

PAYPAL HACKS™

100 Industrial-Strength Tips & Tools



O'REILLY®

*Shannon Sofield,
Dave Nielsen & Dave Burchell*

HACK
#88

Make Your First API Call

Make your first API call by issuing a refund from the command line.

As a programmer, you know that web services are the “next big thing.” They’re supposed to make it easy for two computers to exchange information. PayPal Web Services, however, handle money and therefore require an extra level of security. The extra layers are quite easy to implement, but you’ll need to take the following configuration steps prior to executing your first call:

1. Set up an SSL certificate issued by PayPal.
2. Install Simple Object Access Protocol (SOAP) libraries or set up a web reference to SOAP-enable your application

Setting Up the SSL Certificate

Your web site might already have an SSL certificate that it uses for secure communication, but at the time of this writing, PayPal does not support using certificates from other certificate authorities (CAs). This means that you’ll need to generate an SSL certificate from the [PayPal Sandbox \[Hack #87\]](#), and then later, the PayPal live site when your application goes live [\[Hack #100\]](#). Here’s how to request an API certificate:

1. Log into your PayPal Sandbox Business account and click the Profile tab.
2. Click the API Access link and then click the API Certificate Request link.
3. In the Certificate Profile section, enter your merchant information (First Name, Last Name, Company, Volume, and Expected Use are required fields). While the Volume and Expected Use fields are required, they are mainly for PayPal informational purposes only.
4. In the Account Name and Password section, enter a password.



Make sure to write down your account name and password, because there will be no way to get a reminder later on. This account name and password, along with a certificate file, will be required when you connect to the PayPal API. If you do forget your password, you will need to create a new SSL certificate request.

5. In the Terms of Use section, check Yes and click Continue. Review your Certificate Profile and click Generate Certificate. Your API Certificate file will be created and made available for you to download.

6. Once the API certificate file is generated, click Download and save the text file (*cert_key_pem.txt*) to your local hard drive.

This API certificate file is a text file, but it is not yet in the format required to connect to the PayPal API. You'll need to convert it into a PKCS12 (.cer) file using a cryptographic tool such as OpenSSL (<http://www.openssl.org>). To avoid having to compile the OpenSSL source code yourself, you can download a precompiled Windows version, as described in the "Installing OpenSSL for Windows" sidebar.

Download and install Shining Light Productions' Win32 OpenSSL from <http://www.slproweb.com> (at the time of this writing, v0.9.7d is the recommended version).

To convert the text certificate file into SSL (PKCS12) format using OpenSSL, open the Windows command prompt (*cmd.exe* in Windows XP/2000, or *command.com* in Windows 9x/Me). Start OpenSSL by typing *c:\openssl\bin\openssl* at the prompt (the pathname may be different on your system). At the OpenSSL prompt, type the following command, where *c:\cert_key_pem.txt* is the location of your text certificate file and *c:\mycert.p12* is the location of your new SSL (PKCS12) file to create:

```
pkcs12 -export -in c:\cert_key_pem.txt -out c:\mycert.p12
```

The next step involves installing the certification and is dependent upon the type of application you're creating (e.g., a desktop application or a web application) and the development tool you're using to create it. This hack connects to the PayPal API from a desktop application created from within the Microsoft Visual Studio .NET development environment. If, however, you are using another development environment such as Java, or if you are developing a web application under Apache, you'll need to see the developer tool documentation at <http://www.paypalhacks.com/resources>.

SOAP-Enabling Your Application

In order for your application to access PayPal's Web Services, you'll need to install a module or code library that can call a SOAP-based web service. Some development tools, such as Visual Studio .NET, are set up to support web services out of the box.

To access PayPal's API using Visual Studio .NET, you need to import the *.p12* certificate file you created into Internet Explorer to register the certificate in the computer's registry.

Before you access the secure PayPal API with Microsoft development tools, Microsoft requires that you create a valid security certificate. To do this, import the *.p12* certificate file into Internet Explorer and then export the certificate as a *.cer* file, all from within Windows.

To import the *.p12* certificate, double-click the *.p12* file (e.g., *mycert.p12*) to open the Windows Import Certificate Wizard. Follow the prompts and accept the defaults. You will be required to enter the password you provided when you created the PayPal API certificate file earlier in this hack. When finished, you will see a confirmation message that the import was successful. Click OK.

