



Sandbox User Guide

For Professional Use Only
Currently only available in English.

A usage Professionnel Uniquement
Disponible en Anglais uniquement pour l'instant.

PayPal Sandbox User Guide

Document Number: 100007.en_DE-200710

© 2007 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. PayPal (Europe) Ltd. is authorised and regulated by the Financial Services Authority in the United Kingdom as an electronic money institution. PayPal FSA Register Number: 226056.

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.

Preface	5
This Document	5
Intended Audience	5
Organization of This Document	5
Notational Conventions	6
Revision History	7
 Chapter 1 Overview to the PayPal Sandbox	 9
Get Started Quickly: Integration Center	9
At a Glance: Differences between the Sandbox and Live PayPal	9
 Chapter 2 Accessing the PayPal Sandbox	 11
Signing Up for Sandbox Access	11
Welcome to the PayPal Sandbox	13
Test Email	13
 Chapter 3 Setting up Test Users	 15
Planning the Types of Test Accounts You Need	15
Managing Test Accounts	15
Creating a Premium Account	16
Creating a Business Account	16
Confirming an Email Address	16
“Unverified” Account Status	17
Adding a Funding Source	17
Adding a Bank Account	17
Adding a Credit Card	18
 Chapter 4 Testing PayPal Website Features (non-API)	 21
Website Payments with Buy Now Button	21
Encrypted Website Payments	22
Testing Payments with Buy Now Button	22
Verifying a Test Payment	22
Testing Payments with a Third Party Shopping Cart (Cart upload feature)	23
Handling Pending Transactions	23
Creating a Pending Transaction	23
Completing or Canceling a Pending Transaction	24

Instant Payment Notification (IPN)	24
Setting up IPN in the Sandbox.	24
Testing Refunds	25
Issuing a Test Refund	25
Verifying a Test Refund	26
Chapter 5 Testing PayPal NVP APIs27
Testing Express Checkout	27
Testing giropay and EFT Transactions.	30
Chapter 6 Testing Error Conditions35
API Testing	35
Negative Testing Using an Amount-Related Trigger Field.	37
Negative Testing Using a Non-Amount Trigger Field	37
Negative Testing With Multiple Messages	38
Chapter 7 Technical Support39
Contacting Customer Service for Live PayPal Website Help	39
Contacting Developer Technical Support for API Help.	39
Index.41



Preface

This Document

This document describes the PayPal virtual test environment called the Sandbox.

Intended Audience

This document is written for merchants and developers who want to test their PayPal-based applications before using them in production.

Organization of This Document

[Chapter 1, “Overview to the PayPal Sandbox,”](#) describes the Sandbox in general and lists the differences between the Sandbox and the production PayPal services.

[Chapter 2, “Accessing the PayPal Sandbox,”](#) describes how to gain access to the Sandbox and your Sandbox email.

[Chapter 3, “Setting up Test Users,”](#) describes the different kinds of business roles and corresponding test accounts you need to set up on the Sandbox for effective testing of your PayPal-based applications.

[Chapter 4, “Testing PayPal Website Features \(non-API\),”](#) describes how you can use the Sandbox (<https://www.sandbox.paypal.com>) to test features that correspond to the features on the main PayPal interactive site <https://www.paypal.com/>.

[Chapter 5, “Testing PayPal NVP APIs,”](#) describes how to test the Express Checkout NVP API in the Sandbox.

[Chapter 6, “Testing Error Conditions,”](#) describes how to use negative testing to simulate API error codes, address verification errors, and credit card validation errors.

[Chapter 7, “Technical Support,”](#) provides information on contacting Technical Support.

Notational Conventions

This document uses typefaces to identify the characteristics of text. These typefaces and the characteristics they imply are described below:

Typeface	How Used
<i>serif italics</i>	<p>A document title.</p> <p>A term being discussed or defined. For example: A file is a readable or writable stream of characters ...</p> <p>Boolean values (not keywords). For example: The function returns true if it encounters an error.</p>
monospaced	<p>Pathnames or file names that appear in body text frames.</p> <p>Code-related names that appear in body text frames. Such names are used for functions, callbacks, arguments, data structures, and fields. For example: <code>AbstractResponseType</code> is the SOAP response type definition on which all PayPal API response methods are based.</p> <p>Components of Internet protocol requests and responses, such as HTTPS and FORM variables. For example: The PayPal system uses a <code>method=POST</code> request to return IPN status variables related to subscriptions, such as <code>txn_type</code>.</p>
Serif bold	<p>User interface names, such as window names or menu selections. For example: On the Profile page, click Email to confirm your email address.</p>
<i>San-serif oblique</i>	<p>Placeholders used in the context of a format or programming standard or formal descriptions of PayPal system syntax. Placeholders indicate values or names that the reader should provide. Example: For example, amount is the variable for a single-item shopping cart, but amount_X is the name of the variable for a multi-item shopping cart. <i>amount_3</i> is the item amount for the third item in a multiple-item shopping cart.</p>

To convey additional information, this document may also apply color and underlining to words or phrases that use the typefaces described above. Such use is described below:

Text attribute	How Used
xxxxxx	Hypertext link to a page in the current document or to another document in the set.
xxxxxx	Hypertext link to a URL or that initiates a web action, such as sending mail.

Revision History

Revision history for *PayPal Sandbox User Guide*.

TABLE P.1 *Revision History*

Date	Description
October 2007	Added giropay and EFT example.
August 2007	Changed PayPal logo and corrected all known problems.
May 2007	Revised manual to reflect new “autocreated” accounts feature.
April 2007	Added chapters on negative testing and testing APIs using the Express Checkout NVP API. Added new section on handling pending transactions and made miscellaneous changes.
July 2006	Correction of variable name ‘ipn_test,’ which should be “test_ipn”.
June 2006	Correction of Sort Code necessary to test UK accounts in Sandbox. Proper Sort Code is 609204.
December 2005	Miscellaneous corrections.



Revision History

1

Overview to the PayPal Sandbox

The PayPal Sandbox is a self-contained environment within which you can prototype and test your PayPal applications. The PayPal Sandbox is an almost identical copy of the live PayPal website. Its purpose is to give developers a shielded environment for testing and integration purposes and to help avoid problems that might occur while testing PayPal integration solutions on the live site. Before moving any PayPal-based application into production, you should test the application in the Sandbox to ensure that it functions as you intend and within the guidelines and standards set forth by the PayPal Developer Network (PDN).

Get Started Quickly: Integration Center

PayPal's Integration Center at <https://www.paypal.de/integrationcenter> has step-by-step details for getting started with the PayPal Software Development Kits (SDKs), Express Checkout, Website Payments Standard, Instant Payment Notification, and more.



Visit the Integration Center at: <https://www.paypal.de/integrationcenter>

At a Glance: Differences between the Sandbox and Live PayPal

The following table compares the Sandbox and Live PayPal. This is an at-a-glance view of the differences from the perspective of an in-house or third-party developer for a business.

You can also use this table as a checklist.

