A Developer's Guide to Using the PayPal APIs

Getting Started with

PayPal

Free Sampler

O'REILLY®

Michael Balderas

O'Reilly Ebooks—Your bookshelf on your devices!

When you buy an ebook through oreilly.com, you get lifetime access to the book, and whenever possible we provide it to you in four, DRM-free file formats—PDF, .epub, Kindle-compatible .mobi, and Android .apk ebook—that you can use on the devices of your choice. Our ebook files are fully searchable and you can cut-and-paste and print them. We also alert you when we've updated the files with corrections and additions.

Learn more at http://oreilly.com/ebooks/

You can also purchase O'Reilly ebooks through iTunes, the Android Marketplace, and Amazon.com.

Preface

The surging demand for transaction convenience is being felt across virtually every application delivery model. In this book, Michael Balderas introduces PayPal's APIs with instruction and resources for its integration in different environments including websites and mobile applications.

Goals Of This Book

The goal of this book is to help you understand what PayPal has to offer. Lets face it, your looking to get money from your customers into your bank account as fast as possible, and I want to help you accomplish this. By the end of this book you will have a better understanding of what PayPal is, how PayPal can streamline your Payments, and how to get the most out of PayPal for your particular payment situation.

Who Should Read This Book

Anyone looking for a solution to accepting payments for their goods or services using PayPal. You can be an individual with an open source project looking to accept donations, a multi million dollar corporation, a non profit looking for donations to help a cause, or a software developer writing mobile apps for cell phones. PayPal can provide you with the solutions you need no matter who you are. Code Samples will be provided pirmarily in PHP and some Java. An understanding of using API's is recommended but not required.

How This Book Is Organized

Here is a brief summary of the chapters in the book and what you can expect from each:

Chapter 1
: An introduction to PayPal, Why you should choose Paypal, PayPal Account types and Products.

Chapter 2
: Introduction to PayPal's API and how to get started using it's powerful features for accepting payments.

Chapter 3
: Introduction to Express Checkout and using the API to execute Express Checkout Payments.

Chapter 4
: Introduction to Website Payment Pro and using the API to execute Website Payment Pro Payments.

Chapter 5
: Introduction to Adaptive Payments and using the API to execute Adaptive Payments.

Chapter 6
: Introduction to Mobile Checkout and using the API to execute Mobile Payments.

Chapter 7
: This chapter will cover use cases and sample code referenced throughout the book.

Chapter 8
: This chapter will cover additional Developer resources for using PayPal to answering your Payment needs.

Conventions Used In This Book

The following typographical conventions are used in this book:

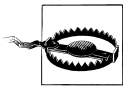
Italic
: Indicates new terms, URLs, email addresses, filenames, and file extensions.

`Constant width`
: Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

`Constant width bold`
: Shows commands or other text that should be typed literally by the user.

Constant width italic
: Shows text that should be replaced with user-supplied values or by values determined by context.

This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission. Selling or distributing a CD-ROM of examples from O'Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product's documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: "*Book Title* by Some Author. Copyright 2008 O'Reilly Media, Inc., 978-0-596-xxxx-x."

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at *permissions@oreilly.com*.

Safari® Books Online

Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at *http://my.safaribooksonline.com*.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707 829-0104 (fax)

We have a web page for this book, where we list errata, examples, and any additional information. You can access this page at:

http://www.oreilly.com/catalog/<catalog page>

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about our books, conferences, Resource Centers, and the O'Reilly Network, see our web site at:

http://www.oreilly.com

CHAPTER 2

Introducing The PayPal API

Overview Of PayPal API

PayPal provides access to it's payments system for developers via it's Name-Value Pair API, refered to as NVP API for the remainder of this book. The NVP API allows a merchant to access PayPal to do the following tasks:

- Accept PayPal during your checkout process via Express Checkout
- Charge a Credit Card during a Direct Payment session
- Capture previous authorized Express Checkout, Direct Payment and Express Checkout Payments
- Reauthorize or void previous authorizations
- Pay single or multiple recipients via Mass Payment
- Issue full or multiple partial refunds
- Search transactions using specified search criteria
- Retrieve details of a specific transaction
- Accept PayPal for multiparty payments
- Accept PayPal for subsc riptions or freemium models

PayPal's NVP API makes it simple to integrate PayPal for payments into your specific web application. You, the merchant, construct an NVP string and post it via HTTPS (HTTP Secure aka SSL) to the PayPal authorization server. PayPal posts back an NVP formated response that you then parse in your web application for the relevant information in regards to the payment. Figure 2-1shows a basic request and response workflow.

