0).amount=100.00
&receiverList.receiver(0).email=receiver1@domain
&receiverList.receiver(0).primary=false
receiverList.receiver(1).amount=50.00
receiverList.receiver(1).email=receiver2@domain
receiverList.receiver(1).primary=false
&requestEnvelope.errorLanguage=en_US
&returnUrl=http://your_return_url
&reverseAllParallelPaymentsOnError=true
&senderEmail=sender@domain
```

Response:

```
responseEnvelope.timestamp=2009-07-13T12%3A34%3A29.316-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=d615a365bed61
&responseEnvelope.build=DEV
&payKey=AP-42706441J12795911
&paymentExecStatus=COMPLETED
```

Digital goods payment

In this example, the sender makes a payment of $1 to a PayPal-registered receiver. The sender must explicitly approve the payment.

NOTE: The sample code below uses the `insecure` setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/Pay  -d
"requestEnvelope.errorLanguage=en_US
&actionType=PAY
&receiverList.receiver(0).amount=1.00
&currencyCode=USD
&feesPayer=EACHRECEIVER
&memo=Simple payment example.
&cancelUrl=http://your_cancel_url
&returnUrl=http://your_return_url
&ipnNotificationUrl=http://your_ipn_notification_url
&receiverList.receiver(0).paymentType=DIGITALGOODS"
```

Response to a payment for digital goods:

```
responseEnvelope.timestamp=2009-07-13T12%3A34%3A29.316-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=d615a365bed61
&responseEnvelope.build=DEV
&payKey=AP-3TY011106S4428730
&paymentExecStatus=CREATED
```

NOTE: You must redirect the sender to the following PayPal URL to complete the payment for digital goods:
`https://www.paypal.com/webapps/adaptivepayment/flow/pay?paykey=...`

Pay Errors

Code	Message	Additional Information
500000	There is a system error	
520002	Internal error	
520003	User name/password is incorrect	
520003	• The caller's credential doesn't have permission for the fee payer <value> • The caller's credential doesn't have permission for the payment type <value> • Authentication failed. API credentials are incorrect.	Since 1.6.0

Code	Message	Additional Information
520005	Merchant account is locked	
520006	This call is not defined in the database	
529038	There was an error while making this payment	
539012	The preapproval key has not been authorized yet	
539041	The email account is based in a country that is not enabled to receive payments	
539043	The email account is based in a country that is not enabled to send payments	
540031	You don't have permission to cancel this preapproval	
550001	• You are not allowed to confirm the preapproval for this key • User is not allowed to perform this action.	Since 1.6.0
559044	Account setting on the receiver prohibited the payment	
560027	The argument value is unsupported	
569000	Split payments are not supported at this time	
569013	The preapproval key has been canceled	
569016	Preapproval PIN functionality is not enabled	
569017	The preapproval key has been suspended	
569018	Preapproved payments have been disabled	
569019	The preapproval has been suspended due to too many PIN failures	
569042	The email account is not confirmed by PayPal	
579007	The maximum number of receivers is <number>	
579010	If a preapproval key is specified, the sender's email address must be, too	
579014	The preapproval key specifies a different sender than the payment request	
579017	The amount for the primary receiver must be greater than or equal to the total of other chained receiver amounts	
579024	The preapproval key cannot be used before the start date or after the end date	
579025	The preapproval key cannot be used on this weekday	
579026	The preapproval key cannot be used on this day of the month	
579027	The preapproval key specifies a different currency than the payment request	

Code	Message	Additional Information
579028	The payment amount exceeds the maximum amount per payment	
579030	The number of payments made this period exceeds the maximum number of payments per period	
579031	The total amount of all payments exceeds the maximum total amount for all payments	
579033	The sender and each receiver must have different accounts	
579040	The receivers cannot belong to the same PayPal account	
579042	The tracking ID already exists and cannot be duplicated	
579045	The email account exceeds the receiving limit	
579047	The email account exceeds the purse limit	
579048	The email account exceeds the sending limit	
580001	Invalid request	
580001	• Invalid request: both email and phone cannot be set <value> <value> <value> • Invalid request: neither email nor phone is set <value> • To use currency <value>, currency must be held by sender and all receivers • Invalid request: More than one field cannot be used to specify a sender • Invalid request: <value>	Since 1.6.0
580022	Your request was invalid. Check the parameter of the error to see which is invalid	
580022	• Invalid request parameter: <value> • Invalid Request Parameter: institutionId with value <value> • The institution id <value> is not activated • The clientDetails parameter is missing • Invalid request parameter: payKey with value <value> • Invalid Phone Country Code <value> • Invalid Country Code <value> for Receiver • Invalid Receiver Phone Type <value> <value> <value> • Invalid Phone Format <value> <value> • Invalid Phone Number <value> <value> • Invalid Receiver Phone Number <value> <value> • Invalid Area or Exchange Code <value> <value> • <value> has already been set and cannot be set multiple times • Invalid request parameter: <value> with value <value> • Invalid request parameter: <value> <value> is invalid • Invalid request parameter: <value> <value> is duplicated • Invalid request parameter: <value> <value> cannot be negative or zero	Since 1.6.0

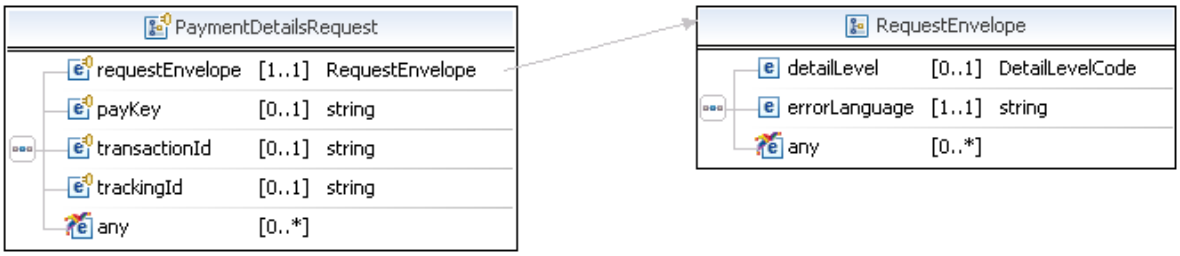
Code	Message	Additional Information
580023	Invalid request	
580023	• The fee payer <value> cannot be used if a primary receiver is specified • The fee payer <value> can only be used if a primary receiver is specified • The fee payer <value> must have exactly 2 receivers • The reverseAllParallelPaymentOnError field can be true only if a primary receiver is not specified • The fee payer <value> cannot be used if a preapproval key is specified • Invalid request: FundingType cannot be specified when PaymentSubType is specified • Your Application is not approved to use the PaymentSubType: <value> • Your Application is not approved to use the PaymentType: <value> • Invalid request parameter: PaymentType should be the same for all receivers <value> • Inconsistant request parameter for digitalgoods • Invalid request parameter: PaymentSubType should be the same for all receivers <value> • Invalid Request: RECEIVABLE cannot be used with other funding types • For chained payments, receivers must have premier or business accounts	Since 1.6.0
580027	The argument is unsupported	
580027	• Currently the system does not accept the currency <value> • Unsupported country code <value> • Currently the system does not accept the currency <value> • The parameter is not supported	Since 1.6.0
580028	The URL <value> is malformed	
580029	Invalid request	
580030	This transaction cannot be processed at this time. Please try again later.	Since 1.3.0
589009	This payment cannot be processed because no payment source is available	
589019	The preapproval key is invalid	
589023	If a fractional amount is rounded due to currency conversion, funds could be lost	
589039	The email address is invalid. It may not be registered in PayPal's system yet	

4 PaymentDetails API Operation

Use the `PaymentDetails` API operation to obtain information about a payment. You can identify the payment by your tracking ID, the PayPal transaction ID in an IPN message, or the pay key associated with the payment.

PaymentDetailsRequest Message

The `PaymentDetailsRequest` message contains the identifier used to retrieve information about the payment.



PaymentDetailsRequest Fields

The `payKey`, `transactionId`, and `trackingId` fields identify the payment. You can supply either the `payKey` field or the `trackingId` field in your request. If you supply both fields, the `payKey` field will be used to identify the payment and the `trackingId` field will be ignored.

Field	Description
`payKey`	`xs:string` *(Optional)* The pay key that identifies the payment for which you want to retrieve details. This is the pay key returned in the `PayResponse` message.
`requestEnvelope`	`common:RequestEnvelope` *(Required)* Information common to each API operation, such as the language in which an error message is returned.
`transactionId`	`xs:string` *(Optional)* The PayPal transaction ID associated with the payment. The IPN message associated with the payment contains the transaction ID.

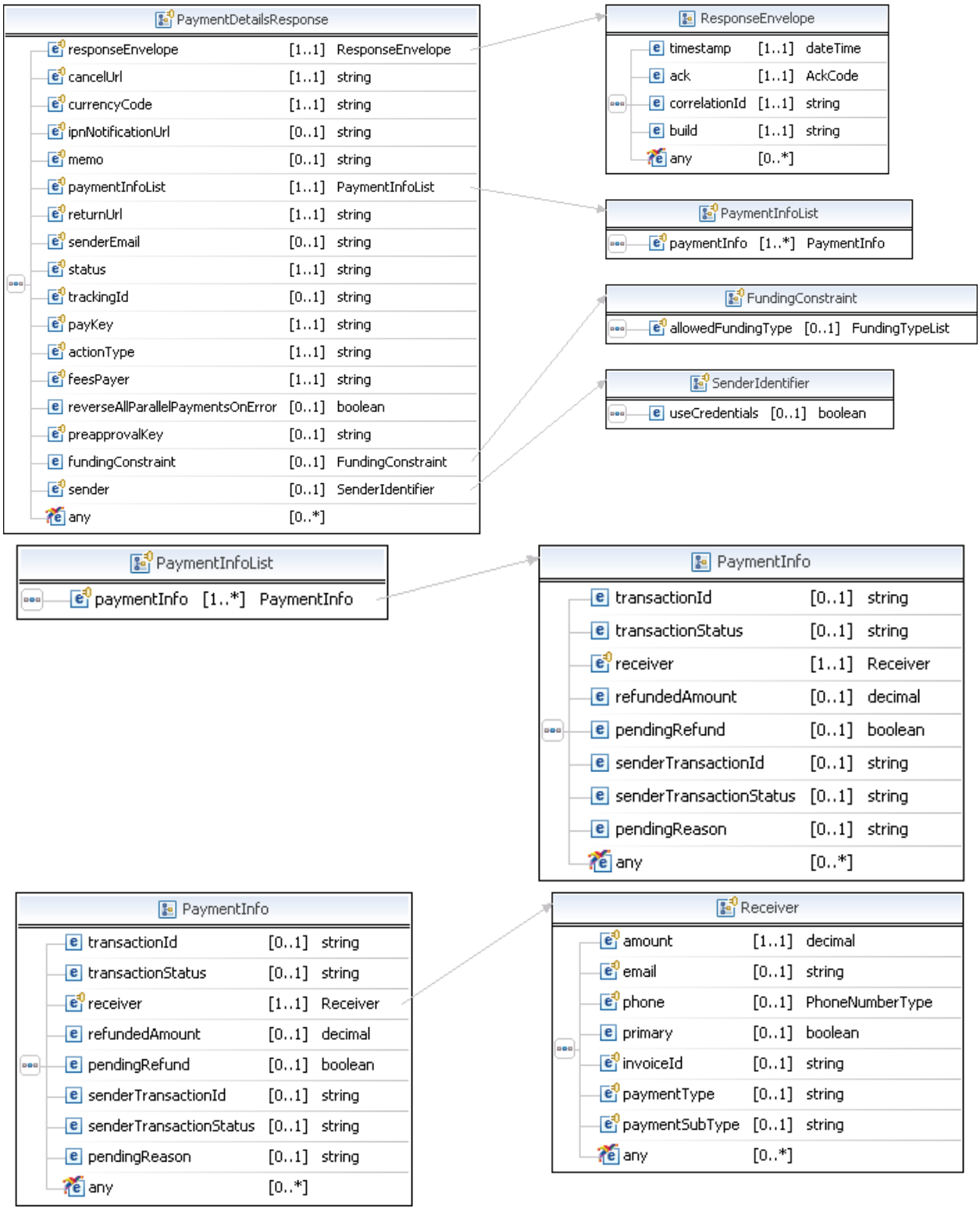
Field	Description
`trackingId`	`xs:string` *(Optional)* The tracking ID that was specified for this payment in the `PayRequest` message. Maximum length: 127 characters

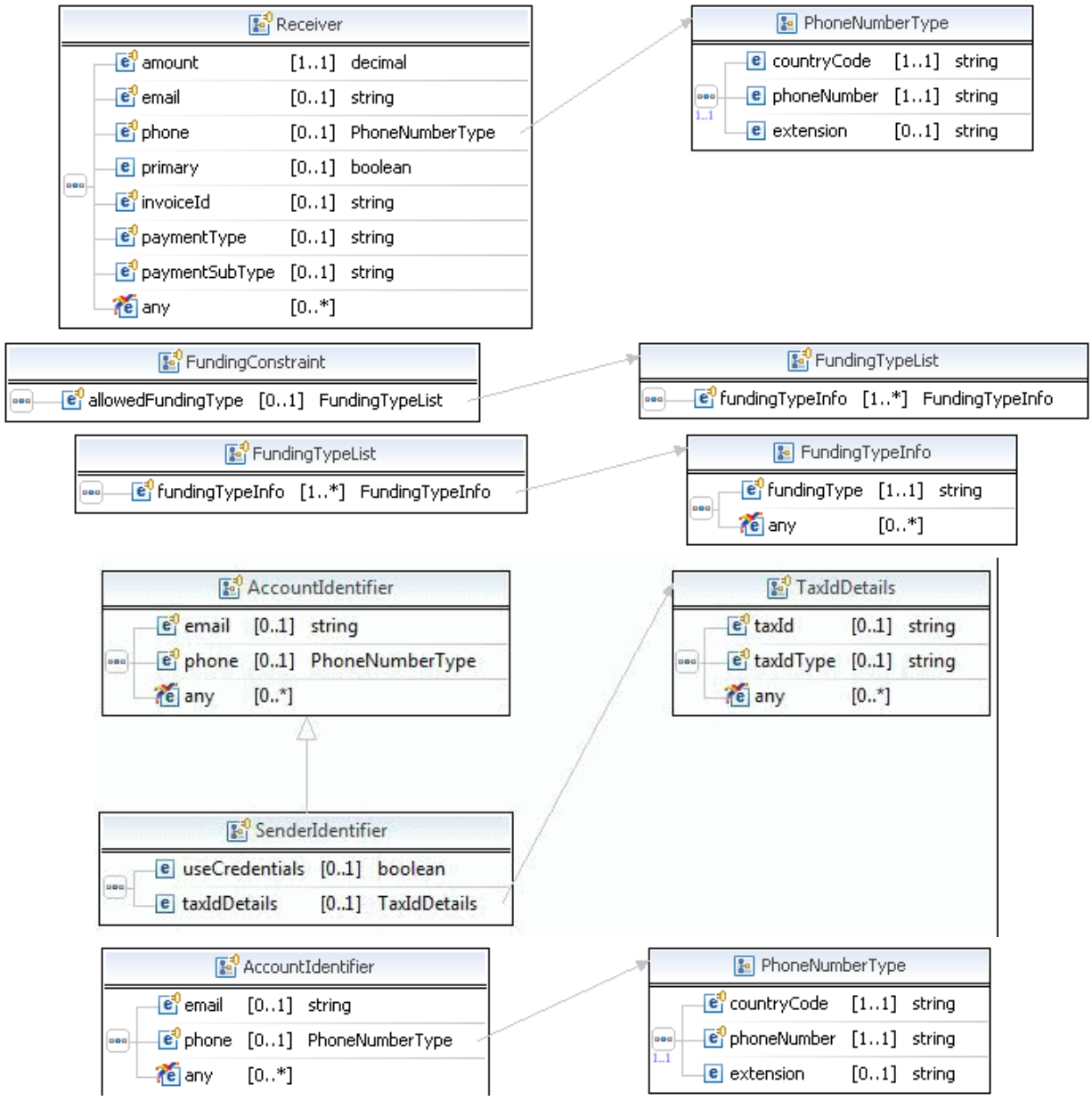
RequestEnvelope Fields

Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.
`errorLanguage`	`xs:string` *(Required)* RFC 3066 language in which error messages are returned; by default it is `en_US`, which is the only language currently supported.

PaymentDetailsResponse Message

The `PaymentDetailsResponse` message contains information about the payment.


PaymentDetailsResponse
responseEnvelope [1..1] ResponseEnvelope
cancelUrl [1..1] string
currencyCode [1..1] string
ipnNotificationUrl [0..1] string
memo [0..1] string
paymentInfoList [1..1] PaymentInfoList
returnUrl [1..1] string
senderEmail [0..1] string
status [1..1] string
trackingId [0..1] string
payKey [1..1] string
actionType [1..1] string
feesPayer [1..1] string
reverseAllParallelPaymentsOnError [0..1] boolean
preapprovalKey [0..1] string
fundingConstraint [0..1] FundingConstraint
sender [0..1] SenderIdentifier
any [0..*]
ResponseEnvelope
timestamp [1..1] dateTime
ack [1..1] AckCode
correlationId [1..1] string
build [1..1] string
any [0..*]
PaymentInfoList
paymentInfo [1..*] PaymentInfo
FundingConstraint
allowedFundingType [0..1] FundingTypeList
SenderIdentifier
useCredentials [0..1] boolean
PaymentInfoList
paymentInfo [1..*] PaymentInfo
PaymentInfo
transactionId [0..1] string
transactionStatus [0..1] string
receiver [1..1] Receiver
refundedAmount [0..1] decimal
pendingRefund [0..1] boolean
senderTransactionId [0..1] string
senderTransactionStatus [0..1] string
pendingReason [0..1] string
any [0..*]
PaymentInfo
transactionId [0..1] string
transactionStatus [0..1] string
receiver [1..1] Receiver
refundedAmount [0..1] decimal
pendingRefund [0..1] boolean
senderTransactionId [0..1] string
senderTransactionStatus [0..1] string
pendingReason [0..1] string
any [0..*]
Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]


Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]
PhoneNumberType
countryCode [1..1] string
phoneNumber [1..1] string
extension [0..1] string
FundingConstraint
allowedFundingType [0..1] FundingTypeList
FundingTypeList
fundingTypeInfo [1..*] FundingTypeInfo
FundingTypeList
fundingTypeInfo [1..*] FundingTypeInfo
FundingTypeInfo
fundingType [1..1] string
any [0..*]
AccountIdentifier
email [0..1] string
phone [0..1] PhoneNumberType
any [0..*]
TaxIdDetails
taxId [0..1] string
taxIdType [0..1] string
any [0..*]
SenderIdentifier
useCredentials [0..1] boolean
taxIdDetails [0..1] TaxIdDetails
AccountIdentifier
email [0..1] string
phone [0..1] PhoneNumberType
any [0..*]
PhoneNumberType
countryCode [1..1] string
phoneNumber [1..1] string
extension [0..1] string

PaymentDetailsResponse Fields

Field	Description
`actionType`	`xs:string` Whether the `Pay` request is set up to create a payment request with the `SetPaymentOptions` request, and then fulfill the payment with the `ExecutePayment` request. Possible values are: • `PAY` – Use this option if you are not using the `Pay` request in combination with `ExecutePayment`. • `CREATE` – Use this option to set up the payment instructions with `SetPaymentOptions` and then execute the payment at a later time with the `ExecutePayment`. • `PAY_PRIMARY` – For chained payments only, specify this value to delay payments to the secondary receivers; only the payment to the primary receiver is processed.
`cancelUrl`	`xs:string` The URL to which the sender's browser is redirected if the sender cancels the approval for a payment on paypal.com. Use the pay key to identify the payment as follows: `payKey=${payKey}`.

Field	Description
currencyCode	xs:string The currency code. Possible values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD
feesPayer	xs:string The payer of PayPal fees. Possible values are: • SENDER – Sender pays all fees (for personal, implicit simple/parallel payments; do not use for chained or unilateral payments) • PRIMARYRECEIVER – Primary receiver pays all fees (chained payments only) • EACHRECEIVER – Each receiver pays their own fee (default, personal and unilateral payments) • SECONDARYONLY – Secondary receivers pay all fees (use only for chained payments with one secondary receiver)

Field	Description
fundingtypeList	ap:FundingTypeList Specifies a list of allowed funding selections for the payment. This is a list of funding selections that can be combined in any order to allow payments to use the indicated funding type. If this Parameter is omitted, the payment can be funded by any funding type that is supported for Adaptive Payments.
ipnNotificationUrl	xs:string The URL to which all IPN messages for this payment are sent.
memo	xs:string A note associated with the payment.
payKey	xs:string The pay key that identifies this payment. This is a token that is assigned by the Pay API after a PayRequest message is received and can be used in other Adaptive Payments Methods, as well as the cancelURL and returnURL to identify this payment. The pay key is **valid for 3 hours**.
paymentInfoList	ap:PaymentInfoList Represents the payment attempt made to a receiver of a PayRequest. If the execution of the payment has not yet completed, there are no transaction details returned.
preapprovalKey	xs:string The preapproval key that identifies the preapproval for this payment. This key can be used in other Adaptive Payment APIs to identify the preapproval.
responseEnvelope	common:ResponseEnvelope Common response information, including a timestamp and the response acknowledgement status.
returnUrl	xs:string The URL to which the sender's browser is redirected after approving a payment on paypal.com. Use the pay key to identify the payment as follows: payKey=${payKey}.
reverseAllParallelPaymentsOnError	xs:string Whether to reverse parallel payments. Possible values are: • true – Each parallel payment is reversed if an error occurs • false – Each parallel payment is not reversed if an error occurs
sender	ap:SenderIdentifer Sender information.
senderEmail	xs:string The sender's email address.

Field	Description
status	xs:string The status of the payment. Possible values are: • CREATED – The payment request was received; funds will be transferred once the payment is approved • COMPLETED – The payment was successful • INCOMPLETE – Some transfers succeeded and some failed for a parallel payment or, for a delayed chained payment, secondary receivers have not been paid • ERROR – The payment failed and all attempted transfers failed or all completed transfers were successfully reversed • REVERSALERROR – One or more transfers failed when attempting to reverse a payment • PROCESSING – The payment is in progress • PENDING – The payment is awaiting processing
trackingId	xs:string The tracking ID that was specified for this payment in the PaymentDetailsRequest message.

FundingTypeList Fields

Field	Description
fundingTypeInfo	ap:FundingTypeInfo Specifies a list of allowed funding selections for the payment. This is a list of funding selections that can be combined in any order to allow payments to use the indicated funding type. If this field is omitted, the payment can be funded by any funding type that is supported for Adaptive Payments.

FundingTypeInfo Fields

Field	Description
fundingType	Specifies a list of allowed funding selections for the payment. This is a list of funding selections that can be combined in any order to allow payments to use the indicated funding type. If this field is omitted, the payment can be funded by any funding type that is supported for Adaptive Payments. Possible values are: • ECHECK – Electronic check • BALANCE – PayPal account balance • CREDITCARD – Credit card

PaymentInfoList Fields

Fields	Description
paymentInfo	ap:PaymentInfo Represents the payment attempt made to a receiver of a PayRequest. If the execution of the payment has not yet completed, there are no transaction details returned.

PaymentInfo Fields

Field	Description
pendingReason	xs:string The reason why a payment is pending. the payment has a pending refund. Possible values are: • ADDRESS_CONFIRMATION – The payment is pending because the sender does not have a confirmed address on file and receiver's Payment Receiving Preferences is set for manually accepting payments or denying each of these payments. • ECHECK – The payment is pending because it was made by an eCheck that has not yet cleared. • INTERNATIONAL – The payment is pending because the receiver holds a non-U.S. account and does not have a withdrawal mechanism. The receiver must manually accept or deny this payment from the Account Overview. • MULTI_CURRENCY – The receiver does not have a balance in the currency sent, and does not have the Payment Receiving Preferences set to automatically convert and accept this payment. Receiver must manually accept or deny this payment from the Account Overview. • UNILATERAL – The payment is pending because it was made to an email address that is not yet registered or confirmed. • UPGRADE – The payment is pending because it was made via credit card and the receiver must upgrade the account to a Business account or Premier status to receive the funds. It can also mean that receiver has reached the monthly limit for transactions on the account. • VERIFY – The payment is pending because the receiver is not yet verified. • RISK – The payment is pending while it is being reviewed by PayPal for risk. • OTHER – The payment is pending for a reason other than those listed above. For more information, contact PayPal Customer Service.
pendingRefund	xs:boolean Whether the payment has a pending refund. Possible values are: • true – The payment has a pending refund. • false – The payment does not have a refund pending.

Field	Description
receiver	ap:Receiver The receiver of the payment.
refundedAmount	xs:decimal The total amount refunded for this payment since the payment occurred.
senderTransactionId	xs:string The transaction identification number of the sender of the payment.
senderTransactionStatus	xs:string The transaction status of the payment. Possible values are: • COMPLETED – The sender's transaction has completed • PENDING – The transaction is awaiting further processing • CREATED – The payment request was received; funds will be transferred once approval is received • PARTIALLY_REFUNDED– Transaction was partially refunded • DENIED – The transaction was rejected by the receiver • PROCESSING – The transaction is in progress • REVERSED – The payment was returned to the sender • REFUNDED – The payment was refunded • FAILED – The payment failed
transactionId	xs:string PayPal transaction ID for this payment to the specified receiver.
transactionStatus	xs:string PayPal transaction status. Possible values are: • COMPLETED – The sender's transaction has completed • PENDING – The transaction is awaiting further processing • CREATED – The payment request was received; funds will be transferred once approval is received • PARTIALLY_REFUNDED– Transaction was partially refunded • DENIED – The transaction was rejected by the receiver • PROCESSING – The transaction is in progress • REVERSED – The payment was returned to the sender • REFUNDED – The payment was refunded • FAILED – The payment failed

Receiver Fields

Field	Description
amount	xs:decimal Amount to be paid to the receiver.

Field	Description
email	xs:string Receiver's email address. Maximum length: 127 characters
invoiceId	xs:string The invoice number for the payment. This data in this field shows on the Transaction Details report. Maximum length: 127 characters
paymentType	xs:string The transaction type for the payment. Possible values are: • GOODS – This is a payment for non-digital goods • SERVICE – This is a payment for services (default) • PERSONAL – This is a person-to-person payment • CASHADVANCE – This is a person-to-person payment for a cash advance • DIGITALGOODS – This is a payment for digital goods • BANK_MANAGED_WITHDRAWAL – This is a person-to-person payment for bank withdrawals, available only with special permission.
paymentSubType	xs:string The transaction subtype for the payment.
phone	common:PhoneNumberType The receiver's phone number.
primary	xs:boolean Whether this receiver is the primary receiver. If this field is set to true, this is a chained payment. If this field shows false, this is a simple or parallel payment. Possible values are: • true – Primary receiver (chained payment) • false – Secondary receiver (simple/parallel payment)

SenderIdentifier Fields

Field	Description
useCredentials	xs:boolean If true, credentials identify the sender; default is false.
taxIdDetails	ap:TaxIdDetails Tax ID details (For Brazil only)

TaxIdDetails Fields

Field	Description
taxId	xs:string Tax ID of the Merchant (For Brazil only)
taxIdType	xs:string Type of the Tax ID. Either CPF or CNPJ (For Brazil only)

AccountIdentifier Fields

Field	Description
email	xs:string Sender's email address. Maximum length: 127 characters
phone	common:PhoneNumberType Sender's phone number.

PhoneNumberType Response Fields

Field	Description
countryCode	xs:string Telephone country code.
phoneNumber	xs:string Telephone number.
extension	xs:string Telephone extension.

ResponseEnvelope Fields

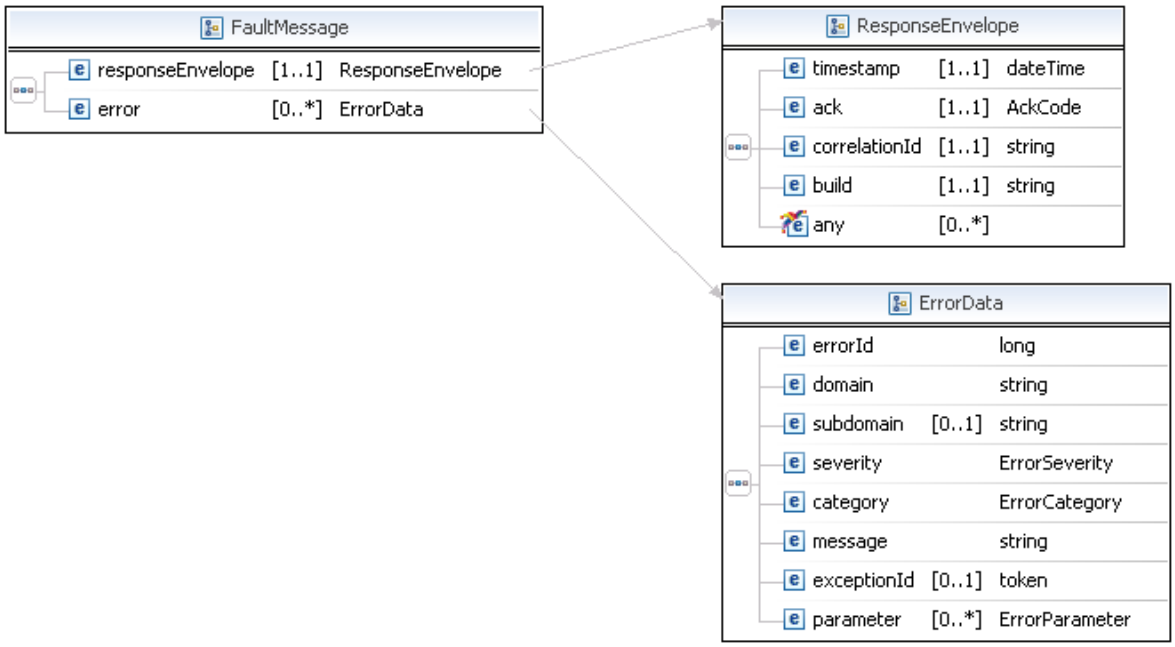
Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.

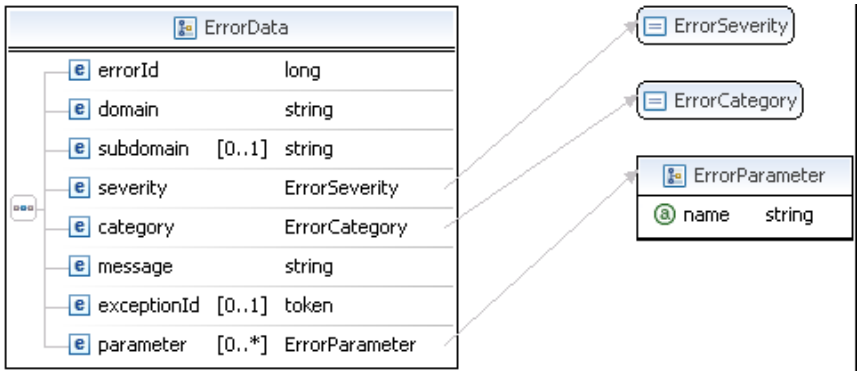
Field	Description
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The `PPFaultMessage` returns `ErrorData` and the `ResponseEnvelope` information to your application if an error occurs.





FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.

Field	Description
severity	common:ErrorSeverity The severity of the error encountered. Possible values are: • Error – Processing of the request was interrupted • Warning – Processing of the request was completed
subdomain	This field is not used.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PaymentDetails Examples Using NVP and CURL

These examples use NVP for the data binding and CURL to deliver the HTTP request to the PayPal sandbox endpoint. Line breaks are provided for ease of reading; each CURL command

is a single line and each request and response is a string without line breaks or extra whitespace.