TABLE 1.1 Differences between Developer Central, Sandbox, and Live PayPal

	PayPal Developer Central and Sandbox	Live PayPal Website and API Service
Type of PayPal Accounts	Depending on the feature you want to develop and test, you need a Business or Premier account.	Business or Premier account
URL and Site logos in upper left corner	https://www.sandbox.paypal.com 	https://www.paypal.de 
NVP API Servers	https://api.sandbox.paypal.com/nvp/	For API Certificate security: https://api.paypal.com/nvp/ For API Signature security: https://api-3t.paypal.com/nvp/

Overview to the PayPal Sandbox

At a Glance: Differences between the Sandbox and Live PayPal

TABLE 1.1 Differences between Developer Central, Sandbox, and Live PayPal

	PayPal Developer Central and Sandbox	Live PayPal Website and API Service
SOAP API Servers	https://api.sandbox.paypal.com/2.0/	For API Certificate security: https://api.paypal.com/2.0/ For API Signature security: https://api-3t.paypal.com/2.0/
Business roles	You fill all roles you need to test: merchant, buyer, and seller.	Real-world people fill these roles.
Company and people's names and postal addresses	Completely fictitious. Before you begin working with the Sandbox, create the details for all the business roles you must fulfill. The Sandbox simulates verification of postal addresses and names.	Real companies' and people's names and postal addresses.
Email addresses and email inboxes	The Sandbox has a special-purpose email inbox for your testing, contained in the Sandbox itself.	Real email address and inbox for each business role
Bank account and credit card numbers	The Sandbox creates bank accounts, credit card numbers, and CVV2 numbers you need in order to develop and test; all of which are fictitious and only used within the Sandbox. The Sandbox simulates the verification of these numbers. Transactions do not affect real accounts and actual money is never exchanged.	Actual verification of bank account numbers, credit card numbers, and CVV2 numbers
PayPal transactions	All transaction in Sandbox are fictitious transactions that are not connected to live PayPal processes.	Live transactions, cleared by live PayPal processes
Fraud detection	Fraud detection is not enabled for the Sandbox.	Full protection through PayPal's fraud detection
Digital certificates	After you request digital certificates for use with the PayPal Web Services API, the Sandbox automatically generates them. They are available for immediate downloading.	To safeguard your and your customers' security, requests for digital certificates for use with the Live PayPal Web Services API must be verified by PayPal before they are issued. You are notified in email when your request has been approved.
PayPal Merchant Features supported	All features of the live PayPal website, except closing an account, auction features, monthly statements, shipping preferences, and PayPal Shops.	
Technical Support	Developer Technical Support is available from Developer Central. No telephone support.	Customer Service is available either by email or by telephone. See “Contacting Customer Service for Live PayPal Website Help” on page 39.

2

Accessing the PayPal Sandbox

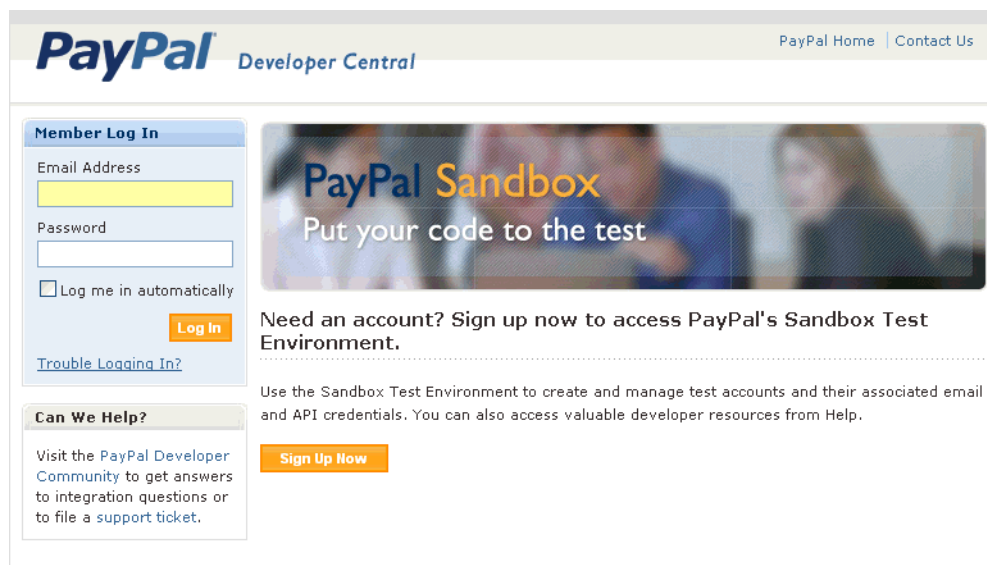
To access the PayPal Sandbox, sign up for an account at <https://developer.paypal.com>. After signing up, you access the Sandbox either programmatically or by logging in.

Depending on the PayPal feature you want to test with an application, you need to set up different types of test accounts. See “Planning the Types of Test Accounts You Need” on [page 15](#).

Signing Up for Sandbox Access

To sign up for Sandbox access:

1. Go to <https://developer.paypal.com>. The log in screen is shown below:



The screenshot shows the PayPal Developer Central login page. At the top, the PayPal logo is followed by "Developer Central" and links for "PayPal Home" and "Contact Us". On the left, there is a "Member Log In" section with fields for "Email Address" and "Password", a checkbox for "Log me in automatically", a "Log In" button, and a link for "Trouble Logging In?". Below this is a "Can We Help?" section with text about the PayPal Developer Community and a "Sign Up Now" button. On the right, there is a large banner for "PayPal Sandbox" with the text "Put your code to the test". Below the banner, it says "Need an account? Sign up now to access PayPal's Sandbox Test Environment." and "Use the Sandbox Test Environment to create and manage test accounts and their associated email and API credentials. You can also access valuable developer resources from Help."

2. If you already have an account, enter your Log In Email and Password and click **Log In**.

3. If you do not already have an account, click **Sign Up Now** and provide the requested information shown below:

Sign Up for Access to the Sandbox Test Environment

This account will allow you to use the PayPal Sandbox Test Environment to try out Website Payments, Instant Payment Notification, PayPal APIs, and other features.

First Name:

Last Name:

Email Address:
Do not use your PayPal account login email.

Password:
At least 8 characters long, case sensitive.

Confirm Password:

Security Question: -- select a question --

Security Answer:

Communications: ☒ Please keep me informed on PayPal's Web Services, the PayPal Sandbox, and Developer Central.

Terms of Use - The User Agreement and Privacy Policy are designed to protect and inform you of your rights within the PayPal Developer Central service.

User Agreement [\(Printer Friendly Version\)](#)

THE FOLLOWING DESCRIBES THE TERMS ON WHICH PAYPAL OFFERS YOU ACCESS TO OUR PAYPAL DEVELOPER CENTRAL SERVICES.

