const http = require('http');

const port = '8084';

const url = require('url');

const fs = require('fs');

const server = http.createServer(processRequest);

server.listen(port);

function getPeople(res) {

fs.readFile('./csv/people.csv', (err, data) => {

if (err) {

res.writeHead(404);

res.end();

} else {

var peopleList = data.toString().split('\r\n');

var peopleArray = [];

for (var i = 1; i < peopleList.length; i++) {

temp = peopleList[i].split(',');

let people = {

id: temp[0],

name: temp[1],

height: temp[2],

mass: temp[3],

skin\_color: temp[4],

hair\_color: temp[5],

birth\_year: temp[6],

gender: temp[7]

}

peopleArray.push(people);

}

res.writeHead(200, { 'content-type': 'text/html' });

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

res.end();

}

})

}

function getPlanets(resplanets) {

fs.readFile('./csv/planets.csv', (err, data) => {

if (err) {

resplanets.writeHead(404);

resplanets.end();

} else {

var planetList = data.toString().split('\r\n');

var planetArray = [];

for (var i = 1; i < planetList.length; i++) {

temp2 = planetList[i].split(',');

let planet = {

id: temp2[0],

name: temp2[1],

climate: temp2[2],

diameter: temp2[3]

}

planetArray.push(planet);

}

resplanets.writeHead(200, { 'content-type': 'text/html' });

resplanets.write(JSON.stringify(planetArray));

resplanets.end();

}

})

}

function processRequest(request, response) {

const parsedUrl = url.parse(request.url, true);

console.log(parsedUrl);

switch (parsedUrl.pathname) {

case '/api/people': {

getPeople(response);

} break;

case '/api/planets': {

getPlanets(response);

} break;

case '/api/people?id': {

getPeople(response.filter(person=>person.id=query.id));

} break;

case '/api/planets?id': {

const data = getPlanets.filter(planet => planet.id = parsedUrl.query.id);

response.writeHead(200);

response.write(data);

response.end();

} break;

case '/api/people?name': {

const data3 = getPeople(response).filter(person => person.name = query.name);

response.writeHead(200);

response.write(data3);

response.end();

} break;

case '/api/planets?name': {

const data4 = getPlanets(response).filter(planet => planet.name = query.name);

response.writeHead(200);

response.write(data4);

response.end();

} break;

}

}