Using the payment key to obtain payment details

In this example, the caller of the `Pay` API operation specifies the payment key to obtain information about the entire payment:

NOTE: The sample code below uses the `insecure` setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/PaymentDetails  -d
"payKey=AP-3TY011106S4428730
&requestEnvelope.errorLanguage=en_US"
```

Response:

```
responseEnvelope.timestamp=2009-08-14T09%3A00%3A37.748-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=7967b2d03745a
&responseEnvelope.build=DEV
&cancelUrl=your_cancel_url
&currencyCode=USD
&logDefaultShippingAddress=false
&memo=Simple payment example.
&paymentInfoList.paymentInfo(0).receiver.amount=100.00
&paymentInfoList.paymentInfo(0).receiver.email=receiver@domain
&paymentInfoList.paymentInfo(0).receiver.primary=false
&returnUrl=your_return_url
&senderEmail=sender@domain
&status=CREATED
&payKey=AP-3TY011106S4428730
&actionType=PAY
&feesPayer=EACHRECEIVER
```

Payment Details Errors

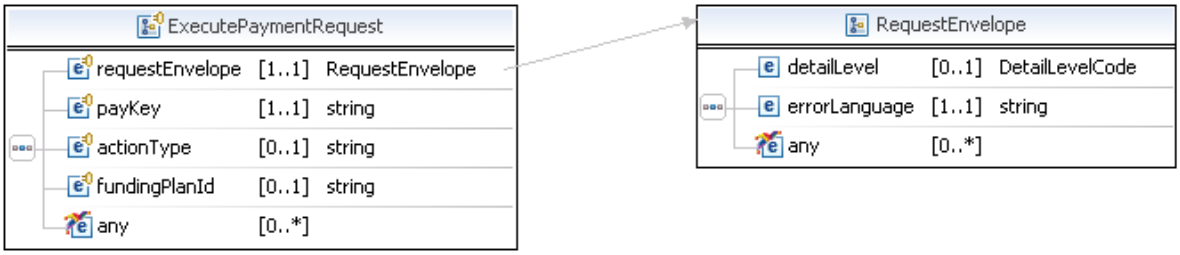
Code	Message	Additional Information
500000	There is a system error	
520002	Internal error	
520003	User name/password is incorrect	
520003	Authentication failed. API credentials are incorrect.	Since 1.6.0
520006	This call is not defined in the database	
540031	You do not have permission to get these payment details	Since 1.6.0
579049	The payment request you are looking for was not found in our system	
579051	The tracking ID is invalid	
580001	Invalid request	
580022	Invalid request parameter: payKey with value <value>	
580029	Invalid request	
580030	This transaction cannot be processed at this time. Please try again later.	Since 1.3.0
589052	The transaction ID is invalid	

5 ExecutePayment API Operation

The ExecutePayment API operation lets you execute a payment set up with the Pay API operation with the actionType CREATE. To pay receivers identified in the Pay call, set the pay key from the PayResponse message in the ExecutePaymentRequest message.

ExecutePaymentRequest Message

The ExecutePaymentRequest message lets you execute a payment requested using the Pay API operation.



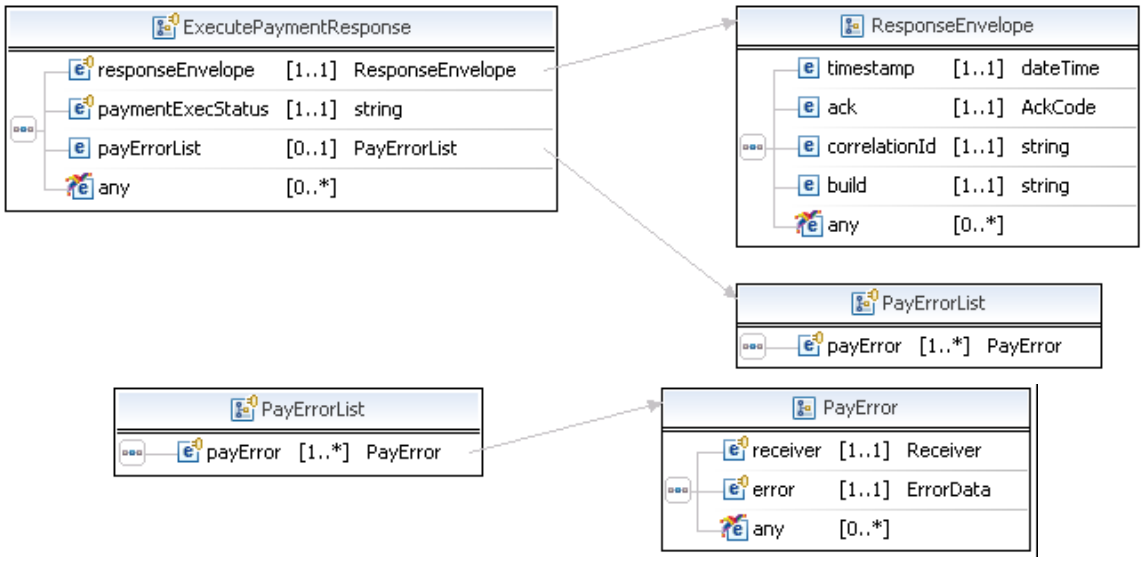
ExecutePaymentRequest Fields

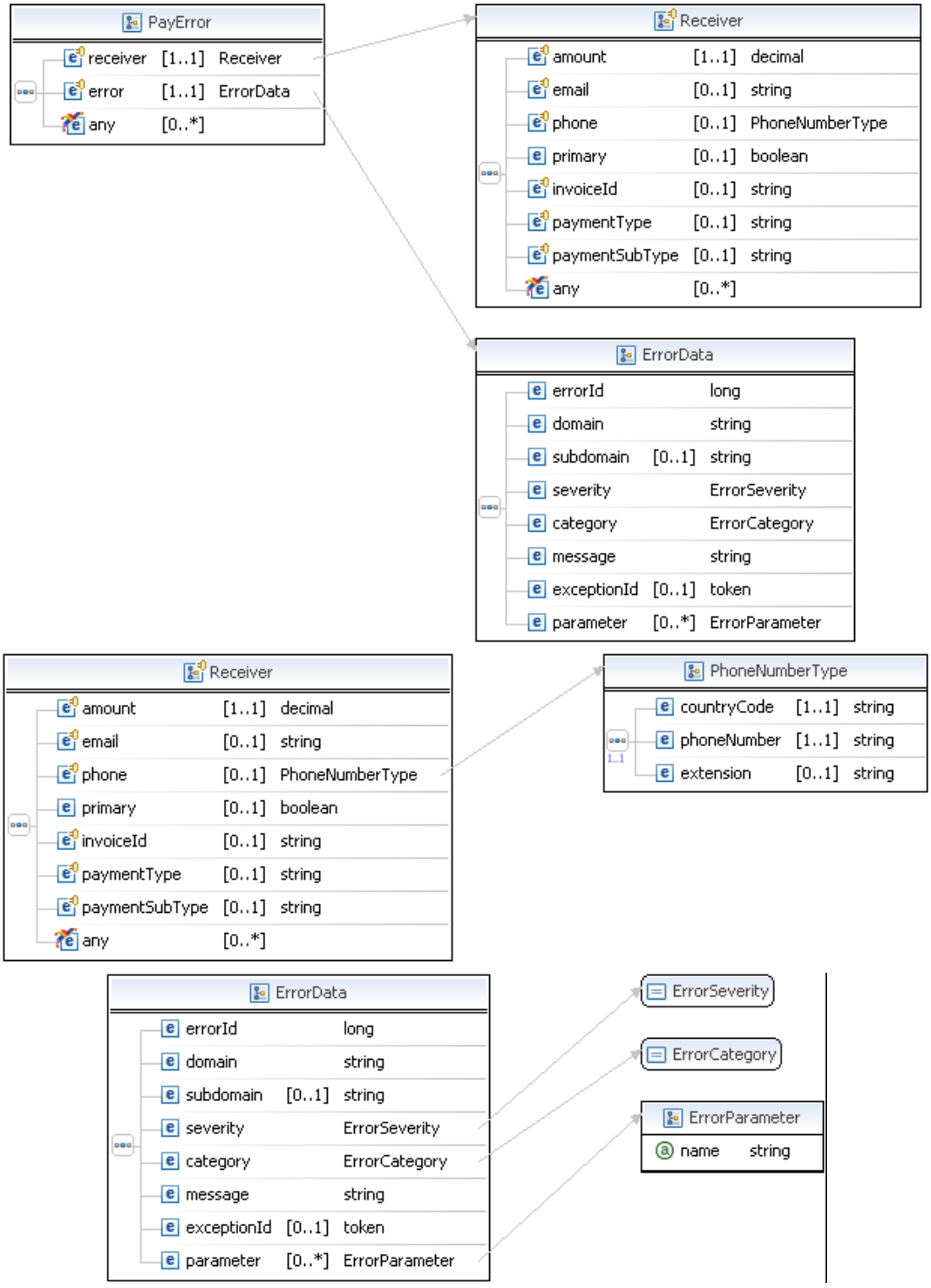
Field	Description
payKey	xs:string *(Optional)* The pay key that identifies the payment to be executed. This is the pay key returned in the PayResponse message.
fundingPlanId	xs:string *(Optional)* The ID of the funding plan from which to make this payment.
requestEnvelope	common:requestEnvelope Information common to each API operation, such as the language in which an error message is returned.

RequestEnvelope Fields

Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.
`errorLanguage`	`xs:string` *(Required)* RFC 3066 language in which error messages are returned; by default it is `en_US`, which is the only language currently supported.

ExecutePaymentResponse Message




PayError
receiver [1..1] Receiver
error [1..1] ErrorData
any [0..*]
Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]
ErrorData
errorId long
domain string
subdomain [0..1] string
severity ErrorSeverity
category ErrorCategory
message string
exceptionId [0..1] token
parameter [0..*] ErrorParameter
Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]
PhoneNumberType
countryCode [1..1] string
phoneNumber [1..1] string
extension [0..1] string
1..1
ErrorData
errorId long
domain string
subdomain [0..1] string
severity ErrorSeverity
category ErrorCategory
message string
exceptionId [0..1] token
parameter [0..*] ErrorParameter
ErrorSeverity
ErrorCategory
ErrorParameter
name string

ExecutePaymentResponse Fields

Field	Description
payErrorList	ap:PayErrorList Information about why a payment failed.
paymentExecStatus	xs:string The status of the payment. Possible values are: • CREATED – The payment request was received; funds will be transferred once the payment is approved • COMPLETED – The payment was successful • INCOMPLETE – Some transfers succeeded and some failed for a parallel payment • ERROR – The payment failed and all attempted transfers failed or all completed transfers were successfully reversed • REVERSALERROR – One or more transfers failed when attempting to reverse a payment **IMPORTANT:** You must test the value of paymentExecStatus for an error even if responseEnvelope.ack is Success. If the PaymentExecStatus is ERROR, the Pay Key can no longer be used.
responseEnvelope	common:ResponseEnvelope Common response information, including a timestamp and the response acknowledgement status.

PayErrorList Fields

Field	Description
payError	PayError indicates the error, if any, that resulted on an attempted payment to a receiver.

PayError Fields

Field	Description
error	Detailed error information.
receiver	ap:Receiver Receiver is the party where funds are transferred to. A primary receiver receives a payment directly from the sender in a chained split payment. A primary receiver should not be specified when making a single or parallel split payment.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.
`severity`	`common:ErrorSeverity` The severity of the error encountered. Possible values are: • `Error` – Processing of the request was interrupted • `Warning` – Processing of the request was completed
`subdomain`	This field is not used.

Receiver Fields

Field	Description
`amount`	`xs:decimal` Amount to be paid to the receiver.
`email`	`xs:string` Receiver's email address. Maximum length: 127 characters
`invoiceId`	`xs:string` The invoice number for the payment. This data in this field shows on the Transaction Details report. Maximum length: 127 characters

Field	Description
paymentType	xs:string The transaction type for the payment. Possible values are: • GOODS – This is a payment for non-digital goods • SERVICE – This is a payment for services (default) • PERSONAL – This is a person-to-person payment • CASHADVANCE – This is a person-to-person payment for a cash advance • DIGITALGOODS – This is a payment for digital goods • BANK_MANAGED_WITHDRAWAL – This is a person-to-person payment for bank withdrawals, available only with special permission.
paymentSubType	xs:string The transaction subtype for the payment.
phone	common:PhoneNumberType The receiver's phone number.
primary	xs:boolean Whether this receiver is the primary receiver. If this field is set to true, this is a chained payment. If this field shows false, this is a simple or parallel payment. Possible values are: • true – Primary receiver (chained payment) • false – Secondary receiver (simple/parallel payment)

ResponseEnvelope Fields

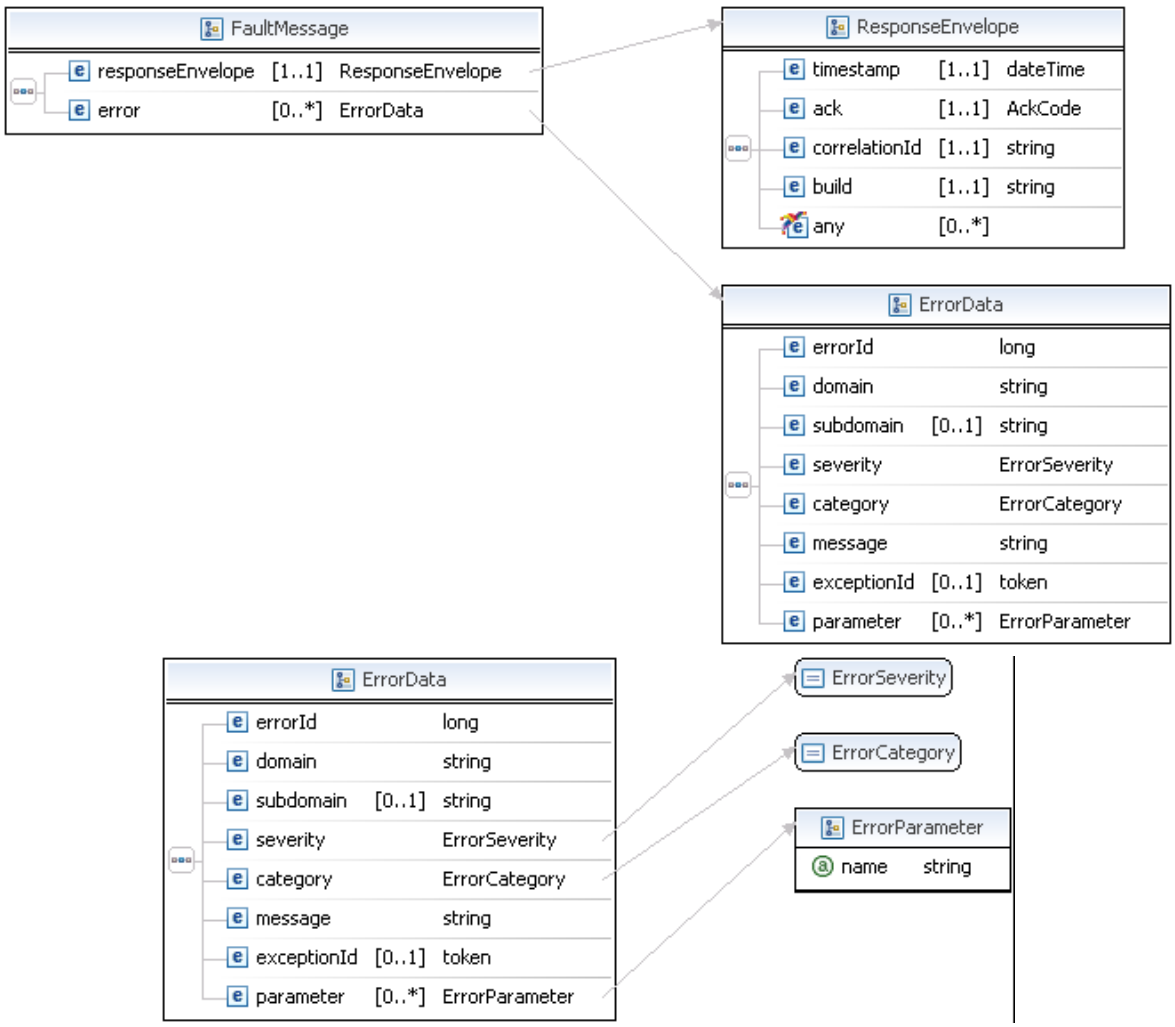
Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

Field	Description

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The `PPFaultMessage` returns `ErrorData` and the `ResponseEnvelope` information to your application if an error occurs.



FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.
`severity`	`common:ErrorSeverity` The severity of the error encountered. Possible values are: • `Error` – Processing of the request was interrupted • `Warning` – Processing of the request was completed
`subdomain`	This field is not used.

ResponseEnvelope Fields

Field	Description
`ack`	`common:AckCode` Acknowledgement code. It is one of the following values: • `Success` – The operation completed successfully. • `Failure` – The operation failed. • `SuccessWithWarning` – The operation completed successfully; however, there is a warning message. • `FailureWithWarning` – The operation failed with a warning message.
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

ExecutePayment Errors

Code	Message	Additional Information
500000	There is a system error	
520002	Internal error	
520003	User name/password is incorrect	
520003	Authentication failed. API credentials are incorrect.	Since 1.6.0
520005	Merchant account is locked	
520006	This call is not defined in the database	
529038	There was an error while making this payment	
539012	The preapproval key has not been authorized yet	

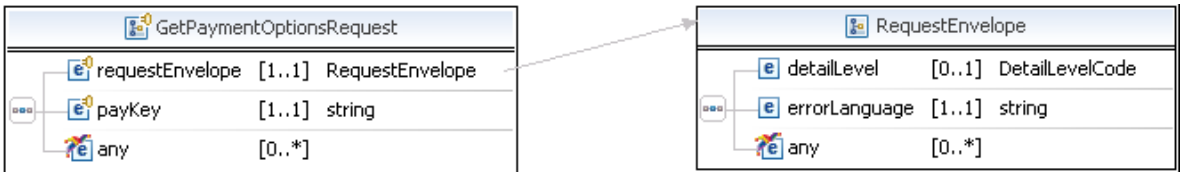
Code	Message	Additional Information
539041	The email account is based in a country that is not enabled to receive payments	
539043	The email account is based in a country that is not enabled to send payments	
540031	You do not have permission to execute this payment	
540031	You don't have permission to cancel this preapproval	
540031	You do not have permission to execute this payment Operation is not permitted because the credentials do not match those of the initial operation	Since 1.6.0
550001	User is not allowed to perform this action	
550001	This payment request must be authorized by the sender	
550001	You do not have permission to execute this payment implicitly	Since 1.6.0
559044	Account setting on the receiver prohibited the payment	
560027	The argument value is unsupported	
569000	Split payments are not supported at this time	
569013	The preapproval key has been canceled	
569016	Preapproval PIN functionality is not enabled	
569017	The preapproval key has been suspended	
569018	Preapproved payments have been disabled	
569019	The preapproval has been suspended due to too many PIN failures	
569042	The email account is not confirmed by PayPal	
579007	The maximum number of receivers is <number>	
579010	If a preapproval key is specified, the sender’s email address must be, too	
579014	The preapproval key specifies a different sender than the payment request	
579017	The amount for the primary receiver must be greater than or equal to the total of other chained receiver amounts	
579024	The preapproval key cannot be used before the start date or after the end date	
579025	The preapproval key cannot be used on this weekday	
579026	The preapproval key cannot be used on this day of the month	

Code	Message	Additional Information
579027	The preapproval key specifies a different currency than the payment request	
579028	The payment amount exceeds the maximum amount per payment	
579030	The number of payments made this period exceeds the maximum number of payments per period	
579031	The total amount of all payments exceeds the maximum total amount for all payments	
579033	The sender and each receiver must have different accounts	
579040	The receivers cannot belong to the same PayPal account	
579042	The tracking ID already exists and cannot be duplicated	
579045	The email account exceeds the receiving limit	
579047	The email account exceeds the purse limit	
579048	The email account exceeds the sending limit	
580001	Invalid request	
580022	Your request was invalid. Check the parameter of the error to see which is invalid	
580022	• Invalid request parameter: payKey with value <value> • payKey <value> has already been used to make a payment	
580022	The funding plan data is not present	Since 1.6.0. Tried to pass in a non-zero funding plan and there are no funding plans present
580023	Invalid request	
580027	The argument is unsupported	
580028	The URL <value> is malformed	
580029	Invalid request	
580029	A pay key, transaction ID, or tracking ID is missing	Since 1.6.0
580030	This transaction cannot be processed at this time. Please try again later.	Since 1.3.0
589009	This payment cannot be processed because no payment source is available	
589023	If a fractional amount is rounded due to currency conversion, funds could be lost	
589039	The email address is invalid. It may not be registered in PayPal's system yet	

6 GetPaymentOptions API Operation

You use the `GetPaymentOptions` API operation to retrieve the payment options passed with the `SetPaymentOptionsRequest`.

GetPaymentOptionsRequest Message



GetPaymentOptionsRequest Fields

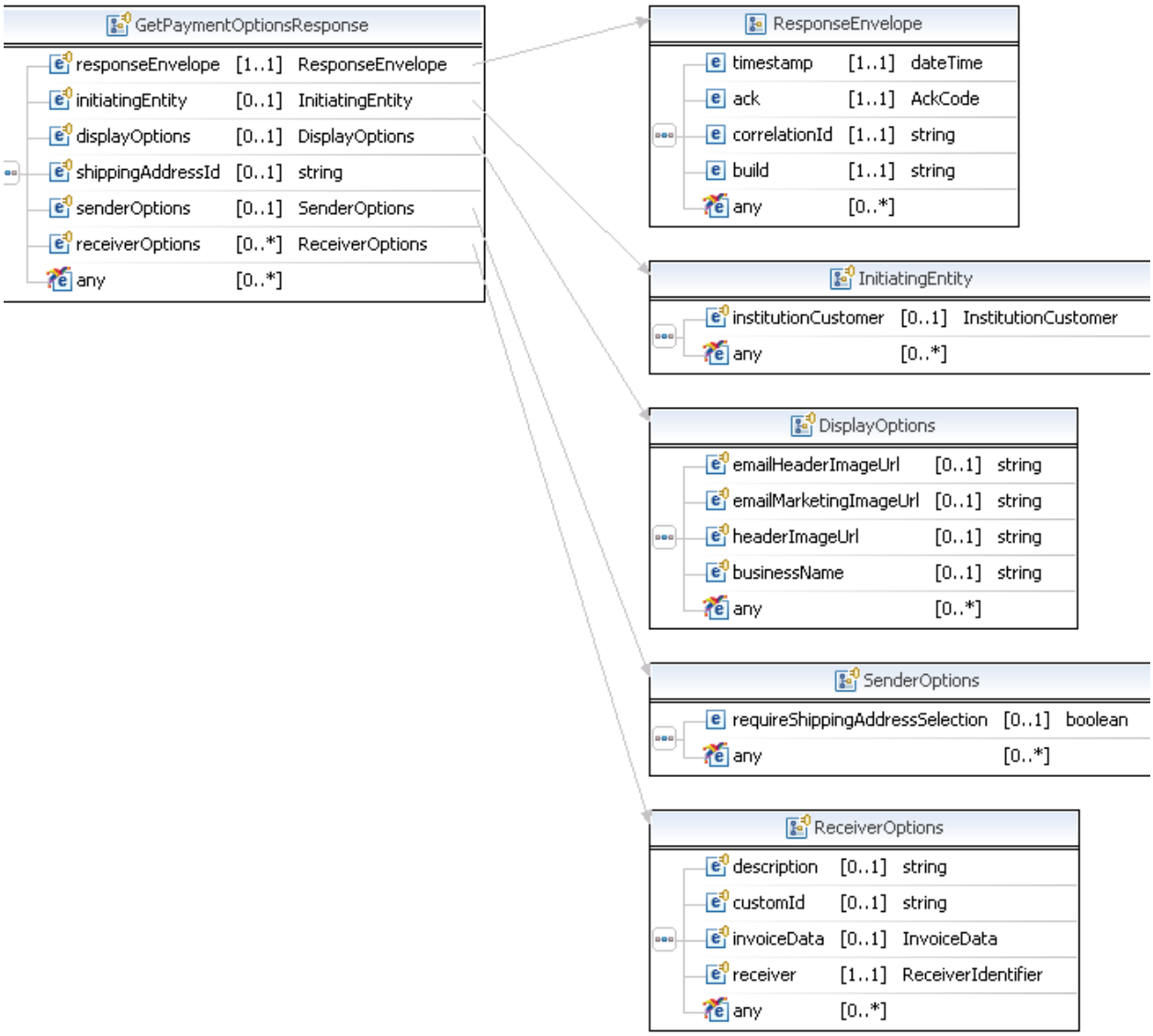
Field	Description
`payKey`	`xs:string` *(Required)* The pay key that identifies the payment for which you want to get payment options. This is the pay key you used to set the payment options.
`requestEnvelope`	`common:requestEnvelope` *(Required)* Information common to each API operation, such as the language in which an error message is returned.

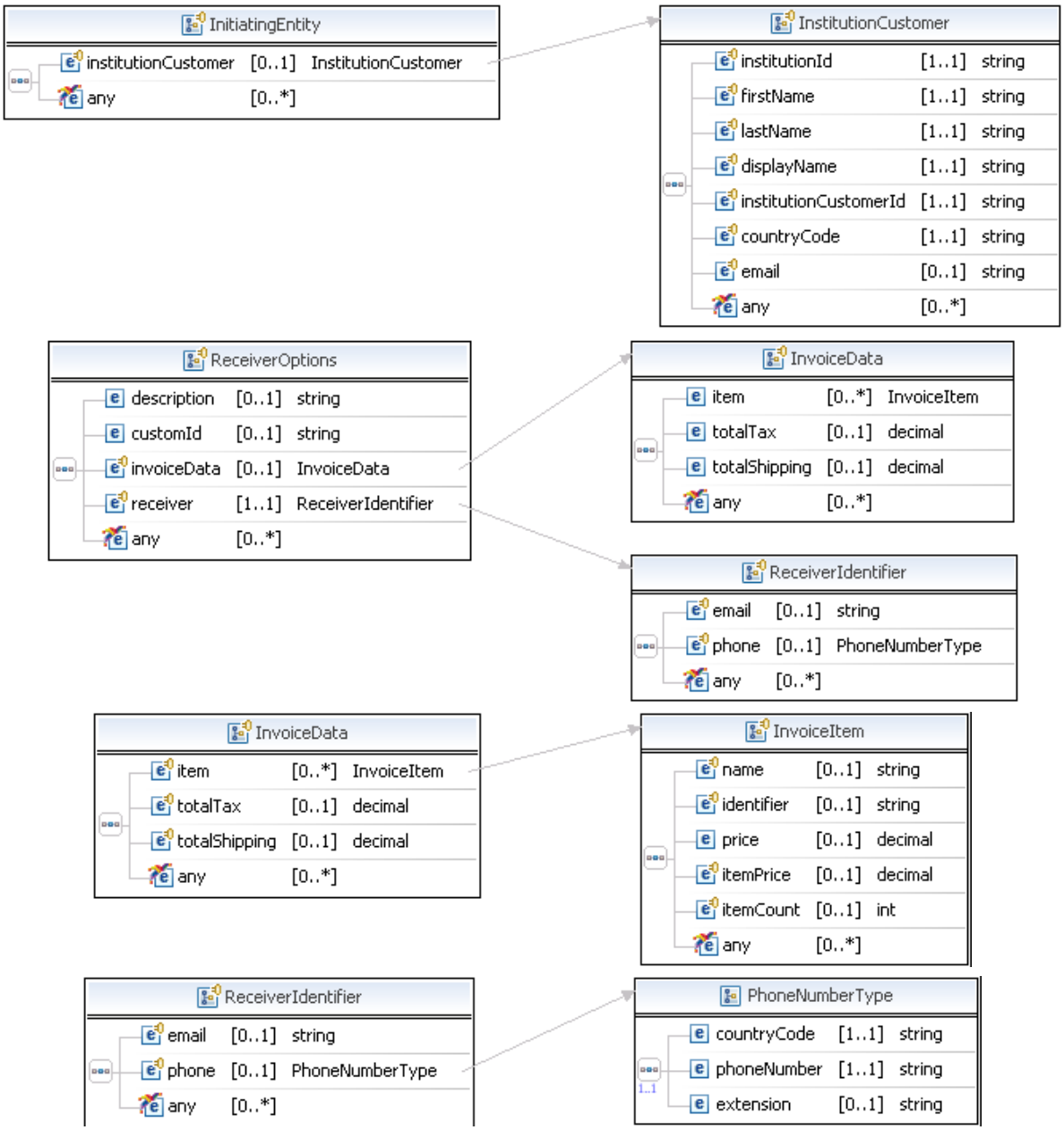
RequestEnvelope Fields

Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.

Field	Description
errorLanguage	xs:string *(Required)* RFC 3066 language in which error messages are returned; by default it is en_US, which is the only language currently supported.

GetPaymentOptionsResponse Message





GetPaymentOptionsResponse Fields

Field	Description
payKey	xs:string The pay key that identifies the payment for which you set payment options.

Field	Description
displayOptions	ap:DisplayOptions Specifies display items in payment flows and emails.
initiatingEntitity	ap:InitiatingEntity The PayPal financial partner that is initiating the payment. Financial partners must first be set up in the Admin tool before they can use the PayPal APIs.
shippingAddressId	xs:string Sender's shipping address ID.
senderOptions	ap:SenderOptions Specifies information to be provided by the sender.
receiverOptions	ap:ReceiverOptions Specifies information about each receiver.
requestEnvelope	common:requestEnvelope Information common to each API operation, such as the language in which an error message is returned.

DisplayOptions Fields

DisplayOptions fields allow the API caller to provide PayPal with a link to an image.

Field	Description
emailHeaderImageUrl	xs:string The URL of the image that displays in the in the header of customer emails. The image dimensions are 43 pixels high x 240 pixels wide.
emailMarketingImageUrl	xs:string The URL of the image that displays in the in customer emails. The image dimensions are 80 pixels high x 530 pixels wide.
headerImageUrl	xs:string The URL of the image that displays in the header of a payment page. Use this to configure payment flows by passing a different image URL for different scenarios. If set, it overrides the header image URL specified in your account's Profile. The image dimensions are 90 pixels high x 750 pixels wide. **NOTE:** The headerImageUrl and businessName parameters are mutually exclusive; only one of these fields can be used. If you specify both, the image will take precedence over the business name.
businessName	xs:string The business name to display. **NOTE:** The headerImageUrl and businessName parameters are mutually exclusive; only one of these fields can be used. If you specify both, the image will take precedence over the business name.

InitiatingEntity Fields

Field	Description
`institutionCustomer`	`ap:InstitutionCustomer` *(Optional)* Details about the party that initiated this payment. This payment is made by the API caller on behalf of the initiating party. The initiating party can be an institution or a customer of the institution. The initiating party must be set up by PayPal Merchant Services.

Institution Customer Fields

Field	Description
`countryCode`	`xs:string` *(Required)* The two-character country code of the home country of the end consumer
`displayName`	`xs:string` *(Required)* The full name of the consumer as known by the institution. Maximum length: 200 characters
`email`	`xs:string` *(Optional)* The email address of the consumer as known by the institution. Maximum length: 127 characters
`firstName`	`xs:string` *(Required)* The first name of the consumer as known by the institution. Maximum length: 64 characters
`institutionCustomerId`	`xs:string` *(Required)* The unique identifier assigned to the consumer by the institution. Maximum length: 64 characters
`institutionId`	`xs:string` *(Required)* The unique identifier assigned to the institution. Maximum length: 64 characters
`lastName`	`xs:string` *(Required)* The last name of the consumer as known by the institution. Maximum length: 64 characters

SenderOptions Fields

Field	Description
`requireShippingAddressSelection`	`xs:boolean` If `true`, require the sender to select a shipping address during the embedded payment flow; default is `false`.
`referrerCode`	`xs:string` A code that identifies the partner associated with this transaction. Maximum length: 32 characters.

ReceiverOptions Fields

Field	Description
`description`	`xs:string` A description you want to associate with the payment. Maximum length: 1000 characters
`customId`	`xs:string` An external reference or identifier you want to associate with the payment. Maximum length: 1000 characters
`invoiceData`	`ap:InvoiceData` Item information for the payment, which could appear on an invoice.
`receiver`	`ap:ReceiverIdentifier` Receiver ID associated with these options.

InvoiceData Fields

Field	Description
`item`	`ap:InvoiceItem` Any number of items associated with the payment.
`totalTax`	`xs:decimal` Total tax associated with the payment.
`totalShipping`	`xs:decimal` Total shipping charge associated with the payment.

InvoiceItem Fields

Field	Description
`name`	`xs:string` Name of item.
`identifier`	`xs:string` External reference to item or item ID.
`price`	`xs:decimal` Price of item.
`itemPrice`	`xs:decimal` Price of an individual item.
`itemCount`	`xs:int` Item quantity.

ReceiverIdentifier Fields

Field	Description
`email`	`xs:string` Receiver's email address. Maximum length: 127 characters
`phone`	`common:PhoneNumberType` Receiver's phone number.

PhoneNumberType Response Fields

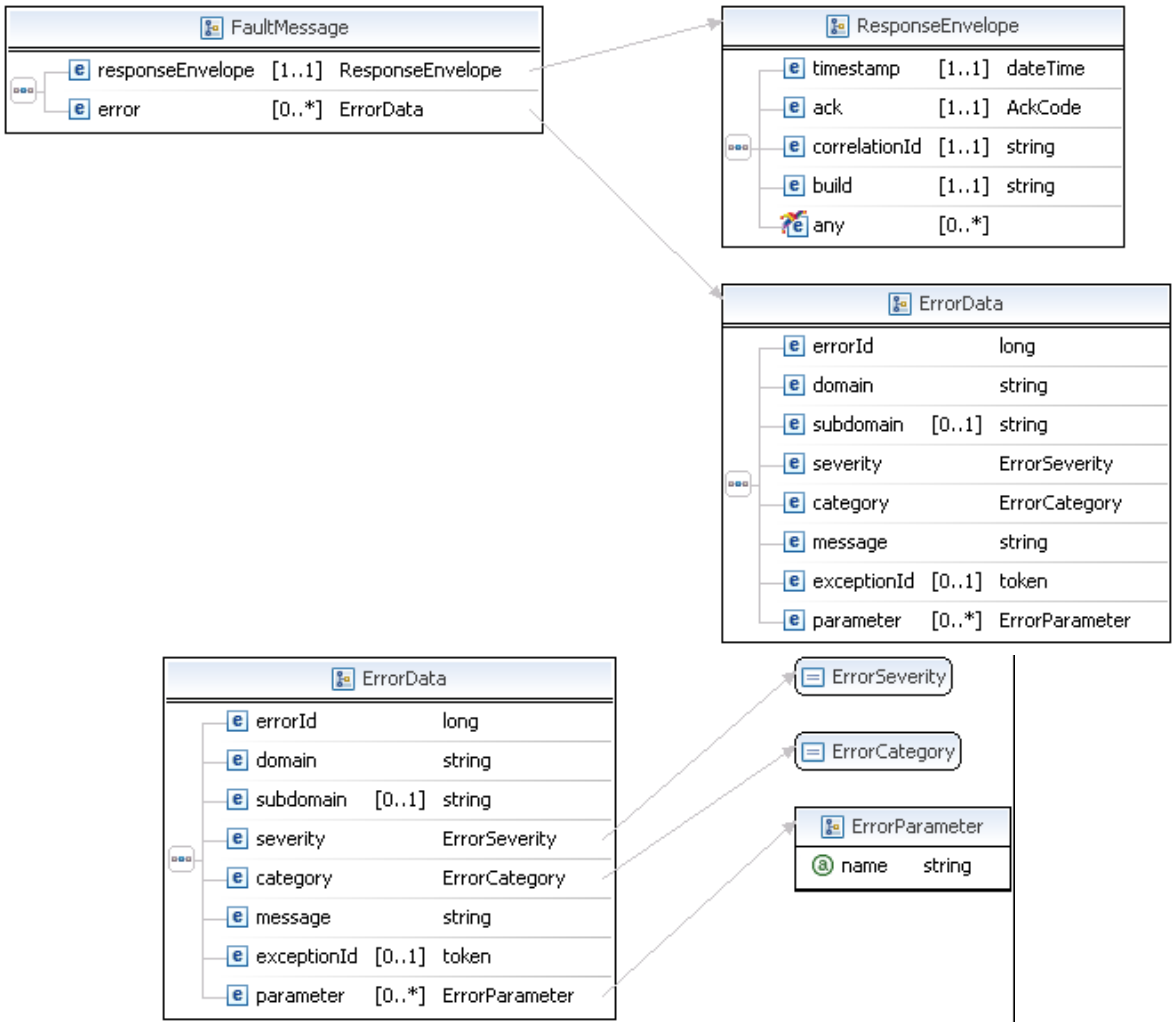
Field	Description
`countryCode`	`xs:string` Telephone country code.
`phoneNumber`	`xs:string` Telephone number.
`extension`	`xs:string` Telephone extension.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The PPFaultMessage returns ErrorData and the ResponseEnvelope information to your application if an error occurs.



FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.
`severity`	`common:ErrorSeverity` The severity of the error encountered. Possible values are: • `Error` – Processing of the request was interrupted • `Warning` – Processing of the request was completed
`subdomain`	This field is not used.

ResponseEnvelope Fields

Field	Description
`ack`	`common:AckCode` Acknowledgement code. It is one of the following values: • `Success` – The operation completed successfully. • `Failure` – The operation failed. • `SuccessWithWarning` – The operation completed successfully; however, there is a warning message. • `FailureWithWarning` – The operation failed with a warning message.
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

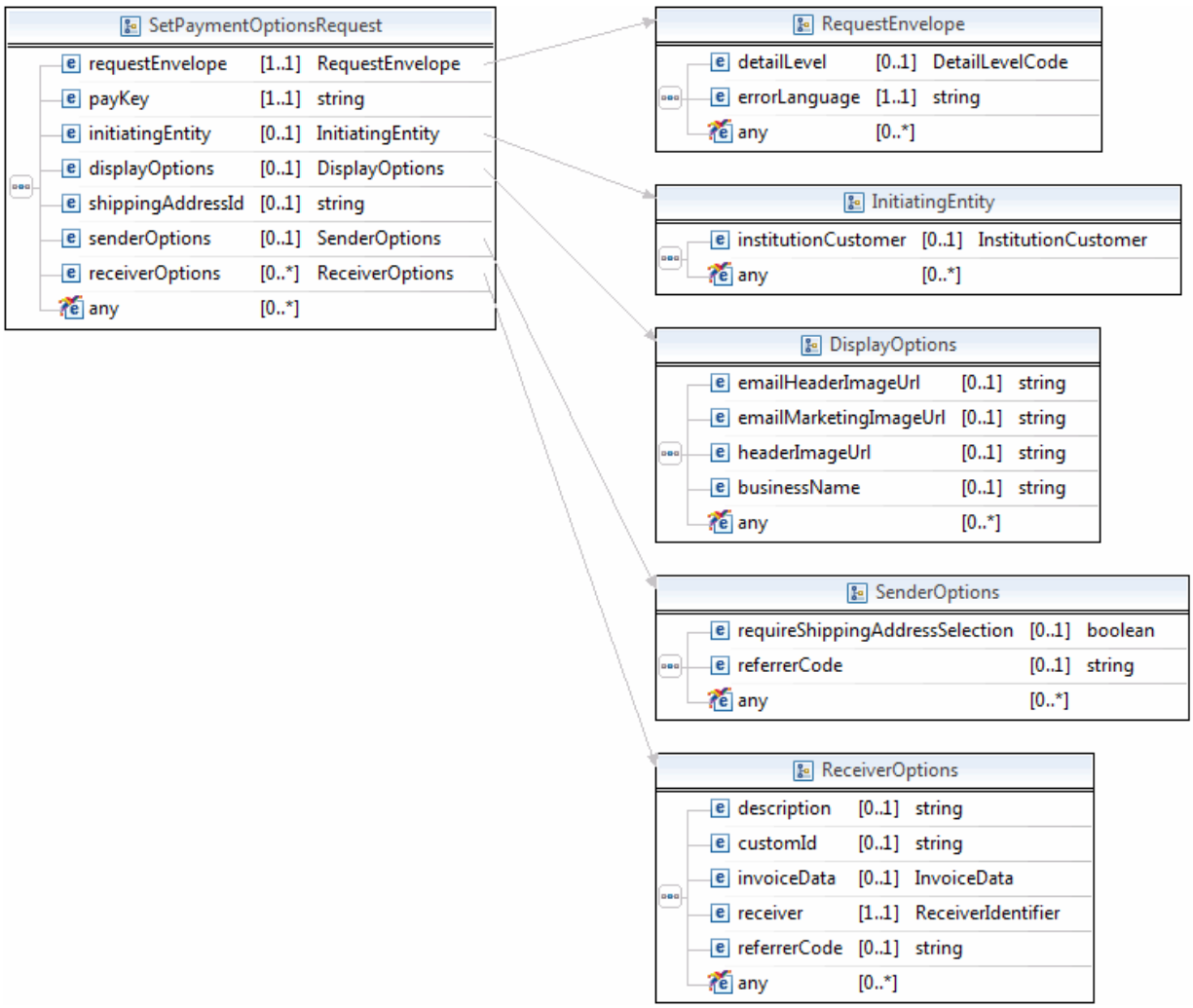
GetPaymentOptions Errors

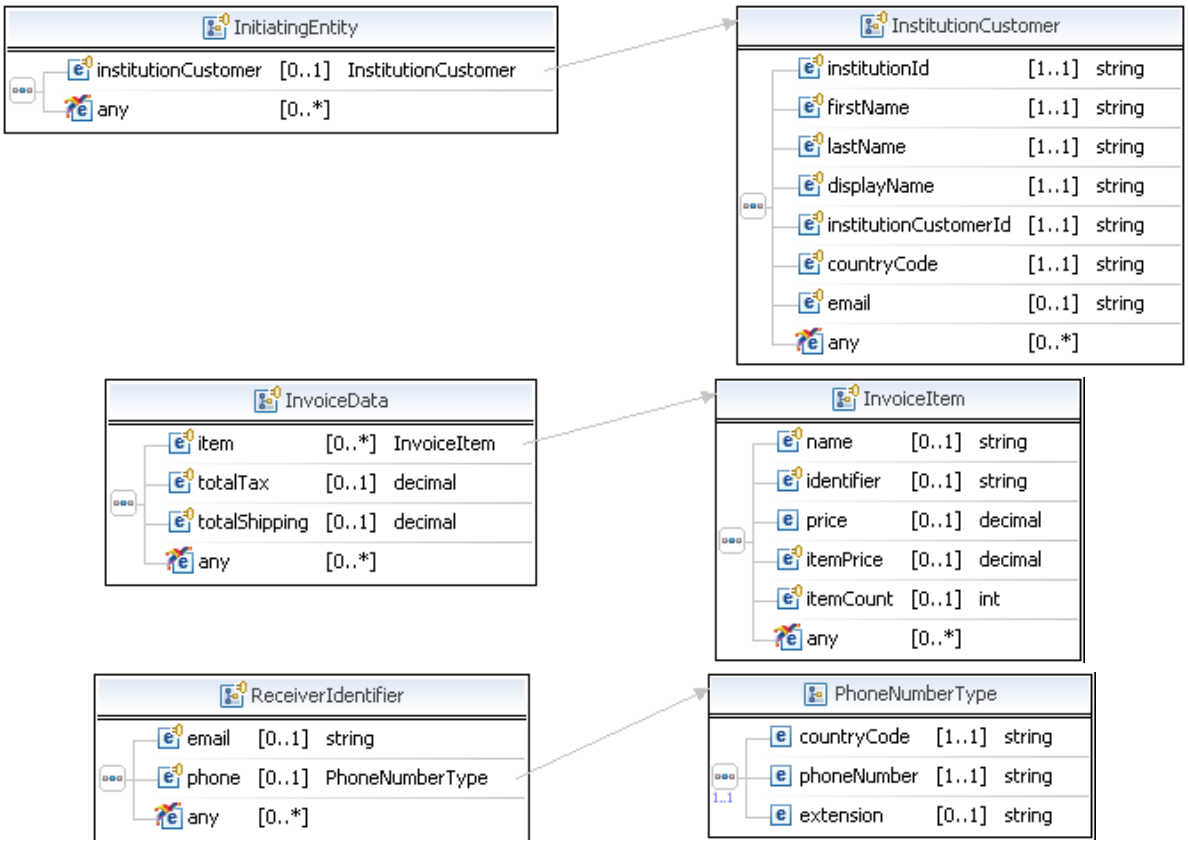
Code	Message	Additional Information
520002	Internal error	
520003	Authentication failed. API credentials are incorrect.	Since 1.6.0
540031	You do not have permission to get these payment options	
540031	You do not have permission to get these payment options Operation is not permitted because the credentials do not match those of the initial operation	Since 1.6.0
550001	User is not allowed to perform this action	
580001	Invalid request: <value>	Since 1.6.0
580022	Invalid request parameter: payKey with value <value>	

7 SetPaymentOptions API Operation

You use the `SetPaymentOptions` API operation to specify settings for a payment of the `actionType` `CREATE`. This `actionType` is specified in the `PayRequest` message.

SetPaymentsOptionsRequest Message





SetPaymentOptionsRequest Fields

Field	Description
`payKey`	`xs:string` *(Required)* The pay key that identifies the payment for which you want to set payment options. This is the pay key returned in the `PayResponse` message.
`displayOptions`	`ap:DisplayOptions` *(Optional)* Specifies display items in payment flows and emails.
`initiatingEntity`	`ap:InitiatingEntity` *(Optional)* The PayPal financial partner that is initiating the payment. Financial partners must first be set up in the Admin tool before they can use the PayPal APIs.
`shippingAddressId`	`xs:string` *(Optional)* Sender's shipping address ID.
`senderOptions`	`ap:SenderOptions` *(Optional)* Specifies information to be provided by the sender.

Field	Description
receiverOptions	ap:ReceiverOptions *(Optional)* Specifies information about each receiver.
requestEnvelope	common:requestEnvelope *(Required)* Information common to each API operation, such as the language in which an error message is returned.

DisplayOptions Fields

Field	Description
emailHeaderImageUrl	xs:string *(Optional)* The URL of the image that displays in the in the header of customer emails. The URL cannot exceed 1,024 characters. The image dimensions are 43 pixels high x 240 pixels wide.
emailMarketingImageUrl	xs:string *(Optional)* The URL of the image that displays in the in customer emails. The URL cannot exceed 1,024 characters. The image dimensions are 80 pixels high x 530 pixels wide.
headerImageUrl	xs:string *(Optional)* The URL of an image that displays in the header of a payment page. If set, it overrides the header image URL specified in your account's Profile. The URL cannot exceed 1,024 characters. The image dimensions are 90 pixels high x 750 pixels wide.
businessName	xs:string *(Optional)* The business name to display. The name cannot exceed 128 characters.

InitiatingEntity Fields

Field	Description
institutionCustomer	ap:InstitutionCustomer *(Optional)* Details about the party that initiated this payment. This payment is made by the API caller on behalf of the initiating party. The initiating party can be an institution or a customer of the institution. The initiating party must be set up by PayPal Merchant Services.

Institution Customer Fields

Field	Description
`countryCode`	`xs:string` *(Required)* The two-character country code of the home country of the end consumer
`displayName`	`xs:string` *(Required)* The full name of the consumer as known by the institution. Maximum length: 200 characters
`email`	`xs:string` *(Optional)* The email address of the consumer as known by the institution. Maximum length: 127 characters
`firstName`	`xs:string` *(Required)* The first name of the consumer as known by the institution. Maximum length: 64 characters
`institutionCustomerId`	`xs:string` *(Required)* The unique identifier assigned to the consumer by the institution. Maximum length: 64 characters
`institutionId`	`xs:string` *(Required)* The unique identifier assigned to the institution. Maximum length: 64 characters
`lastName`	`xs:string` *(Required)* The last name of the consumer as known by the institution. Maximum length: 64 characters

SenderOptions Fields

Field	Description
`requireShippingAddressSelection`	`xs:boolean` *(Optional)* If `true`, require the sender to select a shipping address during the embedded payment flow; default is `false`.
`referrerCode`	`xs:string` *(Optional)* A code that identifies the partner associated with this transaction. Maximum length: 32 characters.

ReceiverOptions Fields

Field	Description
`description`	`xs:string` *(Optional)* A description you want to associate with the payment. This overrides the value of the memo in Pay API for each receiver. If this is not specified the value in the memo will be used. Maximum length: 1000 characters
`customId`	`xs:string` *(Optional)* An external reference or identifier you want to associate with the payment. Maximum length: 1000 characters
`invoiceData`	`ap:InvoiceData` *(Optional)* Item information for the payment, which could appear on an invoice.
`receiver`	`ap:ReceiverIdentifier` *(Optional)* Receiver ID associated with these options.

InvoiceData Fields

Field	Description
`item`	`ap:InvoiceItem` *(Optional)* Any number of items associated with the payment.
`totalTax`	`xs:decimal` *(Optional)* Total tax associated with the payment.
`totalShipping`	`xs:decimal` *(Optional)* Total shipping charge associated with the payment.

InvoiceItem Fields

Field	Description
`name`	`xs:string` *(Optional)* Name of item.
`identifier`	`xs:string` *(Optional)* External reference to item or item ID.
`price`	`xs:decimal` *(Optional)* Total of item line.

Field	Description
itemPrice	xs:decimal *(Optional)* Price of an individual item.
itemCount	xs:int *(Optional)* Item quantity.

ReceiverIdentifier Fields

Field	Description
email	xs:string *(Optional)* Receiver's email address. Maximum length: 127 characters
phone	common:PhoneNumberType *(Optional)* Receiver's phone number.

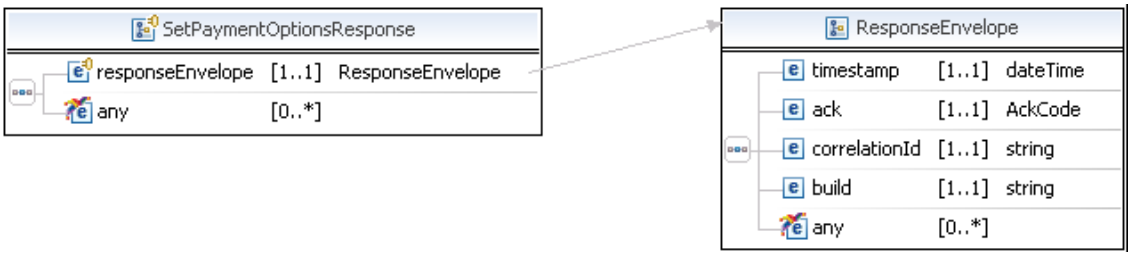
PhoneNumberType Fields

Field	Description
countryCode	xs:string *(Required)* Telephone country code.
phoneNumber	xs:string *(Required)* Telephone number.
extension	xs:string *(Optional)* Telephone extension.

RequestEnvelope Fields

Field	Description
detailLevel	common:DetailLevelCode *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is ReturnAll, which provides the maximum level of detail.
errorLanguage	xs:string *(Required)* RFC 3066 language in which error messages are returned; by default it is en_US, which is the only language currently supported.

SetPaymentOptionsResponse Message



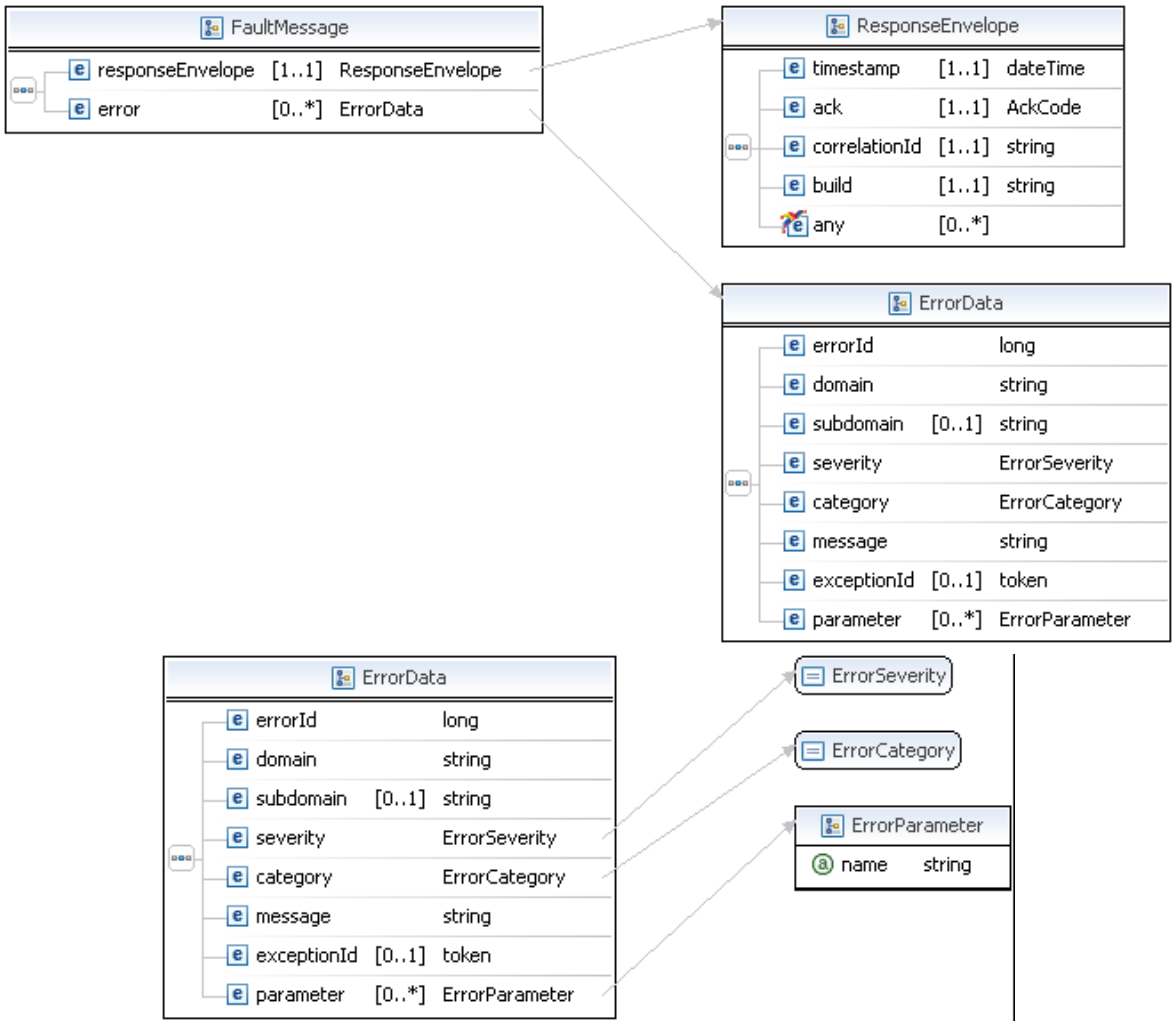
ResponseEnvelope Fields

Field	Description
`ack`	`common:AckCode` Acknowledgement code. It is one of the following values: • `Success` – The operation completed successfully. • `Failure` – The operation failed. • `SuccessWithWarning` – The operation completed successfully; however, there is a warning message. • `FailureWithWarning` – The operation failed with a warning message.
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The `PPFaultMessage` returns `ErrorData` and the `ResponseEnvelope` information to your application if an error occurs.



FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.
`severity`	`common:ErrorSeverity` The severity of the error encountered. Possible values are: • `Error` – Processing of the request was interrupted • `Warning` – Processing of the request was completed
`subdomain`	This field is not used.

ResponseEnvelope Fields

Field	Description
`ack`	`common:AckCode` Acknowledgement code. It is one of the following values: • `Success` – The operation completed successfully. • `Failure` – The operation failed. • `SuccessWithWarning` – The operation completed successfully; however, there is a warning message. • `FailureWithWarning` – The operation failed with a warning message.
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

SetPaymentOptions Examples Using NVP and CURL

These examples all use NVP for the data binding and CURL to deliver the HTTP request to the PayPal sandbox endpoint. Line breaks are provided for ease of reading; each CURL command is a single line and each request and response is a string without line breaks or extra whitespace.

Specifying Invoice data example

In this example, the details about the items or service being purchased in a given payment transaction are set up to be displayed in the payment review page before the user authorizes the payment. The item details include, the item description, item count, item price, total shipping and total tax.

NOTE: The sample code below uses the `insecure` setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/SetPaymentOptions -d
"requestEnvelope.errorLanguage=en_US&receiverOptions[0].receiver.email=test
@test.com
&receiverOptions[0].invoiceData.item[0].name=ITEM1&receiverOptions[0].invoi
ceData.item[0].price=50.0&receiverOptions[0].invoiceData.item[0].itemCount=
2&receiverOptions[0].invoiceData.item[0].itemPrice=25.0&receiverOptions[0].
invoiceData.totalTax=25.0&receiverOptions[0].invoiceData.totalShipping=25.0
&payKey=... "
```