This User Agreement ("Agreement") is a contract between you and PayPal, Inc. and applies to your use of the PayPal Developer

Privacy Policy [\(Printer Friendly Version\)](#)

The Privacy Policy below governs your PayPal Developer Central account and any information you provide on the PayPal Developer Central site.

A. Overview

Do you agree to the User Agreement and Privacy Policy, and terms incorporated therein?

☐ Yes ☒ No

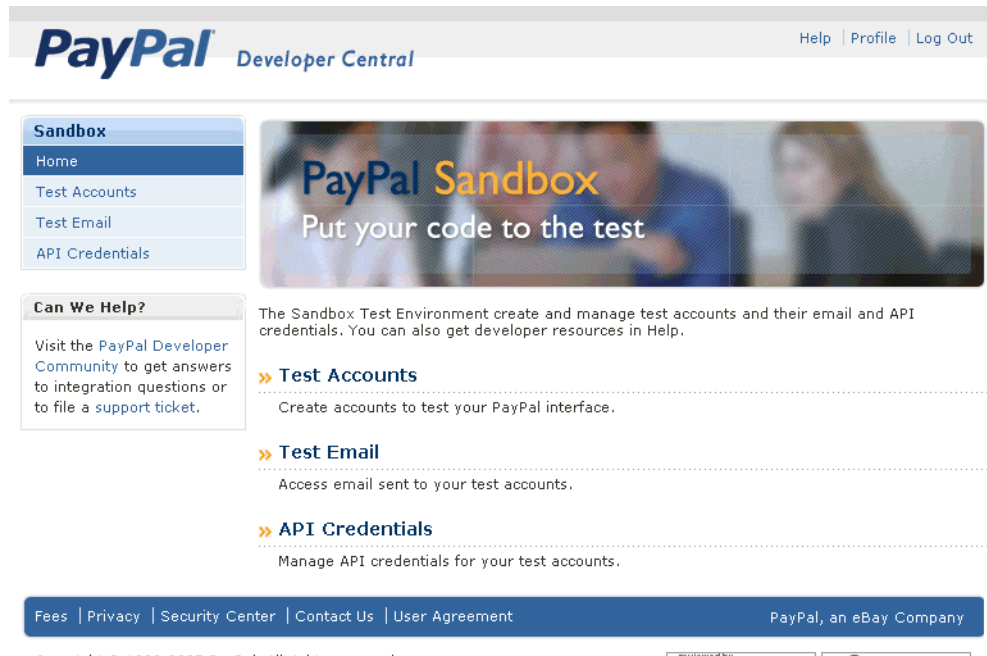
IMPORTANT: Do not use the same log in email address or password that you use for logging into the live paypal.com site because later you may allow someone to work in the Sandbox on your behalf but not want to allow access to your regular PayPal account.

After you sign up, PayPal sends login instructions to the email address you used to sign-up. If you have mail filtering enabled in your mail software, the email sent by PayPal might be filtered out or stored in a folder where you are not expecting it to be. For instance, with Microsoft Outlook mail software, your filtering might cause the email to be stored in "Junk" or "Spam."

4. Respond to the confirmation e-mail and log in.

Welcome to the PayPal Sandbox

When you log in to the Sandbox, the Sandbox Test Environment home page appears, as follows:



On this page, you can perform the following actions:

- Manage test accounts from the **Test Accounts** tab. From this tab you can create and delete test accounts. From this tab, you can also enter the Sandbox Test Site, which simulates the live `paypal.com` site. For more information, see [“Setting Up Test Accounts” on page 17](#).
- Access email sent to test accounts from the **Test Email** tab. For more information, see [“Test Email” on page 13](#).
- View API credentials for business test accounts from the **API Credentials** tab. An API signature, which is the preferred kind of credential, is automatically created when you create a Business test account. You need the information on this tab when you test APIs.
- Obtain technical information about PayPal products and APIs using the **Help** link.
- Change the log in password using the **Profile** link.

NOTE: You cannot change the Log In Email address.

Test Email

When certain kinds of transactions occur in the live PayPal system, PayPal sends email messages to the real email addresses of the participants. From these email messages, the

recipient or initiator of an event or transaction can verify that the event took place and that the monetary amounts associated with the event are correct.

PayPal test email, however, is a self-contained email system in the Sandbox itself. You see email messages addressed only to the Sandbox test accounts you set up. Up to 30 of the latest email messages are listed on the **Test Email** tab. The subject line of email messages you have not read are in bold. Click a subject line to read the message.

3

Setting up Test Users

Depending on the business application you are developing and testing, you need different types of test accounts. There are two types of test accounts: Premium and Business.

Planning the Types of Test Accounts You Need

Determine the types of test accounts you need to test the applications you are developing. In addition, determine the number of different accounts you need. You might need several different Premium or Business PayPal test accounts to test your application.

The information you need to supply when you create test accounts is as follows:

- Mailing address
- Email address and password for the test PayPal account. You can use the same password (not email address) for all your test accounts so that you can more easily remember it.
- Security questions and answers. You can use the same security questions and answers for all your test accounts so that you can more easily remember them.

IMPORTANT: The email address and password of a test account should never be those of a real account on the live PayPal site. The same applies to your answers to the security questions. All of this data should be fictional.

- Premium or Business account
- Your agreement to the terms of using the Sandbox

For Business accounts, you also need to supply the following fictitious information for testing:

- Business name and address
- Customer service contact information
- Business owner contact information
- Business owner address

The email address and password of a test account should never be those of a real account on the live PayPal site. The same applies to your answers to the security questions. All of this data should be fictional.

Managing Test Accounts

In Developer Central, you can view, work with, or launch the Sandbox for all your test accounts. You can also create new accounts or remove test email addresses from your view.

- To work with test accounts, log in to Developer Central, and click the **Sandbox** tab.
- To create a new account, click the Create Account link in the upper left corner of the list of accounts.
- To work with the account, select the account by clicking the radio button associated with it on the left.

You can start the Sandbox for the selected account by clicking **Enter Sandbox Test Site**. When you logged in to Developer Central, you might have set the **Log me in automatically** checkbox to allow direct access to <https://www.sandbox.paypal.com/>, in which case you do not have to launch the Sandbox to access it.

NOTE: The **Delete** button does not delete the test account. It removes the test account from your list of accounts, but the email address for the test account is still on file for the Sandbox. You cannot reuse an email address that is still on file for the Sandbox.

Creating a Premium Account

To create a test user with a Premium account:

1. After logging in, select **Test Accounts** and click the **Create Test Account** link.
2. Choose **Germany** for the **Country** and make other selections or accept the defaults.
3. Choose **Buyer** for the **Account Type**.
4. Make other selections or accept the defaults.
5. Click **Create Account**.
6. Add a bank account; see “[Adding a Bank Account](#)” on page 17.

Creating a Business Account

To create a test user with a Business account:

1. After logging in, select **Test Accounts** and click the **Create Test Account** link.
2. Choose **Germany** for the **Country** and make other selections or accept the defaults.
3. Choose **Seller** for the **Account Type**.
4. Make other selections or accept the defaults.
5. Click **Create Account**.
6. Add a bank account; see “[Adding a Bank Account](#)” on page 17.

