# GoCoin: Merchant integration guide

*More information can be found at help.gocoin.com*

## Preface

This guide is intended for Merchants who wish to use the GoCoin API to accept and process payments in Cryptocurrency. Depending on your desired functionality, this document will cover common structural choices for using the API, as well as useful features to improve the user experience for consumers.

### Invoices and Bills

The GoCoin API revolves around 'Invoices.' An Invoice is a commercial document that can be paid in Cryptocurrency.

**Invoice:** An *Invoice* is a request for payment that contains information necessary to remit payment to the issuer.

Any time a customer is paying you in Bitcoin, they will be paying a GoCoin invoice. Each invoice is assigned a single, never used payment address for the cryptocurrency the customer has chosen to pay with.

Invoices that have not been paid are said to be in an 'unpaid' state. An invoice can remain unpaid for 15 minutes. Rather, it has a 15 minute window in which payments are acceptable. This is how GoCoin is able to lock in a spot rate (guaranteed exchange rate) for fiat currencies.

**Bill:** A *Bill* is a request for payment that requires more input before a payment can be completed.

A 'Bill' is a special type of invoice. It has not yet been assigned a price currency (currency it will be paid in), payment address, or a spot rate, and therefore must have additional data supplied before it can be paid. This also means that a Bill does not have a 15 minute payment window *UNTIL* that information is supplied.

### The Hosted Gateway

GoCoin provides a hosted application that can display Invoices, so you do not have to render them inside your site or application.

- Fully Responsive
- Can be iframed into your site

It is not required to use the Gateway - if you desire white-label control over the UI or the payment page, you can easily render your own using the API calls discussed in this Guide.

## Are you ready to integrate?

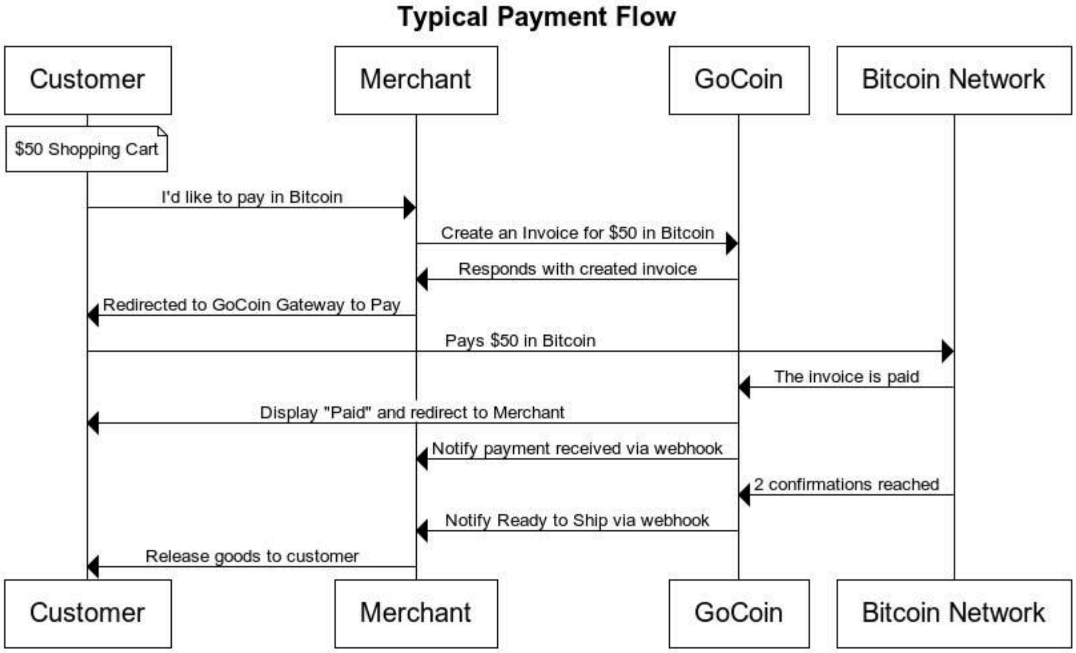
GoCoin currently requires that we have a way to pay a Merchant before they can generate invoices. This means that your merchant account must have at least one Payout Account on file with us before you can create invoices.