SetPaymentOptions Errors

Code	Message	Additional Information
520002	Internal error	
520003	Authentication failed. API credentials are incorrect.	Since 1.6.0
550001	User is not allowed to perform this action	
550001	This payment request must be authorized by the sender	
540031	You do not have permission to set these payment options	
540031	You do not have permission to set these payment options Operation is not permitted because the credentials do not match those of the initial operation	Since 1.6.0
580001	Invalid request: <value>	Since 1.6.0
580022	• Invalid request parameter: payKey with value <value> • payKey <value> has already been used to make a payment • displayOptions has already been set and cannot be set multiple times • initiationEntity has already been set and cannot be set multiple times • Invalid Request Parameter: institutionId with value <value> • The instituion ID <value> is not activated	

Code	Message	Additional Information
580022	• Invalid request parameter: <value> • Invalid Request Parameter: institutionId with value <value> • The institution id <value> is not activated • The clientDetails parameter is missing • Invalid request parameter: payKey with value <value> • Invalid Phone Country Code <value> • Invalid Country Code <value> for Receiver • Invalid Receiver Phone Type <value> <value> <value> • Invalid Phone Format <value> <value> • Invalid Phone Number <value> <value> • Invalid Receiver Phone Number <value> <value> • Invalid Area or Exchange Code <value> <value> • <value> has already been set and cannot be set multiple times • Invalid request parameter: <value> with value <value> • Invalid request parameter: <value> <value> is invalid • Invalid request parameter: <value> <value> is duplicated • Invalid request parameter: <value> <value> cannot be negative or zero	Since 1.6.0
580027	• Currently the system does not accept the currency <value> • Unsupported country code <value> • Currently the system does not accept the currency <value> • The parameter is not supported	Since 1.6.0
580028	The URL <value> is malformed	

8 Preapproval API Operation

Use the `Preapproval` API operation to set up an agreement between yourself and a sender for making payments on the sender's behalf.

You set up a preapprovals for a specific maximum amount over a specific period of time and, optionally, by any of the following constraints: the number of payments, a maximum per-payment amount, a specific day of the week or the month, and whether or not a PIN is required for each payment request.

Preapproval Overview

To set up a preapproval request, you must specify values for the following fields:

Required preapproval fields

Field	Description
`endingDate`	Last date for which the preapproval is valid. It cannot be later than one year from the starting date. Contact PayPal if you do not want to specify an ending date.
`startingDate`	First date for which the preapproval is valid. It cannot be before today's date or after the ending date.
`maxTotalAmountOfAllPayments`	The preapproved maximum total amount of all payments. It cannot exceed $2,000 USD or its equivalent in other currencies. Contact PayPal if you do not want to specify a maximum amount.
`currencyCode`	The code for the currency in which the payment is made; you can specify only one currency, regardless of the number of receivers
`cancelUrl`	URL to redirect the sender's browser to after canceling the preapproval
`returnUrl`	URL to redirect the sender's browser to after the sender has logged into PayPal and confirmed the preapproval
`requestEnvelope.errorLanguage`	The code for the language in which errors are returned, which must be `en_US`.

Preapproval Notifications

Notifications are sent after preapproval is complete:

- PayPal sends an email to the sender that confirmed the approval.
- PayPal sends an IPN message to the URL specified in the `ipnNotificationUrl` field of the `Preapproval` request.

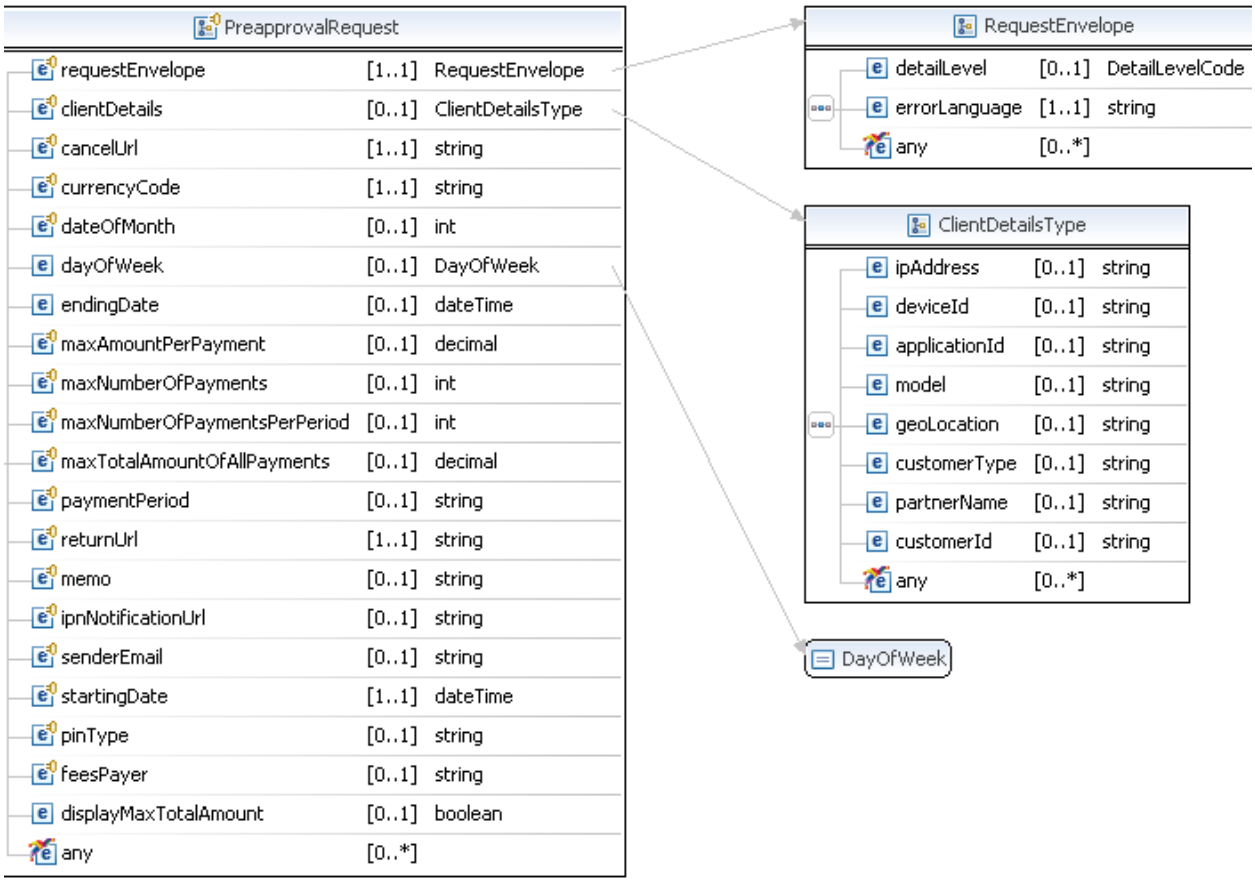
Additional Notes About the PreApproval API Operation

1. Preapproval constraints are additive; thus, for example, if you specify a preapproval that allows payments only on Fridays and on the 13th day of the month, the preapproval would be valid only on Friday the 13th of months within the specified time period.
2. The `startingDate` and `endingDate` can be in either Zulu or GMT offset formats. as in the following respective examples:

```
2010-09-10Z
2010-09-10T17:24:03.874-07:00
```

PreapprovalRequest Message

The `PreapprovalRequest` message contains the fields to set up a preapproval agreement between yourself and a sender.



PreapprovalRequest Fields

Field	Description
`cancelUrl`	`xs:string` *(Required)* The URL to which the sender's browser is redirected after the sender cancels the preapproval at paypal.com. Maximum length: 1024 characters
`clientDetails`	`common:ClientDetailsType` *(Required)* Information about the sender.

Field	Description
currencyCode	xs:string *(Required)* The currency code. Allowable values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD
dateOfMonth	xs:int *(Optional)* The day of the month on which a monthly payment is to be made. Allowable values are numbers between 0 and 31. A number between 1 and 31 indicates the date of the month. Specifying 0 indicates that payment can be made on any day of the month.

Field	Description
dayOfWeek	common:DayOfWeek *(Optional)* The day of the week that a weekly payment is to be made. Allowable values are: • NO_DAY_SPECIFIED • SUNDAY • MONDAY • TUESDAY • WEDNESDAY • THURSDAY • FRIDAY • SATURDAY
endingDate	xs:dateTime *(Optional)* Last date for which the preapproval is valid. It cannot be later than one year from the starting date. **NOTE:** You must specify a value unless you have specific permission from PayPal to omit this value.
ipnNotificationUrl	xs:string *(Optional)* The URL to which you want all IPN messages for this preapproval to be sent. This URL supersedes the IPN notification URL in your profile. Maximum length: 1024 characters
maxAmountPerPayment	xs:decimal *(Optional)* The preapproved maximum amount per payment. It cannot exceed the preapproved maximum total amount of all payments.
maxNumberOfPayments	xs:int *(Optional)* The preapproved maximum number of payments. It cannot exceed the preapproved maximum total number of all payments.
maxNumberOfPaymentsPerPeriod	xs:int *(Optional)* The preapproved maximum number of all payments per period. You must specify a value unless you have specific permission from PayPal.
maxTotalAmountOfAllPayments	xs:decimal *(Optional)* The preapproved maximum total amount of all payments. It cannot exceed $2,000 USD or its equivalent in other currencies. **NOTE:** You must specify a value unless you have specific permission from PayPal to omit this value.
memo	xs:string *(Optional)* A note about the preapproval. Maximum length: 1000 characters, including newline characters

Field	Description
paymentPeriod	xs:string *(Optional)* The payment period. It is one of the following values: • NO_PERIOD_SPECIFIED • DAILY – Each day • WEEKLY – Each week • BIWEEKLY – Every other week • SEMIMONTHLY – Twice a month • MONTHLY – Each month • ANNUALLY – Each year
pinType	xs:string *(Optional)* Whether a personal identification number (PIN) is required. It is one of the following values: • NOT_REQUIRED – A PIN is not required (default) • REQUIRED – A PIN is required; the sender must specify a PIN when setting up the preapproval on PayPal
requestEnvelope	common:RequestEnvelope *(Required)* Information common to each API operation, such as the language in which an error message is returned.
returnUrl	xs:string *(Required)* The URL to which the sender's browser is redirected after the sender approves the preapproval on paypal.com. Maximum length: 1024 characters
senderEmail	xs:string *(Optional)* Sender's email address. If not specified, the email address of the sender who logs in to approve the request becomes the email address associated with the preapproval key. Maximum length: 127 characters
startingDate	xs:dateTime *(Required)* First date for which the preapproval is valid. It cannot be before today's date or after the ending date.
displayMaxTotalAmount	xs:boolean *(Optional)* Whether to display the maximum total amount of this preapproval. It is one of the following values: • TRUE – Display the amount • FALSE – Do not display the amount (default)

Field	Description
feesPayer	xs:string *(Optional)* The payer of PayPal fees. Allowable values are: • SENDER – Sender pays all fees (for personal, implicit simple/parallel payments; do not use for chained or unilateral payments) • PRIMARYRECEIVER – Primary receiver pays all fees (chained payments only) • EACHRECEIVER – Each receiver pays their own fee (default, personal and unilateral payments) • SECONDARYONLY – Secondary receivers pay all fees (use only for chained payments with one secondary receiver)
requireInstantFundingSource	xs:boolean *(Optional)* Whether the PayPal user account must have an instant funding source for preapproval to be available. Allowable values are: • TRUE – Require an instant funding source in the PayPal user account • FALSE – Allow a non-instant funding source such as eCheck for preapproval (default)

ClientDetails Fields

Field	Description
applicationId	xs:string *(Optional)* Your application's identification, such as the name of your application
customerId	xs:string *(Optional)* Your ID for this sender Maximum length: 127 characters
customerType	xs:string *(Optional)* Your identification of the type of customer Maximum length: 127 characters
deviceId	xs:string *(Optional)* Sender's device ID, such as a mobile device's IMEI number or a web browser cookie. If a device ID was passed with the PayRequest, use the same ID here. Maximum length: 127 characters
geoLocation	xs:string *(Optional)* Sender's geographic location Maximum length: 127 characters
ipAddress	xs:string *(Optional)* Sender's IP address. If an IP addressed was passed with the PayRequest, use the same ID here.

