DEV PORTAL

Articles

API troubleshooting

In this article, we will cover:

What is the rate limit?

As of August 2023, all free public API usage has been reduced to 1 requests per second (rps).

> **INFO**
>
> If you are an enterprise with significant API request volumes, please complete this application so we can assign you a custom API endpoint. The enterprise endpoint will offer significantly better performance across market rates and response times.

"Insufficient Liquidity" error

This error can occur for a number of reasons:

- The liquidity pool has not been queried - make sure one of the aggregated sources has at least 10k of a connector token listed below.
- If you're getting a quote, please make sure to take into account the decimals of the token. Do not include decimal places in the "from token amount". The API returns the number of decimal places the token supports when calling /quote or /swap
- There's an error in the server and is unable to find a path, this may require contacting the 1inch support team for more investigation.

"Cannot Estimate"

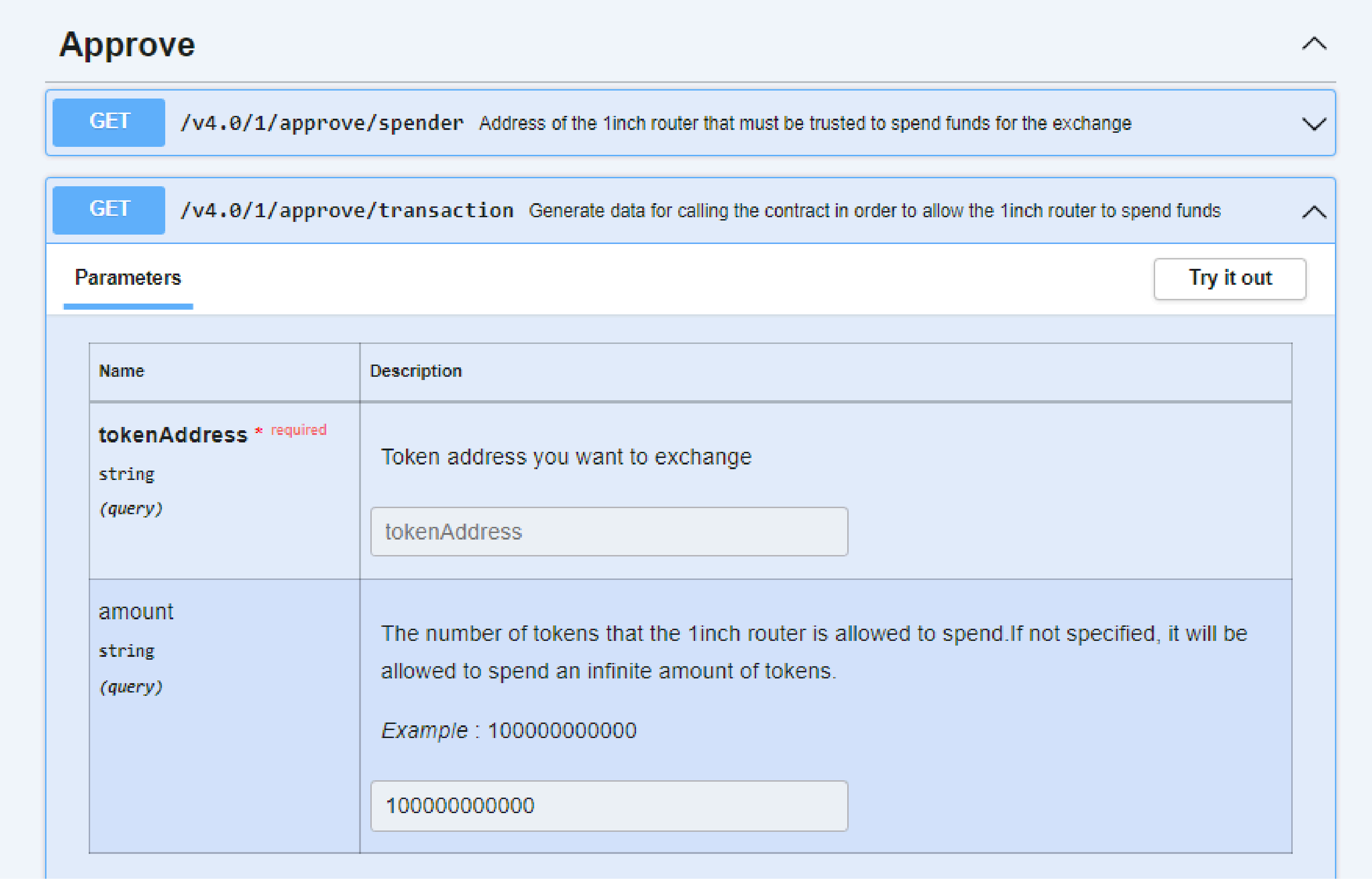
This error occurs when the eth_estimateGas RPC call throws an error. There are many reasons why this can throw an error, and they all tend to mean the transaction is guaranteed to fail. This is a protection, so a transaction is broadcast that's going to fail. To find the precise error, disableEstimate=true can be used in the query and an eth_estimateGas RPC call can be done locally. The transaction can also be simulated with a platform like tenderly or blocknative for a better look at the transaction trace.