- If you want to convert payments to Fiat Currency
  - Upload KYC Documents and get your merchant account verified.
  - Add a Bank Account

- If you want to keep payments in Crypto
  - Add a Payout Address for each currency you would like to accept payments in

- If you would like to control the percentage converted to fiat
  - All of the above

## Payment flow


Typical Payment Flow
Customer
Merchant
GoCoin
Bitcoin Network
$50 Shopping Cart
I'd like to pay in Bitcoin
Create an Invoice for $50 in Bitcoin
Responds with created invoice
Redirected to GoCoin Gateway to Pay
Pays $50 in Bitcoin
The invoice is paid
Display "Paid" and redirect to Merchant
Notify payment received via webhook
2 confirmations reached
Notify Ready to Ship via webhook
Release goods to customer
Customer
Merchant
GoCoin
Bitcoin Network

Merchants can release goods to the customer at two points in this flow. Either when the payment is initially received, or after GoCoin detects that the payment has been confirmed on the network (Recommended, aka "ready_to_ship"). We advise that all merchants wait until the payment is confirmed, to prevent bad actors from attempting to "double-spend" Bitcoin on your site. More on this later.

# API Overview

The GoCoin API is a JSON API based on RESTful Priciples. It uses Standard HTTP methods (GET, POST, PUT, DELETE) as well as utilizing the PATCH method for updating partial resources.

Resources are protected via OAuth 2.0 Authentication. The GoCoin API requires Merchants to obtain an Access Token from an Authorization Code Grant type. All resources are protected by unique scopes, to prevent access to sensitive data if a token is compromized.

In addition, most resources in the GoCoin API are identified by UUIDs.

```
//An example Merchant ID (uuid)
78966b46-fe9d-47ac-af61-e1ea1cfab855
```

### Understanding the URL Format

```
//The Base URL for any API Endpoint is formatted:
https://HOST/api/VERSION/

// Therefore, the current Base URL for V1 of the GoCoin API is:
https://api.gocoin.com/api/v1/

//the path for a collection of resources will always be
"/resources"

//the path for a particular element of a resource will always be
"/resources/:id"

//Therefore, the URL to GET this merchant would be
https://api.gocoin.com/api/v1/merchants/78966b46-fe9d-47ac-af61-e1ea1
cfab855
```

## Accessing the API

The GoCoin API requires an access token. Depending on how much functionality your App or Website requires from the API, you can choose the best way to obtain a token.

**Understanding Scopes** Scopes define the privileges for an access token.

| Scope | Description |
|---|---|
| `account_read` | Access to read accounts data. |
| `user_read` | Access to read users data. |
| `user_read_write` | Access to read and write user data. Cannot modify password. |
| `user_password_write` | Can update user password. Must be combined with `user_read_write` to access other attributes. |
| `merchant_read` | Can read all merchant info associated with a user. |
| `merchant_read_write` | Can read and write to all merchants associated with a user. |
| `merchant_settings_read` | Can read all merchant settings associated with a user. |
| `merchant_settings_read_write` | Can read and write to all merchant settings associated with a user. |

| `invoice_read` | Can read all invoice info. | `invoice_write` | Can create an invoice (but cannot read an existing one). | `invoice_read_write` | Can read and write to invoices.

Scopes can be defined in a space delimited string. ie:

```
scope = "user_read user_read_write merchant_read_write"
```

As a best practice, do not request an access token with more privileges than your application will need.

### API Keys

The GoCoin Dashboard gives you the ability to obtain an API Key. An API Key is an Access token with a predefined scope of `"user_read invoice_read_write"` This gives you access to read your user information (which contains your User ID and Merchant ID) and also create and read invoices.

API Keys are ideal for typical implementations. If you have a standard shopping cart and checkout experience, you will not need a token with any more scope than this.