Confirming an Email Address

To confirm an email address for a test user:

1. Log in to Developer Central.
2. Click the **Email** tab.
3. On the next page, click the email link to open the email window.
4. In the window that appears, find the email for the new test user with a subject line of “Activate Your PayPal Account!”
5. Open the email and click the confirmation link.

“Unverified” Account Status

At this point your new test users exist and have confirmed email addresses. However, they have an “Unverified” status. To set a user to a status of “Verified”, add a bank account for it. For more information, see [“Adding a Bank Account” on page 17](#).

Adding a Funding Source

To test transactions, you must add a source of funds to your buyer test account. The following sections describe your choices:

1. [“Adding a Bank Account” on page 17](#). You can add bank accounts, but they will not contain funds unless you use Send Money to send the bank account holder money.
2. [“Adding a Credit Card ” on page 18](#). For testing, this is the most efficient way to add funds.

NOTE: No money or funds are actually transferred in the Sandbox; however, to protect confidentiality, you should not use actual credit card numbers or bank accounts if you allow other people to log in to your Developer Central account.

Adding a Bank Account

You add a bank account to the Sandbox test account representing a customer or buyer so that you can test transactions between the buyer’s account and another account; typically, the other account is a business account that represents yourself as a merchant. Adding a bank account also changes the account status from “Unverified” to “Verified.”

The bank account is a source of funds for the test user’s PayPal account, and thus for transactions between that test user and other test users. A test user can have multiple bank accounts, but at least one is required in order to verify the test user.

To add a bank account for a new test user:

1. After logging into Developer Central, launch the Sandbox as the test user.
2. Navigate to **My Account > Overview**.

3. In the Overview window, click the **Add Bank Account** link in the Activate Account box on the left.
4. In the Add Bank Account window:
 - Enter a fake bank account number
The account number must consist of 1-10 numeric values
 - Enter a fake Bankleitzahl
The Bankleitzahl must consist of 8 numeric values
 - Make a note of the test bank account number, because it will be handy to have when you do your testing.
 - Click **Add Bank Account**.
5. In the resulting success page, click **Continue** at the bottom.
The **My Account > Overview** page opens.
6. Click the **Confirm Bank Account** link in the **Activate Account** box at the left side.
7. In the Confirm Bank Account window, click **Submit**.
To create additional bank accounts for an existing test user:
 1. Repeat the same steps 1-7 as described above for adding a bank account for a test user.

Adding a Credit Card

A credit card is a source of funds for the test user's PayPal account, and thus can be used for transactions between that test user and other test users. A test user can have multiple credit cards.

Test credit card numbers cannot be used to pay for real-world transactions.

NOTE: Skip this step if you signed up for a Business Account. Business Accounts automatically get a credit card.

Adding a Credit Card for a New Test User

To add a credit card account for a new test user:

1. After logging into Developer Central, launch the Sandbox as the test user.
2. Navigate to **My Account > Overview**.
3. In the **Overview** window, click the **Add Credit Card** link in the **Activate Account** box on the left.
4. In the **Add Credit Card** window, leave all information as it is (automatically generated) and click **Add Credit Card**.

Adding More Credit Cards for an Existing Test User

To create additional credit card accounts for an already existing test user:

1. Log in to the Sandbox as the test user.
2. Navigate to **My Account > Profile**.
3. Under the Financial Information header, click the Credit Cards link.
4. In the **Credit Cards** window, click the **Add** button.
5. In the **Add Credit Card** window, leave the automatically generated information as is.
6. Make a note of the credit card number for your use in later testing.
7. Click **Add Credit Card**.

Generating a Credit Card Number to Test PayPal Account Optional

To obtain a test credit card number for testing PayPal Account optional:

1. Log in to the Sandbox as the test user.
2. Navigate to **My Account > Profile**.
3. Under the **Financial Information** header, click the **Credit Cards** link.

Make a note of the credit card number for your use in later testing.

4

Testing PayPal Website Features (non-API)

This chapter describes PayPal products features you can test in the Sandbox without PayPal APIs:

- **Website Payments with Buy Now Buttons:** Use the Sandbox to test accepting PayPal as a payment mechanism on a website.
- **Payments with a Third Party Shopping Cart (Cart upload feature):** Use the Sandbox to test the purchase of multiple items in a single transaction using a single payment.
- **Instant Payment Notification (IPN):** Use the Sandbox to test IPN for updates and payment notifications.
- **Refunds:** Use the Sandbox to test refunding payments from a test buyer.
- **Subscriptions:** Use the Sandbox to test subscription buttons.

IMPORTANT: To execute test transactions on Sandbox you need to complete a purchase as a test buyer with your buyer test account. Typically, you go through your website purchase flow as a buyer. You must ensure that you execute your test on `www.sandbox.paypal.com` instead of `www.paypal.com`.

Website Payments with Buy Now Button

You can use the Sandbox to familiarize yourself with the PayPal **Buy Now** button, with which you can associate PayPal with a specific item you sell on your website.

To create a test **Buy Now** button:

1. After logging into Developer Central, launch the Sandbox as the test seller account.
2. Go to the **Merchant Services** tab.
3. Select the **Buy Now Buttons** link under the **Key Features** heading to get to the Button Factory. You can also search the Help for “Button Factory.”
4. Follow the online instructions to create a Buy Now button. For more information, see the [Website Payments Standard Integration Guide](#).
5. Copy and paste the code into your web page file wherever you would like the button image to appear. Typically, the button should be located next to the description of the item or service. Your web page does not have to be published to your web server for you to check the button placement; it can be on your own local hard drive.

IMPORTANT: You must change the form action to redirect to the Sandbox, using the following URL:

Testing PayPal Website Features (non-API)

Website Payments with Buy Now Button

```
https://www.sandbox.paypal.com/cgi-bin/webscr"  
method="post"
```

Use the PayPal Help link to answer related questions, such as “How do I make a Buy Now Button compatible with the Shopping Cart feature?” For general information, see <https://www.paypal.com/pdn-item>. For general information about shopping cart purchases, see <https://www.paypal.com/shoppingcart>. For general information about subscriptions, see <https://www.paypal.com/pdn-recurring>.

Encrypted Website Payments

The Sandbox also supports Encrypted Website Payments (EWP), as does the PayPal SDK console.

For information about what EWP is and how to use it, see the [Website Payments Standard Integration Guide](#)

For information about using the PayPal SDK console to generate EWP HTML, see the [PayPal SDK Guide](#) for any of the supported platforms.

Testing Payments with Buy Now Button

For the purposes of testing the Buy Now button, your web page does not need to be published to your web server. It can reside on your local hard drive. However, you do need to be logged in to the Sandbox.

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Enter Sandbox Test Site**.
2. Open the HTML file containing the Buy Now Button.
3. Click the **Buy Now** Button.
4. Log in using your test buyer account.
5. Follow the on-screen instructions to complete your test payment.