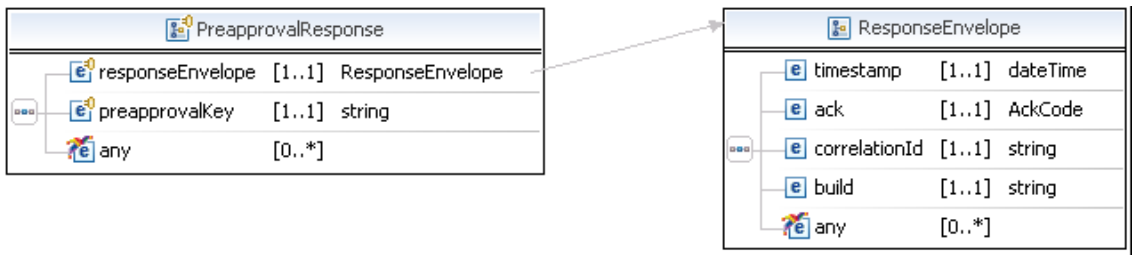
Field	Description
`model`	`xs:string` *(Optional)* A sub-identification of the application Maximum length: 127 characters
`partnerName`	`xs:string` *(Optional)* Your organization's name or ID Maximum length: 127 characters

RequestEnvelope Fields

Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.
`errorLanguage`	`xs:string` *(Required)* RFC 3066 language in which error messages are returned; by default it is `en_US`, which is the only language currently supported.

PreapprovalResponse Message

The `PreapprovalResponse` message contains a preapproval key. This key uniquely identifies the preapproval for this request, and can be used in other Adaptive Payment APIs to identify this specific preapproval.



PreapprovalResponse Fields

Field	Description
`preapprovalKey`	`xs:string` A preapproval key that identifies the preapproval requested. You can use this key in other Adaptive Payment requests to identify this preapproval.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the acknowledgement status.

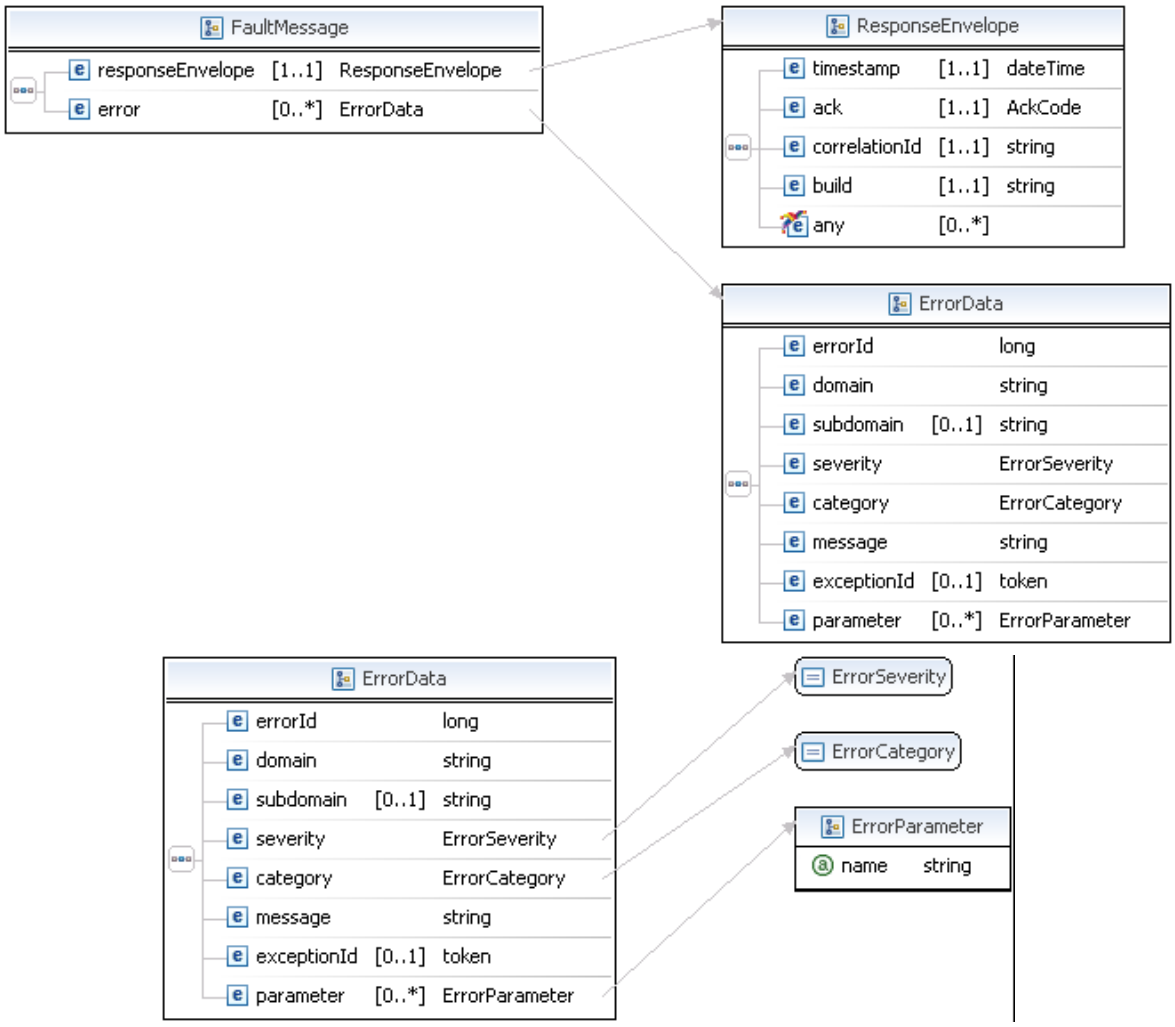
ResponseEnvelope Fields

Field	Description
`ack`	`common:AckCode` Acknowledgement code. It is one of the following values: • `Success` – The operation completed successfully. • `Failure` – The operation failed. • `SuccessWithWarning` – The operation completed successfully; however, there is a warning message. • `FailureWithWarning` – The operation failed with a warning message.
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The `PPFaultMessage` returns `ErrorData` and the `ResponseEnvelope` information to your application if an error occurs.



FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
category	common:ErrorCategory The location where the error occurred. Possible values are: • System – The system encountered errors; try again • Application – The application encountered errors; try again • Request – The request was incorrect
domain	xs:string The domain to which this service belongs.
errorId	xs:long A 6-digit number that uniquely identifies a particular error.
exceptionID	This field is not used.
message	xs:string A description of the error.
parameter	common:ErrorParameter Represents contextual information about the error.
severity	common:ErrorSeverity The severity of the error encountered. Possible values are: • Error – Processing of the request was interrupted • Warning – Processing of the request was completed
subdomain	This field is not used.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

Preapproval Examples Using NVP and CURL

These examples all use NVP for the data binding and CURL to deliver the HTTP request to the PayPal sandbox endpoint. Line breaks are provided for ease of reading; each CURL command is a single line and each request and response is a string without line breaks or extra whitespace.

Setting up a preapproval

In this example, the caller of the `Preapproval` API sets up a preapproval that is valid from July 17, 2009 through December 12, 2009. The maximum amount of the preapproval is $1,500, of which a maximum of 30 payments are authorized with the largest payment not to exceed $200. The sender need not include a PIN. The response contains the preapproval key that the caller can use in the `Pay` API operation.

NOTE: The sample code below uses the `insecure` setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/Preapproval  -d
"cancelUrl=http://your_cancel_url
&currencyCode=USD
&endingDate=2009-12-13T08%3A00%3A00.000Z
&maxAmountPerPayment=200.00
&maxNumberOfPayments=30
&maxTotalAmountOfAllPayments=1500.00
&pinType=NOT_REQUIRED
&requestEnvelope.errorLanguage=en_US
&returnUrl=http://your_return_url
&startingDate=2009-07-13T07%3A00%3A00.000Z
&senderEmail=sender@domain
```

Response:

```
responseEnvelope.timestamp=2009-08-14T09%3A00%3A37.748-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=7967b2d03745a
&responseEnvelope.build=DEV
&preapprovalKey=PA-9JR04288NR0519129
```

Preapproval Errors

Code	Message	Additional Information
500000	There is a system error	
520002	Internal error	
520003	User name/password is incorrect	
520006	This call is not defined in the database	
569016	Preapproval PIN functionality is not enabled	
569018	Preapproved payments have been disabled	
570026	The starting date is too far in the future	
579038	The date range between the start date and the end date is too wide	
579055	The limit for total amount of all payments is above PayPal's limit	

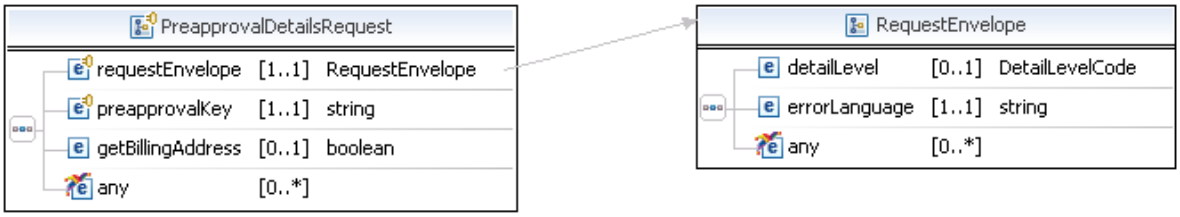
Code	Message	Additional Information
580001	Invalid request	
580022	Account already exists	Since 1.3.0
580024	The start date must be in the future	
580025	The start date must be earlier than the end date	
580027	The argument is unsupported	
580028	The URL <value> is malformed	
589023	If a fractional amount is rounded due to currency conversion, funds could be lost	
589039	The email address is invalid. It may not be registered in PayPal's system yet	

9 PreapprovalDetails API Operation

Use the `PreapprovalDetails` API operation to obtain information about an agreement between you and a sender for making payments on the sender's behalf.

PreapprovalDetailsRequest Message

The `PreapprovalDetailsRequest` message specifies the key of the preapproval agreement whose details you want to obtain.



PreapprovalDetailsRequest Fields

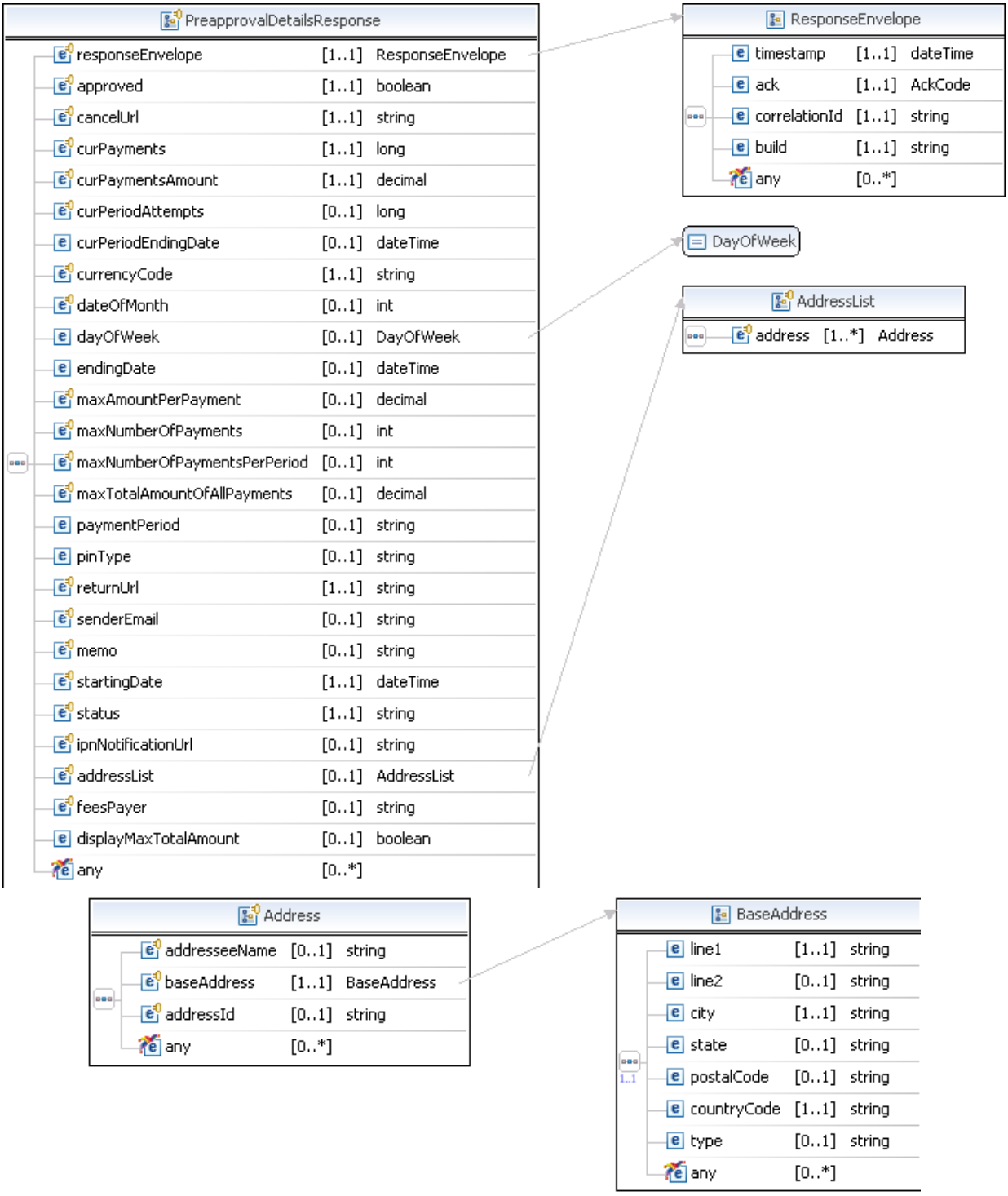
Field	Description
`getBillingAddress`	`xs:boolean` *(Optional)* An option that lets you retrieve a list of billing addresses for the sender. • `true` – Includes the billing address in the response • `false` – Omits the billing address from the response (default) **NOTE:** This field is available only to API callers with advanced permission levels. For information, refer to the section Adaptive Payments Permission Levels.
`preapprovalKey`	`xs:string` *(Required)* A preapproval key that identifies the preapproval for which you want to retrieve details. The preapproval key is returned in the `PreapprovalResponse` message.
`requestEnvelope`	`common:requestEnvelope` *(Required)* Information common to each API operation, such as the language in which an error message is returned.

RequestEnvelope Fields

Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.
`errorLanguage`	`xs:string` *(Required)* RFC 3066 language in which error messages are returned; by default it is `en_US`, which is the only language currently supported.

PreapprovalDetailsResponse Message

The `PreapprovalDetailsResponse` message provides details about the requested preapproval.


PreapprovalDetailsResponse
responseEnvelope [1..1] ResponseEnvelope
approved [1..1] boolean
cancelUrl [1..1] string
curPayments [1..1] long
curPaymentsAmount [1..1] decimal
curPeriodAttempts [0..1] long
curPeriodEndingDate [0..1] dateTime
currencyCode [1..1] string
dateOfMonth [0..1] int
dayOfWeek [0..1] DayOfWeek
endingDate [0..1] dateTime
maxAmountPerPayment [0..1] decimal
maxNumberOfPayments [0..1] int
maxNumberOfPaymentsPerPeriod [0..1] int
maxTotalAmountOfAllPayments [0..1] decimal
paymentPeriod [0..1] string
pinType [0..1] string
returnUrl [1..1] string
senderEmail [0..1] string
memo [0..1] string
startingDate [1..1] dateTime
status [1..1] string
ipnNotificationUrl [0..1] string
addressList [0..1] AddressList
feesPayer [0..1] string
displayMaxTotalAmount [0..1] boolean
any [0..*]
ResponseEnvelope
timestamp [1..1] dateTime
ack [1..1] AckCode
correlationId [1..1] string
build [1..1] string
any [0..*]
DayOfWeek
AddressList
address [1..*] Address
Address
addresseeName [0..1] string
baseAddress [1..1] BaseAddress
addressId [0..1] string
any [0..*]
BaseAddress
line1 [1..1] string
line2 [0..1] string
city [1..1] string
state [0..1] string
postalCode [0..1] string
countryCode [1..1] string
type [0..1] string
any [0..*]

PreapprovalDetailsResponse Fields

Field	Description
`addresslist`	`ap:AddressList` Returns a list of billing addresses.
`approved`	Whether the preapproval request was approved. Possible values are: • `true` – The request was approved • `false` – The request was denied
`cancelUrl`	The URL to which the sender's browser is redirected if the sender decides to cancel the preapproval as requested. Use the preapproval key to identify the preapproval as follows: `preapprovalKey=${preapprovalKey}`.
`curPayments`	The current number of payments by the sender for this preapproval.
`curPaymentsAmount`	The current total of payments by the sender for this preapproval.
`curPeriodAttempts`	The current number of payments by the sender this period for this preapproval.
`curPeriodEndingDate`	Ending date for the current period. Time is currently not supported.

Field	Description
currencyCode	The currency code represented by the following values: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD
dateOfMonth	The day of the month on which a monthly payment is to be made. A number between 1 and 31 indicates the day of the month. 0 indicates that the payment can be made on any day.

Field	Description
dayOfWeek	The day of the week that a weekly payment is to be made. Possible values are: • NO_DAY_SPECIFIED • SUNDAY • MONDAY • TUESDAY • WEDNESDAY • THURSDAY • FRIDAY • SATURDAY
endingDate	Last date for which the preapproval is valid. Time is currently not supported. **NOTE:** You must specify a value unless you have specific permission from PayPal to omit this value.
ipnNotificationUrl	The URL to which all IPN messages for this preapproval are sent.
maxAmountPerPayment	The preapproved maximum amount per payment.
maxNumberOfPayments	The preapproved maximum number of payments. This field is only returned if provided.
maxNumberOfPaymentsPerPeriod	The preapproved maximum number of payments per period.
maxTotalAmountOfAllPayments	The preapproved maximum total amount of all payments. **NOTE:** You must specify a value unless you have specific permission from PayPal to omit this value.
memo	A note about the preapproval.
paymentPeriod	The payment period. Possible values are: • NO_PERIOD_SPECIFIED • DAILY • WEEKLY • BIWEEKLY • SEMIMONTHLY • MONTHLY • ANNUALLY
pinType	Whether a personal identification number (PIN) is required. It is one of the following values: • NOT_REQUIRED – A PIN is not required • REQUIRED – A PIN is required
responseEnvelope	Common response information, including a timestamp and the acknowledgement status.
returnUrl	The URL to which the sender's browser is redirected after the sender approves the preapproval on paypal.com. Use the preapproval key to identify the preapproval as follows: preapprovalKey=${preapprovalKey}.

Field	Description
senderEmail	Sender's email address. If not specified, the email address of the sender who logs in to approve the request becomes the email address associated with the preapproval key.
startingDate	First date for which the preapproval is valid.
status	Whether this preapproval is active, represented by the following values: • ACTIVE – The preapproval is active • CANCELED – The preapproval was explicitly canceled by the sender or by PayPal • DEACTIVATED – The preapproval is not active; you can reactivate it by resetting the personal identification number (PIN) or by contacting PayPal
displayMaxTotalAmount	xs:boolean Whether to display the maximum total amount of this preapproval. It is one of the following values: • TRUE – Display the amount • FALSE – Do not display the amount (default)
feesPayer	xs:string The payer of PayPal fees. Allowable values are: • SENDER – Sender pays all fees (for personal, implicit simple/parallel payments; do not use for chained or unilateral payments) • PRIMARYRECEIVER – Primary receiver pays all fees (chained payments only) • EACHRECEIVER – Each receiver pays their own fee (default, personal and unilateral payments) • SECONDARYONLY – Secondary receivers pay all fees (use only for chained payments with one secondary receiver)

AddressList Fields

Field	Description
AddressList	ap:Address Returns a billing address.

Address Fields

Field	Description
addresseeName	xs:string The name associated with the address.

Field	Description
baseAddress	common:BaseAddress Street address.
addressId	xs:string The ID associated with the address.

BaseAddress Fields

Field	Description
city	xs:string The city of the address.
countryCode	xs:string The country code of the address.
line1	xs:string The first line of the address.
line2	xs:string An second line of the street address.
postalCode	xs:string The postal code of the address.
state	xs:string The state for the address
type	xs:string. The type of address.

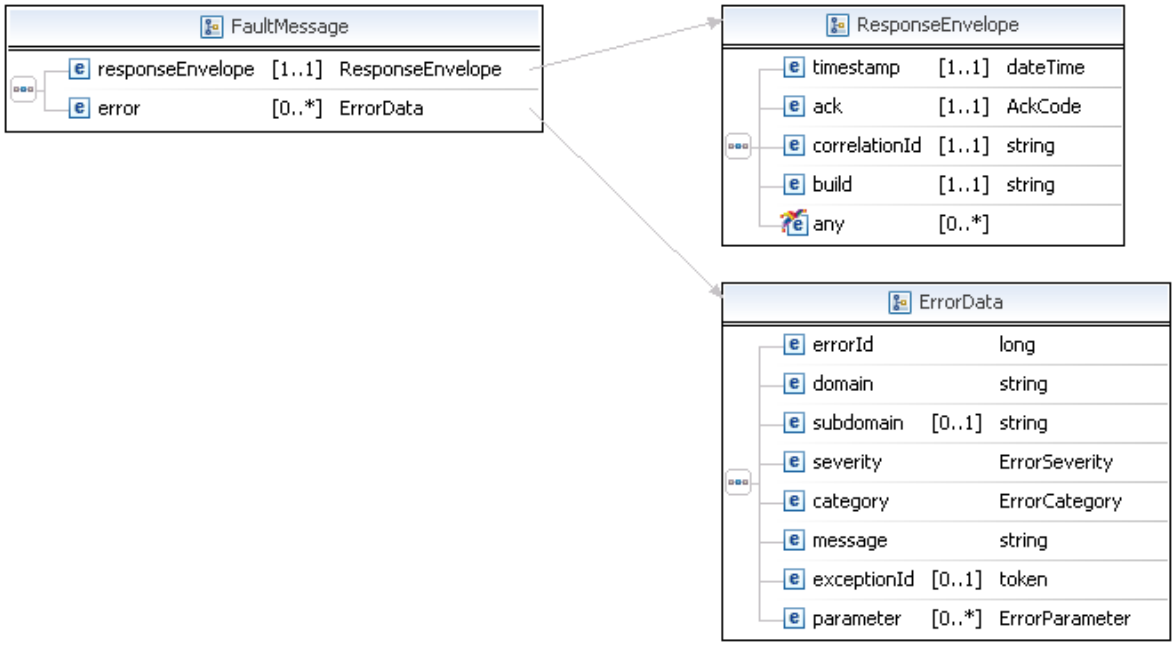
ResponseEnvelope Fields

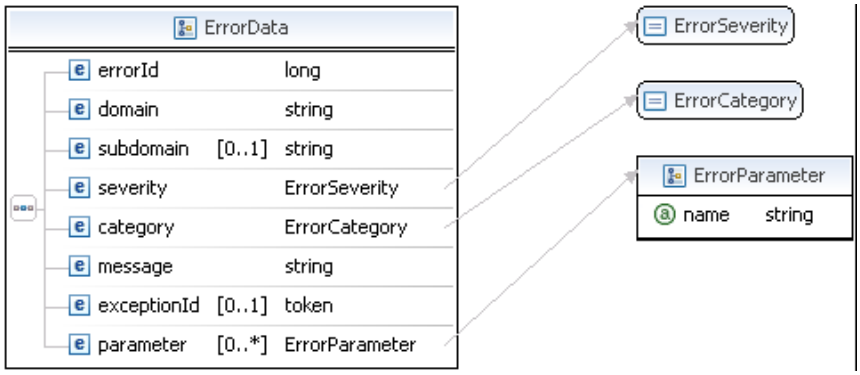
Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The PPFaultMessage returns ErrorData and the ResponseEnvelope information to your application if an error occurs.





FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.

Field	Description
severity	common:ErrorSeverity The severity of the error encountered. Possible values are: • Error – Processing of the request was interrupted • Warning – Processing of the request was completed
subdomain	This field is not used.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PreapprovalDetails Examples Using NVP and CURL

These examples all use NVP for the data binding and CURL to deliver the HTTP request to the PayPal sandbox endpoint. Line breaks are provided for ease of reading; each CURL

command is a single line and each request and response is a string without line breaks or extra whitespace.

Obtaining information about a preapproval

In this example, the caller of the `PreapprovalDetails` API operation specifies a preapproval key. The result shows whether or not the status has been approved, which in this case is `false` because the person making the approval has not completed the process on PayPal.

NOTE: The sample code below uses the `insecure` setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV"
-H "X-PAYPAL-APPLICATION-ID: app_id"
https://svcs.sandbox.paypal.com/AdaptivePayments/PreapprovalDetails  -d
"preapprovalKey=PA-9JR04288NR0519129
&requestEnvelope.errorLanguage=en_US "
```

Response:

```
responseEnvelope.timestamp=2009-07-13T13%3A50%3A40.496-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=16dbc35f1aea7
&responseEnvelope.build=DEV
&approved=false
&cancelUrl=your_cancel_url
&curPayments=0
&curPaymentsAmount=0.00
&curPeriodAttempts=0
&currencyCode=USD
&dateOfMonth=0
&dayOfWeek=NO_DAY_SPECIFIED
&endingDate=2009-12-13T13%3A40%3A37.0-08%3A00&maxAmountPerPayment=200.00
&maxNumberOfPayments=30
&maxTotalAmountOfAllPayments=1500.00
&paymentPeriod=NO_PERIOD_SPECIFIED
&pinType=NOT_REQUIRED
&returnUrl=your_cancel_url
&startingDate=2009-07-13T13%3A40%3A37.0-07%3A00
&status=ACTIVE
```

Preapproval Details Errors

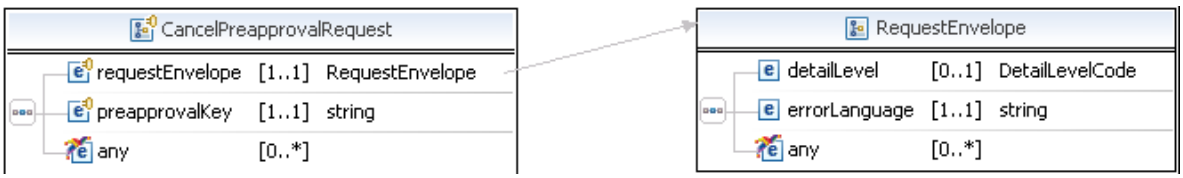
Code	Message	Additional Information
500000	There is a system error	
520002	Internal error	
520003	User name/password is incorrect	
520003	Authentication failed. API credentials are incorrect.	Since 1.6.0
520006	This call is not defined in the database	
540031	You do not have permission to get these preapproval details	Since 1.6.0
550001	User is not allowed to perform this action	
580001	Invalid request	
580001	Invalid request: <value>	Since 1.6.0
589019	The preapproval key is invalid	
589019	The preapproval key <value> is invalid	Since 1.6.0

10 CancelPreapproval API Operation

Use the `CancelPreapproval` API operation to handle the canceling of preapprovals. Preapprovals can be canceled regardless of the state they are in, such as active, expired, deactivated, and previously canceled.

CancelPreapprovalRequest Message

The `CancelPreapprovalRequest` message specifies which preapproval is to be canceled by identifying the preapproval with its the preapproval key.



CancelPreapprovalRequest Fields

The table below lists the fields for the `CancelPreapprovalRequest` message.

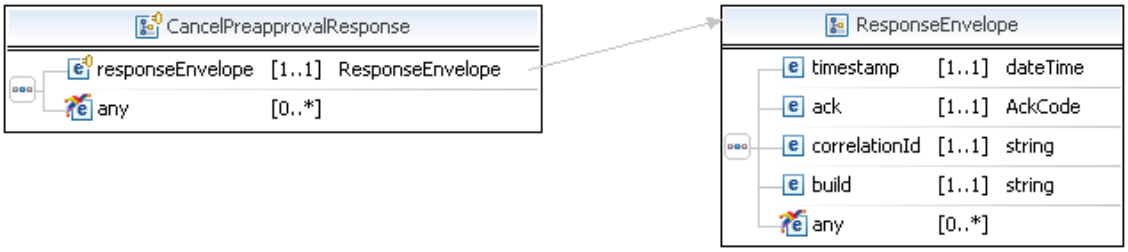
Field	Description
`preapprovalKey`	`xs:string` *(Required)* The preapproval key that identifies the preapproval to be canceled.
`requestEnvelope`	`common:requestEnvelope` *(Required)* Information common to each API operation, such as the language in which an error message is returned.

RequestEnvelope Fields

Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.
`errorLanguage`	`xs:string` *(Required)* RFC 3066 language in which error messages are returned; by default it is `en_US`, which is the only language currently supported.

CancelPreapprovalResponse Message

The `CancelPreapprovalResponse` message contains information returned in response to a `CancelPreapprovalRequest` message.



CancelPreapprovalResponse Fields

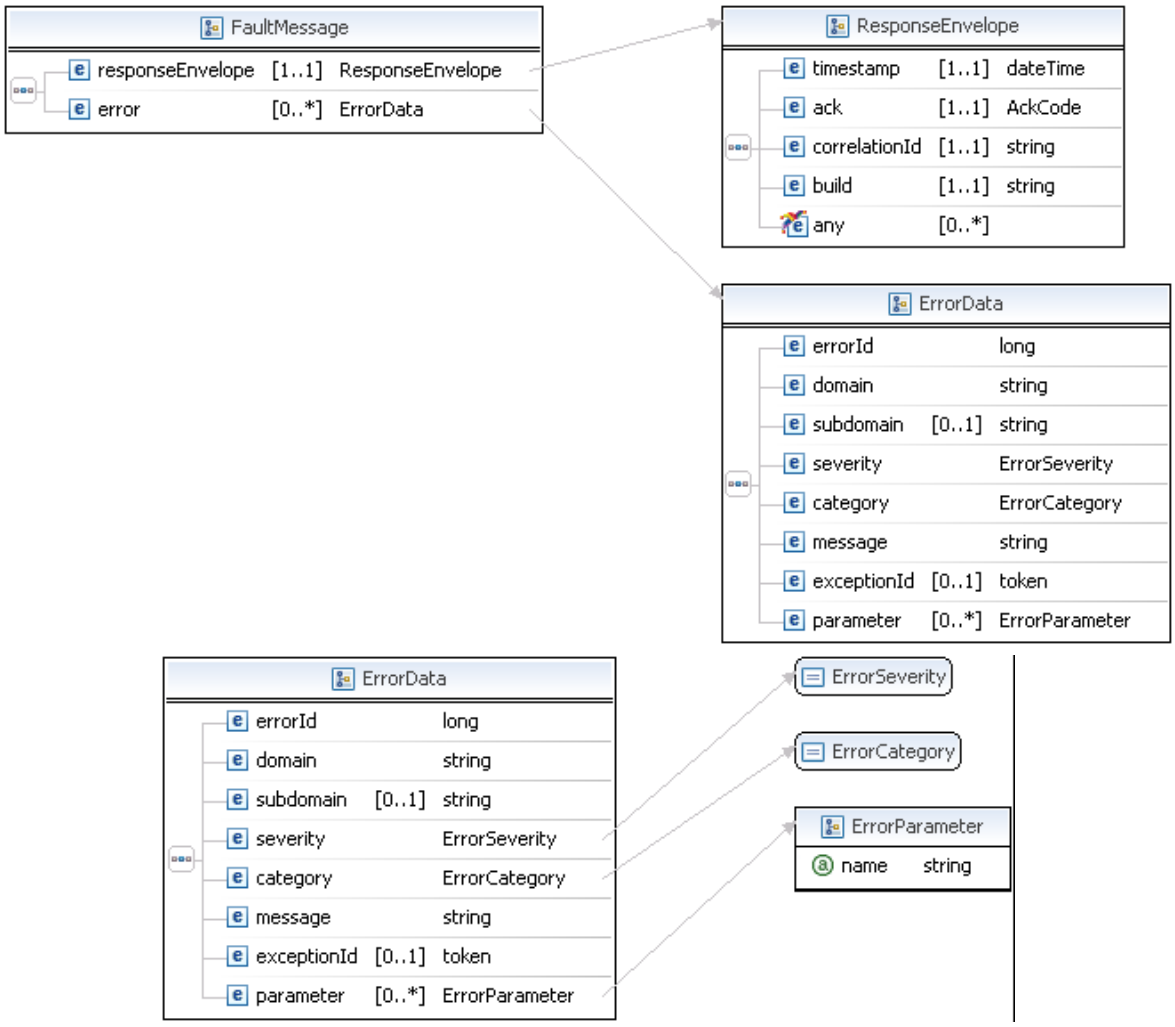
Field	Description
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The PPFaultMessage returns ErrorData and the ResponseEnvelope information to your application if an error occurs.



FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
category	common:ErrorCategory The location where the error occurred. Possible values are: • System – The system encountered errors; try again • Application – The application encountered errors; try again • Request – The request was incorrect
domain	xs:string The domain to which this service belongs.
errorId	xs:long A 6-digit number that uniquely identifies a particular error.
exceptionID	This field is not used.
message	xs:string A description of the error.
parameter	common:ErrorParameter Represents contextual information about the error.
severity	common:ErrorSeverity The severity of the error encountered. Possible values are: • Error – Processing of the request was interrupted • Warning – Processing of the request was completed
subdomain	This field is not used.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

CancelPreapproval Errors

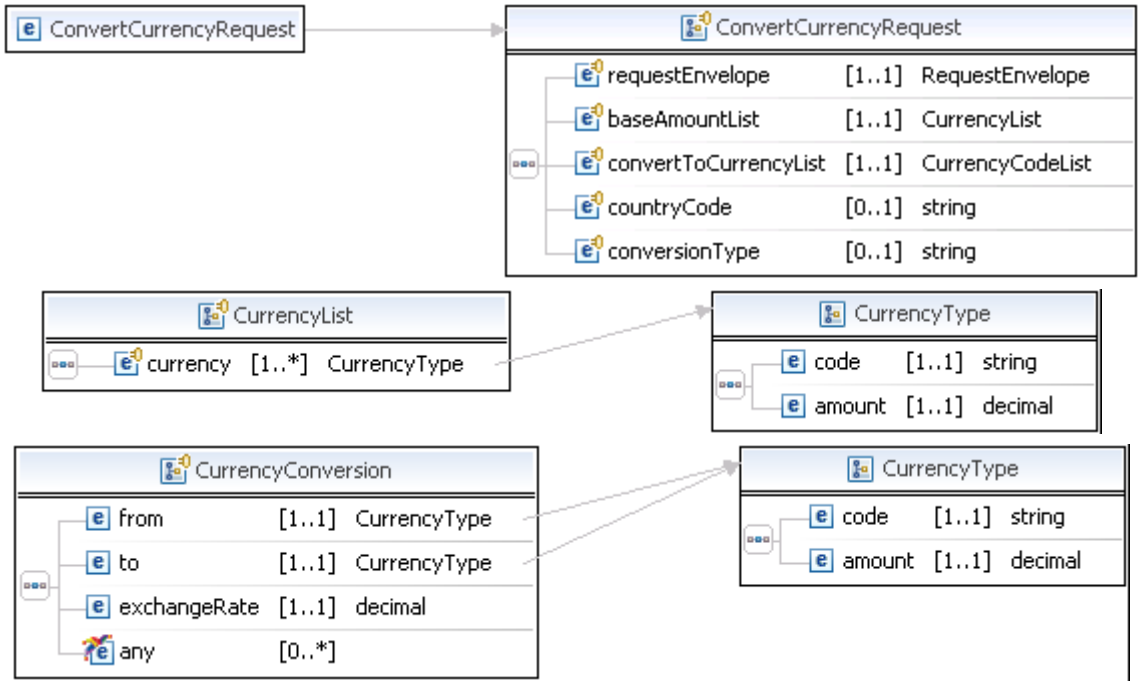
Code	Message	Additional Information
520002	Internal error	
540031	You don't have permission to cancel this preapproval	
589019	The preapproval key is invalid	

11 ConvertCurrency API Operation

Use the `ConvertCurrency` API operation to obtain the estimated current foreign exchange (FX) rate for a specific amount and currency. This is an estimated rate, which may be different from the FX rate used at the time of the transaction.

ConvertCurrencyRequest Message

The `ConvertCurrencyRequest` message enables you to have your application get an estimated exchange rate for a list of amounts. This API operation does not affect PayPal balances.



ConvertCurrencyRequest Fields

The table below describes the fields for the `ConvertCurrencyRequest` message.

Field	Description
baseAmountList	ap:CurrencyList *(Required)* A list of amounts with associated currencies to be converted.
convertToCurrencyList	ap:CurrencyCodeList *(Required)* A list of currencies to convert to.
requestEnvelope	common:RequestEnvelope *(Required)* Information common to each API operation, such as the language in which an error message is returned
countryCode	xs:string *(Optional)*The two-character ISO code for the country where the function is supposed to happen. The default value is US.
conversionType	xs:string *(Optional)*The conversion type allows you to determine the converted amounts for a PayPal user in different currency conversion scenarios, e.g., sending a payment in a different currency than what this user holds, accepting payment in a different currency than what the user holds, or converting a balance to a different currency than the user holds.. The default value is SENDER_SIDE.

CurrencyType Fields

Field	Description
amount	xs:decimal *(Required)* The amount to be converted.

Field	Description
code	`xs:string` *(Required)* The currency code. Allowable values are: • Australian Dollar – `AUD` • Brazilian Real – `BRL` **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – `CAD` • Czech Koruna – `CZK` • Danish Krone – `DKK` • Euro – `EUR` • Hong Kong Dollar – `HKD` • Hungarian Forint – `HUF` • Israeli New Sheqel – `ILS` • Japanese Yen – `JPY` • Malaysian Ringgit – `MYR` **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – `MXN` • Norwegian Krone – `NOK` • New Zealand Dollar – `NZD` • Philippine Peso – `PHP` • Polish Zloty – `PLN` • Pound Sterling – `GBP` • Singapore Dollar – `SGD` • Swedish Krona – `SEK` • Swiss Franc – `CHF` • Taiwan New Dollar – `TWD` • Thai Baht – `THB` • Turkish Lira – `TRY` **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – `USD`

CurrencyList Fields

Field	Description
currency	ap:CurrencyType *(Required)* The currency code. Allowable values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD

CurrencyCodeList Fields

Field	Description
currencyCode	xs:string *(Required)* The currency code. Allowable values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD

RequestEnvelope Fields

Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.
`errorLanguage`	`xs:string` *(Required)* RFC 3066 language in which error messages are returned; by default it is `en_US`, which is the only language currently supported.

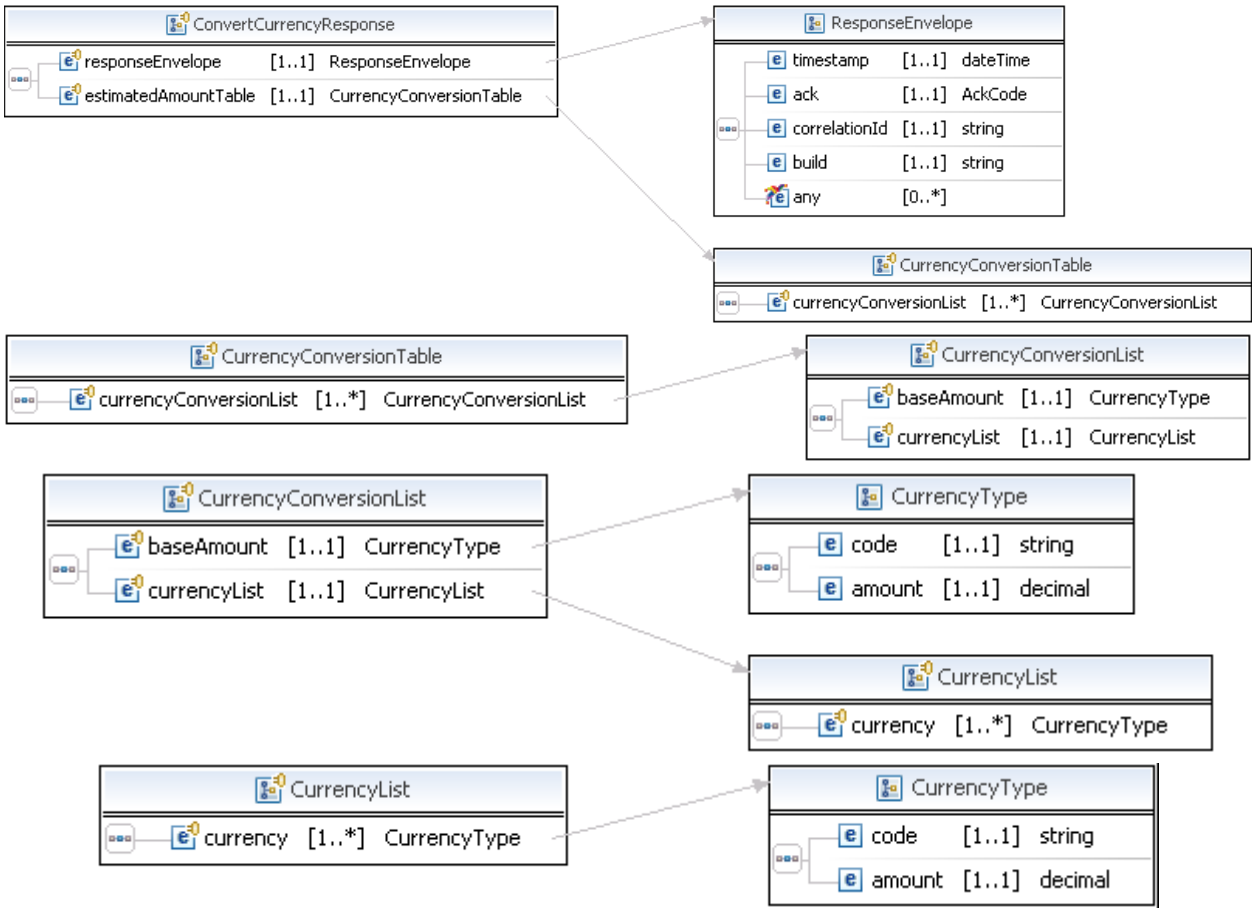
ConversionType Fields

The table below describes the `ConversionType` field.

Field	Description
`conversionType`	`xs:string` *(Optional)* The conversion type allows you to determine the converted amounts for a PayPal user in different currency conversion scenarios. Possible values are: • `SENDER_SIDE` – Pass this conversion type when you convert amounts for a PayPal user who is sending a payment in a different currency from what he/she holds in PayPal. • `RECEIVER_SIDE` – Pass this conversion type when you when you convert amounts for a PayPal user who is accepting a payment in a currency that he/she does not hold and who wants to convert the received payment. • `BALANCE_TRANSFER` – Pass this conversion type when you convert amounts for a PayPal user who is converting a balance in one currency to a balance in a different currency on his/her profile.

ConvertCurrencyResponse Message

The `ConvertCurrencyResponse` message returns the following information to the application in response to the `ConvertCurrencyRequest` message.



ConversionCurrencyResponse Fields

Field	Description
`estimatedAmountTable`	`ap:CurrencyConversionTable` A list of the requested amounts converted to the requested currencies.
`responseEnvelope`	`common:responseEnvelope` Information common to each API operation, such as the language in which an error message is returned.

CurrencyConversionTable Fields

Field	Description
`currencyConversionList`	`ap:CurrencyConversionList` The list of converted currencies.

CurrencyConversionList Fields

Field	Description
`baseAmount`	`ap:CurrencyType` The base amount and currency code.
`currencyList`	`ap:CurrencyList` The list of amounts converted to the requested currencies, and the currency codes.

CurrencyType Fields

Field	Description
`amount`	`xs:decimal` The converted amount.

Field	Description
code	xs:string The currency code for the converted amount. Possible values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD

CurrencyList Fields

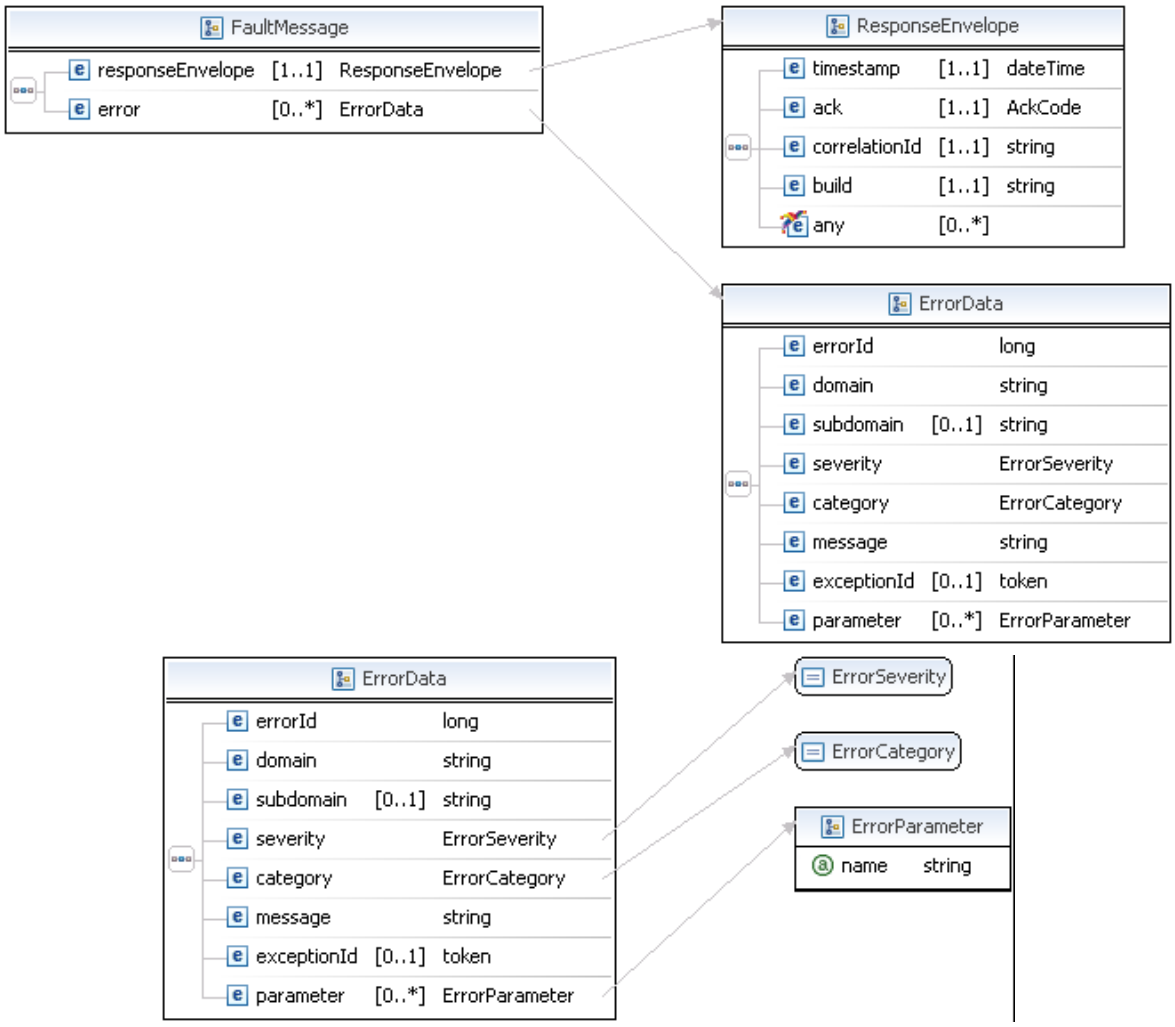
Field	Description
currency	ap:CurrencyType The list of converted amounts with their target currency codes.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The PPFaultMessage returns ErrorData and the ResponseEnvelope information to your application if an error occurs.



FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.
`severity`	`common:ErrorSeverity` The severity of the error encountered. Possible values are: • `Error` – Processing of the request was interrupted • `Warning` – Processing of the request was completed
`subdomain`	This field is not used.

ResponseEnvelope Fields

Field	Description
`ack`	`common:AckCode` Acknowledgement code. It is one of the following values: • `Success` – The operation completed successfully. • `Failure` – The operation failed. • `SuccessWithWarning` – The operation completed successfully; however, there is a warning message. • `FailureWithWarning` – The operation failed with a warning message.
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

ConvertCurrency Errors

Code	Message	Additional Information
520002	Internal error	
580022	Your request was invalid. Check the parameter of the error to see which is invalid	
580027	The argument is unsupported	

12 Refund API Operation

Use the Refund API operation to refund all or part of a payment. You can specify the amount of the refund and identify the accounts to receive the refund by the payment key or tracking ID, and optionally, by transaction ID or the receivers of the original payment.

Refund API Overview

PayPal supports refunds made manually, via the PayPal account interface, or by means of the RefundTransaction API. The terms *sender* and *receiver* refer to sender and receivers of the original payment using the Pay API operation. When making a refund, the sender's account receives the refund and the receivers' accounts are the source of the refund. Refunds are made from one or more receivers to a sender.

To make a refund using the Refund API operation, you must:

- have made a Pay request for which payment you want to refund the payment, or you must have received the part of the payment you want to refund, and
- have permission to make a refund on behalf of the receiver, unless you are also the receiver; in other words, each receiver whose account is the source of the refund must have granted you third-party access permission to call the Refund API operation.

  NOTE: A receiver can grant you third-party access to make a refund by logging in to PayPal, choosing **API Access** on the Profile page, then clicking the link to Grant API permission and selecting **Refund** after clicking **Configure a custom API authorization**.

  NOTE: The Adaptive Payments Refund API call does not support pay keys generated by the Mobile Payment Library (MPL).

For each kind of refund, you must specify values for the following fields:

Common fields for all refunds

Field	Description
`currencyCode`	The currency code. You must specify the currency code that matches the currency code of the original payment unless you also specify the payment key.
`requestEnvelope.errorLanguage`	The code for the language in which errors are returned, which must be `en_US`

You can refund a payment from all receivers associated with a payment by specifying the payment's payment key or tracking ID:

Refunds that apply to all receivers

Field	Description
`payKey`	The key used to create the payment that you want to refund
`trackingId`	The tracking ID associated with the payment that you want to refund

You can make a refund to the sender of a payment from specific receivers associated with a payment key or tracking ID. In this case, you must specify:

- either the payment key or tracking ID that identifies the original payment
- for each receiver making a refund, the receiver's email address
- for each receiver, the amount to refund to the sender from the receiver's account.

NOTE: The receiver can't make a partial refund if the original payment was a cash advance, personal payment, or unilateral payment.

Additional fields for refunds of specific amounts to specific receivers

Field	Description
`receiverList.receiver(`*n*`).email`	The receiver's email address, where *n* is between 0 and 5 for a maximum of 6 receivers
`receiverList.receiver(`*n*`).amount`	Amount to refund from the receiver identified by email address

A payment can also be identified as a set of PayPal transactions between the sender and each specific receiver. You can refund the amount of the transaction by specifying the transaction ID:

Refunds that apply to a PayPal transaction associated with the payment

Field	Description
`transactionId`	A PayPal transaction ID associated with the receiver whose payment you want to refund to the sender. Use field name characters exactly as shown.

Refund Notifications

Notifications are sent after the refund is complete:

- PayPal sends an email to all PayPal account holders affected by the refund.
- PayPal sends an IPN message to the URL specified in the `ipnNotificationUrl` field of the `Pay` request.

Additional Notes About the Refund API Operation

1. Payments made using the Pay API operation cannot be refunded using the `RefundTransaction` API operation. You must use the `Refund` API operation or log in to PayPal to issue the refund.
2. A personal account holder cannot grant API access to a third party. A personal account holder must issue the refund by specifying the transaction ID, or by logging in to paypal.com to make the refund.
3. A payment made using the `Pay` API operation cannot be refunded by logging in to paypal.com if the fees cannot be refunded. In these cases, you must use the `Refund` API operation to issue the refund.
4. In the refund request, specify the payment using either the pay key, transaction ID, or tracking ID; it is not recommended to specify more than one of these arguments, unless they identify the same payment. If the arguments do not belong to the same payment, the transaction ID is used; next, it uses the payment key. If more than one way is specified, the ID is never used.
5. With the Adaptive Payments Pay API operation, you can make payments to a person who does not have a PayPal account. In essence, you send a payment request that includes an email address for a receiver, and this email address is not linked to a registered PayPal account. The receiver receives an email notifying the receiver to create an account and claim the payment. Supporting features include email notifications and IPNs to the sender, receiver, and API caller, plus the ability to cancel unclaimed payments.

   You can make such a payment in the same way that you make any other kind of payment; however, PayPal holds a payment to a receiver whose email address is not yet registered or confirmed until the receiver creates a PayPal account and confirms the email address. If a refund specifies a receiver whose email address is not yet registered or confirmed, the payment to the receiver is canceled.

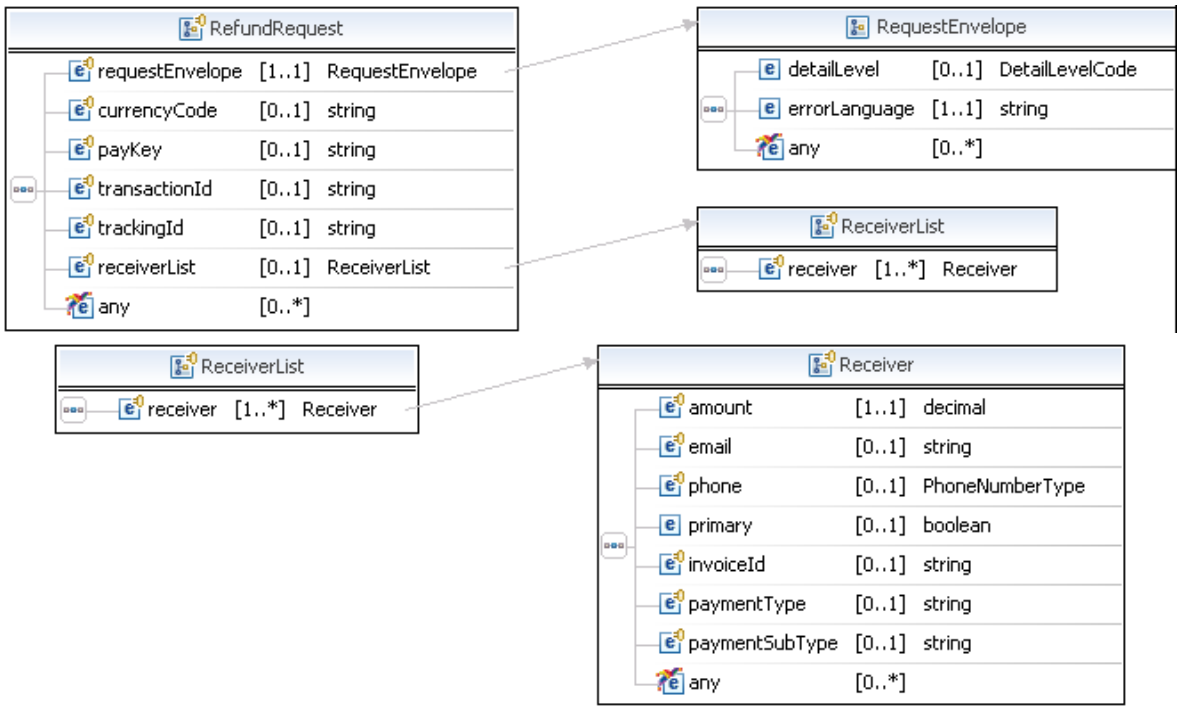
6. If the refund involves chained payments, PayPal first moves the money from the secondary receiver to the primary, refunding the secondary's share back to the primary. PayPal then attempts to refund the full amount from primary to buyer.

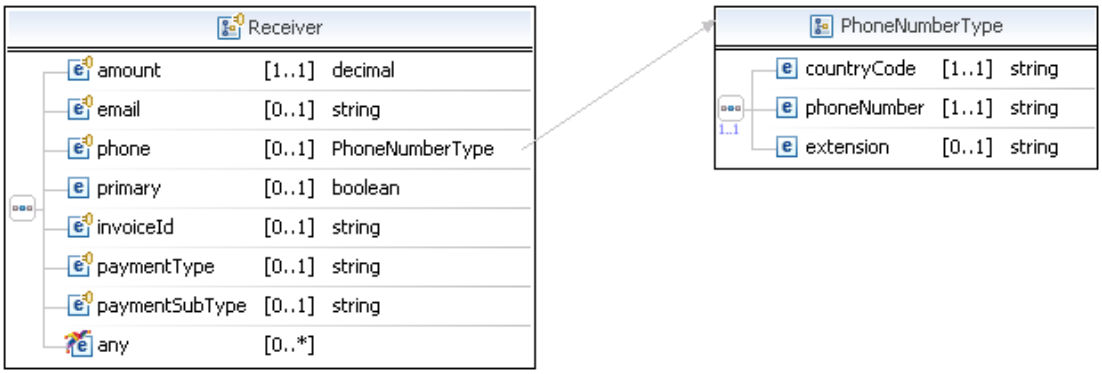
   At this point, if the primary receiver's account still does not have sufficient funds to cover the refund, PayPal initiates a draft from the Primary's bank account to get enough money to complete the refund. The bank draft is usually completed within a couple of days. At this time, since the primary has enough funds in his account, PayPal completes the refund to the buyer. If the bank draft fails, the refund request is cancelled.

   NOTE: The Refund API operation does not support iACH.

RefundRequest Message

The `refundRequest` message specifies information about the refund and how it is to be allocated to PayPal account holders.





RefundRequest Fields

You must specify a value for only one of the following fields to identify the part of the payment to refund: `payKey`, `transactionId`, or `trackingId`.

Field	Description
currencyCode	*(Required)* The currency code. You must specify the currency code used for the original payment. You do not need to specify this field if you use a payKey to refund a complete transaction. Allowable values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD
payKey	xs:string *(Optional)* The key used to create the payment that you want to refund.

Field	Description
receiverList	ap:ReceiverList *(Optional)* Information about the receivers of the payment that you want to refund; the payment is identified by payment key, tracking ID, or transaction ID. Specify each receiver to whom you want to issue a refund. You can specify the refund amount for specific receivers. If you do not specify any receivers, all receivers will be fully refunded if you specify the payment key or tracking ID; otherwise, only the affected receiver will be fully refunded if you specify the transaction ID.
requestEnvelope	common:requestEnvelope *(Required)* Information common to each API operation, such as the language in which an error message is returned.
transactionId	xs:string *(Optional)* The PayPal transaction ID associated with the payment that you want to refund.
trackingId	xs:string *(Optional)* The tracking ID associated with the payment that you want to refund.

ReceiverList Fields

Field	Description
receiver	ap:Receiver *(Required)* Receiver is the party whose account is credited.

Receiver Fields

Field	Description
amount	xs:decimal *(Required)* Amount to be credited to the receiver's account.
email	xs:string *(Required)* Receiver's email address. Maximum length: 127 characters
invoiceId	xs:string *(Optional)* This fields is not used. Maximum length: 127 characters
paymentType	xs:string *(Optional)* This field is not used.

Field	Description
paymentSubType	xs:string *(Optional)* This field is not used.
phone	common:PhoneNumberType *(Optional)* A type to specify the receiver's phone number. The PayRequest must pass either an email address or a phone number as the payment receiver.
primary	xs:boolean *(Optional)* Whether this receiver is the primary receiver, which makes this a refund for a chained payment. You can specify at most one primary receiver. Omit this field for refunds for simple and parallel payments. Allowable values are: • true – Primary receiver • false – Secondary receiver (default)

PhoneNumberType Fields

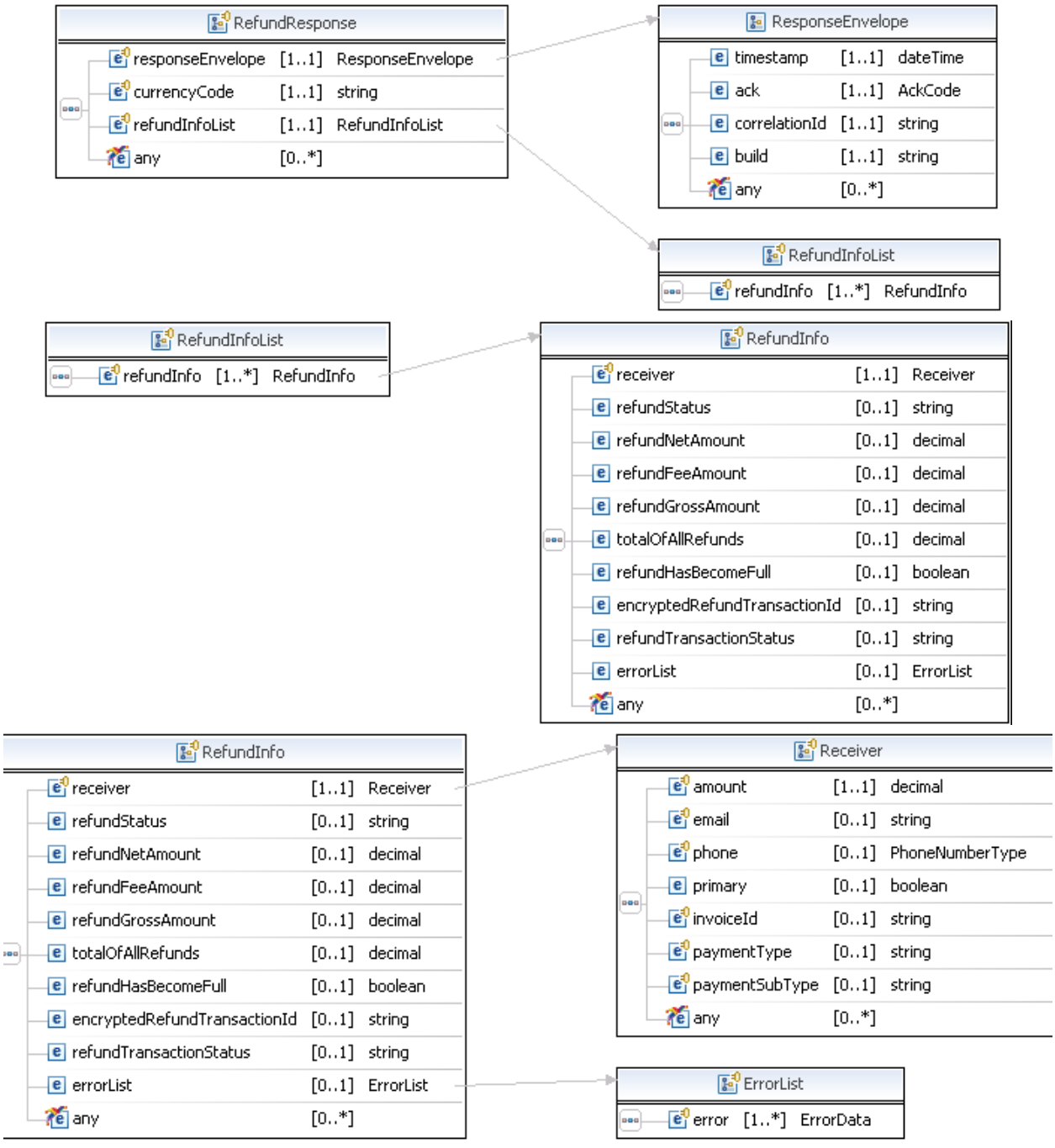
Field	Description
countryCode	xs:string *(Required)* Telephone country code.
phoneNumber	xs:string *(Required)* Telephone number.
extension	xs:string *(Optional)* Telephone extension.

