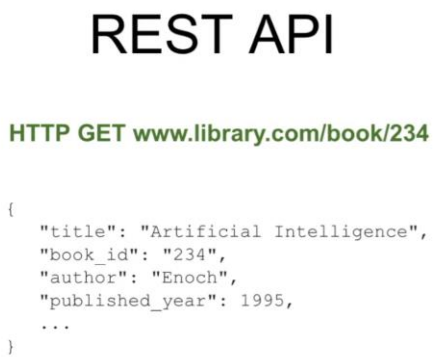
REST, NodeJS and Express Library

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.

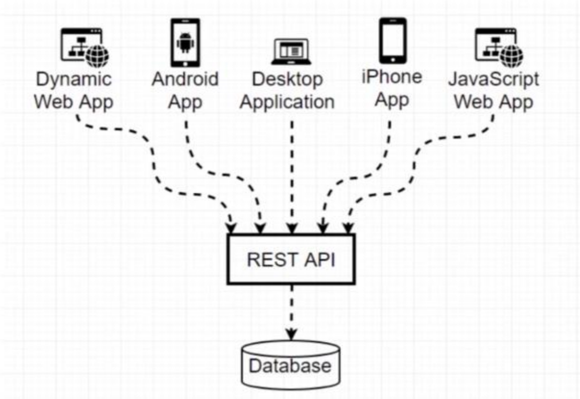
1. Resources (entities, includes text, image, music, services, etc.)
    a. Every **URL** in REST represents one type of resource
2. Representation (How to represent those entities in data format, text, HTML, JSON, XML)
    a. Usually use "`content-type`" in "header" to specify what it is
3. State Transfer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



REST API

HTTP GET www.library.com/book/234

```
{
    "title": "Artificial Intelligence",
    "book_id": "234",
    "author": "Enoch",
    "published_year": 1995,
    ...
}
```



NodeJS

What is NodeJS

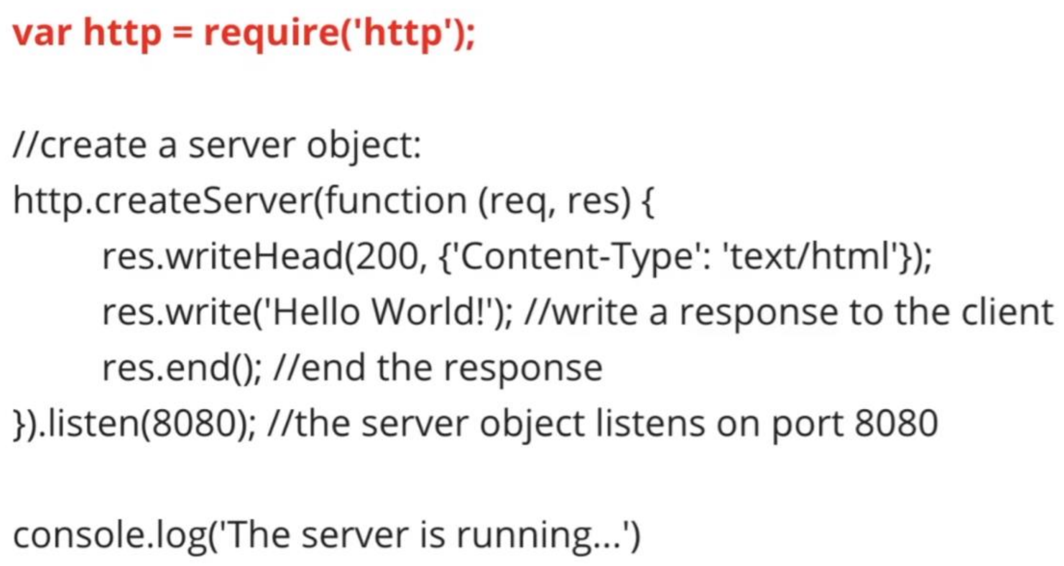
NodeJS is a runtime environment of JavaScript, which allows us use JavaScript in the Server side.it includes

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



```
var http = require('http');

//create a server object:
http.createServer(function (req, res) {
        res.writeHead(200, {'Content-Type': 'text/html'});
        res.write('Hello World!'); //write a response to the client
        res.end(); //end the response
}).listen(8080); //the server object listens on port 8080

console.log('The server is running...')
```