Verifying a Test Payment

1. Log in to Developer Central.
2. Click the **Test Email** tab.

Your Sandbox inbox shows payment confirmation email messages for the seller and buyer.

3. To further verify that the payment was successful:
 - Check your web server for IPN notifications related to the payment.
 - Launch the Sandbox as your test buyer or seller account and navigate to **My Account > Overview** to see the transaction in your **Recent Activity**.

Testing Payments with a Third Party Shopping Cart (Cart upload feature)

Setting your PayPal integration to link to Sandbox:

- In order to test your PayPal integration you need to change the link location to PayPal Sandbox instead of linking to the PayPal live system.
- Linking to sandbox: `<form action="https://www.sandbox.paypal.com/cgi-bin/webscr" method="post">`
- Linking to live PayPal system: `<form action="https://www.paypal.com/cgi-bin/webscr" method="post">`

Doing the transaction:

For the purposes of testing your shopping cart installation, your web page does not need to be published to your web server. It can reside on your local hard drive. However, you do need to be logged in to the Sandbox.

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Enter Sandbox Test Site**.
2. Open the HTML file containing the **Pay** Button.
3. Click the **Pay** Button.
4. Log in using your test buyer account.
5. Follow the on-screen instructions to complete your test payment.

Handling Pending Transactions

Transactions typically are credited to your PayPal account instantly after the buyer completes the transaction; however, the user might select a payment method that is not completed instantly. In these cases, the transaction goes into a pending state and the transaction is completed after a couple of days. The following sections describe how to set up pending status transactions that can either be completed or canceled.

Creating a Pending Transaction

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Enter Sandbox Test Site**.
2. Open the HTML file containing the Pay Button.
3. Click the **Pay** Button.
4. Log in using your test buyer account.
5. On the **Review Purchase Page** click on the link **Change** under funding method.
6. Select **Electronic Funds Transfer** as funding method and click **Continue**.

7. Click **Pay** to create the transaction.

To verify the creation of the transaction, see “[Verifying a Test Payment](#)” on page 22.

Completing or Canceling a Pending Transaction

To complete a test transactions funded by Electronic Funds Transfer:

1. Log in to your buyer Sandbox account.
2. Go to **My Account > History**.
3. Click on **Details** link (in the Details column) of the transaction “Transfer from bank account.”
4. In the Transaction Detail window, there is a link **Clear Transaction** to simulate actual completion of Electronic Funds Transfer. Click this link to complete the transaction. This link appears only in the Sandbox.
5. Click **Return to Log** to see the transfer completed and the money in the Sandbox account.

The **My Account > Overview** page opens.

To cancel a test transactions funded by Electronic Funds Transfer:

1. Log in to your buyer Sandbox account.
2. Go to **My Account > History**.
3. Click on **Details** link (in the Details column) of the transaction “Payment to.”
4. In the Transaction Detail window, there is a button **Cancel Transaction** to simulate actual cancellation of Electronic Funds Transfer. Click this link to cancel the transaction.
5. Click **Return to Log** to see the transfer completed and the money in the Sandbox account.

You can verify the completion/cancellation of the transaction as described in the section “[Verifying a Test Payment](#)” on page 22.

Instant Payment Notification (IPN)

You can use the Sandbox to test Instant Payment Notification, such as the PayPal **Buy Now** button or reversals.

Setting up IPN in the Sandbox

For information about implementing IPN, see the following:

- Technical overview at <https://www.paypal.com/cgi-bin/webscr?cmd=p/xcl/rec/ipn-techview-outside>

- [Order Management Integration Guide](#)

test_ipn Variable

The Sandbox sets the variable `test_ipn` with a value of 1 in the HTTP response back to your IPN page. The purpose of this variable is to clearly differentiate between live and Sandbox IPN, so you can write your processing programs to work with either live or Sandbox IPN. If the `test_ipn` variable is not present in the HTTP response, you are working with Live PayPal.

To enable IPN for a test user:

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Enter Sandbox Test Site**.
2. Click the **Profile** subtab.
3. Click the **Instant Payment Notification Preferences** link in the **Selling Preferences** column.
4. Click **Edit**.
5. Click the checkbox and enter the URL at which you would like to receive your IPN notifications.
6. Click **Save**.
7. Install IPN on your web server. You might want to start with one of PayPal's source code samples available at <https://www.paypal.com/ipn> under the **Code Samples** section. There are source code samples for the several programming languages and development environments. For further information, see the *Order Management Integration Guide*.

Testing Refunds

Issuing a Test Refund

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Enter Sandbox Test Site**.
2. Click the **History** subtab.
3. Find the payment transaction you want to refund.
4. Click the **Details** link for the transaction.
5. Click the **Refund Payment** link on the **Transaction Details** page.
6. Enter the refund amount and click **Submit**.
7. Confirm the refund amount and click **Process Refund**.

Verifying a Test Refund

To verify a test refund, you must have already made a test payment.

1. Log in to Developer Central.
2. Click the **Test Email** tab.

Your Sandbox inbox shows refund confirmation email messages for the seller and buyer.

3. To further verify that the refund was successful:
 - Check your Web server for IPN notifications related to the refund.
 - Launch the Sandbox as your test buyer or seller account and navigate to **My Account > Overview** to see the transaction in your **Recent Activity**.