To export the certificate as a *.cer* file, open the Tools menu in Internet Explorer and select Internet Options. Choose the Content tab and then click the Certificates button to display the Certificates screen. The Certificates screen lists the certificates currently installed on your computer; select the certificate you just imported (it's under the Personal tab) and click Export. Accept the default options. When prompted to select a File Format, select "DER encoded binary X.509 (.CER)" and click Next. Enter the filename and location, click Next, and then click Finish. You'll see a message that the export was successful. Click OK, then Close, and then OK again to close the Internet Options screen. Later, you'll refer to this *.cer* file from your code to access the PayPal API.



For the sake of simplicity and consistency, the rest of this chapter uses code written in C# using Visual Studio .NET. If you are using another language, such as Java, VB, C++, PHP, or Perl, review the PayPal Web Services page (<http://www.paypalhacks.com/resources/>).

To access a web service from within a development environment such as Visual Studio .NET, you need the URL of the Web Service Description Language (WSDL) file that describes the web service and, possibly, a valid security certificate. Typically, you would set up a web reference to abstract the SOAP-specific details of the web service, allowing you to access the web service as you would any other class or function call. Once you validated a web service using its WSDL file in the Visual Studio .NET Web Reference Wizard, a web reference would be added to your project and you'd be able to access its methods just like any other class in your project.

Currently, PayPal does things differently. For security reasons, PayPal requires that you not only install a security certificate, but also provide your digital certificate account name and password to access the PayPal API.

To set up a proxy web reference in Visual Studio .NET, open your Visual C# Windows Application. In your project's Solution Explorer, right-click the References folder and select Add Web Reference. In the Add Web Reference box, type the URL of the appropriate PayPal Sandbox WSDL file:

Sandbox: <http://api.sandbox.paypal.com/wsd/ PayPalSvc.wsdl>

Sandbox (alternate): <http://www.paypalhacks.com/wsd/ PayPalSvc.wsdl>

Live PayPal site: <http://api.paypal.com/wsd/ PayPalSvc.wsdl>

Then click Go. (The wizard does not work well with https, so use http.) If successful, the Web Reference wizard displays the description of the PayPalAPIInterface and the methods it contains. As of this writing, the methods are BillAgreementUpdate(), BillUser(), GetTransactionDetails(), MassPay(), RefundTransaction(), and TransactionSearch(). (BillAgreementUpdate() and BillUser() are not publicly available and are not discussed in this book.)

Change the Web reference name from com.paypal.sandbox.api to PayPalSvc and then click Add Reference. Verify that a new folder named Web References has been created and that it contains a reference named PayPalSvc.

You are now ready to use your PayPalSvc web reference. Using the digital certificate, certificate account name, and password, you can access the PayPal Web Service's methods via this PayPalSvc object.

Getting Started with PayPal's APIClient Tool

PayPal offers immediate gratification for users who can't wait to use the PayPal API. The APIClient application is downloadable from the Help Center tab at Developer Central.



The APIClient was created using Microsoft Visual Studio .NET and is written in C#. The application is a .NET project you'll need to modify and build before you can use it.

Here's how to set up the APIClient application:

1. Download the .NET Code Samples and unzip the *APIClient.zip* file into a folder on your hard drive.
2. Double-click the *APIClient.csproj* file to open the APIClient project in Visual Studio .NET.
3. Expand the Web References folder, right-click on the PayPalSvc reference, and select Properties, as shown in [Figure 8-4](#).