If you need access to additional scopes and resources for your application, see Appendix B

## Making an API Request

All requests must be made over SSL. The Access token will be passed in the Authorization header of any API request.

```
Authorization: Bearer <Access Token>
```

Additionally, we must set the Content-Type header

```
Content-Type: application/json
```

A Full Request

```
// This route retrieves user who owns the current access token.
// Note that the Access token is passed

GET /api/v1/user HTTP/1.1
Host: api.gocoin.com
Content-Type: application/json
Authorization: Bearer 7f661a21a25b6fb1242644b281bb1013d36be2e02a0ea57
86bb656f5f0229a03
Cache-Control: no-cache

200 (OK)
Content-Type: application/json

{
    "id": "dd456fd9-9928-4649-89d2-379368787845",
    "email": "user@gocoin.com",
    "first_name": "GoCoin",
    "last_name": "User",
    "created_at": "2013-08-04T07:05:44.590Z",
    "updated_at": "2013-08-04T07:05:44.590Z",
    "image_url": null,
    "merchant_id": "08d3bedf-7cb3-4ccc-9d20-cd221df9443d"
}
```

### API Errors

Most errors generated by the API will contain a JSON Response Body.

**Error Response Body Format:**

```
{
    "status": "422",
    "message": "Cannot create invoice.",
    "errors": {
        "base_price": [
            "must be greater than 0"
        ]
     }
}
```

# Invoices & Payments

An Invoice is the name (Class) given to a request for payment within GoCoin. Invoices contain all information necessary for payment, unless they are of the special type 'Bill'

Creating an invoice requires an access token with the scope `'invoice_read_write'` or `'invoice_write'`

### Request

```
Create a new invoice.
POST /merchants/:id/invoices
Headers:
 Accept: application/json

Body:
{
  "price_currency": "BTC",
  "base_price": 134.00,
  "base_price_currency": "USD",
  "callback_url": "https://www.example.com/gocoin/callback",
  "redirect_url": "https://www.example.com/redirect"
}
```

### Response

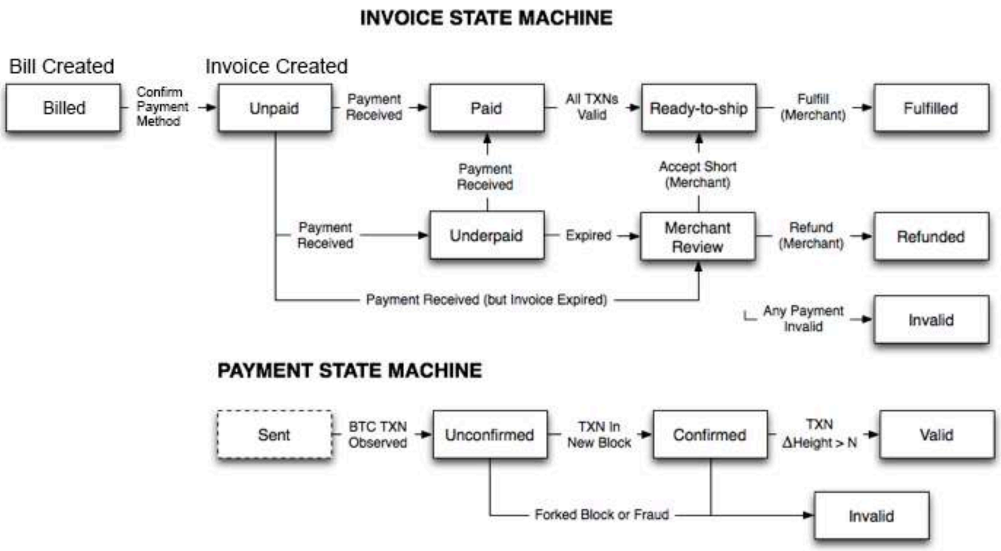
```
Status Code: 201
Body:
{
    "id": "84c4fc04-66f2-49a5-a12a-36baf7f9f450",
    "status": "unpaid",
    "payment_address": "1xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
    "price": "1.00000000",
    "crypto_balance_due": "1.00000000",
    "price_currency": "BTC",
    "valid_bill_payment_currencies": null,
    "base_price": "134.00",
    "base_price_currency": "USD",
    "service_fee_rate": "0.01",
    "usd_spot_rate": "1.0",
    "spot_rate": "0.00746268656716",
    "inverse_spot_rate": "134.0",
    "crypto_payout_split": "100",
    "confirmations_required": 2,
    "crypto_url": null,
    "gateway_url": "https://gateway.gocoin.com/invoices/84c4fc04-66f2
-49a5-a12a-36baf7f9f450",
    "notification_level": null,
    "redirect_url": "http://www.example.com/redirect",
    "order_id": null,
    "item_name": null,
    "item_sku": null,
    "item_description": null,
    "physical": null,
    "customer_name": null,
    "customer_address_1": null,
    "customer_address_2": null,
    "customer_city": null,
    "customer_region": null,
    "customer_country": null,
    "customer_postal_code": null,
    "customer_email": null,
    "customer_phone": null,
    "user_defined_1": null,
    "user_defined_2": null,
    "user_defined_3": null,
    "user_defined_4": null,
    "user_defined_5": null,
    "user_defined_6": null,
    "user_defined_7": null,
    "user_defined_8": null,
    "data": null,
    "expires_at": "2014-01-02T22:08:09.599Z",
    "created_at": "2014-01-02T21:53:10.867Z",
    "updated_at": "2014-01-02T21:53:10.867Z",
    "server_time": "2014-01-02T23:59:12Z",
    "callback_url": "https://www.example.com/gocoin/callback",
    "merchant_id": "7af834d9-aa7a-423c-be16-33ea6a724007"
}
```