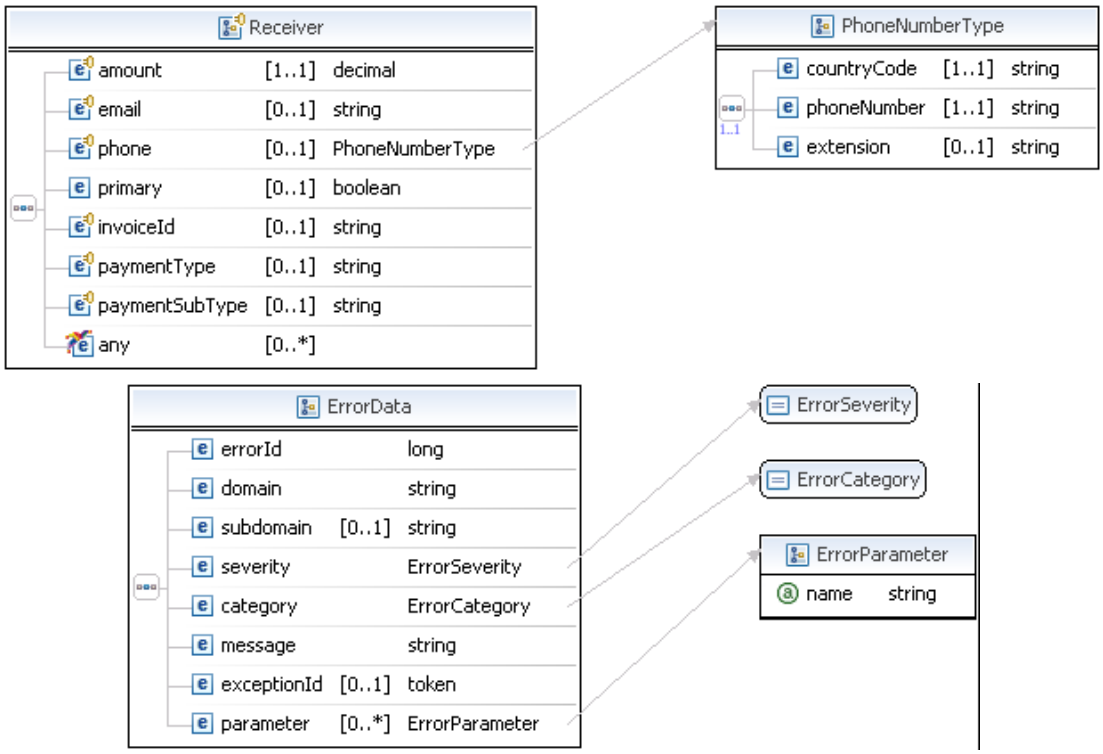
RequestEnvelope Fields

Field	Description
detailLevel	common:DetailLevelCode *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is ReturnAll, which provides the maximum level of detail.
errorLanguage	xs:string *(Required)* RFC 3066 language in which error messages are returned; by default it is en_US, which is the only language currently supported.

Refund Response Message

The `refundResponse` message contains status information about a refund request.




Receiver
amount [1..1] decimal
email [0..1] string
phone [0..1] PhoneNumberType
primary [0..1] boolean
invoiceId [0..1] string
paymentType [0..1] string
paymentSubType [0..1] string
any [0..*]
PhoneNumberType
countryCode [1..1] string
phoneNumber [1..1] string
extension [0..1] string
1..1
ErrorData
errorId long
domain string
subdomain [0..1] string
severity ErrorSeverity
category ErrorCategory
message string
exceptionId [0..1] token
parameter [0..*] ErrorParameter
ErrorSeverity
ErrorCategory
ErrorParameter
name string

RefundResponse Fields

Field	Description
currencyCode	The currency code represented by one of the following values: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD
refundInfoList	ap:RefundInfoList List of refunds associated with the payment.
responseEnvelope	common:ResponseEnvelope Common response information, including a timestamp and the response acknowledgement status.

RefundInfoList Fields

Field	Description
`refundInfo`	`ap:RefundInfo` Represents the refund attempt made to a receiver of a `PayRequest`.

RefundInfo Fields

Field	Description
`encryptedRefundTransactionId`	`xs:string` The PayPal transaction ID for this refund. **NOTE:** This ID is different than the transaction ID for the original payment.
`errorList`	`ap:ErrorList` Information about why a refund failed.
`receiver`	`ap:Receiver` Information about the receiver of the refund.
`refundFeeAmount`	`xs:decimal` Amount of fees that have been refunded.
`refundGrossAmount`	`xs:decimal` Gross amount of the refund, including the total amount from this receiver and fees.
`refundHasBecomeFull`	`xs:boolean` Whether you have refunded the total amount of the payment. Possible values are: • `true` – Entire amount of the payment has been refunded • `false` – Entire amount of the payment has not been refunded **NOTE:** This field is only available when a partial refund has been requested.
`refundNetAmount`	`xs:decimal` The total amount of the refund from this receiver.

Field	Description
refundStatus	xs:string Status of the refund. It is one of the following values: • REFUNDED – Refund successfully completed • REFUNDED_PENDING – Refund awaiting transfer of funds; for example, a refund paid by eCheck. • NOT_PAID – Payment was never made; therefore, it cannot be refunded. • ALREADY_REVERSED_OR_REFUNDED – Request rejected because the refund was already made, or the payment was reversed prior to this request. • NO_API_ACCESS_TO_RECEIVER – Request cannot be completed because you do not have third-party access from the receiver to make the refund. • REFUND_NOT_ALLOWED – Refund is not allowed. • INSUFFICIENT_BALANCE – Request rejected because the receiver from which the refund is to be paid does not have sufficient funds or the funding source cannot be used to make a refund. • AMOUNT_EXCEEDS_REFUNDABLE – Request rejected because you attempted to refund more than the remaining amount of the payment; call the PaymentDetails API operation to determine the amount already refunded. • PREVIOUS_REFUND_PENDING – Request rejected because a refund is currently pending for this part of the payment • NOT_PROCESSED – Request rejected because it cannot be processed at this time • REFUND_ERROR – Request rejected because of an internal error • PREVIOUS_REFUND_ERROR – Request rejected because another part of this refund caused an internal error.
refundTransactionStatus	xs:string The status of the refund transaction. Possible values are: • UNKNOWN – The status of the refund process is unknown. • COMPLETED – The refund process has been completed.
totalOfAllRefunds	xs:decimal The total of all refunds actually paid; does not include pending amounts **NOTE:** This field is only available when a partial refund has been requested.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.
`severity`	`common:ErrorSeverity` The severity of the error encountered. Possible values are: • `Error` – Processing of the request was interrupted • `Warning` – Processing of the request was completed
`subdomain`	This field is not used.

Receiver Fields

Field	Description
`amount`	`xs:decimal` Amount to be refunded to the receiver.
`email`	`xs:string` Receiver's email address. Maximum length: 127 characters
`invoiceId`	`xs:string` This fields is not used. Maximum length: 127 characters
`paymentType`	`xs:string` This field is not used.

Field	Description
paymentSubType	xs:string This field is not used.
phone	common:PhoneNumberType The receiver's phone number.
primary	xs:boolean Whether this receiver is the primary receiver, which makes this a refund for a chained payment. If this field is set to false, this is a refund for a simple or parallel payment. Possible values are: • true – Primary receiver (chained payment) • false – Secondary receiver (simple/parallel payment)

PhoneNumberType Response Fields

Field	Description
countryCode	xs:string Telephone country code.
phoneNumber	xs:string Telephone number.
extension	xs:string Telephone extension.

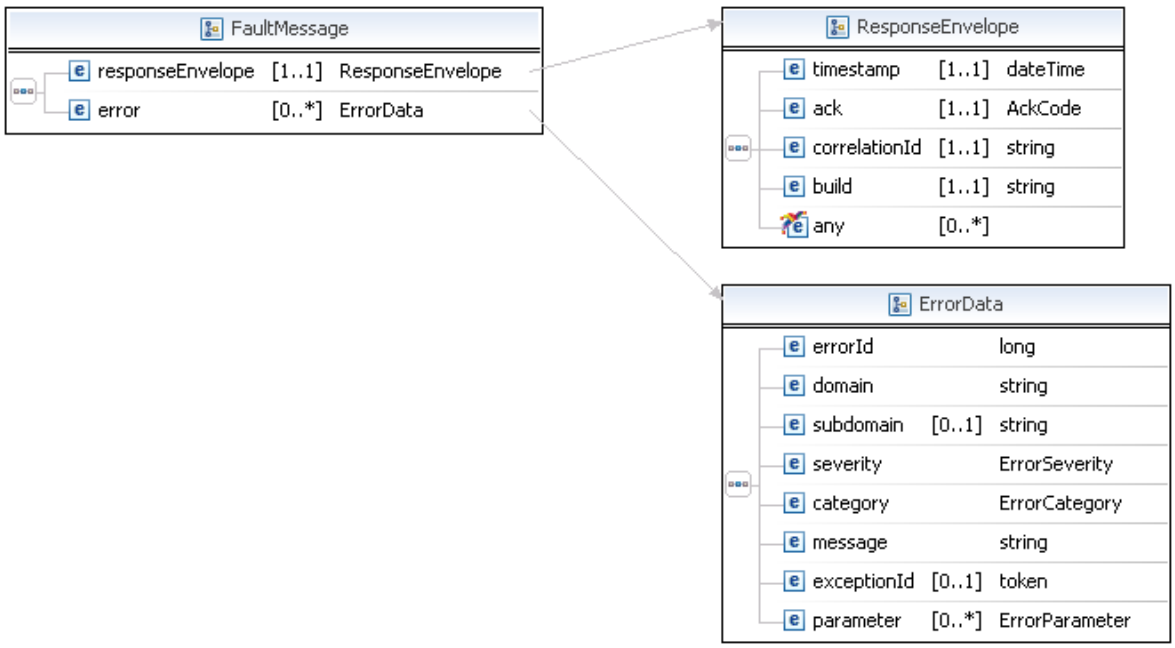
ResponseEnvelope Fields

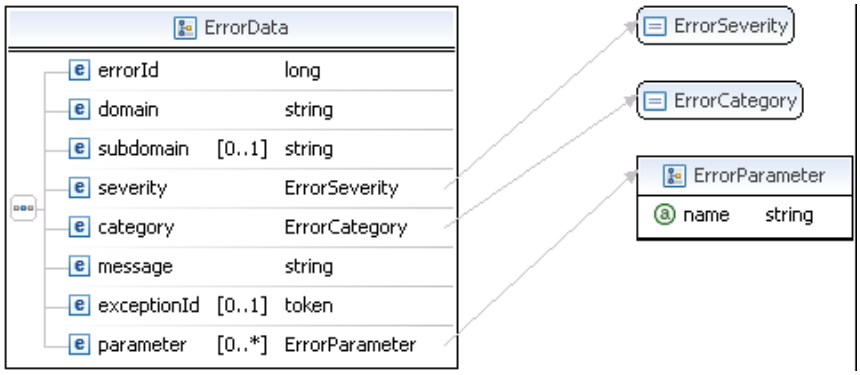
Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The PPFaultMessage returns ErrorData and the ResponseEnvelope information to your application if an error occurs.





FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.

Field	Description
severity	common:ErrorSeverity The severity of the error encountered. Possible values are: • Error – Processing of the request was interrupted • Warning – Processing of the request was completed
subdomain	This field is not used.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

Refund Examples Using NVP and CURL

These examples all use NVP for the data binding and CURL to deliver the HTTP request to the PayPal sandbox endpoint. Line breaks are provided for ease of reading; each CURL

command is a single line and each request and response is a string without line breaks or extra whitespace.

Using the payment key to refund an entire payment

In this example, the caller of the `Refund` API operation specifies a payment key. The total amount from each receiver is refunded to the sender.

NOTE: The sample code below uses the `insecure` setting to work around the certificate for testing in a sandbox environment. For actual implementations, you must specify the location of the certificate.

Request:

```
curl -s --insecure
-H "X-PAYPAL-SECURITY-USERID: api_username"
-H "X-PAYPAL-SECURITY-PASSWORD: api_password"
-H "X-PAYPAL-SECURITY-SIGNATURE: api_signature"
-H "X-PAYPAL-REQUEST-DATA-FORMAT: NV"
-H "X-PAYPAL-RESPONSE-DATA-FORMAT: NV
-H "X-PAYPAL-APPLICATION-ID: your_app_id "
https://svcs.sandbox.paypal.com/AdaptivePayments/Refund  -d
"requestEnvelope.errorLanguage=en_US
&payKey=AP-95V43510SV018561T"
```

Response:

```
responseEnvelope.timestamp=2009-08-14T09%3A00%3A37.748-07%3A00
&responseEnvelope.ack=Success
&responseEnvelope.correlationId=7967b2d03745a
&responseEnvelope.build=DEV
&currencyCode=USD
&refundInfoList.refundInfo(0).receiver.amount=3.15
&refundInfoList.refundInfo(0).receiver.email=receiver1@domain
&refundInfoList.refundInfo(0).refundStatus=REFUNDED
&refundInfoList.refundInfo(0).refundNetAmount=2.81
&refundInfoList.refundInfo(0).refundFeeAmount=0.34
&refundInfoList.refundInfo(0).refundGrossAmount=3.15
&refundInfoList.refundInfo(0).totalOfAllRefunds=3.15
&refundInfoList.refundInfo(0).refundHasBecomeFull=true
&refundInfoList.refundInfo(0).encryptedRefundTransactionId=48K11199GC160...
&refundInfoList.refundInfo(0).refundTransactionStatus=COMPLETED
&refundInfoList.refundInfo(1).receiver.amount=1.78
&refundInfoList.refundInfo(1).receiver.email=receiver2@domain
&refundInfoList.refundInfo(1).refundStatus=REFUNDED
&refundInfoList.refundInfo(1).refundNetAmount=1.43
&refundInfoList.refundInfo(1).refundFeeAmount=0.35
&refundInfoList.refundInfo(1).refundGrossAmount=1.78
&refundInfoList.refundInfo(1).totalOfAllRefunds=1.78
&refundInfoList.refundInfo(1).refundHasBecomeFull=true
&refundInfoList.refundInfo(1).encryptedRefundTransactionId=40Y76985BU687...
&refundInfoList.refundInfo(1).refundTransactionStatus=COMPLETED
```

Refund Errors

Code	Message	Additional Information
500000	There is a system error	
520002	Internal error	
520003	User name/password is incorrect	
520006	This call is not defined in the database	
520008	You do not have a verified ACH	
520009	Account is restricted	
520010	The account for the counter party is locked or inactive	
560011	Because a complaint case exists on this transaction, only a refund of the full or full remaining amount of the transaction can be issued	
560016	You cannot do a partial refund on this transaction	

Code	Message	Additional Information
560018	You cannot refund this type of transaction	
570012	Cannot do a full refund after a partial refund	
570013	The partial refund amount must be less than or equal to the original transaction amount	
570014	The partial refund amount must be less than or equal to the remaining amount	
570015	This transaction has already been fully refunded	
570017	You are over the time limit to perform a refund on this transaction	
579051	The tracking ID is invalid	
580001	Invalid request	
580022	Invalid request parameter: payKey with value <value>	
580027	The argument is unsupported	
589018	The pay key is valid	
589023	If a fractional amount is rounded due to currency conversion, funds could be lost	
580029	A pay key, transaction ID, or tracking ID is missing	Since 1.6.0
580030	This transaction cannot be processed at this time. Please try again later.	Since 1.3.0
589036	The currency code is missing or does not match the code in the payment request	
589037	The refund's receiver was not part of the payment request	
589038	The refund has to be both less than or equal to the original payment request and greater than zero	
589041	All receivers must give full refunds for this type of chained payment	
589052	The transaction ID is invalid	
589061	The receiver <value> is invalid for this refund	Since 1.6.0

13 GetFundingPlans API Operation

Use the GetFundingPlans API operation to determine the funding sources that are available for a specified payment, identified by its key, which takes into account the preferences and country of the receiver as well as the payment amount. You must be both the sender of the payment and the caller of this API operation.

GetFundingPlans Overview

To obtain the funding plans for a payment, specify values for the following fields:

Required fields

Field	Description
payKey	The key used to create the payment whose funding sources you want to determine.
requestEnvelope.errorLanguage	The code for the language in which errors are returned, which must be en_US.

Additional Notes About the Pay API Operation

1. The GetFundingPlans API operation is only avail a be for implicitly approved payments, meaning you must be both the sender and the caller of the Pay API operation identified by the pay key.

GetFundingPlansRequest Message

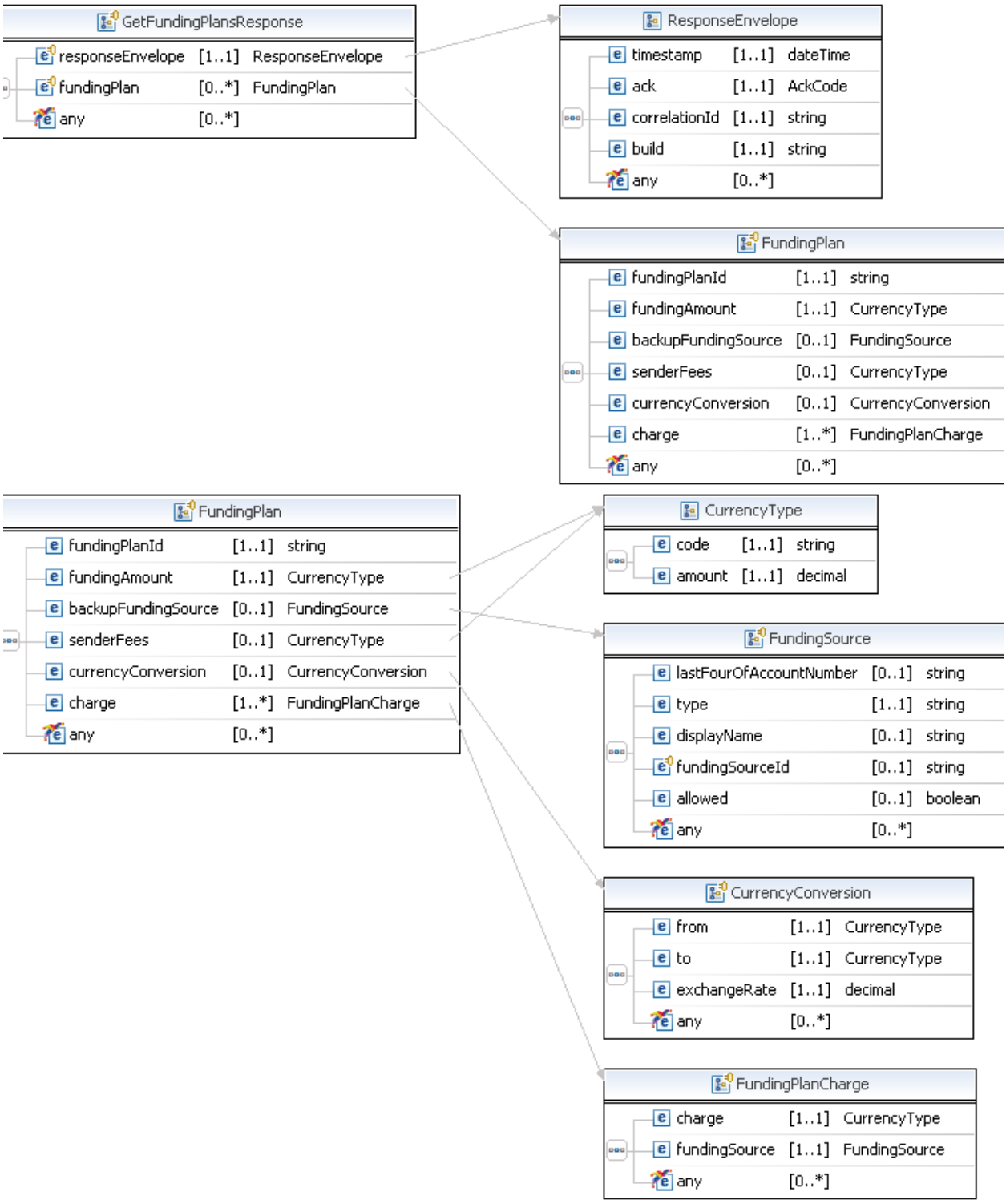
GetFundingPlansRequest Fields

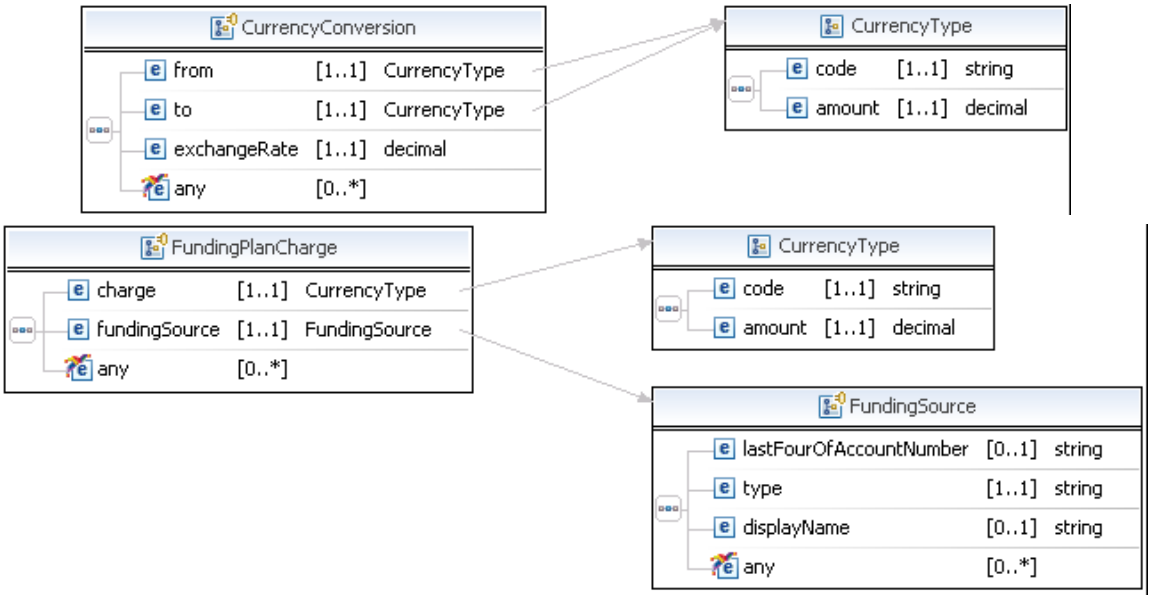
Field	Description
payKey	xs:string *(Required)* The key used to create the payment whose funding sources you want to determine.
requestEnvelope	common:requestEnvelope *(Required)* Information common to each API operation, such as the language in which an error message is returned.

RequestEnvelope Fields

Field	Description
detailLevel	common:DetailLevelCode *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is ReturnAll, which provides the maximum level of detail.
errorLanguage	xs:string *(Required)* RFC 3066 language in which error messages are returned; by default it is en_US, which is the only language currently supported.

GetFundingPlansResponse Message


GetFundingPlansResponse
responseEnvelope [1..1] ResponseEnvelope
fundingPlan [0..*] FundingPlan
any [0..*]
ResponseEnvelope
timestamp [1..1] dateTime
ack [1..1] AckCode
correlationId [1..1] string
build [1..1] string
any [0..*]
FundingPlan
fundingPlanId [1..1] string
fundingAmount [1..1] CurrencyType
backupFundingSource [0..1] FundingSource
senderFees [0..1] CurrencyType
currencyConversion [0..1] CurrencyConversion
charge [1..*] FundingPlanCharge
any [0..*]
FundingPlan
fundingPlanId [1..1] string
fundingAmount [1..1] CurrencyType
backupFundingSource [0..1] FundingSource
senderFees [0..1] CurrencyType
currencyConversion [0..1] CurrencyConversion
charge [1..*] FundingPlanCharge
any [0..*]
CurrencyType
code [1..1] string
amount [1..1] decimal
FundingSource
lastFourOfAccountNumber [0..1] string
type [1..1] string
displayName [0..1] string
fundingSourceId [0..1] string
allowed [0..1] boolean
any [0..*]
CurrencyConversion
from [1..1] CurrencyType
to [1..1] CurrencyType
exchangeRate [1..1] decimal
any [0..*]
FundingPlanCharge
charge [1..1] CurrencyType
fundingSource [1..1] FundingSource
any [0..*]



GetFundingPlansResponse Fields

Field	Description
`FundingPlan`	`ap:FundingPlan` One or more available funding sources for this payment.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

FundingPlan Fields

Field	Description
`fundingPlanId`	`xs:string` ID for this funding plan.
`fundingAmount`	`common:CurrencyType` Amount of available funding.
`backupFundingSource`	`ap:FundingSource` Backup funding source.
`senderFees`	`common:CurrencyType` Fees to be paid by the sender.

Field	Description
currencyConversion	ap:CurrencyConversion The currency conversion associated with the funding plan.
charge	ap:FundingPlanCharge The amount to be charged to this funding plan.

CurrencyType Fields

Field	Description
amount	xs:decimal The converted amount.

Field	Description
code	xs:string The currency code for the converted amount. Possible values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD

FundingSource Fields

Field	Description
lastFourOfAccountNumber	xs:string Last 4 digits or characters in account number.

Field	Description
type	xs:string Type of funding source, which is one of the following values:. • UNDEFINED • BALANCE • BANK_INSTANT • BANK_DELAYED • CREDITCARD • DEBITCARD • ACCOUNTS_RECEIVABLE
displayName	xs:string Display name of funding source.
fundingSourceId	xs:string Funding source ID.
allowed	xs:boolean Whether the funding source is allowed for this payment: • true – You can use this funding source for the payment • false – You cannot use this funding source (default)

CurrencyConversion Fields

Field	Description
from	ap:CurrencyType The currency to be converted.
to	ap:CurrencyType The currency resulting from the conversion.
exchangeRate	xs:decimal The exchange rate for the from currency to the to currency.

ResponseEnvelope Fields

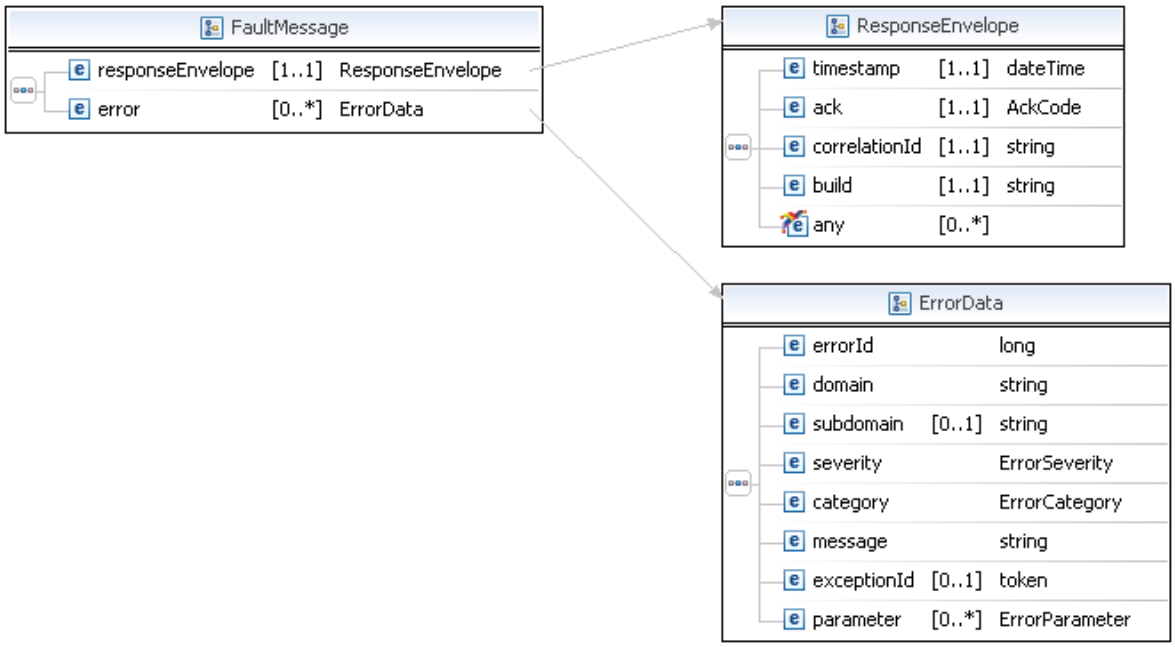
Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.

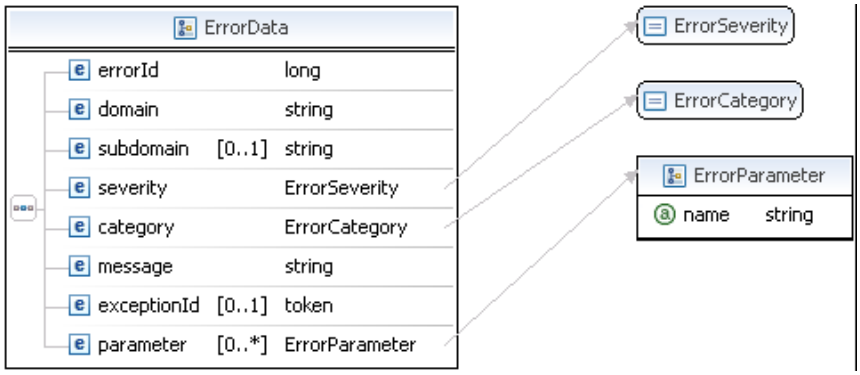
Field	Description
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The `PPFaultMessage` returns `ErrorData` and the `ResponseEnvelope` information to your application if an error occurs.





FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.

Field	Description
severity	common:ErrorSeverity The severity of the error encountered. Possible values are: • Error – Processing of the request was interrupted • Warning – Processing of the request was completed
subdomain	This field is not used.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

GetFundingPlan Errors

Code	Message	Additional Information
520002	Internal error	

Code	Message	Additional Information
520003	Authentication failed. API credentials are incorrect.	Since 1.6.0
540031	Operation is not permitted because the credentials do not match those of the initial operation	Since 1.6.0
550001	• User is not allowed to perform this action. • You are not allowed to get the funding plans for this key • You do not have permission to get these funding plan details	Since 1.6.0
580001	Invalid request: <value>	Since 1.6.0
580022	• Invalid request parameter: <value> • Invalid request parameter: payKey with value <value>	Since 1.6.0
580027	This parameter is not supported	Since 1.6.0
580029	A pay key, transaction ID, or tracking ID is missing	Since 1.6.0
580030	This transaction cannot be processed at this time. Please try again later.	Since 1.3.0

14 GetShippingAddresses API Operation

Use the `GetShippingAddresses` API operation to obtain the selected shipping address. You must have created the payment payKey that identifies the account holder whose shipping address you want to obtain, or be the primary receiver of the payment or one of the parallel receivers of the payment. The shipping address is available only if it was provided during the embedded payment flow.

GetShippingAddressesRequest Message

GetShippingAddressesRequest Fields

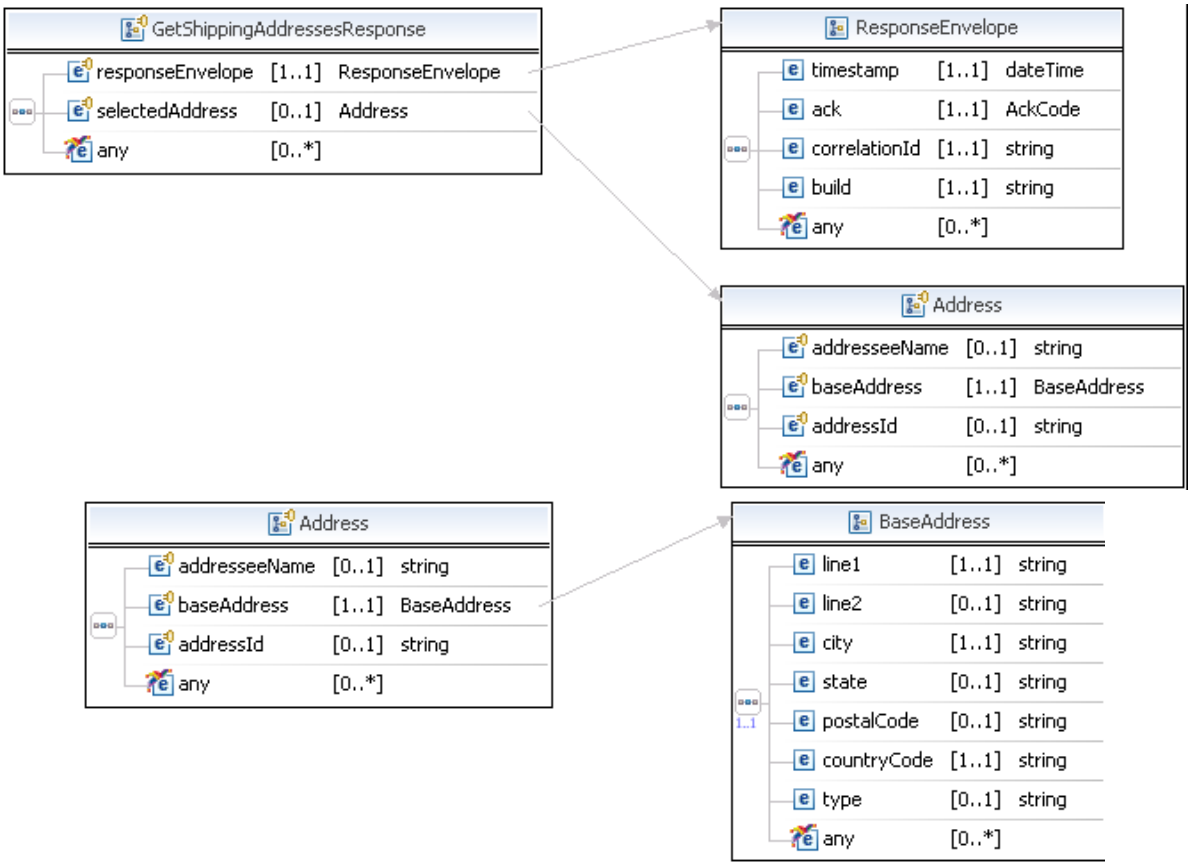
You must specify a value for the `key` field, which uses the payKey to identify the account holder.

