# REST, NodeJS and Express Library

## REST (Representational State Transfer)

### What is REST API

REST API is built on top of HTTP protocol, which defines a specific structure of software development.

1. Resources (entities, includes text, image, music, services, etc.)
   1. Every **URL** in REST represents one type of resource
2. Representation (How to represent those entities in data format, text, HTML, JSON, XML)
   1. Usually use “content-type” in “header” to specify what it is
3. State Transfer (Use four states to manage resources, includes **GET**, **PUT**, **POST**, **DELETE**)

A picture containing graphical user interface

Description automatically generated

## NodeJS

### What is NodeJS

NodeJS is a runtime environment of JavaScript, which allows us use JavaScript in the Server side.it includes

1. JavaScript Virtual Machine
2. Built-in Libraries

It uses Google V8 driver for JavaScript Interpreting and compiling.

### Example

If you want load the built-in library, use “require()” command

Graphical user interface, text, application

Description automatically generated

## ExpressJS

### What is ExpressJS

ExpressJS is a open source library for developing a server using JavaScript, which can be downloaded by **npm** with the following script.

npm install express

### Send local JSON data to clients

Graphical user interface

Description automatically generated with low confidence

### How to handle Query and post request

The query starts by “?” mark, and follows with the criteria

Example: userId=1&id=2

Graphical user interface, text, application, email

Description automatically generated

### Homework

Text

Description automatically generated

const fs = require('fs')

var express = require('express');

var app = express();

app.use(express.json()); // JSON parser for post request

app.get('/json\_file', (req, res) => {

    try {

        let data = fs.readFileSync(`${\_\_dirname}/${req.query.name}.json`) // read data

        res.json(JSON.parse(data)); // transfer json file into JS Objects

    } catch (err) {

        console.error(err);

        res.send({'error': err.toString()});

    }

});

app.post('/json\_file', (req, res) => {

    try {

        const fileName = \_\_dirname + '/' + req.query.name + '.json';

        bodyData = req.body;

        fs.open(fileName, 'r', (err, fd) => {

            if (err) {

                fs.writeFile(fileName, JSON.stringify(bodyData), (err) => { if (err) console.log(err); }); // Create new file

            } else {

                let fileContent = JSON.parse(fs.readFileSync(fileName, 'utf8')); // Read file content

                Object.keys(bodyData).forEach( (key) => {fileContent[key] = bodyData[key];}); // add all data from post to file content

                fs.writeFileSync(fileName, JSON.stringify(fileContent)); // Write content to the file

            }

        })

        res.send({'success': 'File successfully updated.'})

    } catch (err) {

        console.log(err);

        res.send({'error': 'Update json file failed.'})

    }

})

app.delete('/json\_file', (req, res) => {

    try {

        fs.unlinkSync(\_\_dirname + "/" + req.query.name + '.json'); // delete the given json file

        res.send({'success': 'File deleted.'})

    } catch (err) {

        console.log(err);

        res.send({'error': 'Delete file failed.'})

    }

});

const port = 8080;

app.listen(port, () => {

    console.log(`Server listening at http://localhost:${port}`)

})