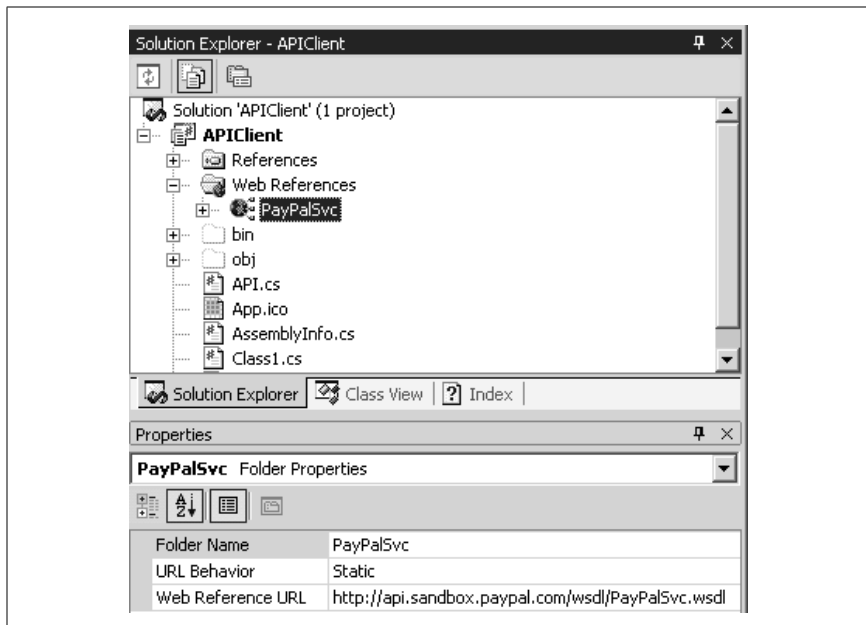


Figure 8-4. Specifying the location of the WSDL file in the properties sheet of the PayPalSvc web reference

4. Point the Web Reference URL to the PayPal Sandbox WSDL file.
5. Right-click the APIClient project name in Visual Studio .NET and select Properties.
6. Select Configuration Properties, and then select Build.
7. In the Properties pane, set the Output Path to C:\ (or whatever drive you are comfortable with; you are going to run this program from the command line, so using something like C:\ is easy on the fingers). Click OK.
8. From the Build menu, select Build APIClient. Visual Studio .NET will build the executable and save it into your Output path; make sure you place it in the same folder as your *certificate.cer* file.

The APIClient is ready to go. All you need now is a transaction to play with.

Setting up a Test Transaction

Before you start using the APIClient, send some money from your Sandbox Personal account to your Sandbox Business account:

1. Log into Developer Central, click the Sandbox tab, click the Launch Sandbox button, and log in with your Personal Sandbox account.
2. Click Send Money and then send some cash (e.g., \$10) to your Business account.

3. Next, log out of your Personal account and log back into your Sandbox Business account.
4. The payment you made from your Personal account will appear on the Overview page. Your balance will have increased by the amount you sent (minus the simulated transaction fee).
5. Click the Details link to bring up the Transaction Details. Record the Transaction ID number for use in the next step.

Making Your First Call

That's it for the prep work. Now, it's time to call the Refund Web Service. The `APIClient` is a .NET console application, so you need to open up a command prompt (`cmd.exe` in Windows XP/2000, or `command.com` in Windows 9x/Me).

Use `cd` to navigate to the directory where the `APIClient.exe` executable is located (e.g., `cd c:\`), and execute the client program:

```
APIClient RefundTransaction -t transaction_number -u your_api_username
-p your_api_password -c certificate_file
```



For a full description of the arguments for the test tool, please see the `APIClient` documentation or type `APIClient help` at the prompt.

If all goes as planned, you will see some output text in your console, as shown in [Figure 8-5](#). Among other things, `Ack` will be set to `Success` to confirm that the transaction has been refunded. Also note the number of errors reported by the call (which, in this case, happens to be zero.)

```

-----RUNNING-----
Certificate Problem with accessing https://api.sandbox.paypal.com/2.0/
Problem code 0x800B0101-Certificateproblem:CertEXPIRED
Certificate Problem with accessing https://api.sandbox.paypal.com/2.0/
Problem code 0x00000000, Unknown Certificate Problem
-----RESULTS-----
+-RefundTransactionResponse:-----
Ack: <Success>
Version: <1.000000>
Build: <1.0006>
CorrelationID: <>
Timestamp: <4/29/2004 5:49:30 PM>
RefundTransaction Errors: <0>
-----
Soap response completed and valid.
C:\>

```

Figure 8-5. Using the `APIClient` to issue refunds

Log into your Sandbox Business account, click History, and look at your transaction log to verify that the payment was refunded successfully.

Make Your First API Call

The APIClient is a nice introduction to the use of the PayPal API, but it demonstrates only a fraction of what the PayPal API can do. In addition, the APIClient was written solely for command-line use and will not scale to other applications. Use the next few hacks to extend the PayPal API into a stand-alone .NET assembly that any client can use.

—*Rob Conery and Dave Nielsen*