



Bit-Pay.com Payment API

v0.2

Table of Contents

[Introduction](#)

[Activating API Access](#)

[Invoice States](#)

[Creating an Invoice](#)

[Getting an Invoice](#)

[Invoice Status Updates](#)

[Sample Client Library](#)

Introduction

The Bit-Pay.com payment API is designed for merchants that need full control over their customers' shopping and checkout experience. A merchant that uses their own shopping cart software can make use of this API to transmit invoice information to Bit-Pay.com and receive notifications when the customer has completed payment and the invoice total has been credited to the merchant account. A merchant can elect to receive a single notification when a payment has been confirmed according to a speed preference (high, medium or low), or notifications can be sent immediately upon full payment, on confirmation and on completion (when the payment has been credited to the merchant account).

There are three interactions with the Bit-Pay.com service that this API enables:

- create an invoice
- fetch an invoice
- receive invoice status updates.

Activating API Access

The Bit-Pay.com JSON API is accessible at <https://bit-pay.com:8443>. The merchant's system must use a client SSL certificate for authentication. Once an SSL certificate has been created, the SHA-1 fingerprint for the certificate must be set in the merchant account (on the order preferences page). The Bit-Pay system will use the fingerprint of the client certificate to lookup the merchant account and authenticate the connection. The merchant's client key must remain private. Should it ever be compromised, the merchant can generate a new certificate and update the fingerprint in their Bit-Pay account. In this guide, we provide instructions for generating a private key and certificate using openssl (see <http://openssl.org> for download and installation instructions), however there are other tools that can be used to generate an SSL key and certificate.

To generate an SSL client key, an SSL certificate, and get the SHA-1 fingerprint, execute the following sequence of commands (the 3 commands are shown in bold):

```
$ openssl genrsa -out ssl-key.pem 1024
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
$ openssl req -new -x509 -nodes -sha1 -days 3650 -key ssl-key.pem -out ssl-cert.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:
Email Address []:
$ openssl x509 -in ssl-cert.pem -fingerprint -noout
SHA1 Fingerprint=20:02:07:AC:45:D4:5A:38:AC:1A:4A:72:8E:EB:3D:6B:CD:50:21:78
```

You can fill in your organization details if you choose, but it is not mandatory. The SHA1 fingerprint highlighted in yellow is the value that you will need to set in your merchant account. This gives the bit-pay.com system the information it will need to authenticate your API connections.

When communicating with the bit-pay.com system, it is highly recommended that you also verify the Bit-Pay.com SSL certificate. Doing so will ensure that you are in fact communicating with the Bit-Pay.com server. Note, the client side example utilities (covered later in this document) do verify the Bit-Pay.com certificate. The SHA-1 fingerprint for the Bit-Pay.com system is:

38:7A:15:D0:DC:D5:69:FE:CE:8A:DA:CA:CB:DA:B9:31:C4:31:FF:19

Invoice States

A Bit-Pay.com invoice can be in one of the following states: new, paid, confirmed, completed, expired or invalid. Payments sent to the bitcoin address associated with an invoice will only be credited to the invoice when it is in the “new” state.

Invoice State	Description
new	An invoice starts in this state. When in this state and only in this state, payments to the associated bitcoin address are credited to the invoice. If an invoice has received a partial payment, it will still reflect a status of new to the merchant (from a merchant system perspective, an invoice is either paid or not paid, partial payments and over payments are handled by bit-pay.com by either refunding the customer or applying the funds to a new invoice.
paid	As soon as full payment (or over payment) is received, an invoice goes into the paid status.
confirmed	The transaction speed preference of an invoice determines when an invoice is confirmed. For the high speed setting, it will confirmed as soon as full payment is received on the bitcoin network (note, the invoice will go from a status of new to confirmed, bypassing the paid status). For the medium speed setting, the invoice is confirmed after the payment transaction(s) have been confirmed by 1 block on the bitcoin network. For the low speed setting, 6 blocks on the bitcoin network are required. Invoices are considered complete after 6 blocks on the bitcoin network, therefore an invoice will go from a paid status directly to a complete status if the transaction speed is set to low.
complete	When an invoice is complete, it means that Bit-Pay.com has credited the merchant's account for the invoice. Currently, 6 confirmation blocks on the bitcoin network are required for an invoice to be complete. Note, in the future (for qualified payers), invoices may move to a complete status immediately upon payment, in which case the invoice will move directly from a new status to a complete status.
expired	An expired invoice is one where payment was not received and the 15 minute payment window has elapsed.
invalid	An invoice is considered invalid when it was paid, but payment was not confirmed within 1 hour after receipt. It is possible that some transactions on the bitcoin network can take longer than 1 hour to be included in a block. In such circumstances, once payment is confirmed, Bit-Pay.com will make arrangements with the merchant regarding the funds (which can either be credited to the merchant account on another invoice, or returned to the buyer).

