

PayIOTA API Documentation

[Edit](#)[New Page](#)[Jump to bottom](#)

Laszlo Molnarfi edited this page 7 days ago · 35 revisions

Welcome to PayIOTA's API documentation!

1. Introduction

The API provides you direct access to our payment gateway's services. It allows you to create new invoices (address/price pairs) from your IOTA seed, and update the IOTA price of an already created invoice. They are implemented as HTTP(s) POST calls to <https://payiota.me/api.php> endpoint.

2. API Setup

All necessary information, such as your API key and verification string can be found on your interface after you sign-up for our service. You need a valid email address to do so. You are responsible for keeping your API key safe.

3. API Response

Our API will return to you a JSON array if the call is an invoice creation, getting invoice, or checking invoice. If it is a price compensation call, it will return the new price in integer format (number of IOTA's).

4. Invoice creation

To create an address/price pair, you will need to send a POST request to the API URL with the following parameters:

```
api_key=[your_api_key]&currency=[3_letter_currency_code_caps]&price=[price]&custom=[custom_string]&ipn_url=[ipn_url]&action=new
```

The parameter ipn_url is optional; if you do not specify it, PayIOTA will use your default IPN URL that you set on the interface page. If the request was successful, you will get a response such as this:

```
["YMG9BPGYKCAHTOLSIOFNLNHRVBHMBVBEQCCUNOXATFZMBURWSXCSRVMQOWWG  
UHRELMCKGRWDUXRHBQQSJBMECXUH","29176011"]
```

It is a JSON-encoded array. In the 0th place, you have the address your user needs to send the IOTA payment to, for the amount in the 1th place. The value you give for price and currency will always be converted to USD currency and price on our end. You can also use 'IOTA' as the currency value, thus giving the price in IOTA. Be aware that invoices have a maximum life of 1 week - after that our service will not check the address to prevent overloading our server and the full node it is checking from.

If the request was not successful, you will get a raw error coded response such as:

ERR_FATAL_3RD_PARTY (nodes offline)

ERR_IPN_URL_INVALID (when you have no default IPN URL set and you have not specified a custom IPN URL, or your specified IPN URL is invalid)

ERR_CUSTOM_INVALID (empty custom variable)

ERR_PRICE_INVALID (price variable not an integer)

ERR_API_KEY_INVALID (invalid API key)

ERR_INCORRECT_METHOD (used GET instead of POST for API)

ERR_PARAMETERS_MISSING (missing parameters for API)

ERR_API_KEY_DISABLED (yearly invoice due date missed)

Sorry, a fatal error has occurred, service is unavailable! (As a string, if a fatal error happens)

5. User Experience

You can either code the user experience of paying yourself, or you can use our external.php page:

Construct a link as such for our payment page:

[https://payiota.me/external.php?price=\[price_iota_returned_by_api\]&address=\[address_returned_by_api\]&success_url=\[success_url\]&cancel_url=\[cancel_url\]](https://payiota.me/external.php?price=[price_iota_returned_by_api]&address=[address_returned_by_api]&success_url=[success_url]&cancel_url=[cancel_url])

The success_url and cancel_url denote the URL's to which the user will be redirect to depending on the success or failure of paying the invoice.

Or if you code your own, see how our external.php page works (with the getinvoice and checkinvoice API calls).

6. Security/Privacy

For security and privacy reasons, you may want to encrypt the custom string on your side. Then, when the payment complete IPN is sent, you can decrypt it on your side. If there is a mismatch, you'd know that the payment was tampered with. It is extremely important to check the verification string when you receive the IPN (Instant Payment Notification).

You should store your verification string from your interface page server-side and verify that the received POST value of 'verification' is the same as in your database. If not, someone is attempting to fake transactions.

7. Speed

Since we now have address pregeneration, invoice creation completes very quickly. To compensate for IOTA price changes, we have another API call for updating the price the moment your user wishes to buy something. That API call is instant. Payments are as fast as the IOTA network with a 5-15 second delay (with the checkinvoice function).

8. Refunds

You are responsible for all refunds unless there is an error on our side. All the addresses are derived from your seed, so you have direct control over your funds. Please see the legal document for more information.

