Server.js

const http = require("http");

const getReq = require("./methods/get-request");

const postReq = require("./methods/post-request");

const putReq = require("./methods/put-request");

const deleteReq = require("./methods/delete-request");

let movies = require("./data/movies.json");

//require("dotenv").config();

const PORT = process.env.PORT || 5001;

const server = http.createServer((req, res) => {

req.movies = movies;

switch (req.method) {

case "GET":

getReq(req, res);

break;

case "POST":

postReq(req, res);

break;

case "PUT":

putReq(req, res);

break;

case "DELETE":

deleteReq(req, res);

break;

default:

res.statusCode = 404;

res.setHeader("Content-Type", "application/json");

res.write(

JSON.stringify({ title: "Not Found", message: "Route not found" })

);

res.end();

}

});

server.listen(PORT, () => {

console.log(`Server started on port : ${PORT}`);

});

**Delete.js method**

const writeToFile = require("../util/write-to-file");

module.exports = (req, res) => {

let baseUrl = req.url.substring(0, req.url.lastIndexOf("/") + 1);

let id = req.url.split("/")[3];

const regexV4 = new RegExp(

/^[0-9A-F]{8}-[0-9A-F]{4}-4[0-9A-F]{3}-[89AB][0-9A-F]{3}-[0-9A-F]{12}$/i

);

if (!regexV4.test(id)) {

res.writeHead(400, { "Content-Type": "application/json" });

res.end(

JSON.stringify({

title: "Validation Failed",

message: "UUID is not valid",

})

);

} else if (baseUrl === "/api/movies/" && regexV4.test(id)) {

const index = req.movies.findIndex((movie) => {

return movie.id === id;

});

if (index === -1) {

res.statusCode = 404;

res.write(

JSON.stringify({ title: "Not Found", message: "Movie not found" })

);

res.end();

} else {

req.movies.splice(index, 1);

writeToFile(req.movies);

res.writeHead(204, { "Content-Type": "application/json" });

res.end(JSON.stringify(req.movies));

}

} else {

res.writeHead(404, { "Content-Type": "application/json" });

res.end(JSON.stringify({ title: "Not Found", message: "Route not found" }));

}

};

**Get.js method**

module.exports = (req, res) => {

let baseUrl = req.url.substring(0, req.url.lastIndexOf("/") + 1);

let id = req.url.split("/")[3];

const regexV4 = new RegExp(

/^[0-9A-F]{8}-[0-9A-F]{4}-4[0-9A-F]{3}-[89AB][0-9A-F]{3}-[0-9A-F]{12}$/i

);

if (req.url === "/api/movies") {

res.statusCode = 200;

res.setHeader("Content-Type", "application/json");

res.write(JSON.stringify(req.movies));

res.end();

} else if (!regexV4.test(id)) {

res.writeHead(400, { "Content-Type": "application/json" });

res.end(

JSON.stringify({

title: "Validation Failed",

message: "UUID is not valid",

})

);

} else if (baseUrl === "/api/movies/" && regexV4.test(id)) {

res.setHeader("Content-Type", "application/json");

let filteredMovie = req.movies.filter((movie) => {

return movie.id === id;

});

if (filteredMovie.length > 0) {

res.statusCode = 200;

res.write(JSON.stringify(filteredMovie));

res.end();

} else {

res.statusCode = 404;

res.write(

JSON.stringify({ title: "Not Found", message: "Movie not found" })

);

res.end();

}

} else {

res.writeHead(404, { "Content-Type": "application/json" });

res.end(JSON.stringify({ title: "Not Found", message: "Route not found" }));

}

};

**Post method**

const crypto = require("crypto");

const requestBodyparser = require("../util/body-parser");

const writeToFile = require("../util/write-to-file");

module.exports = async (req, res) => {

if (req.url === "/api/movies") {

try {

let body = await requestBodyparser(req);

body.id = crypto.randomUUID();

req.movies.push(body);

writeToFile(req.movies);

res.writeHead(201, { "Content-Type": "application/json" });

res.end();

} catch (err) {

console.log(err);

res.writeHead(400, { "Content-Type": "application/json" });

res.end(

JSON.stringify({

title: "Validation Failed",

message: "Request body is not valid",

})

);

}

} else {

res.writeHead(404, { "Content-Type": "application/json" });

res.end(JSON.stringify({ title: "Not Found", message: "Route not found" }));

}

};

**Put method**

const requestBodyparser = require("../util/body-parser");

const writeToFile = require("../util/write-to-file");

module.exports = async (req, res) => {

let baseUrl = req.url.substring(0, req.url.lastIndexOf("/") + 1);

let id = req.url.split("/")[3];

const regexV4 = new RegExp(

/^[0-9A-F]{8}-[0-9A-F]{4}-4[0-9A-F]{3}-[89AB][0-9A-F]{3}-[0-9A-F]{12}$/i

);

if (!regexV4.test(id)) {

res.writeHead(400, { "Content-Type": "application/json" });

res.end(

JSON.stringify({

title: "Validation Failed",

message: "UUID is not valid",

})

);

} else if (baseUrl === "/api/movies/" && regexV4.test(id)) {

try {

let body = await requestBodyparser(req);

const index = req.movies.findIndex((movie) => {

return movie.id === id;

});

if (index === -1) {

res.statusCode = 404;

res.write(

JSON.stringify({ title: "Not Found", message: "Movie not found" })

);

res.end();

} else {

req.movies[index] = { id, ...body };

writeToFile(req.movies);

res.writeHead(200, { "Content-Type": "application/json" });

res.end(JSON.stringify(req.movies[index]));

}

} catch (err) {

console.log(err);

res.writeHead(400, { "Content-Type": "application/json" });

res.end(

JSON.stringify({

title: "Validation Failed",

message: "Request body is not valid",

})

);

}

} else {

res.writeHead(404, { "Content-Type": "application/json" });

res.end(JSON.stringify({ title: "Not Found", message: "Route not found" }));

}};