## Invoice States

An invoice can exist in and transition as described below.

**VERY IMPORTANT**

*It is up to you, the Merchant, when you release your goods to the Consumer. However,* *GoCoin only considers a transaction valid (and credits your account) - after an invoice reaches 'Ready to Ship.' The number of blockchain confirmations required to achieve this state varies by payment currency.*



| State | Sent as | Description |
|---|---|---|
| Billed | "billed" | A request for payment has been made, but a payment method has not been selected. Therefore, no payment address or spot rate has been assigned, and the expiration timer has not started. |
| Unpaid | "unpaid" | A request for payment has been made, the payment method has been declared, and the expiration timer is ticking. |
| Paid | "paid" | Payment has been received in full, but has not been confirmed on the blockchain |
| Ready to Ship | "ready_to_ship" | The Payment is confirmed on the blockchain and the merchant. Typically, this is when the merchant releases the goods to the customer. |
| Fulfilled | "fulfilled" | An optional state that merchants can use for internal tracking |
| Underpaid | "underpaid" | A payment has been received, but does not fulfill the request for payment at the given exchange rate |
| Merchant Review | "merchant_review" | The Expiration window has passed, and the invoice was either underpaid, or a payment was received late |
| Refunded | "refunded" | A Merchant requested that a refund be issued for the invoice |
| Invalid | "invalid" | Any payment on the invoice was flagged as invalid |

*Note: an invoice never passes into an 'Expired' state - the expiration window is based on the value of* `"expires_at"` *in the Invoice.*

**Confirmations required for Ready_to_ship**

The following events are monitored within GoCoin and can generate Webhooks:

| Currency | # Confirmations | Approximate time from 'paid' |
|---|---|---|
| BTC | 2 | 20 minutes |
| LTC | 10 | 25 minutes |
| XDG | 20 | 20 minutes |

# Webhooks (aka Callbacks, Postbacks, or IPN)

GoCoin sends webhook notifications to update your system of invoice related events, like payments. All webhooks for a particular invoice are sent to the `"callback_url"` parameter in invoice.