Some common reasons why a transaction may fail:

- A token has a fee on transfer or swap and the slippage tolerance needs to be increased
- A token has a fee on transfer *and* the fee and referrer parameter is set causing the transaction to always fail
- The rate has expired and changed in the latest block.

Troubleshooting insufficient allowance / balance

In the case of one of these errors, you may not have a token approval or you're missing the token balance. To do a token approval, you can use the API to approve the token to the 1inch router.



Approve

GET `/v4.0/1/approve/spender` Address of the 1inch router that must be trusted to spend funds for the exchange

GET `/v4.0/1/approve/transaction` Generate data for calling the contract in order to allow the 1inch router to spend funds

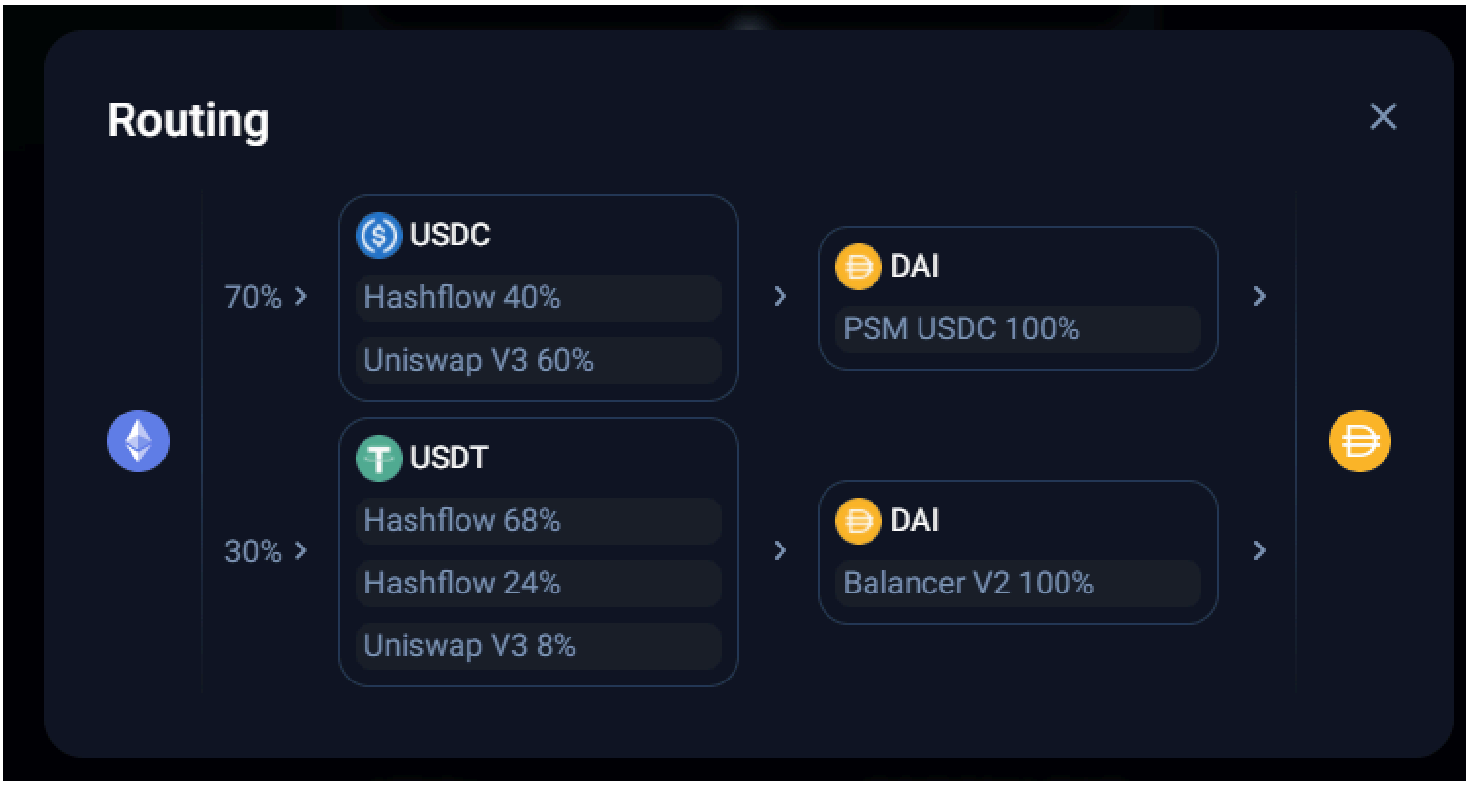
Name	Description
tokenAddress * required string *(query)*	Token address you want to exchange tokenAddress
amount string *(query)*	The number of tokens that the 1inch router is allowed to spend.If not specified, it will be allowed to spend an infinite amount of tokens. *Example* : 100000000000 100000000000

