AEON.to API Documentation

Release 2.0

AEON.to

Jan 16, 2019

Contents

This is the API documentation for AEON.to, which is a service that allows users to pay any bitcoin address anonymously using Aeon.

Note: The current version of the API is 3.

Contents

1.1 Introduction

AEON.to is a service that allows users to pay any bitcoin address anonymously using Aeon. Aeon is a secure, private, untraceable currency that is open-source and freely available to all. AEON.to converts Aeon provided by the user to bitcoins (BTC), which are sent to a given bitcoin address.

AEON.to provides an API that enables developers to use the service in programs, apps, scripts etc. This API therefore allows developers to integrate an option to pay any bitcoin address in their product, for example, in a Aeon wallet service. The API is a REST-like API using *JSON* as data format. It is available anonymously and does not require identification or authentication. This document describes this API and gives examples of its usage.

> **Warning:** You agree to AEON.to's Terms of Service by accessing this API in any way or form. Read the Terms of Service here: https://aeon.to/static/html/tos.html

1.1.1 Overview

In order to use AEON.to, a user must first create an order over a certain amount of bitcoins (BTC). Once the order has been created, AEON.to provides the user with the details for the payment in aeon. Once the user has fully paid using Aeon, AEON.to will create a bitcoin transaction over the given amount to the given bitcoin address.

When using the API, the general flow of events is similar:

- get current order parameters to see if AEON.to is available and to fetch current price, order limits, etc. . .
- create a new order by supplying bitcoin destination address and mount
- check order status to get payment information
- pay order
- continue to repeatedly check order status for processing progress

1.1.2 Naming conventions

API base URL

The base URL of the API is `https://aeon.to/api/`, followed by the version identifier (currently `v3`), followed by the conversion direction (currently only `aeon2btc`). Therefore, the complete API base URL is `https://aeon.to/api/v3/aeon2btc/`.

Note: The current version of the API is 3.

API endpoint names

API endpoints are always named `<noun>_<verb>`. For example, the endpoint to create a new order is called `/order_create/`. For example, for API version 1 the complete URL would be `https://aeon.to/api/v3/aeon2btc/order_create/`.

Field names and values

Fields are named using lower-case nouns separated by underscores. Values are given in their default format. For example, the field `btc_amount` gives the amount of an order in bitcoins rather than satoshis. Data types are not indicated in field names.

1.1.3 Data and formats

Responses

API responses are always *JSON*-formatted data.

Return codes and errors

On success, the API returns the requested data in *JSON* format and *HTTP* return code `200`.

On failure, the API returns an error message formatted in *JSON* and a *HTTP* error code. The error message is formatted as follows:

```
{
    "error": "AEONTO-ERROR-<number>"
    "error_msg": "<error_error_message_as_string>"
}
```

The `error` number is unique per message across all endpoints and can be used to reliably identify errors when they occur. The `error_msg` is a human-readable description of the error and should not be used by a script or program that uses the API. For example, the error below is returned if the user provided a malformed bitcoin address when creating a new order:

```
{
    "error": "AEONTO-ERROR-002"
    "error_msg": "malformed btc address"
}
```

The error messages and *HTTP* return codes specific to the different API endpoint are defined in the appropriate section.

1.1.4 Various parameters

Currently, AEON.to has a few additional parameters and best practice values that are constant. Therefore, we do not expose them via a dedicated API endpoint. However, for the sake of completeness, we list them here:

Description	Value
Number of BTC confirmations required before we purge an order	144
Number of seconds after which an order times out if no or only partial payment occurred	900
Number of recommended mixins to use in Aeon payments	3

1.2 Version 3

This is the current version of the API. Active from January 2019 onwards.

Note: API version 3 is the current version.

1.2.1 Querying order parameters

Reference

API endpoint: https://aeon.to/api/v3/aeon2btc/order_parameter_query/

The order parameter endpoint supplies information about whether new orders can be created. In this case, this endpoint provides the current price, order limits, etc. for newly created orders.

Note: It is possible to query the status of existing orders even if the order parameter endpoint reports *not available*.

Request

Issue a *GET* request to query current order parameters.