### Events

The following events are monitored within GoCoin and can generate Webhooks:

| Event | Trigger |
|---|---|
| invoice_created | Sent when an invoice is created |
| invoice_payment_received | Payment is detected and unconfirmed on the network |
| invoice_ready_to_ship | Payment(s) confirmed for the full amount of the invoice |
| invoice_merchant_review | A Payment was received after the payment window, or the invoice expired while underpaid. Contact support to resolve. |
| invoice_invalid | Invalid payments detected. Purchaser may be attempting to double spend Bitcoin. Contact support. |

### Webhook Contents

Webhooks are sent as HTTP POST requests to your URL with a json body (Content-Type: application/json) for easy parsing in almost any programming language. All Webhooks contain the following keys:

| Key | Description |
|---|---|
| id | A unique, numeric, identifier for the webhook. You can use this value to record which webhooks you’ve already seen or recorded or acted on |
| event | An identifier for the type of event that occurred. See Events, below |
| payload | An object containing pertinent data about the event. |

### Webhook body

```
{
  "id" : "<event id>",
  "event" : "<event>",
  "payload" : {
               "id": "84c4fc04-66f2-49a5-a12a-36baf7f9f450",
               "status": "unpaid",
               "payment_address": "1xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
",
               "price": "1.00000000",
               "crypto_balance_due": "1.00000000",
               "price_currency": "BTC",
               "valid_bill_payment_currencies": null,
               "base_price": "134.00",
               "base_price_currency": "USD",
               "service_fee_rate": "0.01",
               "usd_spot_rate": "1.0",
               "spot_rate": "0.00746268656716",
               "inverse_spot_rate": "134.0",
               "crypto_payout_split": "100",
               "confirmations_required": 2,
               "crypto_url": null,
               "gateway_url": "https://gateway.gocoin.com/invoices/84
c4fc04-66f2-49a5-a12a-36baf7f9f450",
               "notification_level": null,
               "redirect_url": "http://www.example.com/redirect",
               "order_id": null,
               "item_name": null,
               "item_sku": null,
               "item_description": null,
               "physical": null,
               "customer_name": null,
               "customer_address_1": null,
               "customer_address_2": null,
               "customer_city": null,
               "customer_region": null,
               "customer_country": null,
               "customer_postal_code": null,
               "customer_email": null,
               "customer_phone": null,
               "user_defined_1": null,
               "user_defined_2": null,
               "user_defined_3": null,
               "user_defined_4": null,
               "user_defined_5": null,
               "user_defined_6": null,
               "user_defined_7": null,
               "user_defined_8": null,
               "data": null,
               "expires_at": "2013-10-01T18:47:45.150Z",
               "created_at": "2013-10-01T18:32:45.153Z",
               "updated_at": "2013-10-01T18:32:45.153Z",
               "server_time": "2014-06-24T19:03:16Z",
               "callback_url": "https://www.example.com/gocoin/callba
ck",
               "merchant_id": "63d3cd4c-1514-11e3-a3f0-080027fd9579"
              }
```

### Webhook Acknowledgement and Automatic Retries

Upon receipt of a webhook, you should accept it by returning an HTTP “200 OK” response as quickly as possible. Sending any other response (i.e. “500 Internal Server Error”, “404 Not Found”, etc.) OR failing to return a response within approximately 30 seconds will result in automatic retries of the webhooks.

GoCoin will automatically retry the webhook twice, after which we can manually resend if necessary.

### Verifying Authenticity

If you are concerned that an attacker might attempt to emulate a webhook to confirm their purchase, please consider the following.

- A list of IP Addresses to whitelist is available on request. Please message support@gocoin.com for more information.

- You can also self-sign an invoice in one of the User Defined fields. Take a look at this [article](article).