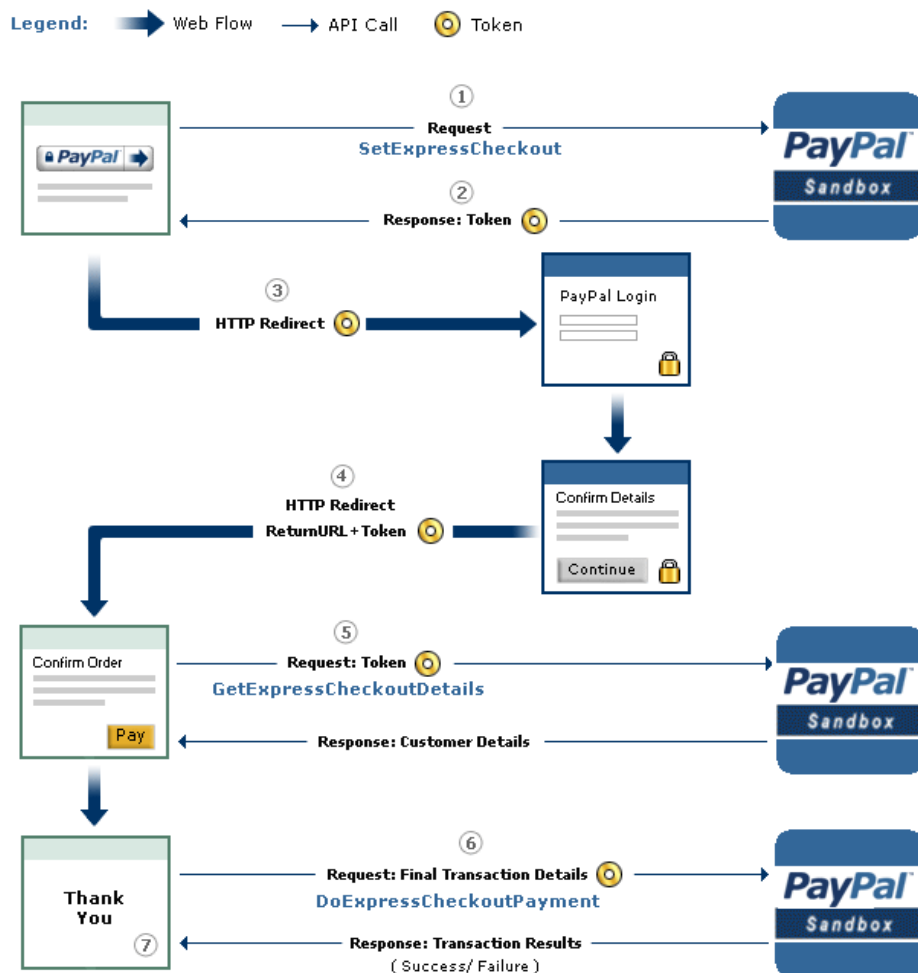
5

Testing PayPal NVP APIs

This chapter describes how to test the Express Checkout name-value pair (NVP) API in the Sandbox. For more sophisticated examples, PayPal recommends you use the PayPal SDK that matches your environment, such as PHP or ASP. You can also use this chapter for ideas on how to establish a general testing procedure for PayPal APIs called from your site.

Testing Express Checkout

The following diagram shows the Express Checkout flow, which uses the Sandbox as the API server. The pages on the left represent your site.



NOTE: For information about Express Checkout, see the *Integrationshandbuch Express-kaufabwicklung* and the *PayPal NVP API Reference*.

The following steps match the circled numbers in the diagram. Perform the actions in each step to test Express Checkout.

1. Invoke a form on your site that calls the `SetExpressCheckout` API on the Sandbox. To invoke the API, set form fields whose names match the NVP names of the fields you want to set, specify their corresponding values, and then post the form to `https://api-3t.sandbox.paypal.com/nvp`, as shown in the following example:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
  <input type=hidden name=USER value= API_username>
  <input type=hidden name=PWD value= API_password>
  <input type=hidden name=SIGNATURE value= API_signature>
  <input type=hidden name=VERSION value=2.3>
  <input type=hidden name=PAYMENTACTION value=Authorization>
  <input name=AMT value=19.95>
  <input type=hidden name=RETURNURL
    value=http://www.YourReturnURL.com>
  <input type=hidden name=CANCELURL
    value=http://www.YourCancelURL.com>
  <input type=submit name=METHOD value=SetExpressCheckout>
</form>
```

NOTE: The API username is a Sandbox business test account for which a signature exists. See the Test Certificates tab of the Sandbox to obtain a signature.

2. PayPal responds with a message, such as the one shown below. Note the status, which should include ACK set to `Success`, and a token that is used in subsequent steps.

```
TIMESTAMP=2007%2d04%2d05T23%3a23%3a07Z
&CORRELATIONID=63cdac0b67b50
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
```

3. If the operation was successful, use the token and redirect your browser to the Sandbox, as follows:

```
https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-checkout
&token=EC-1NK66318YB717835M
```

NOTE: You may need to replace hexadecimal codes with ASCII codes; for example, you may need to replace `%2d` in the token with a hyphen (-).

4. Log into the Sandbox and confirm details. You must log in to Developer Central before you log in to a Sandbox test account. You log in to the test account that represents the buyer, not the *API_username* business test account that represents you as the merchant.

When you confirm, the Sandbox redirects your browser to the return URL you specified when calling `SetExpressCheckout`, as in the following example:

`http://www.YourReturnURL.com/
?token=EC-1NK66318YB717835M&PayerID=7AKUSARZ7SAT8`

5. Invoke a form on your site that calls the GetExpressCheckout API on the Sandbox:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
  <input type=hidden name=USER value= API_username>
  <input type=hidden name=PWD value= API_password>
  <input type=hidden name=SIGNATURE value= API_signature>
  <input type=hidden name=VERSION value=2.3>
  <input name=TOKEN value=EC-1NK66318YB717835M>
  <input type=submit name=METHOD value=GetExpressCheckoutDetails>
</form>
```

If the operation was successful, the GetExpressCheckout API returns information about the payer, such as the following information:

```
TIMESTAMP=2007%2d04%2d05T23%3a44%3a11Z
&CORRELATIONID=6b174e9bac3b3
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&EMAIL=YourSandboxBuyerAccountEmail
&PAYERID=7AKUSARZ7SAT8
&PAYERSTATUS=verified
&FIRSTNAME=...
&LASTNAME=...
&COUNTRYCODE=US
&BUSINESS=...
&SHIPTONAME=...
&SHIPTOSTREET=...
&SHIPTOCITY=...
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOCOUNTRYNAME=United%20States
&SHIPTOZIP=94666
&ADDRESSID=...
&ADDRESSSTATUS=Confirmed
```

6. Invoke a form on your site that calls the DoExpressCheckoutPayment API on the Sandbox:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
  <input type=hidden name=USER value= API_username>
  <input type=hidden name=PWD value= API_password>
  <input type=hidden name=SIGNATURE value= API_signature>
  <input type=hidden name=VERSION value=2.3>
  <input type=hidden name=PAYMENTACTION value=Authorization>
  <input type=hidden name=PAYERID value=7AKUSARZ7SAT8>
  <input type=hidden name=TOKEN value= EC%2d1NK66318YB717835M>
  <input type=hidden name=AMT value= 19.95>
  <input type=submit name=METHOD value=DoExpressCheckoutPayment>
```

</form>

7. If the operation was successful, the response should include ACK set to Success, as follows:

```
TIMESTAMP=2007%2d04%2d05T23%3a30%3a16Z
&CORRELATIONID=333fb808bb23
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&TRANSACTIONID=043144440L487742J
&TRANSACTIONTYPE=expresscheckout
&PAYMENTTYPE=instant
&ORDERTIME=2007%2d04%2d05T23%3a30%3a14Z
&AMT=19%2e95
&CURRENCYCODE=USD
&TAXAMT=0%2e00
&PAYMENTSTATUS=Pending
&PENDINGREASON=authorization
&REASONCODE=None
```

Testing giropay and EFT Transactions

To test giropay and EFT transactions with Express Checkout, perform the following actions:

NOTE: The steps for testing Express Checkout with giropay and EFT are the same as for any other funding method; however, an additional redirect to paypal.com may be required, as specified in the return to the GetExpressCheckoutDetails and DoExpressCheckout APIs.

1. Invoke a form on your site that calls the SetExpressCheckout API on the Sandbox. To invoke the API, set form fields whose names match the NVP names of the fields you want to set, specify their corresponding values, and then post the form to `https://api-3t.sandbox.paypal.com/nvp`, as shown in the following example:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
  <input type=hidden name=USER value= API_username>
  <input type=hidden name=PWD value= API_password>
  <input type=hidden name=SIGNATURE value= API_signature>
  <input type=hidden name=VERSION value=2.3>
  <input type=hidden name=PAYMENTACTION value=Authorization>
  <input name=AMT value=19.95>
  <input name=CURRENCYCODE value=EUR>
  <input type=hidden name=RETURNURL
    value=http://www.YourReturnURL.com>
  <input type=hidden name=CANCELURL
    value=http://www.YourCancelURL.com>
  <input type=submit name=METHOD value=SetExpressCheckout>
```

</form>

NOTE: The API username is a Sandbox business test account for which a signature exists. See the Test Certificates tab of the Sandbox to obtain a signature.