Response

On success (*HTTP* code `200`), the request returns the following *JSON* data:

```
{
    "lower_limit": <lower_order_limit_in_btc_as_float>,
    "price": <price_of_1_aeon_in_btc_as_offered_by_service_as_float>,
    "upper_limit": <upper_order_limit_in_btc_as_float>,
    "zero_conf_enabled": <true_if_zero_conf_is_enabled_as_boolean>,
    "zero_conf_max_amount": <up_to_this_amount_zero_conf_is_possible_as_float>
}
```

Fields should be self-explanatory.

On failure, one of the following errors is returned:

HTTP code	AEON.to error code	Error message/reason	Solution
503	AEONTO-ERROR-001	internal services not available	try again later
503	AEONTO-ERROR-007	third party service not available	try again later
503	AEONTO-ERROR-008	insufficient funds available	try again later
403	AEONTO-ERROR-012	too many requests	try less often

Example

Request

```
curl https://aeon.to/api/v3/aeon2btc/order_parameter_query/
```

or

```
http https://aeon.to/api/v3/aeon2btc/order_parameter_query/
```

Response

```
{
    "price": 0.017666,
    "upper_limit": 20.0,
    "lower_limit": 0.002,
    "zero_conf_enabled": true,
    "zero_conf_max_amount": 0.1
}
```

Rate limitation

Too many requests in a short time will result in a rate limitation of the endpoint.

It is possible to request the endpoint up to **3 times per second**.

1.2.2 Creating a new order

Reference

API endpoint: https://aeon.to/api/v3/aeon2btc/order_create/

The order creation endpoint allows to create a new order at the current price. The user has to supply a bitcoin destination address and amount to create the order.

Note: Please use the `order_check_price` API endpoint if you only want to check the price for a specific Bitcoin amount.

Request

Issue a *POST* request to create a new order supplying the following parameters:

```
{
    "btc_amount": <requested_amount_in_btc_as_float>,
    "btc_dest_address": <requested_destination_address_as_string>
}
```

Note: Make sure that `btc_amount` amount is inside the possible limits for an order. These limits can be queried using the `order_parameter_query` endpoint.

Response

On success (*HTTP* code `201`, "created"), the request returns the following *JSON* data:

```
{
    "state": "TO_BE_CREATED",
    "btc_amount": <requested_amount_in_btc_as_float>,
    "btc_dest_address": <requested_destination_address_as_string>,
    "uuid": <unique_order_identifier_as_12_character_string>
}
```

The field `state` reflects the state of an order. If `state` is `TO_BE_CREATED` in the response, the order has been registered for creation and you can use the order `uuid` to start querying the order's status. All other fields should be self-explanatory.

On failure, one of the following errors is returned:

HTTP code	AEON.to error code	Error message/reason	Solution
`503`	`AEONTO-ERROR-001`	internal services not available	try again later
`400`	`AEONTO-ERROR-002`	malformed bitcoin address	check address validity
`400`	`AEONTO-ERROR-003`	invalid bitcoin amount	check amount data type
`400`	`AEONTO-ERROR-004`	bitcoin amount out of bounds	check min and max amount
`400`	`AEONTO-ERROR-005`	unexpected validation error	contact support
`403`	`AEONTO-ERROR-012`	too many requests	try less often

Example

In this example, we create an order for donating 0.1 BTC to the Aeon project (using Bitcoin, ironically).

Request

```
curl --data '{"btc_dest_address": "12Cyjf3qV6qLyXdzpLSLPdRFPUVidvnzFM", \
    "btc_amount": 0.1}' -H "Content-Type: application/json" https://aeon.to/api/v3/
→aeon2btc/order_create/
```

or

```
http --json https://aeon.to/api/v3/aeon2btc/order_create/ btc_dest_
→address=12Cyjf3qV6qLyXdzpLSLPdRFPUVidvnzFM btc_amount=0.1
```

Hint: Remember to set the *HTTP* Content-Type to `application/json`!

Response

```
{
    "state": "TO_BE_CREATED",
    "btc_amount": 0.1,
    "btc_dest_address": "12Cyjf3qV6qLyXdzpLSLPdRFPUVidvnzFM",
    "uuid": "aeonto-5rpnYP"
}
```

Rate limitation

Too many requests in a short time will result in a rate limitation of the endpoint.

It is possible to request the endpoint up to **4 times per minute**.