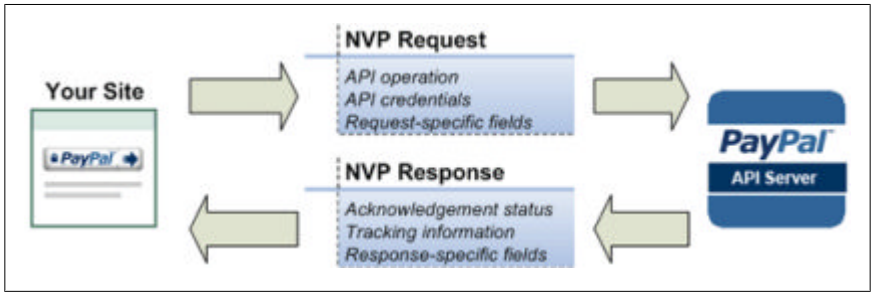


Figure 2-1. Basic NVP Request and Response

The request identifies:

- The Name or Method of the API Operation to be performed and it's version
- PayPal API credentials
- Request-specific information to control the API operation being performed

Adaptive apis require an APP ID as well during the request.

The PayPal API server executes the operation and returns a response containing:

- Acknowledgment status indicating wheter the operation succeeded or failed and if any warning message were returned
- Information that PayPal can use to track execution of the API operation
- Response-specific information required to fulfull the request

Some features of the NVP API, like Express Checkout, require calls to multiple API operations.

Typicall your are required to:

1. Call an API Operation, for example SetExpressCheckout, that sets up the return URL that PayPal uses to redirect your buyer's browser after the buyer finishes on PayPal. Other setup routines can be performed by this same API operation

2. Call additional API operations after receiving the buyers permission on PayPal, such as GetExpressCheckoutDetails or DoExpressCheckoutPayment

Figure 2-2 shows the execution work flow between your application and PayPal.

