Quickstart guide

This guide will help you set up a project to retrieve and display chart data using the 1inch Charts API. You'll create a Node.js backend to handle API requests and serve data, and a React frontend to display this data in a user-friendly way.

Once finished, you'll have built a simple yet powerful NFT collection viewer that you can expand upon and customize!

Prerequisites

- You'll need Node.js and npm installed.
- Basic knowledge of JavaScript, React, and Express.js.

Step 1: Set Up the Node Backend

This backend will handle API requests, fetch data from the Charts API, and serve your frontend React application.

1.1. Create a new project directory

```
mkdir nft-collection && cd nft-collection
```

1.2. Initialize a new Node.js project

This will create the `package.json' file.

```
npm init -y
```

1.3. Install necessary packages

- Express: A web framework for Node.js, used to build the backend server.
- Axios: A HTTP client, used to make API requests.

- CORS: Middleware to handle Cross-Origin Resource Sharing, essential when the frontend and backend are on different ports.
- Path: A Node.js module for working with file and directory paths.

```
npm install express axios cors path
```

1.4. Set up a basic Express server

Create a file named `api.js` and add the following code. This will initialize an Express server to handle API requests and serve the React app (see Step 2 below).

```
const express = require("express");
const axios = require("axios");
const cors = require("cors");
const path = require("path");

const app = express();
const PORT = 3000;
const BASE_URL = "https://api.1inch.dev/charts/v1/...";

app.use(cors()); // Handle CORS issues

// Serves static files from the React app
app.use(express.static(path.join(__dirname, "client/build")));

// Endpoint to fetch chart data
app.get("/charts", async (req, res) => {
  try {
    const response = await axios.get(BASE_URL, {
      headers: { Authorization: `Bearer ${process.env.API_KEY}` },
    });
    res.send(response.data);
  } catch (error) {
    console.error("Error fetching charts data:", error);
    res.status(500).json({ error: "Failed to fetch charts data" });
  }
});

// Serve the React app for other routes
app.get("*", (req, res) => {
  res.sendFile(path.join(__dirname, "client/build", "index.html"));
});

app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

Step 2: Setting Up React Frontend

The frontend will display the data fetched from the backend API.

2.1. Create a new React app

This creates a new React application for your frontend code.

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

2.2. Navigate to the React app directory

```
cd client
```

2.3. Install Axios

This will be used to make HTTP requests from the React app to the backend.

```
npm install axios
```

2.4. Create an NFT list component

This component will handle fetching and displaying NFT data. Create a file with this directory and name: `src/NFTList.js`. Then add the following code:

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

const NFTList = ({ address }) => {
  // Initialize state to store NFT data
  const [nfts, setNfts] = useState([]);

  useEffect(() => {
    // Fetch NFT data from the server
    const fetchData = async () => {
      try {
        // Fetch NFTs for the given address
        const response = await axios.get(`/fetchNfts?address=${address}`);
        // Update the state with the fetched NFT data
        setNfts(response.data.assets);
      } catch (error) {
        console.error("Error fetching NFTs:", error);
      }
    };

    fetchData();
  }, [address]); // Refetch if the address changes

  return (
    <div className="Nft-list">
      {/* Render the list of NFTs */}
      {nfts.map((nft) => (
        <div key={nft.id}>
          <img src={nft.image_url} alt={nft.name} width="150" />
```

```
            <h2>{nft.name}</h2>
            <p>{nft.description}</p>
          </div>
        ))}
      </div>
    );
};

export default NFTList;
```

2.5. Update the main App component

Integrate the NFT list component into the main application. Add this to `src/App.js`:

```
import React from "react";
import "./App.css";
import NFTList from "./NFTList";

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1>My NFT Collection</h1>
      </header>
      <NFTList address="0xd8da6bf26964af9d7eed9e03e53415d37aa96045" />
    </div>
  );
}

export default App;
```

Step 3: Running the Project

3.1. Start the Express server

```
node api.js
```

3.2. Start the React app

Navigate to the client directory and run the React development server:

```
cd client
npm start
```

You should now have a functional application for viewing NFT collections! While primitive, this framework can easily be built upon with features such as web3 wallet connections, auction mechanics, and more.

Previous
Introduction
Next
Allows to get historical line chart data for specific token pair and period