1.2.3 Creating a new order using a payment protocol URL

Reference

API endpoint: https://aeon.to/api/v3/aeon2btc/order_create_pp/

This alternative order creation endpoint allows to create a new order at the current price, but instead of providing an explicit address and amount, the user provides a BIP70 url that once fetched by AEON.to will provide the address and amount.

Request

Issue a *POST* request to create a new order supplying the following parameters:

```
{
    "pp_url": <payment_protocol_url>
}
```

Note: AEON.to is able to correct automatically URLs provided by users to the correct one serving a BIP70-protocol answer. For instance, values such as `https://bitpay.com/invoice?id=xxx`, `bitcoin:?r=https://bitpay.com/i/xxx` will be corrected to the correct one automatically (the correct one being for *Bitpay*: `https://bitpay.com/i/KbMdd4EhnLXSbpWGKsaeo6`.

Response

On success (*HTTP* code `201`, "created"), the request returns the following *JSON* data:

```
{
    "state": "TO_BE_CREATED",
    "btc_amount": <requested_amount_in_btc_as_float>,
    "btc_dest_address": <requested_destination_address_as_string>,
    "uuid": <unique_order_identifier_as_12_character_string>,
    "pp_url": <payment_bip70_protocol_url>
}
```

The field `state` reflects the state of an order. If `state` is `TO_BE_CREATED` in the response, the order has been registered for creation and you can use the order `uuid` to start querying the order's status. All other fields should be self-explanatory.

On failure, one of the following errors is returned:

HTTP code	AEON.to error code	Error message/reason	Solution
`503`	`AEONTO-ERROR-001`	internal services not available	try again later
`400`	`AEONTO-ERROR-002`	malformed bitcoin address	check address validity
`400`	`AEONTO-ERROR-003`	invalid bitcoin amount	check amount data type
`400`	`AEONTO-ERROR-004`	bitcoin amount out of bounds	check min and max amount
`400`	`AEONTO-ERROR-005`	unexpected validation error	contact support
`400`	`AEONTO-ERROR-010`	payment protocol failed	invalid or outdated data served by url
`400`	`AEONTO-ERROR-011`	malformed payment protocol url	url is malformed or cannot be contacted
`403`	`AEONTO-ERROR-012`	too many requests	try less often

Example

In this example, we create an order for donating 0.1 BTC to the Aeon developers (using Bitcoin, ironically).

Request

```
curl --data '{"pp_url ": "https://bitpay.com/invoice?id=<invoice_id>"}' -H "Content-
→Type: application/json" https://aeon.to/api/v3/aeon2btc/order_create_pp/
```

or

```
http --json https://aeon.to/api/v3/aeon2btc/order_create_pp/ pp_url="https://bitpay.
→com/invoice?id=<invoice_id>"
```

Hint: Remember to set the *HTTP* Content-Type to `application/json`!

Response

```
{
    "state": "TO_BE_CREATED",
    "btc_amount": 0.1,
    "btc_dest_address": "12Cyjf3qV6qLyXdzpLSLPdRFPUVidvnzFM",
    "uuid": "aeonto-XCZEsu",
    "pp_url": "https://bitpay.com/i/xxx"
}
```

Rate limitation

Too many requests in a short time will result in a rate limitation of the endpoint.

It is possible to request the endpoint up to **4 times per minute**.

1.2.4 Querying order status

Reference

API endpoint: https://aeon.to/api/v3/aeon2btc/order_status_query/

The order status endpoint allows users to query the status of an order, thereby obtaining payment details and order processing progress.

Request

Issue a *POST* request to query the status of a given order. You have to supply the order's uuid in the request:

```
{
    "uuid": <unique_order_identifier_as_12_character_string>,
}
```