Creating an Invoice

An invoice is created by sending an http POST message to <https://bit-pay.com:8443/api/invoice> with the details of the invoice passed in the body of the request. The body of the message must be JSON encoded and the content-type should be set to "application/json". On successful creation, the invoice details will be provided in a JSON encoded response. If there is an error, you will receive a JSON encoded error response. All error responses will have an "error" field that is an object with two fields called "type" and "message". A merchant is restricted to creating no more than 100 invoices per hour (there are also per second and per minute limits). The fields in the request are described below:

Name	Required	Default	Description
orderId	No		Used to display an orderId to the buyer. In the account summary view, this value is used to identify a ledger entry if present.
itemDesc	No		Used to display an item description to the buyer.
itemCode	No		Used to display an item code to the buyer.
notificationEmail	No		An email address used for status update notifications.
notificationURL	No		Optional. A URL to send status update messages to (this must be an https URL, unencrypted http URLs or any other type of URL is not supported). Bit-Pay.com will send a POST request with a JSON encoding of the invoice to this URL when the status changes.
redirectURL	No		Optional. This is the URL for a link that is displayed on the receipt once payment has been completed which is intended to be a URL that returns the customer to the merchant's website (it could be to the main storefront landing page, or a page specific to the order).
posData	No		This content of this field is included in status updates or requests to get an invoice. It is intended to be used by the merchant to uniquely identify an order associated with an invoice in their system. Aside from that, Bit-Pay does not use the data in this field. The data in this field can be anything that is meaningful to the merchant.
price	Yes		This is the amount that is required to be collected from the buyer. Note, if this is specified in a currency other than BTC, the price will be converted into BTC at market exchange rates to determine the amount collected from the buyer.
currency	Yes		This is the currency used for the price setting. The pricing currencies currently supported are BTC and USD.
physical	No	false	<true false> Indicates whether anything is to be shipped with the order (if false, the buyer will be informed that nothing is to be shipped)
fullNotifications	No	false	<true false> If set to false, then notifications are only

			sent when an invoice is confirmed (according the the transactionSpeed setting). If set to true, then a notification will be sent on every status change.
transactionSpeed	No	account setting	<high medium low> Controls when an invoice is considered to be “confirmed” (high=immediate, medium=1 block, low=6 blocks). Note, this setting does not affect when the invoice is credited to the merchant’s account. If not set, this setting defaults to the setting in the merchant account.
buyerName buyerAddress1 buyerAddress2 buyerCity buyerState buyerZip buyerEmail buyerPhone	No		These fields are used for display purposes only and will be shown on the invoice if provided.

The response to a create invoice request, the response to a get invoice request, and the content of a status update notification are all identical JSON representations of the invoice object. The fields are described below:

Name	Description
id	The unique id of the invoice.
url	An https URL where the invoice can be viewed.
posData	A data field provided by the merchant and designed to be used by the merchant to correlate the invoice with an order or other object in their system.
status	The current invoice status. The possible states are described earlier in this document.
btcPrice	The amount of bitcoins being requested for payment of this invoice (same as the price if the merchant set the price in BTC).
price	The price set by the merchant (in terms of the provided currency).
currency	The 3 letter currency code in which the invoice was priced.
invoiceTime	The time the invoice was created in milliseconds since midnight January 1, 1970.
expirationTime	The time at which the invoice expires and no further payment will be accepted (in milliseconds since midnight January 1, 1970). Currently, all invoices are valid for 15 minutes.
currentTime	The current time on the Bit-Pay.com system (by subtracting the current time from the expiration time, the amount of time remaining for payment can be determined).

Getting an Invoice

To get the current state of an invoice, an http GET request can be sent to <https://bit-pay.com:8443/api/invoice/<id>> where the id is the invoice id provided when the invoice was created. The format of the response is exactly the same as that which is returned when creating an invoice.

Invoice Status Updates

Invoice status updates can be sent either via email, https or both. The notificationEmail and notificationURL settings control the destination for the notification. Note, email notification is a human readable format and not intended for use as a system interface. For https notification, Bit-Pay.com sends a POST request to the given URL with a JSON encoding of the invoice that is identical to the format returned from a create invoice or get invoice request. If fullNotifications are enabled, then an update will be sent for every change in status. If fullNotifications are disabled, then an update is only sent when an invoice is confirmed (according to the transactionSpeed setting).

Sample Client Library

For convenience, a sample client library is provided to demonstrate how to interact with the Bit-Pay.com JSON API. You can use this client library as is on your server, you can customize it, or you can use it as a guide for developing a client library in another language. The sample client library is written in JavaScript and is designed to run using nodejs. Nodejs can be downloaded from <http://nodejs.org>. The examples have been tested on version 0.4.9, but should work on later versions as well. The sample client library can be obtained from <https://bit-pay.com/downloads/bitpayApi-0.2.zip>. The zip file contains 4 files: createInvoice, getInvoice, invoiceListener, and sampleInvoice.json. The first three of those files are executable and invoke the node runtime using typical Unix shebang notation. They can also be started by passing them as the first argument to the "node" runtime. The files themselves are JavaScript source code.

You must first create an SSL key and certificate for these utilities to work. You must also set the SSL certificate fingerprint in your merchant account at Bit-Pay.com. The procedure for doing so is outlined earlier in this document. These sample utilities look for two files in the same directory called ssl-key.pem and ssl-cert.pem. These are the SSL key and certificate that you generated respectively.

To create an invoice, run the createInvoice utility and pass is an invoice description on stdin. A sample invoice description is provided in the file sampleInvoice.json. To create an invoice using this sample, run the following command:

```
$ ./createInvoice <sampleInvoice.json
```

The newly created invoice will be output on a single line in JSON format.

To get an invoice, run the getInvoice utility and pass the invoice id as the sole argument as follows:

```
$ ./getInvoice 5_TU2V-M0glicVcZuQkkkq9aiA7qP0MjxRkhdc1MRSY=
```

Just as before, the invoice will be output on a single line in JSON format.

To receive notifications of invoice status updates, use the invoiceListener utility. It takes a single parameter on the command line to specify the port number and listens for incoming notifications from Bit-Pay.com. If you create an invoice with a notificationURL to your server and port, notifications of status changes on that invoice will be delivered to this utility. When a notification is received, the utility will print the JSON encoded invoice on stdout (one line per notification).

With these utilities, it is easy to craft a solution that can create a Bit-Pay.com invoice and receive payment notifications.