What are the "parts" parameters?

The parts parameters is comprised of 3 things

- mainRouteParts
- virtualParts
- parts

Each of these has their own unique meaning. The main route parts denotes the initial split and can be seen on the UI. In this example it's the 70%/30% split



Routing

70% >

USDC

Hashflow 40%

Uniswap V3 60%

DAI

PSM USDC 100%

30% >

USDT

Hashflow 68%

Hashflow 24%

Uniswap V3 8%

DAI

Balancer V2 100%

The parts parameter is how many individual blocks there can be. In the picture above we can see 4 blocks with the top being the first main route part with 2 parts to it and the bottom one being the second main route part with 2 parts in it.

The virtual parts is the splits *inside* of a part. We can see in the first main route part above, in the first block there are 2 virtual parts splitting the USDC between hashflow and uniswap. In the second main route part we can see three virtual parts in the first block, a hashflow route with 2 different rates and one to uniswap.

What is the "Complexity level?"

Complexity level is how many steps a swap can go through, below are the levels and what it means for the swap. Each level adds to the previous level.

- Complexity level 0: the source token can be wrapped or unwrapped and swapped through one liquidity pool per main route part.
- Complexity level 1: the source token can be wrapped or unwrapped and swapped through two liquidity pools. This means there can be up to one connector token in each main route part.

- Complexity level 2 (Default): the source token can be wrapped or unwrapped and swapped through three liquidity pools. This means there can be up to two connector tokens in each main route part.
- Complexity level 3: the source token can be wrapped or unwrapped and swapped through four liquidity pools. This means there can be up to three connector tokens in each main route part.

How can I limit the liquidity sources in the query?

To limit the liquidity sources to a query, you can manually enter the pool-contacts that you to use using the "protocols" parameter. All other liquidity sources will be excluded by default.

Still have questions?

Feel free to reach out to us in the live support chat!

Previous
‹ Common API Error Messages

Next
CORS Errors ›