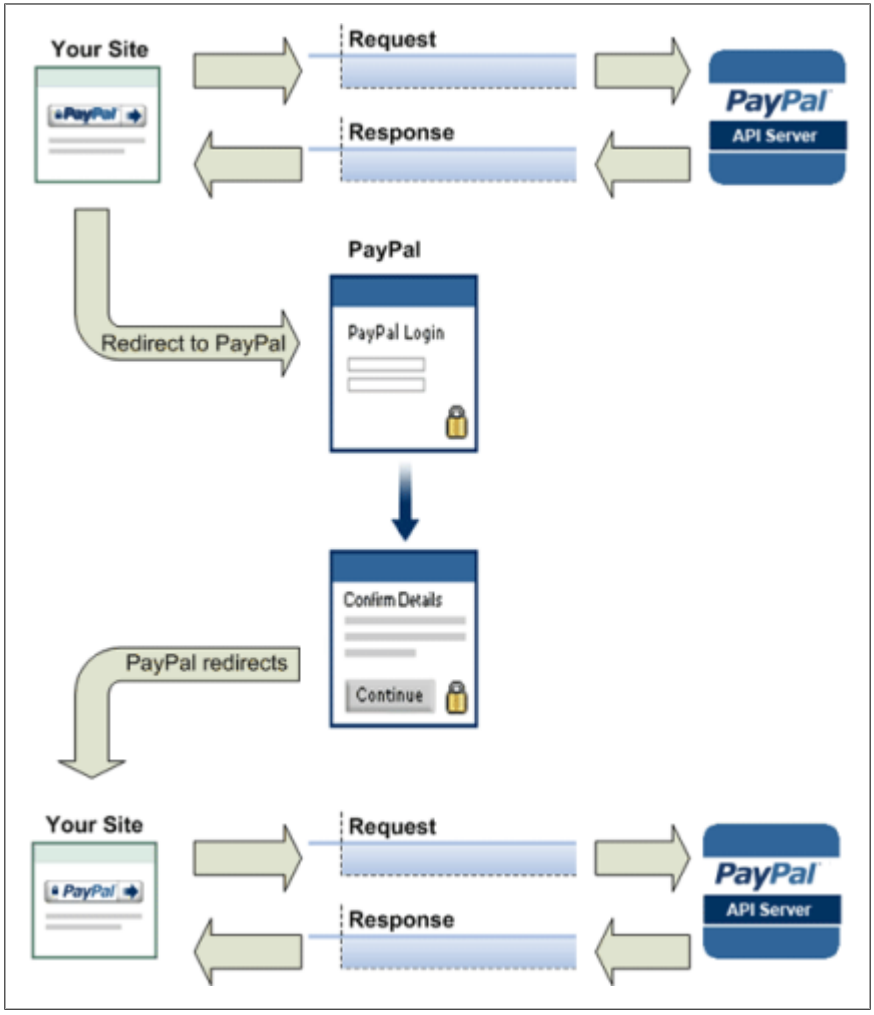


Figure 2-2. Advanced Express Checkout NVP Request and Response

Getting Started

There are two methods to integrate PayPal's NVP API into your application. They are Direct and using a Software Development Kit, refered to as SDK for the remainder of this book. We will focus on doing direct integration into your website and applications.

Direct vs SDK integration.

Direct integration allows you to use the programming language of your choice in communicating via the NVP API. This allows the most flexible approach and allows you direct access to the Name-Value Pair elements of the API. SDK integration provides you with simple functions for integration using the NVP API and SDK's are provided for JAVA, ASP.NET, PHP, Coldfusion and Ruby.

Testing vs Live Implementation

PayPal provides a sandbox enviroment for you to use during your development of your application. The enviroment replicates the live enviroment. The difference is the aspect that true payment processing doesn't occur when using the sandbox. Once you have fully developed and debugged your application you can then switch to the live enviroment and start taking payments. Switching between the two is as simple as changing the target server and the API Credentials used to access the server. The rest of your application will remain unchanged.

I would recommend setting up your API Credentials in seperate files that you include in your application. This will allow you to have your Sandbox Credentials in one file and your Production Credentials in another file. During the development process you can call an include to the Sandbox Credential file and when ready to go live change the include to the Production credential file. This will save you time and headache as you work thru your application and make the transition from Development to Live as simple as a 1 line code change. For added security I would locate these files on your server outside the default webroot so they cannot be called directly from the webbrowser.

Obtaining API Credentials

To get access to the NVP API you first need to establich credentials to identify who you are in order to make use payments get to where they need to go. This

is accomplished using either an API signature or API certificate. You will need two sets of API credentials, one for development and one for production.

Creating API Signature

To develop your application you only need access to the PayPal API sandbox. You can signup for access to the sandbox at *http://developer.paypal.com* or *http://x.com*. Once your account is established you can create your test accounts and obtain your API Credentials. Obtaining your credentials is different for a sandbox account or live account. On a sandbox account it is accomplished thru the following steps:

1. Go to *https://developer.paypal.com* and click "Sign Up Now"
2. Enter the requested information and click "Agree and Submit"
3. PayPal will send you an email to complete the sign-up process
4. Once you confirmed your email address click the "Sign Up Now" to get access to the sandbox
5. Login to your Sandbox Account (this can be accessed directly in the future by going to https://www.sandbox.paypal.com)
6. Click "Test Accounts" link
7. Click "Create Test Account" link
8. Choose Seller for the account type and make other selections (going with the defaults is highly recommended)
9. When using the defaults, API credentials are automatically created
10. Click the API credentials link to access your API credentials

PayPal recommends you use a different login and password then your live PayPal account for you developer account. This will allow you to give someone on your development team access to work in the sandbox to test your application without giving them access to your regular PayPal account.