2. PayPal responds with a message, such as the one shown below. Note the status, which should include ACK set to Success, and a token that is used in subsequent steps.

```
TIMESTAMP=2007%2d04%2d05T23%3a23%3a07Z
&CORRELATIONID=63cdac0b67b50
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
```

3. If the operation was successful, use the token and redirect your browser to the Sandbox, as follows:

```
https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-checkout
&token=EC-1NK66318YB717835M
```

NOTE: Use the value of the token returned in the previous step. You may need to replace hexadecimal codes with ASCII codes; for example, you may need to replace %2d in the token with a hyphen (-).

4. Log into the buyer's Sandbox test account and choose giropay or EFT as the funding method by selecting **Change** funding method and confirming the transaction.

When you confirm, the Sandbox redirects your browser to the return URL you specified when calling SetExpressCheckout.

5. Invoke a form on your site that calls the GetExpressCheckoutDetails API on the Sandbox using the token that you received in the SetExpressCheckout response:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
  <input type=hidden name=USER value= API_username>
  <input type=hidden name=PWD value= API_password>
  <input type=hidden name=SIGNATURE value= API_signature>
  <input type=hidden name=VERSION value=2.3>
  <input name=TOKEN value=EC-1NK66318YB717835M>
  <input type=submit name=METHOD value=GetExpressCheckoutDetails>
</form>
```

If the operation was successful, the GetExpressCheckout API returns information about the payer:

```
TIMESTAMP=2007%2d04%2d05T23%3a44%3a11Z
&CORRELATIONID=6b174e9bac3b3
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&EMAIL=YourSandboxBuyerAccountEmail
&PAYERID=7AKUSARZ7SAT8
&PAYERSTATUS=verified
&FIRSTNAME=...
&LASTNAME=...
&COUNTRYCODE=US
&BUSINESS=...
&SHIPTONAME=...
&SHIPTOSTREET=...
&SHIPTOCITY=...
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOCOUNTRYNAME=United%20States
&SHIPTOZIP=94666
&ADDRESSID=...
&ADDRESSSTATUS=Confirmed
&redirectrequired=true
```

NOTE: When `redirectrequired` is true, you must redirect to `paypal.com` after the buyer has completed the order on your final page, as shown in Step 8. The `redirectrequired` field is only set to true if the funding source is giropay or EFT.

6. Invoke a form on your site that calls the `DoExpressCheckoutPayment` API on the Sandbox:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
  <input type=hidden name=USER value= API_username>
  <input type=hidden name=PWD value= API_password>
  <input type=hidden name=SIGNATURE value= API_signature>
  <input type=hidden name=VERSION value=2.3>
  <input type=hidden name=PAYMENTACTION value=Sale>
  <input type=hidden name=PAYERID value=7AKUSARZ7SAT8>
  <input type=hidden name=TOKEN value= EC%2d1NK66318YB717835M>
  <input type=hidden name=AMT value= 19.95>
  <input type=submit name=METHOD value=DoExpressCheckoutPayment>
</form>
```

7. If the operation was successful, the response should include `ACK` set to `Success`, as follows:

TIMESTAMP=2007%2d04%2d05T23%3a30%3a16Z
&CORRELATIONID=333fb808bb23
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&TRANSACTIONID=043144440L487742J
&TRANSACTIONTYPE=expresscheckout
&PAYMENTTYPE=instant
&ORDERTIME=2007%2d04%2d05T23%3a30%3a14Z
&AMT=19%2e95
&CURRENCYCODE=EUR
&TAXAMT=0%2e00
&REASONCODE=None
&redirectrequired=true

8. If calling the DoExpressCheckout API returned **redirectrequired** as true, you must redirect to the Sandbox to complete the checkout, as follows:

`https://www.sandbox.paypal.com/webscr
?cmd=_complete-express-checkout&token=EC-6A1952025E6618811`

9. Check for an IPN notification that indicates success. For more information, see [“Instant Payment Notification \(IPN\)” on page 24](#).

6

Testing Error Conditions

In default operation, the Sandbox mimics the live PayPal site as closely as possible, which means that an error can be replicated only by creating the exact conditions and sequence of events to raise an error. This *positive test* environment is well-suited for testing logic that follows the typical error-free path; however, it can be difficult to raise error conditions and test logic to handle errors.

The Sandbox can be set to allow *negative testing*, which enables you to simulate an error. You can test against errors that result from calling a PayPal API.

IMPORTANT: Negative testing is only available for Version 2.4 and later of PayPal APIs.

You raise an error condition by setting a value in a field passed to an API or setting a value in a field submitted to Virtual Terminal. The value triggers a specific error condition. Negative testing is available only in the Sandbox; you cannot force or simulate an error on the live site.

You must create a Business test account and enable negative testing; otherwise, setting a value in the API or transaction will not raise an error unless the error would be raised in the default positive test environment. To enable negative testing, set **Test Mode** to **Enabled**. The following screen shows two Business accounts. The first test account enables negative testing; the second account disables negative testing.

Home

Sandbox

Test Certificates



Email

Forums

Help Center

Sandbox

The Sandbox is a safe testing environment where you can test PayPal payments and API calls. The Sandbox is a mirror of the real PayPal site except that real financial transactions are not made. [Learn More...](#)

Test Accounts								Create Account
User	Type	CountryCode	Country	Balance	Confirmed	Verified	Test Mode	
	Business	us	U.S.	0.00 USD	Yes	Unverified	Enabled	
	Business	us	U.S.	0.00 USD	Yes	Unverified	Disabled	

Severe error conditions, such as bad arguments or invalid login, preempt negative testing because the error cannot be handled by either negative testing or positive testing. In these cases, the error condition for positive testing is raised, regardless of whether the account was enabled for negative testing.