Response

On success (*HTTP* code 200), the request returns the following *JSON* data:

```
{
    "state": <order_state_as_string>,
    "btc_amount": <requested_amount_in_btc_as_float>,
    "btc_dest_address": <requested_destination_address_as_string>,
    "uuid": <unique_order_identifier_as_12_character_string>
    "btc_num_confirmations": <btc_num_confirmations_as_integer>,
    "btc_num_confirmations_before_purge": <btc_num_confirmations_before_purge_as_
→integer>,
    "btc_transaction_id": <btc_transaction_id_as_string>,
    "created_at": <timestamp_as_string>,
    "expires_at": <timestamp_as_string>,
    "seconds_till_timeout": <seconds_till_timeout_as_integer>,
    "incoming_amount_total": <amount_in_aeon_for_this_order_as_float>,
    "remaining_amount_incoming": <amount_in_aeon_that_the_user_must_still_send_as_
→float>,
    "incoming_num_confirmations_remaining": <num_aeon_confirmations_remaining_before_
→bitcoins_will_be_sent_as_integer>,
    "incoming_price_btc": <price_of_1_btc_in_aeon_as_offered_by_service_as_float>,
    "receiving_address": <aeon_old_style_address_user_can_send_funds_to_as_string>,
    "receiving_integrated_address": <aeon_integrated_address_user_needs_to_send_funds_
→to_as_string>,
    "aeon_recommended_mixin": <aeon_recommended_mixin_as_integer>,
    "required_payment_id_long": <aeon_payment_id_user_needs_to_include_when_using_old_
→stlye_address_as_string>
    "required_payment_id_short": <aeon_payment_id_included_in_integrated_address_as_
→string>
}
```

The user has to pay the order using the integrated address. In case the user's wallet does not support integrated addresses, the user can pay via the old-style address while specifying the long payment id.

Presence of some of these fields depend on state, which can take the following values:

Value	Meaning
TO_BE_CREATED	order creation pending
UNPAID	waiting for Aeon payment by user
UNDERPAID	order partially paid
PAID_UNCONFIRMED	order paid, waiting for enough confirmations
PAID	order paid and sufficiently confirmed
BTC_SENT	bitcoin payment sent
TIMED_OUT	order timed out before payment was complete
NOT_FOUND	order wasn't found in system (never existed or was purged)

All other fields should be self-explanatory.

On failure, one of the following errors is returned:

HTTP code	AEON.to error code	Error message/reason	Solution
400	AEONTO-ERROR-009	invalid request	check request parameters
404	AEONTO-ERROR-006	requested order not found	check order UUID
403	AEONTO-ERROR-012	too many requests	try less often

Example

Continuing from our previous example, we can query the order by supplying the order's unique identifier `uuid`.

Request

```
curl --data '{"uuid": "aeonto-VkT2yM"}' -H "Content-Type: application/json" \
    https://aeon.to/api/v3/aeon2btc/order_status_query/
```

or

```
http --json https://aeon.to/api/v3/aeon2btc/order_status_query/ uuid=aeonto-VkT2yM
```

Response

The response gives the current status of the order:

```
{
    "incoming_price_btc": 0.01760396,
    "uuid": "aeonto-XCZEsu",
    "state": "UNPAID",
    "btc_amount": 0.1,
    "btc_dest_address": "12Cyjf3qV6qLyXdzpLSLPdRFPUVidvnzFM",
    "receiving_address":
↪"Sm6XWv4nmULGYQyHqRH54fgtXb3m3wnQ52EftauD2aMm1rmnrBwYDsudZj7MDgqB6Q1DgVR4jBgJdTFDZmssEdX711h3WyGzE
↪",
    "receiving_integrated_address":
↪"Sz4Dra9LBvxGYQyHqRH54fgtXb3m3wnQ52EftauD2aMm1rmnrBwYDsudZj7MDgqB6Q1DgVR4jBgJdTFDZmssEdX71ES97Z3M9!
↪",
    "required_payment_id_long":
↪"6db2414c87fd343fdd6ef4d73aa5f899372868da08faf4c0f5950a666ed08ad7",
    "required_payment_id_short": "7c855c2228e19671",
    "created_at": "2019-01-11T12:46:03Z",
    "expires_at": "2019-01-11T12:51:03Z",
    "seconds_till_timeout": 298,
    "incoming_amount_total": 14.4265105,
    "remaining_amount_incoming": 14.4265105,
    "incoming_num_confirmations_remaining": -1,
    "recommended_mixin": 4,
    "btc_num_confirmations_before_purge": 144,
    "btc_num_confirmations": 0,
    "btc_transaction_id": ""
}
```

In this example, the next step would require the user to pay *14.4265105* AEON to the (integrated) Aeon address *Sz4Dra. . . .*

In case the user's wallet does not support integrated addresses, the user can pay via the old-style address *Sm6XWv. . .* while providing the (long) payment ID *6db241. . . .*

Note: The payment **must** be made before the order expires, in this case, inside *298* seconds.

