import express from "express";

import cors from "cors";

import save\_router from "./routers/save\_router.js";

import fetch\_router from "./routers/fetch\_router.js";

const app = express();

const PORT = process.env.PORT || 8000;

// middleware

app.use(express.urlencoded({ extended: true }));

app.use(express.json());

app.use(cors());

// routes

app.use("/save", save\_router);

app.use("/fetch", fetch\_router);

app.get("/", (req, res) => {

res.send("Welcome to the server, Check /api-list for available routes");

});

app.get("/api-list", (req, res) => {

const apiList = {

save\_routes: ["/save/single", "/save/multiple", "save/dog"],

fetch\_routes: ["/fetch/single", "/fetch/multiple"],

};

res.send(apiList);

});

app.listen(PORT, () => {

console.log(`http://localhost:${PORT}`);

});

app.use("", (req, res) => {

res.send(`No request for ${req.url} exists`);

});

import { useState } from "react";

const App = () => {

// what do we need to track

const [singleFile, setSingleFile] = useState(null);

// const [multipleFiles, setMultipleFiles] = useState([]);

const [displayImage, setDisplayImage] = useState(null);

const [displayImages, setDisplayImages] = useState([]);

const [displayDogImage, setDisplayDogImage] = useState("");

const [message, setMessage] = useState("");

// Handlers

const handleSingleFileChange = (e) => {

if (e.target.files.length > 0) {

setSingleFile(e.target.files[0]);

}

};

// fetch functions -> fetch a random single image

const fetchSingleFile = async () => {

try {

const response = await fetch(`http://localhost:8000/fetch/single`);

const blob = await response.blob(); // we made a blob - Binary Large Object

// but thats not an image, so we need to make an image element

// using createObjectURL

const imageUrl = URL.createObjectURL(blob);

setDisplayImage(imageUrl);

} catch (error) {

console.error("Error fetching single file:", error);

}

};

// fetch functions -> save single

const handleSubmitSingleFile = async (e) => {

e.preventDefault();

if (!singleFile) {

setMessage("Please select a file before uploading.");

return;

}

try {

const formData = new FormData();

formData.append("file", singleFile);

const response = await fetch(`http://localhost:8000/save/single`, {

method: "POST",

body: formData,

});

const data = await response.json();

if (!response.ok) {

throw new Error(data.error || "Image upload failed");

}

setMessage("File uploaded successfully!");

} catch (error) {

console.log("Error:", error);

}

};

// fetch functions -> fetch multiple [TODO]

const fetchMultipleFiles = async () => {

try {

// fetch -/fetch/multiple => [01, 02, 03, 04, 05]

const response = await fetch(`http://localhost:8000/fetch/multiple`);

const data = await response.json();

console.log(data);

// fetch -/fetch/file/filename variable

const filePromises = data.map(async (filename) => {

const fileResponse = await fetch(

`http://localhost:8000/fetch/file/${filename}`

);

const fileBlob = await fileResponse.blob();

const imageUrl = URL.createObjectURL(fileBlob);

return imageUrl;

});

const imageUrls = await Promise.all(filePromises);

setDisplayImages(imageUrls);

} catch (error) {

console.error(error);

}

};

// fetch functions -> fetch dog image [TODO]

const fetchDogImage = async () => {

try {

const response = await fetch(`https://dog.ceo/api/breeds/image/random`);

const data = await response.json();

setDisplayDogImage(data.message);

} catch (error) {

console.error(error);

}

};

// fetch functions -> save dog image [TODO]

const saveDogImage = async () => {

try {

const fileResponse = await fetch(displayDogImage);

const blob = await fileResponse.blob();

const formData = new FormData();

formData.append("file", blob, "dog-img.jpg");

// <button onClick={saveDogImage}>Save Dog Image</button>

const response = await fetch(`http://localhost:8000/save/single`, {

method: "POST",

body: formData,

});

const data = await response.json();

console.log(data);

} catch (error) {

console.error(error);

}

};

return (

<div>

<p>{message}</p>

<h2>Fetch Single Random Image</h2>

<button onClick={fetchSingleFile}>Fetch Single File</button>

{displayImage && (

<div>

<h3>Single File</h3>

<img

src={displayImage}

alt="Display Image"

style={{ width: "200px", marginTop: "10px" }}

/>

</div>

)}

<form onSubmit={handleSubmitSingleFile}>

<h2>Upload Single File</h2>

<input type="file" onChange={handleSingleFileChange} />

<button type="submit">Upload Single File</button>

</form>

<button onClick={fetchMultipleFiles}>Fetch Multiple Files</button>

{displayImages.length > 0 ? (

displayImages.map((imageUrl, index) => (

<div key={index}>

<img

src={imageUrl}

style={{ width: "200px" }}

alt={`Image ${index}`}

/>

</div>

))

) : (

<p>No Images to display</p>

)}

<button onClick={fetchDogImage}>Fetch Dog Image</button>

{displayDogImage && (

<div>

<img src={displayDogImage} style={{ width: "200px" }} />

<button onClick={saveDogImage}>Save Dog Image</button>{" "}

</div>

)}

</div>

);

};

export default App;