API Testing

For APIs, you trigger an error condition by setting a field to the value of the error you want to trigger. The value you specify depends on the kind of field:

- for amount-related fields, specify a value as a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error 10755
- for other kinds of fields, specify the actual PP API error; for example, 10755 triggers PP API error 10755

The following table identifies the API, the NVP name or SOAP element of the field that triggers the error, and a description of how to set the value in the field:

TABLE 6.1 API Fields That Trigger Error Conditions

API Name	NVP Field Name	SOAP Element	Description
RefundTransaction	AMT	Amount	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error code 10755.
GetTransaction Details	TRANSACTIONID	TransactionID	Specify the error code to trigger as all digits in the field; for example, an ID of 10755 triggers PP API error code 10755.
TransactionSearch	INVNUM	InvoiceID	Specify the error code to trigger as all digits in the field; for example, an ID of 10755 triggers PP API error code 10755.
SetExpressCheckout	MAXAMT	MaxAmount	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error code 10755.
GetExpressCheckout Details	TOKEN	Token	Specify the error code to trigger as all digits in the field; for example, a token value of 10755 triggers PP API error code 10755.
DoExpressCheckout	TOKEN	Token	Specify the error code to trigger as all digits in the field; for example, a token value of 10755 triggers PP API error code 10755.
MassPay	EMAILSUBJECT	EmailSubject	Specify the error code to trigger as all digits in the field; for example, a subject of 10755 triggers PP API error code 10755.

NOTE: If the trigger value is not a valid error code for the API being tested, positive testing occurs for the request, which might result in another error occurring.

Negative Testing Using an Amount-Related Trigger Field

Consider an example that sets up testing for error 10623 for DoAuthorization, in which the error code is specified in the AMT field:

```
METHOD=DoAuthorization
&TRANSACTIONID=O-1GU0288989807143B&
AMT=106.23&
TRANSACTIONENTITY=Order&
VERSION=2.4&
USER=username&
PWD=password&
SIGNATURE=signature
```

The request invokes the following response:

```
TIMESTAMP=2007%2d04%2d04T03%3a10%3a19Z&
CORRELATIONID=447d121150529&
ACK=Failure&
L_ERRORCODE0=10623&
L_SHORTMESSAGE0=Maximum%20number%20of%20authorization%20allowed%20for%20the%20order%20is%20reached%2e&
L_LONGMESSAGE0=Maximum%20number%20of%20authorization%20allowed%20for%20the%20order%20is%20reached%2e&
L_SEVERITYCODE0=Error&
VERSION=2%2e400000&
BUILD=1%2e0006
```

Negative Testing Using a Non-Amount Trigger Field

Consider an example that sets up testing for error 10603 for DoVoid, in which the error code is specified in the AUTHORIZATIONID field:

```
METHOD=DoVoid&
AUTHORIZATIONID=10603&
VERSION=2.4&
USER=username&
PWD=password&
SIGNATURE=signature
```

The request invokes the following response:

```
TIMESTAMP=2007%2d04%2d04T03%3a10%3a22Z&
CORRELATIONID=51b0c5054dee6&
ACK=Failure&
L_ERRORCODE0=10603&
L_SHORTMESSAGE0=The%20buyer%20is%20restricted%2e&
L_LONGMESSAGE0=The%20buyer%20account%20is%20restricted%2e&
L_SEVERITYCODE0=Error
&VERSION=2%2e400000&
BUILD=1%2e0006
```

Negative Testing With Multiple Messages

Consider an example that sets up testing for error 10009 for RefundTransaction, which returns 14 possible error message sets:

```
METHOD=RefundTransaction&
TRANSACTIONID=asdf&
REFUNDTYPE=Partial&
AMT=100.09&
VERSION=2.4&
USER=username&
PWD=password&
SIGNATURE=signature
```

The request invokes the following response:

```
TIMESTAMP=2007%2d04%2d04T03%3a10%3a23Z&
CORRELATIONID=81ccc18eaec49&
ACK=Failure&
L_ERRORCODE0=10009&
L_SHORTMESSAGE0=Transaction%20refused&
L_LONGMESSAGE0=You%20can%20not%20refund%20this%20type%20of%20transaction&
L_SEVERITYCODE0=Error&
L_ERRORCODE1=10009&
L_SHORTMESSAGE1=Transaction%20refused&
L_LONGMESSAGE1=You%20are%20over%20the%20time%20limit%20to%20perform%20a%20r
efund%20on%20this%20transaction&
L_SEVERITYCODE1=Error&
L_ERRORCODE2=10009&
L_SHORTMESSAGE2=Transaction%20refused&
L_LONGMESSAGE2=Account%20is%20restricted&
L_SEVERITYCODE2=Error&
...
L_ERRORCODE13=10009&
L_SHORTMESSAGE13=Transaction%20refused&
L_LONGMESSAGE13=The%20partial%20refund%20amount%20must%20be%20less%20than%2
0or%20equal%20to%20the%20remaining%20amount&
L_SEVERITYCODE13=Error&
VERSION=2%2e400000&
BUILD=1%2e0006
```

7

Technical Support

Depending on the PayPal product you need assistance with, contact either Customer Service or Developer Technical Support.

Contacting Customer Service for Live PayPal Website Help

Use PayPal Help to find an answer to any problem you might encounter with live products such as Website Payments or Instant Payment Notification.

To contact Customer Service about issues with the Live PayPal website:

1. Go to <https://www.paypal.com/>
2. Click **Help** in the upper right corner of the page.
3. Click **Contact Us** in the lower left of the page.
4. Chose either **Help by Email** or **Help by Phone**.
5. Follow the remaining instructions.

Contacting Developer Technical Support for API Help

For information about PayPal Web Services API, Developer Central, and using the Sandbox, refer to the following resources:

- **Help Center:** In Developer Central, click **Help Center** to access developer manuals and links.
- **Forums:** In Developer Central, click **Forums** to share information with the PayPal developer community.
- **Online Developer Support Portal** at <https://paypal.com/mts>

Use the **Forums** first to find answers about any questions or problems you might have. Another developer might have already posted information about your question or problem.

To contact Developer Technical Support about the PayPal Web Services API:

1. Log in to your account at <https://developer.paypal.com/> by entering your email address and password in the Member Log In box
2. Click **Help Center** at the bottom of the box on the right side of the page.
3. Click **Email PayPal Support**.

4. Complete the form.



Index

A

- adding a bank account 17
- API server for Sandbox 9, 10
- API testing 35

B

- bank accounts, adding 17
- Business account 16
- Buy Now 21

E

- EFT transactions 30
- email in Sandbox 14
- errors, testing 35

F

- fraud detection in Sandbox 10

G

- giro transactions 30

I

- Instant Payment Notification. See *IPN*.
- Integration Center 9
- IPN
 - ipn_test variable 25
 - technical overview 24
 - testing in Sandbox 25

N

- name-value pair (NVP) APIs 27
- negative testing 35

P

- PayPal Developer Network 9
- PDN 9
- pending transactions 23
- positive test 35
- Premium account 16

S

- Sandbox
 - API server for 9, 10
 - email 14
- SDKs 9
- source code samples for IPN 25

T

- test accounts 15
- test email 14
- test_ipn 25
- testing
 - APIs 27, 35
 - Buy Now 22
 - IPN 25
 - negative 35
 - NVP 27
 - verifying a payment 22
 - verifying a refund 26
 - Website Payments 21

W

- Website Payments 21