And on a live account it is accomplished via the following steps:

1. Log into your PayPal Account. Under "My Account" click the "Profile" option
2. Click "API Access"
3. Click "Request API Credentials"
4. Check the "Request API signature" option and click "Agree and Submit"

We will work with using the API Signature method of specifying credentials thru out the book. An API Signature is composed of three elements. Refer to Table 2-1

Table 2-1. NVP API Signature Components

API username	sdk-three_api1.sdk.com
API password	QFZCWN5HZM8VBG7Q
API signature	A-IzJhZZjhg29XQ2qnhapuwxIDzyAZQ92FRP5dqBzVesOkzbdUONzmOU

When you are ready to go live you will need to activate either the Website Payments Standard or Website Payments Pro Product on your account and establish your credentials for that account. You can signup for your account at http://www.paypal.com. (*http://www.paypal.com*)

Website Payments Pro requires additional vetting before being activated.

Creating Name Value Pair (NVP) Request

There are three key steps that your application must accomplish to post to the NVP API. They are URL Encoding, Constructing the Request in a Format the NVP API can interpet, and posting the request via HTTPS to the server.

URL Encoding

Both the request to the PayPal server as well as the response from the server are URL encoded. It is a method of ensuring that you can transmit special characters, characters not typically allowed in a URL and characters that have reserved meanings in a URL. For example:

```
NAME=John Doe&COMPANY= Acme Goods & Services
```

is URL encoded as follows:

```
NAME=John+Doe&Company=Acme+Goods+%26+Services
```

Depending on what application language you are developing in, there will be a specific URL Encode method typically built into the language. Refer to Table 2-2

Table 2-2. URL-Encoding Methods

ASP.NET	Encode	System.Web.HttpUtility.UrlEncode(buffer, Encoding.Default)
Classic ASP	Encode	Server.URLEncode

Java	Encode	java.net.URLEncoder.encode
PHP	Encode	urlencode()
ColdFusion	Encode	URLEncodedFormatstring [, charset]

Request Format

Each NVP API request is composed of required as well as optional parameters and their corresponding values. Parameters are not case sensitive however values like the API Password, PWD, are case sensitive. The Required parameters for all NVP API transactions are USER, PWD, METHOD and VERSION. The METHOD, or type of transaction, you are calling the NVP API to process has an associated VERSION. Together the METHOD and VERSION define the exact behavior of the API Operation you want to be performed. This will be followed by the information posted from your application including things like Item, Quantity and Cost.

API operations can change between versions so it is recommended you retest your application code when you change a version number before going live.

Figure 2-3 outlines the API operation of an NVP Request and Figure 2-4 shows the same transaction with credentials provided.

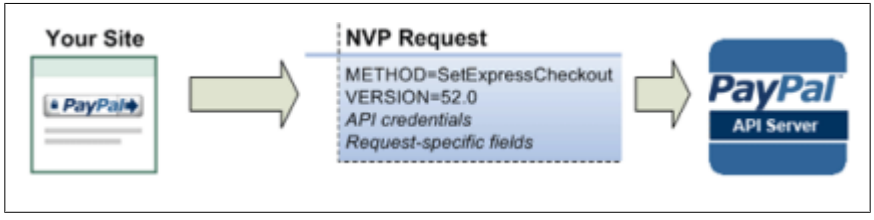


Figure 2-3. NVP Request

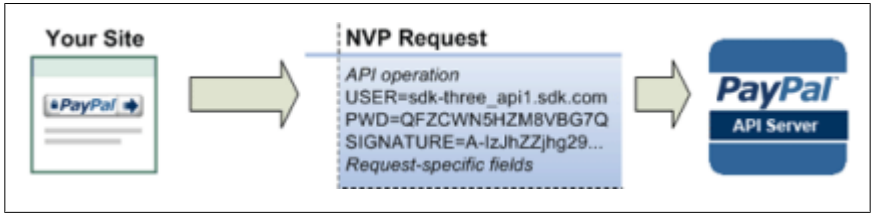


Figure 2-4. NVP Request with Credentials

Putting it Together

Now that we have the basic elements laid out lets put together a sample url Encoded NVP Request via PHP.