## Sample Basic Integration

This block of **Pseudocode** outlines the basics of creating an Invoice and redirecting to the Hosted Gateway. It also touched on the bare minimum of handling webhook notifications.

```
/*
* Initializer
*/

AccessToken = YOUR ACCESS TOKEN
MerchantID  = YOUR MERCHANT ID


Unless AccessToken? then Error "No AccessToken available"

// Get MerchantID if you don't know it explicitly
Unless MerchantID?
    CurrentUser = GoCoin::Self AccessToken
    if CurrentUser.errors?
        // Handle Errors
    else
        MerchantID = CurrentUser.merchant_id



/*
* Checkout Controller
*/

// Create an order or checkout in your system
Order = NewOrder

// Create an invoice for 50 USD in BTC
InvParams = {
    base_price: 50,
    base_price_currency: "USD",
    price_currency: "BTC",
    order_id: Order.id // Pass in an order ID (optional)
    callback_url: base_url + "/gocoin_callback"
}

NewInvoice = GoCoin::CreateInvoice AccessToken, MerchantID, InvParams
if NewInvoice.errors?
    // Handle errors
else
    redirect_to NewInvoice.gateway_url



/*
* Webhook/IPN Controller "/gocoin_callback"
*/

// Assume RequestParams is incoming request as JSON
e = RequestParams.event
invoice = RequestParams.payload

Switch e
case "invoice_created"
    // Do nothing
case "invoice_payment_received"
    if invoice.status is "underpaid"
        // Do Nothing
    else if invoice.status is "paid"
        // Mark Order Payment as Pending or Similar Status
case "invoice_merchant_review"
    // Do nothing
case "invoice_invalid"
    // Mark Order Payment as Failed
case "invoice_ready_to_ship"
    // Mark Order Payment as Completed
```

# Appendix A: HTTP Response Codes

The GoCoin API returns standard HTTP response codes to each API call, depending on the type of operation, its success or failure, and the reason for the failure.

For an exact list of response codes by endpoint, see the API Specification.

## Success Codes

### HTTP 200 (OK)

A `200` code is returned on nearly all API calls when they are processed successfully. The only exceptions are on endpoints that create a resource (using a POST) and return a `201`, and certain other requests such as `POST /users/request_password_reset`, which returns a `204`.

### HTTP 201 (created)

A `201` code is returned on successful API calls that create a new resource. For example, `POST /merchants/:merchant_id/invoices` will return a `201` upon successful invoice creation.

### HTTP 204 (no body)

A `204` code is returned on certain successful API calls that, although they create a resource, such as an association between two other resources, or a stored token for confirming or resetting a password, the resource is not something accessed by the API in a typical CRUD pattern.

## Error Codes

### HTTP 401 (unauthorized)

A `401` code is returned when an invalid (including empty) OAuth token is used in a request, or if the OAuth scope on the token does not include the endpoint.

### HTTP 404 (not found)

A `404` code is returned when a resource is not found, or in certain cases when the requestor is not authorized to access the endpoint.

A simple typo could cause the API to return a `404`, such as `GET /invoicesss/:id`, as well as when providing a resource ID that either doesn't exist or the requestor does not have permission to access.

For example, if another merchant created an invoice with id a931d410-1cde-11e4-8c21-0800200c9a66 and you attempted to access `GET /invoices/a931d410-1cde-11e4-8c21-0800200c9a66`, the API would return a 404 despite the fact that the resource exists.

### HTTP 409 (resource conflict)

A `409` code is returned when the resource is in a state such that the requested action is not permitted.

For example, `PUT /invoices/:id/fulfill` on an invoice with status `unpaid` would return a `409` because the status must be `ready_to_ship` for that action to be performed.

### HTTP 422 (invalid parameters)

A `422` code is returned when an endpoint has been routed correctly, as well as authorized (if applicable), but one or more parameters contained in the query string or the request's JSON body are invalid.

For example, `GET /invoices/search?start_time=not%20a%20valid%20datetime%20parameter` would return a `422` because the API would fail to parse the start_time parameter provided.

### HTTP 500 (internal error)

A `500` code is returned when the API encounters an error that is not due to an error attributable to the request or the authorization. In all cases, GoCoin Administrators are alerted when `500` errors occur and strive to solve them and prevent future `500` errors as soon as possible.

# Appendix B: Authentication with Third Party Applications

You can create a 'Third-party application in the GoCoin Dashboard

You will need the your application's ID & Secret Key to proceed.

