const express = require("express");

const { createActor } = require("xstate");

const apiMachine = require('./src/Machines/machine')

const cors=require('cors');

const app = express();

app.use(express.json());

app.use(cors())

app.post("/api/login", async (req, res) => {

const endpoint = 'http://10.51.41.12:8082/users/login';

const payload = req.body;

console.log("Payload before hitting api" + JSON.stringify(payload));

if (!endpoint) return res.status(400).json({ error: "URL is required" });

try {

const service = createActor(apiMachine);

service.subscribe((snapshot)=>{

console.log("FSM Initial State:" +snapshot.value)

})

// const state = service.getSnapshot();

service.start();

// service.stop();

} catch (error) {

res.status(500).json({ error: "Unexpected error occurred" });

}

});

app.listen(3000, () => console.log("Server running on port 3000"));

const { setup,createMachine, createActor, assign } = require("xstate");

const axios = require("axios");

const apiMachine = setup({}).createMachine({

id: "apiRequest",

initial: "active",

context: {

data: null,

error: null,

endpoint: null,

},

states: {

active: {

on:{

toggle: {target:'loading'},

}

},

loading: {

invoke: {

src: "fetchData",

onDone: { target: "success", actions: "setData" },

onError: { target: "failure", actions: "setError" },

},

},

success: {

type: "final",

},

failure: {

type: "final",

},

},

}, {

actions: {

setData: assign((context, event) => ({ data: event.output, error: null })),

setError: assign((context, event) => ({ error: event.error })),

},

services: {

services: {

fetchData: async (context) => {

console.log("fetchData service invoked with context:", context); // <---- ADD THIS LOG

console.log("Attempting to make POST request to:", context.endpoint); // <---- ADD THIS LOG

console.log("Request body:", JSON.stringify(context.data)); // <---- ADD THIS LOG

try {

const response = await axios.post(context.endpoint, context.data);

return response.data;

} catch (error) {

console.error("Error in fetchData service:", error);

throw error;

}

},

},

},

});

module.exports = { apiMachine, createActor };