

Quickstart guide

In this quick-start example, we will show you how to use the 1inch Portfolio API (v4) to retrieve the current value of tokens, their profit and loss (PnL) with return on investment (ROI), and detailed statistics for each token held.

Calls and endpoints used:

getCurrentValue: Retrieves the current value of ERC20 tokens for a specific wallet and blockchain.

https://api.1inch.dev/portfolio/portfolio/v4/overview/erc20/current_value

getProfitAndLoss: Retrieves the PnL and ROI for those tokens over a specified time range.

https://api.1inch.dev/portfolio/portfolio/v4/overview/erc20/profit_and_loss

getTokenDetails: Retrieves detailed statistics about the tokens.

<https://api.1inch.dev/portfolio/portfolio/v4/overview/erc20/details>

First, we will initialize the project and install dependencies:

```
npm init -y
```

```
npm install dotenv
```

Next, let's create a `.env` file and add the following:

```
API_KEY = PASTE_YOUR_API_KEY_HERE;  
WALLET_ADDRESS = PASTE_YOUR_WALLET_ADDRESS_HERE;  
CHAIN_ID = 1; //using Ethereum mainnet in this example
```

Next, we're going to reference the environment variables from the `.env` file, define our HTTP request helper function, and add a time delay to stay within the free account tier's 1 request per second limit during testing.

```

require("dotenv").config();

const delay = (ms) => new Promise((resolve) => setTimeout(resolve, ms));

async function getCurrentValue(walletAddress, chainId) {
  const endpoint = `https://api.1inch.dev/portfolio/portfolio/v4/overview/erc20/current_value?addresses=${walletAddress}&chain_id=${chainId}`;
  const data = await fetch(endpoint, {
    headers: { Authorization: `Bearer ${process.env.API_KEY}` },
  }).then((res) => res.json());
  return data;
}

```

Then, we'll get the current value for supported ERC20 tokens:

```

async function getCurrentValue(walletAddress, chainId) {
  const endpoint = `https://api.1inch.dev/portfolio/portfolio/v4/overview/erc20/current_value?addresses=${walletAddress}&chain_id=${chainId}`;
  const data = await fetch(endpoint, {
    headers: { Authorization: `Bearer ${process.env.API_KEY}` },
  }).then((res) => res.json());
  return data;
}

```

Next, we will get the profit and loss (PnL) and return on investment (ROI) for supported tokens:

```

async function getProfitAndLoss(
  walletAddress,
  chainId,
  fromTimestamp,
  toTimestamp,
) {
  const endpoint = `https://api.1inch.dev/portfolio/portfolio/v4/overview/erc20/profit_and_loss?addresses=${walletAddress}&chain_id=${chainId}&from_timestamp=${fromTimestamp}&to_timestamp=${toTimestamp}`;
  const data = await fetch(endpoint, {
    headers: { Authorization: `Bearer ${process.env.API_KEY}` },
  }).then((res) => res.json());
  return data;
}

```

Then, we'll request further details for supported tokens, in addition to PnL and ROI:

```

async function getTokenDetails(walletAddress, chainId) {
  const endpoint = `https://api.1inch.dev/portfolio/portfolio/v4/overview/erc20/details?addresses=${walletAddress}&chain_id=${chainId}`;
}

```

```

ss}&chain_id=${chainId}`;
    const data = await fetch(endpoint, {
      headers: { Authorization: `Bearer ${process.env.API_KEY}` },
    }).then((res) => res.json());
    return data;
  }
}

```

Finally, we'll add logic to our checker to incorporate configuration items and execute the calls:

- Executes the above calls sequentially, with a 2-second delay between each call.
- Logs the retrieved data to the console.

```

// Main portfolio checker logic
async function main() {
  console.log(
    "Current Value:",
    await getCurrentValue(process.env.WALLET_ADDRESS, process.env.CHAIN_ID),
  );
  await delay(2000);
  console.log(
    "Profit and Loss:",
    await getProfitAndLoss(
      process.env.WALLET_ADDRESS,
      process.env.CHAIN_ID,
      "2023-01-01T00:00:00Z",
      "2023-01-31T23:59:59Z",
    ),
  );
  await delay(2000);
  console.log(
    "Token Details:",
    await getTokenDetails(process.env.WALLET_ADDRESS, process.env.CHAIN_ID),
  );
}

main();

```

You can then run the project using the `node <Your Filename>` command.

Previous

[< Introduction](#)

Next

[Check if portfolio service is available >](#)