Field	Description
`key`	`xs:string` *(Required)* The payment paykey that identifies the account holder for whom you want to obtain the selected shipping address. **NOTE:** Shipping information can only be retrieved through the payment payKey; not through the preapprovalKey.
`requestEnvelope`	`common:requestEnvelope` *(Required)* Information common to each API operation, such as the language in which an error message is returned.

RequestEnvelope Fields

Field	Description
`detailLevel`	`common:DetailLevelCode` *(Optional)* Level of detail required by the client application pertaining to a particular data component. The detail level is specified as a detail level code, which has all the enumerated values of the detail level for the component. By default, the detail level code is `ReturnAll`, which provides the maximum level of detail.
`errorLanguage`	`xs:string` *(Required)* RFC 3066 language in which error messages are returned; by default it is `en_US`, which is the only language currently supported.

GetShippingAddressesResponse Message



GetShippingAddressesResponse Fields

Field	Description
`selectedAddress`	`ap:Address` Any number of available shipping addresses. The first address selected in the embedded payment flow.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

Address Fields

Field	Description
`addresseeName`	`xs:string` The name associated with the address.
`baseAddress`	`common:BaseAddress` Street address.
`addressId`	`xs:string` The ID associated with the address.

BaseAddress Fields

Field	Description
`city`	`xs:string` The city of the address.
`countryCode`	`xs:string` The country code of the address.
`line1`	`xs:string` The first line of the address.
`line2`	`xs:string` An second line of the street address.
`postalCode`	`xs:string` The postal code of the address.
`state`	`xs:string` The state for the address

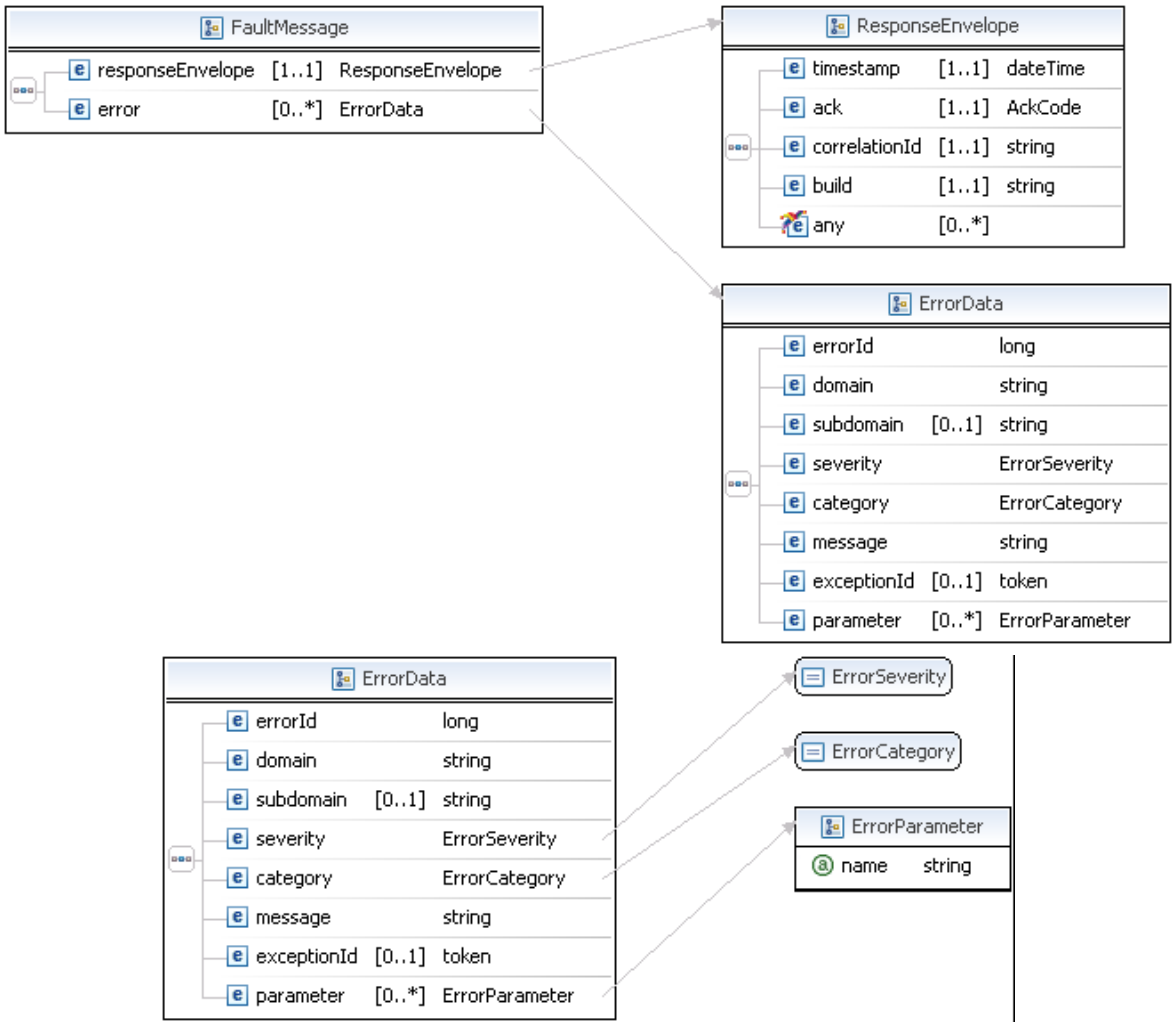
Field	Description
type	xs:string. The type of address.

ResponseEnvelope Fields

Field	Description
ack	common:AckCode Acknowledgement code. It is one of the following values: • Success – The operation completed successfully. • Failure – The operation failed. • SuccessWithWarning – The operation completed successfully; however, there is a warning message. • FailureWithWarning – The operation failed with a warning message.
build	xs:string Build number. It is used only by PayPal Merchant Technical Support.
correlationId	xs:string Correlation identifier. It is a 13-character, alphanumeric string (for example, db87c705a910e) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.
timestamp	xs:datetime Date on which the response was sent, for example: 2012-04-02T22:33:35.774-07:00 **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

PPFault Message

The PPFaultMessage returns ErrorData and the ResponseEnvelope information to your application if an error occurs.



FaultMessage Fields

Field	Description
`error`	`common:ErrorData` Detailed error information.
`responseEnvelope`	`common:ResponseEnvelope` Common response information, including a timestamp and the response acknowledgement status.

ErrorData Fields

Field	Description
`category`	`common:ErrorCategory` The location where the error occurred. Possible values are: • `System` – The system encountered errors; try again • `Application` – The application encountered errors; try again • `Request` – The request was incorrect
`domain`	`xs:string` The domain to which this service belongs.
`errorId`	`xs:long` A 6-digit number that uniquely identifies a particular error.
`exceptionID`	This field is not used.
`message`	`xs:string` A description of the error.
`parameter`	`common:ErrorParameter` Represents contextual information about the error.
`severity`	`common:ErrorSeverity` The severity of the error encountered. Possible values are: • `Error` – Processing of the request was interrupted • `Warning` – Processing of the request was completed
`subdomain`	This field is not used.

ResponseEnvelope Fields

Field	Description
`ack`	`common:AckCode` Acknowledgement code. It is one of the following values: • `Success` – The operation completed successfully. • `Failure` – The operation failed. • `SuccessWithWarning` – The operation completed successfully; however, there is a warning message. • `FailureWithWarning` – The operation failed with a warning message.
`build`	`xs:string` Build number. It is used only by PayPal Merchant Technical Support.

Field	Description
`correlationId`	`xs:string` Correlation identifier. It is a 13-character, alphanumeric string (for example, `db87c705a910e`) that is used only by PayPal Merchant Technical Support. **NOTE:** You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

`timestamp`	`xs:datetime`
	Date on which the response was sent, for example:
	`2012-04-02T22:33:35.774-07:00`
	NOTE: You must log and store this data for every response you receive. PayPal Technical Support uses the information to assist with reported issues.

GetShippingAddresses Errors

Code	Message	Additional Information
520002	Internal error	
520003	Authentication failed. API credentials are incorrect.	Since 1.6.0
540031	Operation is not permitted because the credentials do not match those of the initial operation	Since 1.6.0
550001	• User is not allowed to perform this action. • You are not allowed to get the addresses for this key	Since 1.6.0
580001	Invalid request: <value>	Since 1.6.0
580022	• Invalid request parameter: <value> • Invalid request parameter: payKey with value <value> • No sender is associated with this key	Since 1.6.0
580029	A pay key, transaction ID, or tracking ID is missing	Since 1.6.0
580030	This transaction cannot be processed at this time. Please try again later.	Since 1.3.0

15 Adaptive Payment Commands and Redirects

The web flows associated with Adaptive Payments may require you to redirect a sender's browser to PayPal. When you redirect to the `webscr` endpoint, you must send a command that identifies the flow and associated parameters, or in the case of the embedded payment flow, you must redirect to a specific endpoint for the flow.

Adaptive Payment Commands

When redirecting to `https://www.paypal.com/webscr`, you must specify a command that identifies the flow.

Command	Description
`_ap-payment`	Invokes PayPal login to approve a payment using Adaptive Payments
`_ap-preapproval`	Invokes PayPal login to set up a preapproval using Adaptive Payments

_ap-payment Command

The sender can log in to paypal.com and approve a payment when you redirect the sender's browser to paypal.com with the `_ap-payment` command.

Redirect to PayPal using the _ap-payment command

`https://www.paypal.com/webscr?cmd=_ap-payment&paykey=`*value*

Parameter	Description
cmd	*(Required)* `_ap-payment`
paykey	*(Required)* The payment key from the `Pay` response message; for example, `AP-12345678901234567`.

Response to _ap-payment command from PayPal

PayPal redirects to the return URL you set in the `Pay` request.

NOTE: Any additional parameters you specify in the URL are not returned.

_ap-preapproval Command

The sender can log in to paypal.com and approve a preapproval request when you redirect the sender's browser to PayPal using the `_ap-preapproval` command.

Redirect to PayPal using the _ap-preapproval command

```
https://www.paypal.com/webscr?cmd=_ap-preapproval&preapprovalkey=value
```

Parameter	Description
cmd	*(Required)* `_ap-preapproval`
preapprovalkey	*(Required)* The preapproval key from the `Preapproval` response message; for example, `PA-12345678901234567`.

Response to _ap-preapproval command from PayPal

PayPal redirects to the return URL you set in the `Preapproval` request.

NOTE: Any additional parameters you specify in the URL are not returned.

Embedded Payment Flow Redirect

The sender of a payment can log in to paypal.com and approve a payment using the embedded payment flow. Your web pages must use PayPal-provided JavaScript to launch the flow.

Redirect to PayPal

```
https://www.paypal.com/webapps/adaptivepayment/flow/pay?
        paykey=value&preapprovalkey=value
```

Parameter	Description
paykey	*(Required)* The payment key from the `Pay` response message; for example, `AP-12345678901234567`.
preapprovalkey	*(Optional)* The preapproval key from the `Preapproval` response message, which enables the **Preapprove future payments** option; for example, `PA-12345678901234567`.

Response from PayPal

PayPal redirects to the return URL you set in the `Pay` request.

NOTE: Any additional parameters you specify in the URL are not returned.

JavaScript Functions for Embedded Payments

You must include `https://www.paypalobjects.com/js/external/dg.js` for a lightbox or `https://www.paypalobjects.com/js/external/apdg.js` for a minibrowser on any page that invokes or terminates the embedded payment flow. The JavaScript functions in `dg.js` and `apdg.js` set up and control the PayPal login and payment experience.

Function and Signature	Description
`PAYPAL.apps.DGFlow` `= function (trigger: null, expType: null)`	Sets up the embedded payment flow, where `trigger` is one or more HTML ids of events that trigger the flow and `expType` is one of the following experience type requests: • `light` requests that the embedded payment flow displays in a lightbox (default) • `mini` requests that the embedded payment flow displays in a minibrowser PayPal handles other values, including misspellings, as `light`. Opening a minibrowser in a lightbox causes errors during the flow. **NOTE:** PayPal does not honor the lightbox experience request if the sender is required to log into PayPal.
`setTrigger``: function (`*id*`)`	Add a trigger for the flow, where `id` is the HTML id of the event that triggers the flow. The flow must first be set up by calling `PAYPAL.apps.DGFlow`.
`startFlow``: function (`*url*`)`	Explicitly start the embedded payment flow, where `url` is the URL that triggers the flow. **NOTE:** You are only required to explicitly start the flow when there is no triggering event; for example, when you want to associate the embedded payment flow with playing a Flash movie.
`closeFlow``: function ()`	Close the minibrowser or lightbox associated with the flow. Use this function after the sender completes or cancels the payment. **NOTE:** In addition to calling this function, you must explicitly close the PayPal window using the `close()` JavaScript function.
`isOpen``: function ()`	Determine whether the minibrowser or lightbox is still open.

16 Instant Payment Notifications

PayPal sends an IPN message in response to a payment, a refund, creation of a preapproval, and cancellation of a preapproval. Your IPN message handler must be set up in the way described in the *Instant Payment Notification Guide*.

Specifically, it must guard against spoofing and it must be able to handle duplicate messages, which can occur, for example, if your application is both the caller of the Pay API and a receiver of the payment.

The IPN messages listed here do not use the IPN message handler that you can specify in the Profile. IPN messages are only sent to the notification URL specified in the Adaptive Payment API operation's request.

IMPORTANT: IPN variable names, such as `transaction[0].status` are strings and you must manipulate them as strings. Some languages, such as PHP and others, may attempt to interpret the non-alphanumeric characters found in the variable name. To avoid misinterpretation of these strings, you should always to use the raw HTTP request string and process it yourself rather than rely on the support provided by the language to obtain request parameters.

Pay Message Variables

PayPal generates an IPN message that contains information about the pay request or payment in response to the Adaptive Payments `Pay` and `ExecutePayment` API operations.

Variable	Description
`transaction_type`	The type of transaction. Possible values are: • `Adaptive Payment PAY` This notification occurs when is a payment is made due to a `PayRequest`. The variables for the `Adaptive Payment Pay` notification are similar to the `PaymentDetailsResponse` fields. • `Adjustment` This can be for a chargeback, reversal, or refund; check the `reason_code` to see which it is.

Variable	Description
status	The status of the payment. Possible values are: • CREATED – The payment request was received; funds will be transferred once the payment is approved • COMPLETED – The payment was successful • INCOMPLETE – Some transfers succeeded and some failed for a parallel payment or, for a delayed chained payment, secondary receivers have not been paid • ERROR – The payment failed and all attempted transfers failed or all completed transfers were successfully reversed • REVERSALERROR – One or more transfers failed when attempting to reverse a payment • PROCESSING – The payment is in progress • PENDING – The payment is awaiting processing
sender_email	Sender's email address.
action_type	Whether the Pay API is used with or without the SetPaymentOptions and ExecutePayment API operations. Possible values are: • PAY – If you are not using the SetPaymentOptions and ExecutePayment API operations • CREATE – If you are using the SetPaymentOptions and ExecutePayment API operations
payment_request_date	The date on which the payment request was initiated.
reverse_all_parallel_payments_on_error	Whether the payment request specified to reverse parallel payments if an error occurs. Possible values are: • true – Each parallel payment is reversed if an error occurs • false – Only incomplete payments are reversed (default)
transaction[n].id	The transaction ID, where [n] is a number from 0 to 5. For simple, single receiver payments, this number will be 0. Numbers larger than 0 indicate the payment to a particular receiver in chained and parallel payments.
transaction[n].status	The transaction status, where [n] is a number from 0 to 5. For simple single-receiver payments, this number will be 0. Numbers larger than 0 indicate the payment to a particular receiver in chained and parallel payments. Possible values are: • Completed • Pending • Refunded
transaction[n].id_for_sender	The transaction ID for the sender, where [n] is a number from 0 to 5. For simple, single receiver payments, this number will be 0. Numbers larger than 0 indicate the payment to a particular receiver in chained and parallel payments.

Variable	Description
transaction[n].status_for _sender_txn	The transaction status, where [n] is a number from 0 to 5. For simple single-receiver payments, this number will be 0. Numbers larger than 0 indicate the payment to a particular receiver in chained and parallel payments. Possible values are: • COMPLETED – The sender's transaction has completed • PENDING – The transaction is awaiting further processing • CREATED – The payment request was received; funds will be transferred once approval is received • PARTIALLY_REFUNDED– Transaction was partially refunded • DENIED – The transaction was rejected by the receiver • PROCESSING – The transaction is in progress • REVERSED – The payment was returned to the sender • REFUNDED – The payment was refunded • FAILED – The payment failed
transaction[n].refund_id	The identification number for the refund
transaction[n].refund_amount	The amount that was refunded.
transaction[n].refund_account_charged	The email address of the debit account of the refund.
transaction[n].receiver	The receiver's email address for the transaction
transaction[n].invoiceId	The invoice number for this transaction
transaction[n].amount	The payment amount of the transaction
transaction[n].is_primary_receiver	Whether there is a primary receiver for this transaction, which indicates whether the transaction is a chained payment. Possible values are: • true – There is a primary receiver (chained payment) • false – There is no primary receiver (simple or parallel payment)
return_url	The URL to which the sender's browser is redirected after approving a payment on paypal.com. Use the pay key to identify the payment as follows: payKey=${payKey}.
cancel_url	The URL to which the sender's browser is redirected if the sender cancels the approval for a payment on paypal.com. Use the pay key to identify the payment as follows: payKey=${payKey}.
ipn_notification_url	The URL to which all IPN messages for this payment are sent.
pay_key	The pay key that identifies this payment. This is a token that is assigned by the Pay API after a PayRequest message is received and can be used in other Adaptive Payments APIs as well as the cancelURL and returnURL to identify this payment. The pay key is valid for 3 hours.
memo	A note associated with the payment.

Variable	Description
fees_payer	The payer of PayPal fees. Possible values are: • SENDER – Sender pays all fees (for personal, implicit simple/parallel payments; do not use for chained or unilateral payments) • PRIMARYRECEIVER – Primary receiver pays all fees (chained payments only) • EACHRECEIVER – Each receiver pays their own fee (default, personal and unilateral payments) • SECONDARYONLY – Secondary receivers pay all fees (use only for chained payments with one secondary receiver)
trackingId	The tracking ID that was specified for this payment in the PaymentDetailsRequest message.
preapproval_key	The preapproval key returned after a PreapprovalRequest, or the preapproval key that identifies the preapproval key sent with a PayRequest.
reason_code	Whether this transaction is a chargeback, partial, or reversal. Possible values are: • Chargeback Settlement – Transaction is a chargeback • Admin reversal – Transaction was reversed by PayPal administrators • Refund – Transaction was partially or fully refunded

Preapproval Message Variables

PayPal generates an IPN message that contains information about a preapproval in response to the Adaptive Payments Preapproval API operation.

Variable	Description
transaction_type	The type of transaction. For a preapproval, this variable returns Adaptive Payment Preapproval. **NOTE:** If this variable is set to Adaptive Payment Pay or Adjustment, refer to the Pay Message Variable section.
preapproval_key	The preapproval key returned after a PreapprovalRequest.
approved	Whether the preapproval request was approved. Possible values are: • true – The preapproval was approved • false – The preapproval was denied
cancel_url	The URL to which the sender's browser is redirected if the sender decides to cancel the preapproval as requested. Use the preapproval key to identify the payment as follows: preapprovalKey=${preapprovalKey}
current_number_of_payments	The current number of payments made for this preapproval.

Variable	Description
current_total_amount_of_all_payments	The current total of payments made for this preapproval.
current_period_attempts	The current number of attempts this period for this preapproval.
currencyCode	The currency code. Possible values are: • Australian Dollar – AUD • Brazilian Real – BRL **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. • Canadian Dollar – CAD • Czech Koruna – CZK • Danish Krone – DKK • Euro – EUR • Hong Kong Dollar – HKD • Hungarian Forint – HUF • Israeli New Sheqel – ILS • Japanese Yen – JPY • Malaysian Ringgit – MYR **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. • Mexican Peso – MXN • Norwegian Krone – NOK • New Zealand Dollar – NZD • Philippine Peso – PHP • Polish Zloty – PLN • Pound Sterling – GBP • Singapore Dollar – SGD • Swedish Krona – SEK • Swiss Franc – CHF • Taiwan New Dollar – TWD • Thai Baht – THB • Turkish Lira – TRY **NOTE:** The Turkish Lira is supported as a payment currency and currency balance only for Turkish PayPal accounts. • U.S. Dollar – USD
date_of_month	The day of the month on which a monthly payment is to be made. A number between 1 and 31 indicates the day of the month. A value of 0 indicates that the payment can be made on any day.

Variable	Description
day_of_week	The day of the week that a weekly payment is to be made. Possible values are: • NO_DAY_SPECIFIED • SUNDAY • MONDAY • TUESDAY • WEDNESDAY • THURSDAY • FRIDAY • SATURDAY
starting_date	First date for which the preapproval is valid.
ending_date	Last date for which the preapproval is valid. Time is currently not supported.
max_total_amount_of_all_payments	The preapproved maximum total amount of all payments.
max_amount_per_payment	The preapproved maximum amount of all payments.
max_number_of_payments	The maximum number of payments that is preapproved.
payment_period	The payment period. Possible values are: • NO_PERIOD_SPECIFIED • DAILY • WEEKLY • BIWEEKLY • SEMIMONTHLY • MONTHLY • ANNUALLY
pin_type	Whether a personal identification number (PIN) is required. It is one of the following values: • NOT_REQUIRED – A PIN is not required • REQUIRED – A PIN is required
sender_email	The sender's email address.

Older Versions of the Adaptive Payments API

Older versions of the Adaptive Payments API can still be used; however, PayPal recommends using the latest version whenever possible. Unless specifically specified, API operations default to the latest version.

1.8.1 Features

Version 1.8.1 of the Adaptive Payments API adds new fields related to tax ID details for Brazil (shown below). Also a new BANK_MANAGED_WITHDRAWAL value has been added to the `paymentType` field.

NOTE: Changes to API operations are backward-compatible.

SenderIdentifier Fields

Field	Description
`useCredentials`	`xs:boolean` If `true`, credentials identify the sender; default is `false`.
`taxIdDetails`	`ap:TaxIdDetails` Tax ID details (For Brazil only)

TaxIdDetails Fields

Field	Description
`taxId`	`xs:string` Tax ID of the Merchant (For Brazil only)
`taxIdType`	`xs:string` Type of the Tax ID. Either CPF or CNPJ (For Brazil only)

1.8.0 Features

Version 1.8.0 of the Adaptive Payments API introduces new API operation fields for Convert Currency and changes to JavaScript functions.

NOTE: Changes to API operations are backward-compatible.

Changes to SenderOptions

The change to `SenderOptions` request fields corresponds to version 85.0 of the PayPal API:

Field	Description
`referrerCode`	New field. `xs:string` *(Optional)* A code that identifies the partner associated with this transaction. Maximum length: 32 characters.

Changes to ConvertCurrencyRequest Fields for Version 1.8.0

Field	Description
`conversionType`	`xs:string` *(Optional)* The conversion type allows you to determine the converted amounts for a PayPal user in different currency conversion scenarios. Possible values are: • `SENDER_SIDE` – Pass this conversion type when you convert amounts for a PayPal user who is sending a payment in a different currency from what he/she holds in PayPal. • `RECEIVER_SIDE` – Pass this conversion type when you when you convert amounts for a PayPal user who is accepting a payment in a currency that he/she does not hold and who wants to convert the received payment. • `BALANCE_TRANSFER` – Pass this conversion type when you convert amounts for a PayPal user who is converting a balance in one currency to a balance in a different currency on his/her profile.
`countryCode`	`xs:string` *(Optional)*The two-character ISO code for the country where the function is supposed to happen. The default value is US.

Changes to JavaScript Functions for Embedded Payments

You must include `https://www.paypalobjects.com/js/external/dg.js` for a lightbox or `https://www.paypalobjects.com/js/external/apdg.js` for a minibrowser on any page that invokes or terminates the embedded payment flow.

Function and Signature	Description
PAYPAL.apps.DGFlow = function (trigger: null, expType: null)	New expType: • mini requests that the embedded payment flow displays in a minibrowser

New JavaScript Function for Version 1.8.0: AutoRedirectOnDone

After the payment sender using a Merchant site completes a transaction, the **Thank you for using PayPal** page appears, also known as the "Done" page. Previously, from within this page, the Merchant allowed the sender to close the page or to navigate back to a Merchant-specified URL by clicking a button. Now, by using the AutoRedirectOnDone option, the Merchant can automatically redirect the sender from the **Thank you for using PayPal** page to a specified URL, which opens after a waiting period of about five seconds.

The following code sample shows how this optional function would be implemented in a minibrowser, a lightbox, or an iframe.

NOTE: Do not supply a value for autoredirectondone. Be aware that you do not necessarily have to use the button images provided in the sample code.

Minibrowser code

```
<form id="mini-form"
action="https://www.paypal.com/webapps/adaptivepayment/flow/pay"
target="PPDGFrame">
       <br><fieldset>
            <b><legend>Pay - Mini Browser</legend></b>
            <input id="type" name="expType" value="mini" type="hidden">
            <input id="type" name="autoredirectondone" type="hidden">
            <label for="paykey">PayKey</label>
               <input id="paykey" name="payKey"  size="25" type="text">
               <input type="image" id="miniBrowserBtn"
                  value="Pay with PayPal mini"
                  src="https://www.paypal.com/en_US/i/btn/
                  x-click-but06.gif">
        </fieldset>
</form>
```

Lightbox code

```
<form id="lightbox-form"
action="https://www.paypal.com/webapps/adaptivepayment/flow/pay"
target="PPDGFrame">
        <fieldset>
         <b><legend>Pay - Lightbox</legend></b>
         <input id="type" name="expType" value="lightbox" type="hidden">
         <input id="type" name="autoredirectondone" type="hidden">
               <label for="paykey">PayKey</label>
               <input id="paykey" name="payKey" size="25" type="text">
               <input type="submit" id="submitBtn" value="Pay with light"
                src="https://www.paypal.com/en_US/i/btn/x-click-but06.gif">
         </fieldset>
</form>
```

iFrame Code

```
<iframe id="PPDGFrame" name="PPDGFrame"
   src="https://www.paypal.com/webapps/adaptivepayment/flow
   /pay?expType=light&paykey=AP-SAMPLEPAYKEY&autoredirectondone=
"scrolling="no" frameborder="0" style="border: none; height: 550px;
width: 385px;" allowtransparency="true"> </iframe>
```

1.7.0 Features

Version 1.7.0 of the Adaptive Payments API introduces a new API operation and changes several Adaptive Payments API operations for mobile providers.

NOTE: Changes to API operations are backward-compatible.

New API Operations for Version 1.7.0

API Operation	Description
`GetAllowedFundingSources`	Determines the funding sources associated with a preapproval.

Changes to ConfirmPreapprovalRequest Fields for Version 1.7.0

Field	Description
`fundingSourceId`	New field: `xs:string` *(Optional)* Funding source ID.

Changes to JavaScript Functions for Embedded Payments

You must include `https://www.paypalobjects.com/js/external/dg.js` for a lightbox or `https://www.paypalobjects.com/js/external/apdg.js` for a minibrowser on any page that invokes or terminates the embedded payment flow.

Function and Signature	Description
`PAYPAL.apps.DGFlow` `= function (trigger: null, expType: null)`	New `expType`: • `mini` requests that the embedded payment flow displays in a minibrowser

1.6.0 Features

Version 1.6.0 of the Adaptive Payments API introduces new API operations, as well as changes to several Adaptive Payments API operations and additional error messages.

NOTE: Changes to API operations are backward-compatible.

New API Operations for Version 1.6.0

API Operation	Description
`GetFundingPlans`	Determines the funding sources that are available for a specified payment
`GetShippingAddresses`	Obtains the selected shipping address
`GetAvailableShippingAddresses`	Obtains available shipping addresses
`ConfirmPreapproval`	Confirms the specified preapproval

Changes to PayRequest Fields for Version 1.6.0

Field	Description
`sender`	New field: `ap:SenderIdentifier` *(Optional)* Sender's identifying information.

Changes to PayResponse Fields for Version 1.6.0

Field	Description
defaultFundingPlan	New field: ap:FundingPlan Default funding plan.

Changes to ExecutePaymentRequest Fields for Version 1.6.0

Field	Description
fundingPlanId	New field: xs:string *(Optional)* The ID of the funding plan from which to make this payment.

Changes to GetPaymentOptionsResponse Fields for Version 1.6.0

Field	Description
shippingAddressId	New field: xs:string Sender's shipping address ID.

Changes to SetPaymentOptionsRequest Fields for Version 1.6.0

Field	Description
shippingAddressId	New field: xs:string *(Optional)* Sender's shipping address ID.

Changes to PreapprovalRequest Fields for Version 1.6.0

Field	Description
endingDate	xs:dateTime *(Optional)* Last date for which the preapproval is valid. It cannot be later than one year from the starting date. You must specify a value unless you have specific permission from PayPal.

Field	Description
maxNumberOfPaymentsPerPeriod	xs:int *(Optional)* The preapproved maximum number of all payments per period. You must specify a value unless you have specific permission from PayPal.

Changes to Address Structure for Version 1.6.0

Field	Description
addressId	New field: xs:string The ID associated with the address.

Changes to DisplayOptions Structure for Version 1.6.0

Field	Description
headerImageUrl	New field: xs:string *(Optional)* The URL of the image that displays in the header of a payment page. If set, it overrides the header image URL specified in your account's Profile. The image dimensions are 750 pixels high x 90 pixels wide.
businessName	New field: xs:string *(Optional)* The business name to display.

New CurrencyConversion Structure for Version 1.6.0

Field	Description
from	ap:CurrencyType The currency to be converted.
to	ap:CurrencyType The currency resulting from the conversion.
exchangeRate	xs:decimal The exchange rate for the from currency to the to currency.

New InvoiceData Structure for Version 1.6.0

Field	Description
item	ap:InvoiceItem *(Optional)* Any number of items associated with the payment.
totalTax	xs:decimal *(Optional)* Total tax associated with the payment.
totalShipping	xs:decimal *(Optional)* Total shipping charge associated with the payment.

New InvoiceItem Structure for Version 1.6.0

Field	Description
name	xs:string *(Optional)* Name of item.
identifier	xs:string *(Optional)* External reference to item or item ID.
price	xs:decimal *(Optional)* Price of item.

New SenderOptions Structure for Version 1.6.0

Field	Description
requireShippingAddressSelection	xs:boolean *(Optional)* If true, require the sender to select a shipping address during the payment flow; default is false.

New SenderIdentifier Structure for Version 1.6.0

Field	Description
useCredentials	xs:boolean *(Optional)* If true, use credentials to identify the sender; default is false.

New AccountIdentifier Structure for Version 1.6.0

Field	Description
email	xs:string *(Optional)* Sender's email address. Maximum length: 127 characters
phone	common:PhoneNumberType *(Optional)* Sender's phone number.

New ReceiverOptions Structure for Version 1.6.0

Field	Description
description	xs:string *(Optional)* A description you want to associate with the payment.
customId	xs:string *(Optional)* An external reference or identifier you want to associate with the payment.
invoiceData	ap:InvoiceData *(Optional)* Item information for the payment, which could appear on an invoice.
receiver	ap:ReceiverIdentifier *(Optional)* Receiver ID associated with these options.

New ReceiverIdentifier Structure for Version 1.6.0

Field	Description
base	common:AccountIdentifier Account identifier.

Additional Error Messages for Version 1.6.0

Code	Message
520003	Authentication failed. API credentials are incorrect.
520003	• The caller's credential doesn't have permission for the fee payer <value> • The caller's credential doesn't have permission for the payment type <value> • Authentication failed. API credentials are incorrect.