Rate limitation

Too many requests in a short time will result in a rate limitation of the endpoint.

It is possible to request the endpoint up to **3 times per second**.

1.2.5 Querying order price

Reference

API endpoint: https://aeon.to/api/v3/aeon2btc/order_check_price/

The order status endpoint allows users to query the recent price of an order.

Request

Issue a *POST* request to query the price of a given order. You have to supply the amount of BTC `btc_amount` in the request:

```
{
    "btc_amount": <requested_amount_in_btc_as_float>,
}
```

Response

On success (*HTTP* code `200`), the request returns the following *JSON* data:

```
{
    "btc_amount": <requested_amount_in_btc_as_float>,
    "incoming_amount_total": <amount_in_aeon_for_this_order_as_float>,
    "incoming_num_confirmations_remaining": <num_aeon_confirmations_remaining_before_
→bitcoins_will_be_sent_as_integer>,
    "incoming_price_btc": <price_of_1_btc_in_aeon_as_offered_by_service_as_float>
}
```

On failure, one of the following errors is returned:

HTTP code	AEON.to error code	Error message/reason	Solution
`400`	`AEONTO-ERROR-004`	bitcoin amount out of bounds	check min and max amount
`400`	`AEONTO-ERROR-005`	unexpected validation error	contact support
`400`	`AEONTO-ERROR-009`	invalid request	check request parameters
`403`	`AEONTO-ERROR-012`	too many requests	try less often

Example

Imagine we want to check the recent price for an order including the payment of 0.15 BTC.

Request

```
curl --data '{"btc_amount": "0.15"}' -H "Content-Type: application/json" \
    https://aeon.to/api/v3/aeon2btc/order_check_price/
```

---

or

```
http --json https://aeon.to/api/v3/aeon2btc/order_check_price/ btc_amount=0.15
```

Response

The response gives the current price for the order:

```
{
    "btc_amount": 0.15,
    "incoming_amount_total": 2163.9765748,
    "incoming_num_confirmations_remaining": 1,
    "incoming_price_btc": 0.00006932
}
```

In this example, the order including the payment of 0.15 BTC would require the user to pay *2163.9765748* AEON.

Rate limitation

Too many requests in a short time will result in a rate limitation of the endpoint.

It is possible to request the endpoint **once every 3 seconds**.

1.3 Public test instance

AEON.to offers a public test instance to ease development on services depening on its API. The instance is connected to the Aeon stage network and Bitcoin test network. (The Aeon test network is intended for protocol development and is not suitable to test service integration).

You can create an order, send Aeon stagenet coins to AEON.to, and will receive Bitcoin testnet coins on successful order execution.

Hint: The addresses used look different from mainnet ones: Aeon stagenet addresses start with *S*, while Bitcoin testnet addresses start with *m*, *n* or *2*.

1.3.1 Access

The public test instance is accessible under https://test.aeon.to

1.3.2 Funding

As there is no BTC/AEON exchange available on the stagenet, the hotwallet of the public test instance cannot be automatically refilled. If the service runs low (i.e., the max order is very small), simply send Bitcoin testnet coins to the following address:

```
mxRHStxLBEFfL5WavorMjB6ADrD8rCA2hC
```

1.4 Problems?

Please check:

- Are you including the proper parameters?
- Are you using the proper request type *POST* vs. *GET*?
- Are you setting `"Content-Type: application/json"` in headers?
- Getting redirected? Add a / at the end of the API endpoint!

If none of this resolves the problem, please contact support.

1.4.1 Contact

- AEON.to support: support@aeon.to

Preface

Author AEON.to support

Contact support@aeon.to

Organization AEON.to

Copyright Copyright (C) 2015-2019 AEON.to

Abstract AEON.to is a service that allows users to pay any bitcoin address anonymously using Aeon. .. Aeon is a secure, private, untraceable currency that is open-source .. and freely available to all. ..

AEON.to converts Aeon provided by the user to bitcoins (BTC), which are sent to a given bitcoin address.

AEON.to provides an API that enables developers to use the service in programs, apps, scripts etc. This API therefore allows developers to integrate an option to pay any bitcoin address in their product, for example, in a Aeon wallet service. The API is a REST-like API using *JSON* as data format. It is available anonymously and does not require identification or authentication. This document describes this API and gives examples of its usage.