Quickstart guide

Introduction

This guide will walk you through on how you can seamlessly retrieve and display wallet transactions by using the wallet history API.

Prerequisites

- Node.js and npm installed
- Basic knowledge of JavaScript, React, and Express.js

Step-by-step guide

Step 1: Initialization

1. Create a new directory for the project:

```
mkdir wallet-history && cd wallet-history
```

2. Initialize a new Node.js project:

```
npm init -y
```

3. Install Express, CORS, and Axios:

```
npm install express cors axios
```

4. Install dotenv:

```
npm install dotenv
```

Then create a new file called .env and add your api key:

```
API_KEY=YOUR_1INCH_API_KEY
```

5. Create a new file `api.js` and set up a basic Express server by pasting into it the following:

```
// Import required packages
const express = require("express");
const axios = require("axios");
const dotenv = require("dotenv");
const cors = require("cors");
const path = require("path");

// Initialize the app and load environment variables from .env
dotenv.config({ path: path.resolve(__dirname, ".env") });
const app = express();

// Enable CORS for all routes
app.use(cors());

const BASE_URL = "https://api.1inch.dev/history/v2.0/history";

// Endpoint to fetch wallet transaction history
app.get("/api/:address/history", async (req, res) => {
  const address = req.params.address;
  const limit = req.query.limit || 10;

  try {
    const constructedUrl = `${BASE_URL}/${address}/events?chainId=${1}&limit=${limit}`;

    const response = await axios.get(constructedUrl, {
      headers: {
        Authorization: `Bearer ${process.env.API_KEY}`, // Use API key from .env
      },
    });

    // Send the response data back to the client
    res.json(response.data);
  } catch (error) {
    console.error("Error fetching wallet transactions:", error);
    res.status(500).json({ error: "An error occurred while fetching data" });
  }
});

// Start the server
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

1. Add an endpoint to fetch wallet transactions and don't forget to replace API_KEY with your 1inch Developer Portal API key:

```
const BASE_URL = "https://api.1inch.dev/history/v2.0/history";

app.get("/api/:address/history", async (req, res) => {
  const address = req.params.address;
  const limit = req.query.limit || 10;

  try {
```

```
    const constructedUrl = `${BASE_URL}/${address}/events?chainId=${1}&limit=${limit}
`;

    const response = await axios.get(constructedUrl, {
      headers: {
        Authorization: `Bearer ${process.env.API_KEY}`,
      },
    });

    // Send the data from the API back to the client
    res.json(response.data.items);
  } catch (error) {
    console.error("Axios Error: ", error.response);
    res.status(500).json({ error: "Failed to fetch wallet transactions" });
  }
});
```

Step 2: Setting up React frontend

1. Create a new React application:

```
npx create-react-app client
```

2. Navigate to the React application directory:

```
cd client
```

3. Create a `History.js` component inside the `src` directory with the following content:

```
import React, { useState, useEffect } from "react";

const History = ({ address }) => {
  const [transactions, setTransactions] = useState([]);
  const [isLoading, setLoading] = useState(false);

  useEffect(() => {
    const fetchData = async () => {
      setLoading(true);
      const response = await fetch(
        `http://localhost:5001/api/${address}/history`,
      );

      if (!response.ok) {
        console.log("Fetch history error", response);
        return;
      }
```

```
      const transactions = await response.json();

      setTransactions(transactions);
      setLoading(false);
    };

    if (isLoading) {
      return;
    }

    fetchData();
  }, [address]);

  if (isLoading) {
    return <div>Loading...</div>;
  }

  return (
    <ul>
      {transactions.map((tx) => (
        <li key={tx.id}>
          <a href={`https://etherscan.io/tx/${tx.details.txHash}`}>
            {tx.details.type}
          </a>
        </li>
      ))}
    </ul>
  );
};

export default History;
```

4. Import and use `History` in `src/App.js`:

```
import React, { useState } from "react";
import "./App.css";
import History from "./History";

function App() {
  const [inputValue, setSearchTerm] = useState("");
  const [address, setAddress] = useState();

  const handleChange = (event) => {
    setSearchTerm(event.target.value);
  };

  const handleSubmit = (event) => {
```

```
    event.preventDefault();

    setAddress(inputValue);
  };

  return (
    <div className="App">
      <header>
        <form onSubmit={handleSubmit}>
          <input
            type="text"
            value={inputValue}
            placeholder="search by address"
            onChange={handleChange}
          />
          <button type="submit">Search</button>
        </form>
      </header>

      {address && <h1>Ethereum history of {address}</h1>}
      {address && <History address={address} />}
    </div>
  );
}

export default App;
```

Step 3: Running the project

1. Start the Express server:

```
node api.js
```

2. In a new terminal, navigate to the client directory and start the React app:

```
cd client
```

Now, you can view your wallet transaction history at http://localhost:3000.

Described above is a basic setup and you can expand upon this by adding more features, error handling, and styling to get to production.

Previous
‹ Introduction

Next
Returns history events for address ›