Code	Message
540031	You do not have permission to execute this payment Operation is not permitted because the credentials do not match those of the initial operation
540031	You do not have permission to get these preapproval details
540031	You do not have permission to get these payment options Operation is not permitted because the credentials do not match those of the initial operation
540031	You do not have permission to set these payment options Operation is not permitted because the credentials do not match those of the initial operation
540031	Operation is not permitted because the credentials do not match those of the initial operation
540031	You do not have permission to get these payment details
550001	• You are not allowed to confirm the preapproval for this key • User is not allowed to perform this action.
550001	You do not have permission to execute this payment implicitly
550001	• User is not allowed to perform this action. • You are not allowed to get the funding plans for this key • You do not have permission to get these funding plan details
550001	• User is not allowed to perform this action. • You are not allowed to get the addresses for this key
580001	• Invalid request: both email and phone cannot be set <value> <value> <value> • Invalid request: neither email nor phone is set <value> • To use currency <value>, currency must be held by sender and all receivers • Invalid request: More than one field cannot be used to specify a sender • Invalid request: <value>
580001	Invalid request: <value>
580022	• Invalid request parameter: <value> • The clientDetails parameter is missing • The PIN is invalid
580022	• Invalid request parameter: <value> • The clientDetails parameter is missing • Invalid request parameter: payKey with value <value> • payKey <value> is expired • Invalid request parameter: <value> with <value> • No sender is associated with this key • The funding plan data is not present • payKey <value> has already been used to make a payment

Code	Message
580022	• Invalid request parameter: <value> • Invalid Request Parameter: institutionId with value <value> • The institution id <value> is not activated • The clientDetails parameter is missing • Invalid request parameter: payKey with value <value> • Invalid Phone Country Code <value> • Invalid Country Code <value> for Receiver • Invalid Receiver Phone Type <value> <value> <value> • Invalid Phone Format <value> <value> • Invalid Phone Number <value> <value> • Invalid Receiver Phone Number <value> <value> • Invalid Area or Exchange Code <value> <value> • <value> has already been set and cannot be set multiple times • Invalid request parameter: <value> with value <value> • Invalid request parameter: <value> <value> is invalid • Invalid request parameter: <value> <value> is duplicated • Invalid request parameter: <value> <value> cannot be negative or zero
580022	• Invalid request parameter: <value> • Invalid request parameter: payKey with value <value>
580022	• Invalid request parameter: <value> • Invalid request parameter: payKey with value <value> • No sender is associated with this key
580022	The funding plan data is not present
580023	• The fee payer <value> cannot be used if a primary receiver is specified • The fee payer <value> can only be used if a primary receiver is specified • The fee payer <value> must have exactly 2 receivers • The reverseAllParallelPaymentOnError field can be true only if a primary receiver is not specified • The fee payer <value> cannot be used if a preapproval key is specified • Invalid request: FundingType cannot be specified when PaymentSubType is specified • Your Application is not approved to use the PaymentSubType: <value> • Your Application is not approved to use the PaymentType: <value> • Invalid request parameter: PaymentType should be the same for all receivers <value> • Inconsistent request parameter for digitalgoods • Invalid request parameter: PaymentSubType should be the same for all receivers <value> • Invalid Request: RECEIVABLE cannot be used with other funding types • For chained payments, receivers must have premier or business accounts
580027	• Currently the system does not accept the currency <value> • Unsupported country code <value> • Currently the system does not accept the currency <value> • The parameter is not supported
580027	This parameter is not supported

Code	Message
580029	A pin is required to confirm this preapproval Missing required request parameter: <value>
580029	Missing required request parameter: <value>
580029	A pay key, transaction ID, or tracking ID is missing
589018	The pay key <value> is invalid
589019	The preapproval key <value> is invalid
589061	The receiver <value> is invalid for this refund

1.5.0 Features

Adaptive Payments version 1.5.0 provides the new features for the APIs, such as the ability to delay a chained payment and to obtain the reason a payment is pending.

In addition to API changes, version 1.5.0 also offers new features that do not affect the APIs, such as guest payments, and ELV support.

Ability to Delay a Chained Payment Feature in Version 1.5.0

You can now set up chained payments to process the transaction leg to the primary receiver, and delay the payment to the secondary receivers. To do this, you use the `Pay or ExecutePayment` call and pass the new parameter `actionType` with `PAY_PRIMARY`.

PaymentDetails Pending Reason Feature in Version 1.5.0

The `PaymentDetailsResponse` now returns the new `pendingReason` parameter. This parameter returns a string describing the reason why a transaction is pending. Possible values are:

- `ADDRESS_CONFIRMATION` – The payment is pending because the sender does not have a confirmed address on file and receiver's Payment Receiving Preferences is set for manually accepting payments or denying each of these payments.
- `ECHECK` – The payment is pending because it was made by an eCheck that has not yet cleared.
- `INTERNATIONAL` – The payment is pending because the receiver holds a non-U.S. account and does not have a withdrawal mechanism. The receiver must manually accept or deny this payment from the Account Overview.
- `MULTI_CURRENCY` – The receiver does not have a balance in the currency sent, and does not have the Payment Receiving Preferences set to automatically convert and accept this payment. Receiver must manually accept or deny this payment from the Account Overview.

- UNILATERAL – The payment is pending because it was made to an email address that is not yet registered or confirmed.
- UPGRADE – The payment is pending because it was made via credit card and the receiver must upgrade the account to a Business account or Premier status to receive the funds. It can also mean that receiver has reached the monthly limit for transactions on the account.
- VERIFY – The payment is pending because the receiver is not yet verified.
- RISK – The payment is pending while it is being reviewed by PayPal for risk.
- OTHER – The payment is pending for a reason other than those listed above. For more information, contact PayPal Customer Service.

1.4.0 Features

Release 1.4.0 provides the new API operations: ExecutePayment, GetPaymentOptions, and SetPaymentOptions. The new API operations work with the actiontype CREATE in the Pay API operation.

New API Operations for Version 1.4.0

New API Operation	Description
ExecutePayment	Execute a payment created with a PayRequest that had the actionType set to CREATE.
GetPaymentOptions	Retrieve payment settings specified with the SetPaymentOptions API operation.
SetPaymentOptions	Set payment options to be executed with the ExecutePayment API operation.

Changes to the Pay API Operation

Changes to PayRequest and PayResponse Message

Field	Description
actionType	New CREATE value: • PAY – Use this option if you are not using the Pay request in combinations with the ExecutePayment request. • CREATE – Use this option to set up the payment instructions with the Pay request and then execute the payment at a later time with the ExecutePayment request.
paymentType	New person-to-person values PERSONAL and CASHADVANCE. Person-to-person payments can only be made under the following conditions: • simple and parallel payments (not chained) • all transactions within the single Pay request must be person-to-person • feesPayer field set to EACHRECEIVER or SENDER. • this feature requires the appropriate permission level from PayPal
Receiver.PhoneNumberType	New field: common:PhoneNumberType Allows you to pass a receiver's mobile phone number, including the country code and extension. **NOTE:** This feature is available for applications with special permission level.

Changes to the PaymentDetails API Operation

Changes to PaymentDetails response messages

Field	Description
actionType	New CREATE value: • PAY – Use this option if you are not using the Pay request in combinations with the ExecutePayment request. • CREATE – Use this option to set up the payment instructions with the Pay request and then execute the payment at a later time with the ExecutePayment request.

Field	Description
paymentType	New person-to-person values PERSONAL and CASHADVANCE. Allowable values are: • GOODS – This is a payment for goods • SERVICE – This is a payment for services (default) • PERSONAL – This is a person-to-person payment • CASHADVANCE – This is a person-to-person payment for a cash advance Person-to-person payments can only be made under the following conditions: • simple and parallel payments (not chained) • all transactions within the single Pay request must be person-to-person • feesPayer field set to EACHRECEIVER or SENDER. • this feature requires the appropriate permission level from PayPal
Receiver.PhoneNumberType	New field: common:PhoneNumberType Allows you to pass a receiver's mobile phone number, including the country code and extension. **NOTE:** This feature requires the appropriate permission level from PayPal.

Changes to the PreapprovalDetails API Operation

Changes to PreapprovalDetailsRequest Message

Field	Description
getBillingAddress	New field: xs:boolean *(Optional)* An option that lets you retrieve a list of billing addresses for the sender. Allowable values are: • true – Includes the billing address in the response • false – Omits the billing address from the response (default) **NOTE:** This field is available only to API callers with advanced permission levels. For information, refer to the section Adaptive Payments Permission Levels.

Changes to the PreapprovalDetails API Operation

Changes to PreapprovalDetailsResponse Message

Field	Description
addresslist	New field: ap:AddressList Returns a list of billing addresses.
AddressList	New field: ap:Address
Address.addresseeName	xs:string The name of the person that is billed.
Address.baseAddress	New field: common:BaseAddress
BaseAddress.city	xs:string The city of the billing address
BaseAddress.countryCode	xs:string The country code for the billing address
BaseAddress.line1	xs:string The streetname for the billing address
BaseAddress.line2	xs:string An extra line for the street address
BaseAddress.postalCode	xs:string The postal code of the billing address

Field	Description
BaseAddress.state	xs:string The state for the billing address
BaseAddress.type	xs:string The type of billing address. Possible values are: • BILLING – This is a billing address

Changes to the Refund API Operation

Changes to RefundRequest and RefundResponse Message

Field	Description
Receiver.PhoneNumberType	New field: common:PhoneNumberType Allows you to pass a receiver's phone number, including the country code and extension. **NOTE:** This feature is available for applications with special permission level.

1.3.0 Features

Version 1.3.0 includes changes to existing APIs as well new APIs: ConvertCurrency, CancelPreapproval. It also includes the new PPFaultMessage.

Overview of Changes for Version 1.3.0

These are the main changes for version 1.3.0:

- **Consumer signups for new PayPal accounts**

  Consumers can now sign up for a PayPal account during the payment preapproval web flow. Consumers can also sign up for a PayPal account during a payment web flow and then complete the payment in the same session.

- **Preapproval cancellations**

  This release includes a new API operation to allow consumers to cancel preapprovals: the CancelPreapproval API operation. In past releases, consumers could only cancel preapprovals through paypal.com. For more information, refer to the CancelPreapproval API Operation section.

- **Getting Foreign Exchange rates for a list of payment amounts**

  This release provides a new API Operation, ConvertCurrency, that lets you convert a given amount in one currency to an equivalent amount in another currency based on the

current Foreign Exchange rate. For more information, refer to the ConvertCurrenty API Operation section.

- **Payments to unregistered PayPal email addresses**

  With the Adaptive Payments `Pay` API operation, you can make payments to a person who does not have a PayPal account. In essence, you can send a payment request that includes an email address for a receiver in which the receiver's email address is not linked to a registered PayPal account. PayPal notifies the receiver at this email address to create an account and claim the payment. Supporting features include email notifications and IPN messages to the sender, receiver, and API caller, plus the ability to cancel unclaimed payments.

HTTP Header Changes

In previous versions, the application ID was passed in the `ClientDetails` element `applicationId`. Version 1.3.0 requires a new header to be passed to identify the application ID. The `ClientDetails.applicationId` element is no longer necessary.

Changes to the Pay API Operation for Version 1.3.0

Changes to PayRequest Message

Field	Description
`ClientDetails.applicationId`	This field is no longer required. The application ID is passed in a new HTTP header. For information, refer to the Specifying Application and Device Information section.
`ClientDetails.ipAddress`	This field is no longer required.
`fundingConstraint`	New field: `ap:FundingConstraint` Specifies a list of allowed funding types for the payment. This list can be in any order. If this field is omitted, the payment can be funded by any funding type that is supported by Adaptive Payments. **NOTE:** This feature is available for applications with special permission level.
`FundingTypeList.fundingTypeInfo`	New field: `ap:FundingTypeInfo` Specifies a list of allowed funding types for the payment. This list can be in any order. If this field is omitted, the payment can be funded by any funding type that is supported by Adaptive Payments. **NOTE:** This feature is available for applications with special permission level.
`fundingType`	New field: `xs:string` Specifies the allowed funding types that can be used as funding sources for the payment. Allowable values are: • `ECHECK` – Electronic check • `BALANCE` – PayPal account balance • `CREDITCARD` – Credit card **NOTE:** This feature is available for applications with special permission level.
`logdefaultShippingAddress`	This field has been removed. Pay requests that use this field are executed, but the `logdefaultShippingAddress` is ignored.
`Receiver.paymentType`	New field: `xs:string` The transaction subtype for the payment. Allowable values are: • `GOODS` – This is a payment for goods • `SERVICE` – This is a payment for services (default)

Field	Description
Receiver.email	xs:string In this release, the receiver's email address does not need to be registered with paypal.com. If so, the receiver cannot claim the payment until a PayPal account has been created for the email address.
senderEmail	xs:string In previous releases, this field required the email address to be registered with paypal.com. In the 1.3.0 release, you have the option to omit the field from the payment request. The consumer can either log in using an existing PayPal account (that is not associated with a receiver) or create a new account during the payment flow.

Changes to the PaymentDetails API Operation for Version 1.3.0

Changes to PaymentDetailsResponse messages

Field	Description
fundingConstraint	New field: ap:FundingConstraint Specifies a list of allowed funding types for the payment. This list can be in any order. If this field is omitted, the payment can be funded by any funding type that is supported by Adaptive Payments. **NOTE:** This feature is available for applications with a special permission level.
FundingTypeList.fundingTypeInfo	New field: ap:FundingTypeInfo Specifies a list of allowed funding types for the payment. This list can be in any order. If this field is omitted, the payment can be funded by any funding type that is supported by Adaptive Payments. **NOTE:** This feature is available for applications with a special permission level.
fundingType	New field: xs:string Specifies the allowed funding type that can be used as the funding source for the payment. Possible values are: • ECHECK – Electronic check • BALANCE – PayPal account balance • CREDITCARD – Credit card **NOTE:** This feature is available for applications with a special permission level.
PaymentInfo.senderTransactionID	xs:string The identification number of the sender of the payment. For previous releases, the response included only information about the receiver portion of the transaction. With 1.3.0, the response includes information about the sender portion of the transaction. This feature accommodates payments to a receiver's email address that is not linked to a registered PayPal account.

Field	Description
PaymentInfo.senderTransactionStatus	xs:string The status of the sender of the payment. For previous releases, the response included only information about the receiver portion of the transaction. With 1.3.0, the response includes information about the sender portion of the transaction. This feature accommodates payments to a receiver's email address that is not linked to a registered PayPal account.

1.2.0 Features

Version 1.2.0 of the Adaptive Payments API introduces changes to several Adaptive Payments API operations. These changes include backward-compatible changes to request and response messages as well as additional error messages.

Changes to the Pay API Operation for Version 1.2.0

Changes to PayRequest messages

Field	Description
`ClientDetails.deviceID`	This field is now optional; it is no longer required.
`Receiver.invoiceId`	New field: `xs:string` *(Optional)* The invoice number for the payment

Changes to PayResponse message

Field	Description
`Receiver.invoiceID`	New field: `xs:string` The invoice number for the payment

Changes to the Payment Details API Operation for Version 1.2.0

Changes to PaymentDetailsReponse message

Field	Description
`Receiver.invoiceID`	New field: `xs:string` The invoice number for the payment

Changes to Preapproval API Operation for Version 1.2.0

Changes to PreapprovalRequest messages

Field	Description
ipnNotificationUrl	New field: xs:string *(Optional)* The URL to which you want all IPN messages for this preapproval to be sent. This URL supersedes the IPN notification URL in your profile.
memo	New field: xs:string *(Required)* A note about the preapproval.
ClientDetails.deviceID	This field is now optional; it is no longer required.

Changes to Preapproval Details API Operation for Version 1.2.0

Changes to PreapprovalDetailsResponse message

Field	Description
ipnNotificationUrl	New field: xs:string The URL to which all IPN messages for this preapproval are sent.
memo	New field: xs:string A note about the preapproval.

1.1.0 Features

Version 1.1.0 of the Adaptive Accounts API introduces changes to all Adaptive Payments API operations. These changes include backward compatible changes to request and response messages as well as additional error messages.

Changes to the Pay API Operation for Version 1.1.0

Changes to PayRequest message

Field	Description
`logDefaultShippingAddress`	New field: `xs:boolean` *(Optional)* Whether or not to associate the receiver's default shipping address with the transaction.
`clientDetails.deviceId`	This field is now optional; it is no longer required.

Changes to PayResponse message

Field	Description
`payErrorList`	New field: `ap:PayErrorList` Information about why a payment failed.

New Pay error messages

Code	Message	Additional Information
580022	Invalid request: ~1	Invalid request parameter; for example an invalid application ID or language code
580026	The invoice id (~1) already exists for receiver ~2 and cannot be duplicated	
580027	Payment is not allowed to be pending	
580029	Invalid request: ~1	Missing request parameter

Changes to PaymentDetails API Operation for Version 1.1.0

Changes to PaymentDetailsResponse

Field	Description
`logDefaultShippingAddress`	New field: `xs:string` Indicates when the Pay operation was called for this payKey.
`trackingId`	Empty fields are not returned; this is now enforced for `trackingID`.

New PaymentDetails error messages

Code	Message	Additional Information
540031	You do not have permission to get the details of this Payment	PaymentDetails

Code	Message	Additional Information
580022	Invalid request: ~1	Invalid request parameter; for example an invalid application ID or language code
580029	Invalid request: ~1	Missing request parameter

Changes to Preapproval API Operation for Version 1.1.0

Changes to PreapprovalRequest message

Field	Description
`maxNumberOfPayments`	This field is now optional; it is no longer required.
`senderEmail`	This field is now optional; it is no longer required. If not specified, the email address of the sender who logs in to approve the request becomes the email address associated with the preapproval key.
`clientDetails.deviceId`	This field is now optional; it is no longer required.

New Preapproval error messages

Code	Message	Additional Information
580022	Invalid request: ~1	Invalid request parameter; for example an invalid application ID or language code
580029	Invalid request: ~1	Missing request parameter

Changes to PreapprovalDetails API Operation for Version 1.1.0

Changes to PreapprovalDetailsResponse message

Field	Description
`senderEmail`	The sender's email address is only available when you explicitly set the email address in the `PreapprovalRequest` message.
`maxNumberOfPayments`	The preapproved maximum number of payments is only available when you explicitly set the email address in the `PreapprovalRequest` message.

New PreapprovalDetails error messages

Code	Message	Additional Information
540031	You do not have permission to get the details of this Preapproval	
580022	Invalid request: ~1	Invalid request parameter; for example an invalid application ID or language code
580029	Invalid request: ~1	Missing request parameter

Changes to Refund API Operation for Version 1.1.0

Changes to RefundResponse message

Field	Description
`refundInfoList.refundInfo.errorList`	New field: `xs:string` Information about why a refund failed.

New Refund error messages

Code	Message	Additional Information
540031	You do not have permission to get the details of this Refund	
580022	Invalid request: ~1	Invalid request parameter; for example an invalid application ID or language code
580029	Invalid request: ~1	Missing request parameter

Revision History

The revision history identifies changes to Adaptive Payments documentation at a high level.

Date	Description
08/07/2012	Revised draft for version 1.8.2. Added new fields related to Brazil tax ID to `SenderIdentifer`. Added new BANK_MANAGED_WITHDRAWAL value to `paymentType` field.
05/02/2012	Maintenance release. Added information about Embedded Preapproval Experience.
02/07/2012	Added `refferercode` to `SenderOptions`.
11/16/2011	Revised draft for version 1.8.0. Includes new API operation fields for Convert Currency and changes to JavaScript functions.
05/11/2011	Added information about minibrowser embedded payments flow.
04/05/2011	Revised draft for version 1.7.0. Includes information about invoice data and item details, which was introduced in version 1.6.0.
01/05/2011	Revised draft for version 1.6.0. Includes information about preapproved future payments and shipping address information for embedded payments
10/29/2010	Revised draft for version 1.6.0. Includes information about embedded payments and a digital goods example
10/10/2010	Revised draft for version 1.6.0
07/13/2010	Revised draft for version 1.5.0. Includes information about guest payments
06/28/2010	Release draft for version 1.5.0
03/10/2010	Release draft for version 1.4.0
10/09/2009	Release draft for version 1.3.0
09/26/2009	Review draft for version 1.3.0
09/25/2009	Review draft for version 1.3.0
08/27/2009	Revised draft for version 1.3.0
08/17/2009	Revised draft for version 1.2.0.
08/07/2009	Revised draft for version 1.1.0.
07/16/2009	Initial draft for version 1.0.0.

Index

Symbols

_ap-payment flow command 253
_ap-preapproval flow command 254

A

ack response field 96, 98, 120, 123, 132, 135, 146, 148, 157, 159, 171, 173, 184, 187, 193, 195, 206, 208, 225, 228, 239, 242, 248, 250
action_type IPN variable 258
actionType request field 79
actionType response field 113
Address response fields 183, 247
addresseeName response field 183, 247
addressId response field 184, 247
AddressList response field 183
addresslist response field 180
AddressList response fields 183
allowed response field 93, 239
allowedFundingType request field 83
amount request field 198
amount response field 91, 204, 237
approved IPN variable 260
approved response field 180

B

backupFundingSource response field 91, 236
BaseAddress request fields 184, 247
baseAddress response field 184, 247
baseAmount response field 204
baseAmountList request field 198
build response field 96, 98, 121, 123, 132, 135, 146, 148, 157, 159, 171, 173, 184, 187, 193, 195, 206, 208, 225, 228, 240, 242, 248, 250
businessName request field 153
businessName response field 142

C

cancel_url IPN variable 259, 260
CancelPreapproval API operation 191
CancelPreapprovalRequest fields 191
CancelPreapprovalRequest message 191
CancelPreapprovalResponse fields 192
CancelPreapprovalResponse message 192
cancelUrl request field 79
cancelUrl request fields 165
cancelUrl response field 113, 180
category field 94, 98, 122, 131, 134, 148, 159, 173, 186, 195, 208, 224, 227, 241, 250
charge response field 91, 93, 94, 237
city response field 184, 247
clientDetails request field 79
clientDetails request fields 165
closeFlow JavaScript function 255
code request field 199
code response field 92, 205, 238
ConversionCurrencyResponse fields 202, 203, 264
conversionType request field 198
ConvertCurrency API operation 197
ConvertCurrencyRequest fields 198
ConvertCurrencyRequest message 197
ConvertCurrencyResponse message 202
convertToCurrencyList request field 198
correlationId response field 96, 99, 121, 123, 132, 135, 146, 149, 157, 160, 171, 174, 185, 187, 193, 196, 206, 209, 226, 228, 240, 242, 248, 251
countryCode request field 85, 156, 198, 218, 264
countryCode response field 96, 120, 143, 145, 154, 184, 225, 247
curPayments response field 180
curPaymentsAmount response field 180
curPeriodAttempts response field 180
curPeriodEndingDate response field 180
currency request field 200
currency response field 205
currencyCode request field 80, 201, 216
currencyCode request fields 166
currencyCode response field 114, 181, 221
CurrencyCodeList request fields 201
currencyConversion response field 91, 237
CurrencyConversion response fields 93, 239

currencyConversionList response field 203
CurrencyConversionList response fields 204
CurrencyConversionTable response fields 203
CurrencyList request fields 200
currencyList response field 204
CurrencyList response fields 205
CurrencyType request fields 198
CurrencyType response fields 91, 204, 237
current_number_of_payments IPN variable 260
current_period_attempts IPN variable 261
current_total_amount_of_all_payments IPN variable 261
customId request field 155
customId response field 144

D

dateOfMonth request fields 166
dateOfMonth response field 181
dayOfWeek request fields 167
dayOfWeek response field 182
defaultFundingPlan response field 90
description request field 155
description response field 144
detailLevel request field 86, 110, 128, 139, 156, 170, 178, 192, 202, 218, 234, 246
displayMaxTotalAmount request field 168
displayMaxTotalAmount response field 183
displayName response field 93, 143, 154, 239
displayOptions request field 152
DisplayOptions request fields 153
displayOptions response field 142
DisplayOptions response fields 142
domain field 94, 98, 122, 131, 134, 148, 159, 173, 186, 195, 208, 224, 227, 241, 250

E

email request field 86, 156
email response field 120, 143, 145, 154
emailHeaderImageUrl request field 153
emailHeaderImageUrl response field 142
emailMarketingImageUrl request field 153
emailMarketingImageUrl response field 142
embedded payment flow command 254
encryptedRefundTransactionId response field 222
endingDate request fields 167
endingDate response field 182
environments, production sandbox beta-sandbox 66
error PPFault field 97, 122, 134, 147, 158, 172, 186, 194, 207, 227, 241, 249
error response field 94, 130
ErrorData fields 94, 98, 122, 131, 134, 148, 159, 173, 186, 195, 208, 224, 227, 241, 250
errorId field 94, 98, 122, 131, 134, 148, 159, 173, 186, 195, 208, 224, 227, 241, 250
errorLanguage request field 86, 110, 128, 140, 156, 170, 178, 192, 202, 218, 234, 246
errorList response field 222
estimatedAmountTable response field 203
exceptionID field 94, 98, 122, 131, 134, 148, 159, 173, 186, 195, 208, 224, 227, 241, 250
exchangeRate response field 93, 239
ExecutePayment API operation 127
ExecutePaymentRequest message 127
ExecutePaymentRequest request fields 127
ExecutePaymentResponse fields 130
ExecutePaymentResponse message 128
extension request field 85, 156, 218
extension response field 96, 120, 145, 225

F

FaultMessage PPFault fields 97, 122, 134, 147, 158, 172, 186, 194, 207, 227, 241, 249
fees_payer IPN variable 260
feesPayer request field 80, 169, 183
feesPayer response field 114
firstName response field 143, 154
from response field 93, 239
fundingAmount response field 91, 236
fundingConstraint request field 81
FundingConstraint request fields 83
FundingPlan fields 91, 236
FundingPlan response field 236
FundingPlanCharge response fields 93, 94
fundingPlanId request field 127
fundingPlanId response field 91, 236
fundingSource response field 93, 94
FundingSource response fields 92, 238
fundingSourceId request field 266
fundingSourceId response field 93, 239
fundingType request field 84
fundingType response field 116
fundingTypeInfo request field 83

FundingTypeInfo request fields 84
fundingTypeInfo response field 116
FundingTypeInfo response fields 116
FundingTypeList request fields 83
fundingtypeList response field 115
FundingTypeList response fields 116

G

getBillingAddress request field 177
GetFundingPlans API operation 233
GetFundingPlansRequest fields 234
GetFundingPlansRequest message 233
GetFundingPlansResponse fields 236
GetFundingPlansResponse message 234
GetPaymentOptions API operation 139
GetPaymentOptionsRequest fields 139
GetPaymentOptionsRequest message 139
GetPaymentOptionsResponse fields 141
GetPaymentOptionsResponse message 140
GetShippingAddresses API operation 245
GetShippingAddressesRequest fields 245
GetShippingAddressesRequest message 245
GetShippingAddressesResponse fields 247
GetShippingAddressesResponse message 246

H

headerImageUrl request field 153
headerImageUrl response field 142

I

identifier request field 155
identifier response field 145
initiatingEntity request field 152
initiatingEntitity response field 142
InitiatingEntity response fields 143, 153
institu tionCustomer response field 153
Institution Customer response fields 143, 154
institutionCustomer response field 143
institutionCustomerId response field 143, 154
institutionId response field 143, 154
invoiceData request field 155
InvoiceData request fields 155
invoiceData response field 144
InvoiceData response fields 144
InvoiceItem request fields 155
InvoiceItem response fields 145
ipn_notification_url IPN variable 259
ipnNotificationUrl request field 81
ipnNotificationUrl request fields 167
ipnNotificationUrl response field 115, 182
isOpen JavaScript function 255
item request field 155
item response field 144
itemCount request field 156
itemCount response field 145
itemPrice request field 156
itemPrice response field 145

J

JavaScript functions 255, 265, 267

K

key request field 245

L

lastFourOfAccountNumber response field 92, 238
lastName response field 143, 154
line1 response field 184, 247
line2 response field 184, 247

M

maxAmountPerPayment request fields 167
maxAmountPerPayment response field 182
maxNumberOfPayments request fields 167
maxNumberOfPayments response field 182
maxNumberOfPaymentsPerPeriod request fields 167
maxNumberOfPaymentsPerPeriod response field 182
maxTotalAmountOfAllPayments request fields 167
maxTotalAmountOfAllPayments response field 182
memo IPN variable 259
memo request field 81
memo request fields 167
memo response field 115, 182
message field 94, 98, 122, 131, 134, 148, 159, 173, 186, 195, 208, 224, 227, 241, 250

N

name request field 155
name response field 145

P

parameter field 94, 98, 122, 131, 134, 148, 159, 173, 186, 195, 208, 224, 227, 241, 250
Pay API operation 73
Pay Message variables 257
pay_key IPN variable 259
payError response field 91, 130
PayError response fields 94, 130
payErrorList response field 90, 130
PayErrorList response fields 91, 130
payKey request field 127, 139, 152, 216, 234
payKey response field 90, 115, 141
payment_request_date IPN variable 258
PaymentDetails API operation 109
PaymentDetailsRequest message 109
PaymentDetailsResponse fields 110, 113
paymentExecStatus response field 90, 130
paymentInfo response field 117
PaymentInfo response fields 117
paymentInfoList response field 115
PaymentInfoList response fields 117
paymentPeriod request fields 168
paymentPeriod response field 182
PAYPAL.apps.DGFlow JavaScript function 255, 265, 267
PayRequest fields 79
PayRequest message 77
PayResponse fields 90
PayResponse message 86
pendingReason response field 117
pendingRefund response field 117
phone request field 86, 156
phone response field 120, 145
phoneNumber request field 85, 156, 218
PhoneNumber request fields 85, 156, 218
phoneNumber response field 96, 120, 145, 225
PhoneNumber response fields 96, 120, 145, 225
pin request field 81
pinType request fields 168
pinType response field 182
PPFault message 96, 121, 133, 146, 158, 171, 185, 193, 206, 226, 240, 248
Preapproval Message variables 260
preapproval_key IPN variable 260
PreapprovalDetails API operation 177
PreapprovalDetailsRequest fields 177
PreapprovalDetailsRequest message 177
PreapprovalDetailsResponse fields 180
PreapprovalDetailsResponse message 178
preapprovalKey request field 81, 177, 191
preapprovalKey response field 115
preapprovalKey response fields 171
PreapprovalRequest fields 165
PreapprovalRequest message 164
PreapprovalResponse fields 171
price request field 155
price response field 145
production environment 66

R

reason_code IPN variable 260
receiver request field 84, 155, 217
receiver response field 94, 118, 130, 144, 222
ReceiverIdentifier request fields 156
ReceiverIdentifier response fields 145
receiverList request field 81, 217
ReceiverList request fields 84, 217
receiverOptions request field 153
ReceiverOptions request fields 155
receiverOptions response field 142
ReceiverOptions response fields 144
referrerCode request field 154
referrrerCode response field 144
Refund API operation 211
refundedAmount response field 118
refundFeeAmount response field 222
refundGrossAmount response field 222
refundHasBecomeFull response field 222
refundInfo response field 222
RefundInfo response fields 222
refundInfoList response field 221
RefundInfoList response fields 222
refundNetAmount response field 222
RefundRequest fields 216
RefundRequest message 214
RefundResponse fields 221
RefundResponse message 219
refundStatus response field 223

refundTransactionStatus response field 223
req uestenvelope request field 81
requestEnvelope request field 127, 139, 153, 177, 191, 198, 217, 234, 245
RequestEnvelope request fields 86, 110, 128, 139, 156, 170, 178, 192, 202, 218, 234, 246
requestEnvelope request fields 168
requestEnvelope response field 142
requireInstantFundingSource request field 169
requireShippingAddressSelection request field 154, 264
requireShippingAddressSelection response field 144
responseEnvelope PPFault field 97, 122, 134, 147, 158, 172, 186, 194, 207, 227, 241, 249
responseEnvelope response field 90, 115, 130, 182, 192, 203, 221, 236, 247
ResponseEnvelope response fields 96, 98, 120, 123, 132, 135, 146, 148, 157, 159, 171, 173, 184, 187, 193, 195, 206, 208, 225, 228, 239, 242, 248, 250
responseEnvelope response fields 171
return_url IPN variable 259
returnUrl request field 81
returnUrl request fields 168
returnUrl response field 115, 182
reverse_all_parallel_payments_on_error IPN variable 258
reverseAllParallelPaymentsOnError request field 81
reverseAllParallelPaymentsOnError response field 115

S

sandbox environment 66, 99, 101, 102, 103
sandbox, beta 66
selectedAddress response field 247
sender request field 82
sender_email IPN variable 258
senderEmail request field 82
senderEmail request fields 168
senderEmail response field 115, 183
senderFees response field 91, 236
SenderIdentifier request fields 86
SenderIdentifier response fields 120
senderOptions request field 152
senderOptions response field 142
senderTransactionID response field 118
senderTransactionStatus response field 118
SetPaymentOptionsRequest fields 152
SetPaymentOptionsResponse message 151, 157
SetPaymentOptionsResponse operation 151
setTrigger JavaScript function 255
severity field 95, 98, 123, 131, 134, 148, 159, 173, 187, 195, 208, 224, 228, 242, 250
shippingAddressId request field 152
shippingAddressId response field 142
startFlow JavaScript function 255
startingDate request fields 168
startingDate response field 183
state response field 184, 247
status IPN variable 258
status response field 116, 183
subdomain field 95, 98, 123, 131, 134, 148, 159, 173, 187, 195, 208, 224, 228, 242, 250

T

timestamp response field 96, 99, 121, 123, 133, 135, 146, 149, 157, 160, 171, 174, 185, 187, 193, 196, 206, 209, 226, 228, 240, 242, 248, 251
to response field 93, 239
totalOfAllRefunds response field 223
totalShipping request field 155
totalShipping response field 144
totalTax request field 155
totalTax response field 144
trackingId IPN variable 260
trackingId request field 82, 217
trackingId response field 116
transac tion[n].receiver IPN variable 259
transaction_type IPN variable 257, 260
transaction[n].amount IPN variable 259
transaction[n].id IPN variable 258
transaction[n].id_for_sender IPN variable 258
transaction[n].invoiceId IPN variable 259
transaction[n].is_primary_receiver IPN variable 259
transaction[n].refund_account_charged IPN variable 259
transaction[n].refund_amount IPN variable 259
transaction[n].refund_id IPN variable 259
transaction[n].status IPN variable 258
transaction[n].status_for _sender_txn IPN variable 259
transactionId request field 217
transactionId response field 118
transactionStatus response field 118
type response field 93, 184, 239, 248

U

useCredentials request field 86
useCredentials response field 119, 120, 263

W

warningDataList response field 90