Quickstart guide

1inch Transaction Traces API:

1inch's Transaction Traces API provides users with an efficient way to access transaction traces for different chains.

1. Check Synced Interval:

Before fetching any transaction traces, it's useful to know the synced interval. It gives the range of block numbers that have been synced and are available for query.

Endpoint:

```
GET /v1.0/chain/{chain}/synced-interval
```

Parameters:

- `chain` (required): Chain ID. For Ethereum mainnet, it would be `1`.

Sample Request:

```
https://api.1inch.dev/traces/v1.0/chain/1/synced-interval
```

2. Get Block Trace by Number:

To retrieve the transaction traces of all transactions in a specific block.

Endpoint:

```
GET /v1.0/chain/{chain}/block-trace/{blockNumber}
```

Parameters:

- `chain` (required): Chain ID.

- `blockNumber` (required): The block number you wish to retrieve traces from.

Sample Request:

```
https://api.1inch.dev/traces/v1.0/chain/1/block-trace/15000000
```

3. Get Block Trace by Transaction Hash:

If you are specifically interested in the trace of a particular transaction within a block, you can fetch it using the transaction hash.

Endpoint:

```
GET /v1.0/chain/{chain}/block-trace/{blockNumber}/tx-hash/{txHash}
```

Parameters:

- `chain` (required): Chain ID.
- `blockNumber` (required): The block number containing the transaction.
- `txHash` (required): The hash of the specific transaction you want to trace.

Sample Request:

```
https://api.1inch.dev/traces/v1.0/chain/1/block-trace/17378177/tx-hash/0x16897e492b2e
```

Prerequisites:

- Node.js and npm installed.
- A text editor or IDE.
- API key for the Transaction Traces API.

Steps:

1. Initialize a New Node.js Project:

   ```
   mkdir traces_api_tutorial
   cd traces_api_tutorial
   npm init -y
   ```

2. Install the Required Packages:
   We'll be using `axios` to make the HTTP requests:

   ```
   npm install axios
   ```

3. Set Up Your API Key:

For the sake of security, store your API key in a `.env` file:

```
echo "API_KEY=YOUR-API-KEY" > .env
```

Then, install `dotenv` to use this file:

```
npm install dotenv
```

4. Write the Code:

   Create a new file named `index.js` and add the following:

```
require("dotenv").config();
const axios = require("axios");

const BASE_URL = "https://api.1inch.dev/traces/v1.0/chain";

async function getSyncedInterval(chain) {
  const url = `${BASE_URL}/${chain}/synced-interval`;
  const response = await axios.get(url, {
    headers: {
      Authorization: `Bearer ${process.env.API_KEY}`,
    },
  });
  return response.data;
}

async function getBlockTraceByNumber(chain, blockNumber) {
  const url = `${BASE_URL}/${chain}/block-trace/${blockNumber}`;
  const response = await axios.get(url, {
    headers: {
      Authorization: `Bearer ${process.env.API_KEY}`,
    },
  });
  return response.data;
}

async function getBlockTraceByNumberAndTxHash(chain, blockNumber, txHash) {
  const url = `${BASE_URL}/${chain}/block-trace/${blockNumber}/tx-hash/${txHash}`;
  const response = await axios.get(url, {
    headers: {
      Authorization: `Bearer ${process.env.API_KEY}`,
    },
  });
  return response.data;
}

function sleep(ms) {
  return new Promise((resolve) => setTimeout(resolve, ms));
}

(async function () {
  try {
    const chain = 1; // Example chain
    const blockNumber = 15000000; // Example block number
    const txHash =
      "0x16897e492b2e023d8f07be9e925f2c15a91000ef11a01fc71e70f75050f1e03c"; // Example transaction has
h

    const syncedInterval = await getSyncedInterval(chain);
    console.log("Synced Interval:", syncedInterval);

    await sleep(1000); // Sleep for 1 second to avoid rate limit

    const blockTrace = await getBlockTraceByNumber(chain, blockNumber);
    console.log("Block Trace:", blockTrace);

    await sleep(1000); // Sleep for 1 second to avoid rate limit

    const txTrace = await getBlockTraceByNumberAndTxHash(
      chain,
      blockNumber,
      txHash,
    );
    console.log("Transaction Trace:", txTrace);
  } catch (error) {
    console.error(
      "Error fetching data:",
      error.response ? error.response.data : error.message,
    );
  }
})();
```

5. Run Your Script:

```
node index.js
```

Transaction traces are a powerful tool for anyone interacting with or developing on the blockchain. They provide invaluable insights into the workings of smart contracts, helping improve efficiency, security, and understanding. With 1inch's Transaction Traces API, accessing these insights becomes a seamless process, enabling deeper interaction with blockchain data.

With the steps above, you'll be able to interact with the Transaction Traces API using Node.js. Make sure you replace `'API-KEY'` in the `.env` file with your actual API key.

Previous
‹ Introduction

Next
Get synced interval ›

© 2025 1inch Limited | Privacy Policy | Terms of Service | Commercial API Terms of Use