3rd party apps will be authorized with an `authorization_code` grant type. More information about this grant is available in the spec.

Access Tokens obtained through this grant are do not expire (but can be revoked).

Initially, the app should open this address in a browser (redirect_uri must match given app) - note that this request is routed to the GoCoin dashboard, not the api, located at https://dashboard.gocoin.com

```
https://dashboard.gocoin.com/auth?response_type=code&client_id=CLIENT_ID&redirect_uri=REDIRECT_URI&scope=user_read&state=OPTIONAL
```

You will, be asked to Authorize the application you created. Verify that the scope shown is the scope you requested, and click 'Allow'

*Note: If you click deny, you will be redirected to the page with an error message.

After allowing the app to access your account, you will be redirected to the value set as the `redirect_uri`

```
https://YOUR_REDIRECT_URI?code=123987239817239187239187231231231231&state=OPTIONAL
```

**If you get an error for 'invalid redirect_uri' please check your request and make sure that what is being passed is an EXACT match to what was set during application set up.**

The state parameter should be checked to match the one that was in the initial url marked 'OPTIONAL' above.

If the state is valid, you should make a request for an access token using the 'code' in the querystring.

**Example Request using authorization_code grant**

```
POST /oauth/token HTTP/1.1
Host: https://api.gocoin.com
Content-Type: application/json
Cache-Control: no-cache

{
 "grant_type"    : "authorization_code",
 "code"          : "efsdSDASDlkfjoeiwjwekfmwemfwbvlbwi4d",
 "client_id"     : "676YDu5PS2hR8jbGhH2NSpsfGp7swUkWVWhRJnE5SwJKn2deP
dE5rkNUwdve5qYw",
 "client_secret" : "rSMPwVhf2DXvcYh55bEh2exxVThWFgsnMZcyNjMNN8ShcMzab
9smcxVrGbvwU9Ex",
 "redirect_uri"  : "http://etc.com"
}
```

The request above will return an access token. It is a `bearer` token. Responses from both requests above will look like this:

```
{
    "access_token": "68f77b685e710b023afc641c6b9e4f161f67d2eb4b40bd41
47598d2efe442750",
    "token_type": "bearer",
    "scope": "user_read_write"
}
```

# Appendix C: Invoice Request and Response parameters

### Request Parameters

The following parameters can be passed in the request body.

| Label | Type | Description | Default Value | Req. for: |
|---|---|---|---|---|
| base_price | decimal | The value of good or service being purchased. | | All |

| | | | | |
|---|---|---|---|---|
| base_price_currency | string | The value of good or service being purchased | | All |
| price_currency | string | The currency the invoice is to be paid in | null | Invoice |
| type | string | Makes an invoice a bill (type: "bill") | null | Bill |
| redirect_url | string | Valid url where the customer will be sent after a payment through the hosted gateway | null | |
| callback_url | string | Valid url where the webhook notifications will be sent after a payment through the hosted gateway | null | |
| order_id | string | Identifier for tracking the order from your system | null | |
| item_name | string | Label for the item being sold | null | |
| item_sku | string | SKU for item for easier tracking | null | |
| item_description | string | Optional description of item | null | |
| customer_name | string | Optional customer information fields | null | |
| customer_address_1 | string | Optional customer information | null | |
| customer_address_2 | string | Optional customer | null | |

| | | | | |
|---|---|---|---|---|
| | | information | | |
| customer_country | string | Optional customer information | null | |
| customer_region | string | Optional customer information | null | |
| customer_city | string | Optional customer information | null | |
| customer_postal_code | string | Optional customer information | null | |
| customer_phone | string | Optional customer information | null | |
| customer_email | string | Optional customer information | null | |
| customer_contact | string | Smart Param assigns itself to either customer_phone or customer_email | null | |
| user_defined_x | string | (x = 1-8) Optional pass through data fields - these are not displayed to the customer on the hosted gateway | null | |
| data | string | Miscellaneous field | null | |