Example 2-1. developercredentials.php

```
<?php
//PayPal NVP API Test Developer Credentials//
$paypaluser = sdk-three_api1.sdk.com;
$paypalpwd = QFZCWN5HZM8VBG7Q;
$paypalsig = A-IzJhZZjhg29XQ2qnhapuwxIDzyAZQ92FRP5dqBzVesOkzbdUONzmOU;
$paypalserver = api-3t.sandbox.paypal.com/nvp
?>
```

Example 2-2. simpletransactionrequestprocessor.php

```
<?php
// PayPal NVP API Simple Transaction Request Processor//
// Include the developercredentials.php file for relevant information
include("../path/outside/webroot/developercredentials.php");
// Build the credentials format of the Request String
$credentials= "USER=$paypaluser&PWD=$paypalpwd&SIGNATURE=$paypalsig";
// Designate the API Method we are calling to have handled
$method = api_method_to_use;
$version = method_version_to_use;
// Build Initial Request string
$request = $method."&".$version."&".$credentials;
// Walk the posted form elements to gather additional information
// to pass URLEncoded to API via the request string
foreach ($_POST as $key => $value){
$value = urlencode(stripslashes($value));
$request. = "&$key=$value";
};
//build transacation and execute via curl
$ch = curl_init();
// Ensure Communication is done via SSL and over a fully verified
// SSL key and certificate
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, TRUE);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, TRUE);
// Return Response as a String from Server
curl_setopt($ch, CURL_RETURNTRANSFER, 1);
// Post Values to Server via URLEncoded string
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $request);
//Execute Request
$response = curl_exec($ch);
?>
```

Notice that in Example 2-2 we reference the *developercredentials.php* file from a path outside the webroot. As stated earlier this will ensure that no one can possibly directly access your Credentials File from their Webbrowser and ensures that this information stays secure. If we were satisified with this code and wanted to go to production we would then change the path to the file of our Production Credentials.

Parsing a NVP Response

When it comes to Parsing a NVP response, your application really only has to accomplish one major step, and that is URL Decoding.

URL Decoding

URL Decoding the response from PayPal is basically just the reverse of Encoding the Values to pass to paypal. For Example:

```
NAME=John+Doe&Company=Acme+Goods+%26+Services
```

is Decoded as follows:

```
NAME=John Doe&COMPANY= Acme Goods & Services
```

As with URL Encoding, application languages have a URL Decode method typically build into the language. Refer to Table 2-3

Table 2-3. URL-Encoding Methods

ASP.NET	Decode	System.Web.HttpUtility.UrlDecode(buffer, Encoding.Default)
Classic ASP	Decode	No built-in function. Several implementation examples are available on the Internet.
Java	Decode	java.net.URLDecoder.decode
PHP	Decode	urldecode()
ColdFusion	Decode	URLDecodeurlEncodedString[, charset])

Response Format

Each NVP API response is composed of an Acknowledgement or ACK, a Timestamp, a CorreleationID unique to the transaction and a Build number stating the API version used to process the transaction. It will then be followed by a series of Name/Value pairs of transaction data you can parse and handle accordingly in your application. For example you may want to display the response information to your customer. The Acknowledment will be one of the responses outlined in Table 2-4

Table 2-4. ACK Parameter Values

Type of Response	Value
Successful response	Success , SuccessWithWarning
Partially Successful response (only relavent for parallel payments). Some of the payments were successful, others were not.	PartialSuccess
Error Response Code	Failure , FailureWithWarning , Warning

Putting it together

Now that we know how the Response is formatted we can then extend the *simpletransactionrequestprocessor.php* to handle the information returned in the `$response` string.

Example 2-3. simpletransactionrequestprocessor.php

```
//Parse $Response and handle values
$decoderesponse = explode ('&', $response);

foreach($decoderesponse as $key => $value){

    switch ($key){
        case "ACK":
        $ack = htmlspecialchars(urldecode($value));
        break;
        case "var1":
        $var1 = htmlspecialchars(urldecode($value));
        break;
        default:
        break;
        }
    }
//Your code to display or handle values returned.........
```

Want to read more?

You can find this book at **oreilly.com**
in print or ebook format.

It's also available at your favorite book retailer,
including iTunes, the Android Market, Amazon,
and Barnes & Noble.

O'REILLY®

Spreading the knowledge of innovators

oreilly.com