9. Fees

Staying true to IOTA's idea, we have no service fees either, we only charge a yearly membership fee.

10. IPN

When an invoice is paid (meaning the address has a balance equal or more than that of the 1th place in the response when you generated the invoice), our server will send you a POST request with the following variables:

address : The IOTA address the user paid to

realID : Your ID

price : Price in USD

price_iota : Price in IOTA

custom : Your custom string

ipn_url : Your IPN URL

verification : Your verification string (Verify!)

paid_iota : How much IOTA's your user paid

done : Should be 1. Whether the payment is complete

created: Date when invoice was created. In UNIX time.

IPN's are only sent once, but you can see all your invoices on the interface page.

11. Price Compensation/Invoice Update

When you generate an invoice for a later date, it will generate using IOTA's current price (we do not currently have a time machine). So, we have a quick function for updating an invoices price to the current IOTA price. It will update the IOTA price to the USD price you have given it on the original creation, you cannot change that. It will also update the created value (expiration of invoice) in our database, so your user can pay again and PayIOTA.me will check again for one week starting from the time the API is called. The parameters are the following:

```
api_key=[your_api_key]&address=[address_of_saved_invoice]&verification=[verification_string]&action=update
```

If the request was successful, you will get a response like this:

```
180000
```

That integer is the new price in IOTA's. If it was not successful, you will get a raw error coded response:

```
ERR_DISALLOWED (address not yours or is already paid, check your verification string)
```

Or

```
ERR_NOT_FOUND (address not found)
```

or

```
ERR_FATAL_3RD_PARTY (nodes, currency converter or CoinMarketCap offline)
```

12. IOTA to USD API

You can use our API to convert an amount of IOTA to it's price in USD.

It is easy: https://payiota.me/api.php?action=convert_to_usd&iota=amount_you_want_to_convert

It will give you the corresponding price in USD.

13. Getting the number of users using PayIOTA

You can also use our API to do that.

Use: <https://payiota.me/api.php?action=getnumberofusers>

It will return a number.

14. Getting miscellaneous payment statistics

Use: <https://payiota.me/api.php?action=getpaymentstatistics>

Values will be separated by ':'. First one returned will be the total invoices in the system, then the paid ones, then the value of the paid ones and finally the value of the paid ones in dollars (at the time of creation).

15. Getting an invoice's data

You can get the data of an invoice if you have the invoice's address. Here is how you construct this API call:

```
action=getinvoice&invoice=[address_of_invoice]
```

You will get a response like this:

```
{"content":  
[{"address":"LKWFORIMUPR9TLUGYKIDV NKCSKZFGETSNHWPGQZIKRXRMMPIBBSLJNDWIUB  
WKVCIB9YYLVNZINPBHIVTXOSGERTTXX","price":"0","price_iota":"93667","done":"1","created":  
"1537891767"}], "code":"ERR_OK", "message":"Action Successfully  
completed!", "boolean":"success"}
```

It is a JSON-encoded array that you can easily parse.

Please see the external.php example file on how to use the getinvoice API call:

<https://github.com/lacicloud/payiota/blob/master/www/external.php>

16. Triggering a server-side check of invoice payment

You can use trigger a check that will send the IPN and update the paid value in our database if the invoice is paid. It will return the getinvoice array so that you can check client-side whether the invoice was marked as paid or not by our server. Here is how you construct this API call:

```
action=checkinvoice&invoice=[address_of_invoice]
```

It is practically the same as the `getinvoice` API call, but it will also trigger a server-side check before returning the same as `getinvoice`. Please see the `external.php` example file on how to use the `checkinvoice` API call:

<https://github.com/laticloud/payiota/blob/master/www/external.php>

16. Accessing funds in the IOTA wallet

You have to generate every single address by clicking on the `attach to tangle/generate new address` button until you reach the last one you see on the PayIOTA dashboard.

16. Examples

Please Github, <https://github.com/laticloud/payiota>, 'examples' folder for PHP examples.

17. Support

Our support email address is support@payiota.me, feel free to contact us.

+ Add a custom footer

▼ Pages 1

Find a Page...

[PayIOTA API Documentation](#)

+ Add a custom sidebar

Clone this wiki locally

`https://github.com/laticloud/payiota.wiki.git`