### Response Parameters

The following parameters are returned in the invoice response, in addition to those that were passed in the request.

| Label | Type | Description |
|---|---|---|
| id | uuid | A unique id for the invoice within GoCoin |

| | | |
|---|---|---|
| price | decimal | A price to pay, calculated in <price_currency> |
| status | string | The status (state) of the invoice |
| payment_address | string | The address to pay the invoice in <price_currency> |
| crypto_balance_due | decimal | This is equal to , unless the invoice was underpaid and a balance remains |
| crypto_url | string | Wallet friendly url - this can be used to generate payable QR Code. |
| gateway_url | string | URL to this invoice on the hosted gateway. Redirect your customer here, or use an iframe |
| created_at | string | ISO time of Invoice creation |
| updated_at | string | ISO time of Invoice's last update. This is relevant to Bill's only. |
| expires_at | string | ISO time set to 15 minutes from the time the invoice reaches 'unpaid' status (updated_at) |
| server_time | string | ISO time - reference of currency server time |
| merchant_id | uuid | Unique identifier for the issuing merchant |
| valid_bill_payment_currencies | array | An array of 3 letter currency codes for acceptable payment options. |
| inverse_spot_rate | decimal | <base_price_currency>/<price_currency> exchange rate ie. 134 USD/BTC |
| spot_rate | decimal | <price_currency>/<base_price_currency> exchange rate ie. 0.007 BTC/USD |
| usd_spot_rate | decimal | USD/<base_price_currency> exchange rate |

|  |  |  |
|---|---|---|
| confirmations_required | integer | The number of confirmations before the invoice will be marked as 'ready_to_ship.' This is currently a fixed value and is not user editable. |
| service_fee_rate | decimal | Per transaction service fee. 0.01 = 1% |
| notification_level | string | unused, reserved |
| physical | string | unused, reserved |

# Appendix D: Invoice: Special Cases

### Extending the Expiration timer

You can edit the length of the expiration timer only when * You are not creating a 'Bill' * You are accepting 100% Cryptocurrency

It is possible to extend the Expiration timer by passing the parameter `"expires_in"` This field is denominated in seconds.

*The maximum value is 14400 seconds (4 hours)

```
// Create an invoice with a 1 hour expiration timer
InvParams = {
    base_price: 50,
    base_price_currency: "USD",
    price_currency: "BTC",
    expires_in: 3600
}

NewInvoice = GoCoin::CreateInvoice AccessToken, MerchantID, InvParams
```

### Create a Bill

You want to create a Bill when: * You do not know what type of currency the customer will pay in. * You do not want the invoice expriation timer to start * You want to send the invoice directly to the customer (discussed in the next section)

Creating a bill is as easy as sending the parameter `"type" : "bill"` with the invoice request body.

```
// Create an bill for 50 USD
InvParams = {
    base_price: 50,
    base_price_currency: "USD",
    type: "bill"
}

NewInvoice = GoCoin::CreateInvoice AccessToken, MerchantID, InvParams
```

It is not necessary to pass a price_currency in the request when creating a Bill, however you can do so if you wish.

```
// Create an bill for 50 USD in BTC
InvParams = {
    base_price: 50,
    base_price_currency: "USD",
    price_currency: "BTC",
    type: "bill"

NewInvoice = GoCoin::CreateInvoice AccessToken, MerchantID, InvParams
```

### Sending a bill: SMS or Email

Bills can be sent directly to a customer's mobile device (SMS) or Email. Additionally, the smart parameter "customer_contact" can be used to simplify checkout forms.

*This will create an invoice and send an SMS with a link to the Bill to 3105551212. That phone number will be stored in the invoice object as "customer_phone"*

```
// Create an invoice for 50 USD and text it (SMS) to the customer
InvParams = {
    base_price: 50,
    base_price_currency: "USD",
    type: "bill",
    "customer_contact": "3105551212",
    "send_to_customer": true
}

NewInvoice = GoCoin::CreateInvoice AccessToken, MerchantID, InvParams
```