Quickstart guide

Getting Started with the NFT API with React and Node.js

The 'List NFTs' API demo will walk you through on how users can seamlessly retrieve and display a curated selection of non-fungible tokens (NFTs) from a specified collection or platform.

Prerequisites:

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

Steps:

1. Setting Up Node Backend

1.1. Create a new directory for the project:

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

1.2. Initialize a new Node.js project:

```
npm init -y
```

1.3. Install Express and Axios:

```
npm install express axios
```

1.4. Create a new file `api.js` and set up a basic Express server:

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

const app = express();
```

```
const PORT = 5000;
const BASE_URL = "https://api.1inch.dev/nft/v1/byaddress";

app.use(cors()); // To handle CORS issues when making requests to the front end

// Serve static files from the React app
app.use(express.static(path.join(__dirname, "nft-collection/build")));

// We will route all other requests to the nft-collection build
app.get("*", (req, res) => {
  res.sendFile(path.join(__dirname, "nft-collection/build", "index.html"));
});

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

1.5. Add an endpoint to fetch NFTs (replace `API_KEY`):

```
const BASE_URL = "https://api.1inch.dev/nft/v1/byaddress";

app.get("/fetchNfts", async (req, res) => {
  const address = req.query.address || "0xd8da6bf26964af9d7eed9e03e53415d37aa96045";
  const limit = req.query.limit || 50;
  const offset = req.query.offset || 0;
  const chainIds = req.query.chainIds || 1;

  try {
    const constructedUrl = `${BASE_URL}?address=${address}&chainIds=${chainIds}&limit=${limit}&offset=${off
set}`;

    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);
  } catch (error) {
    console.error("Axios Error: ", error.response);
    res.status(500).json({ error: "Failed to fetch NFTs" });
  }
});
```

2. Setting Up React Frontend

2.1. Create a new React app:

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

2.2. Navigate to the React app directory:

```
cd client
```

2.3. Install Axios:

```
npm install axios
```

2.4. Create a component `NFTList.js` inside the `src` directory:

```
import React, { useState, useEffect } from "react";
import { fetchNFTs } from "./api";

const NFTList = ({ address }) => {
  const [nfts, setNfts] = useState([]);

  useEffect(() => {
    const fetchData = async () => {
      try {
        const response = await fetchNFTs(address);
        setNfts(response.data.assets);
      } catch (error) {
        console.error("Error fetching NFTs:", error);
      }
    };
    fetchData();
  }, [address]);

  return (
    <div className="Nft-list">
   {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. Import and use `NFTList` in `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 />
    </div>
    );
}

export default App;
```

3. Running the Project

3.1. Start the Express server:

```
node api.js
```

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

```
cd client
npm start
```

Now, you can view your NFT collection visualization at `http://localhost:3000`.

That's it! This 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
NftDevV2Controller_supportedChain_v2 ›