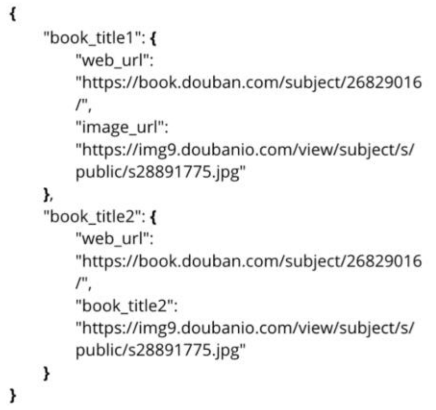
ExpressJS

What is ExpressJS

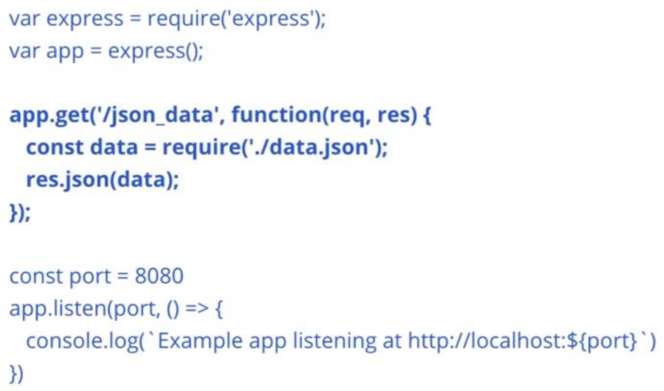
ExpressJS is a open source library for developing a server using JavaScript, which can be downloaded by **npm** with the following script.

```
npm install express
```

Send local JSON data to clients



```
{
    "book_title1": {
        "web_url":
        "https://book.douban.com/subject/26829016
        /",
        "image_url":
        "https://img9.doubanio.com/view/subject/s/
        public/s28891775.jpg"
    },
    "book_title2": {
        "web_url":
        "https://book.douban.com/subject/26829016
        /",
        "book_title2":
        "https://img9.doubanio.com/view/subject/s/
        public/s28891775.jpg"
    }
}
```

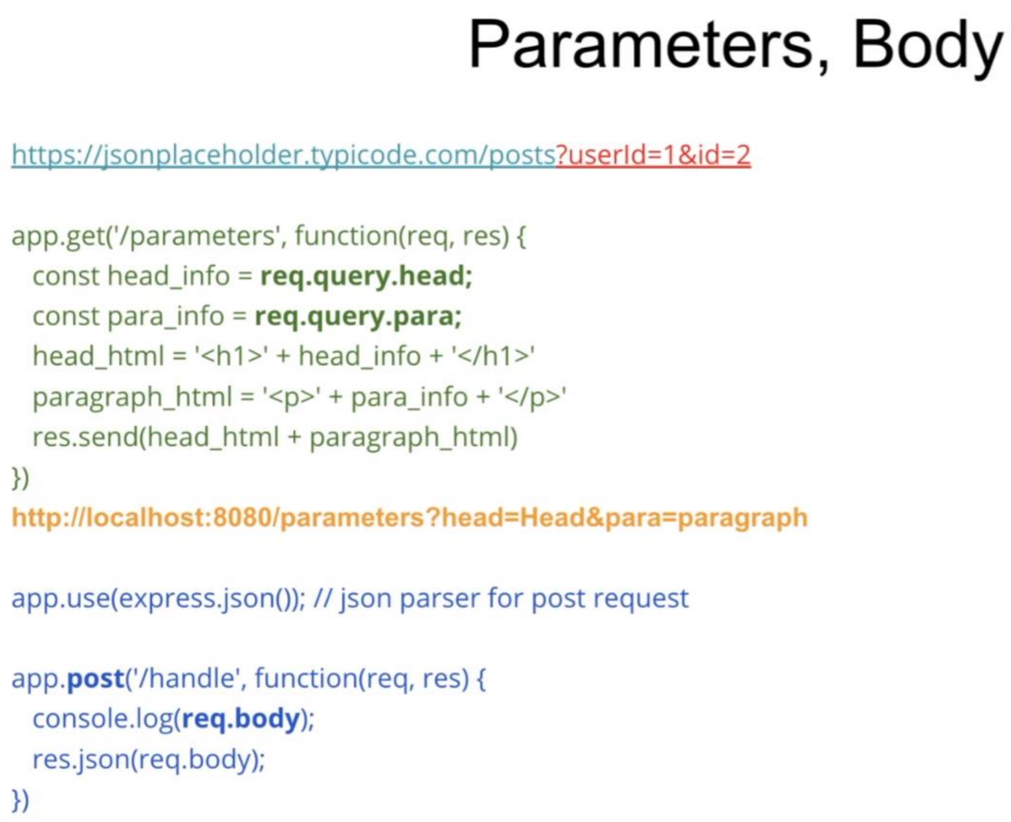


```
var express = require('express');
var app = express();

app.get('/json_data', function(req, res) {
  const data = require('./data.json');
  res.json(data);
});

const port = 8080
app.listen(port, () => {
  console.log(`Example app listening at http://localhost:${port}`)
})
```

How to handle Query and post request

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

Example: userId=1&id=2



Parameters, Body

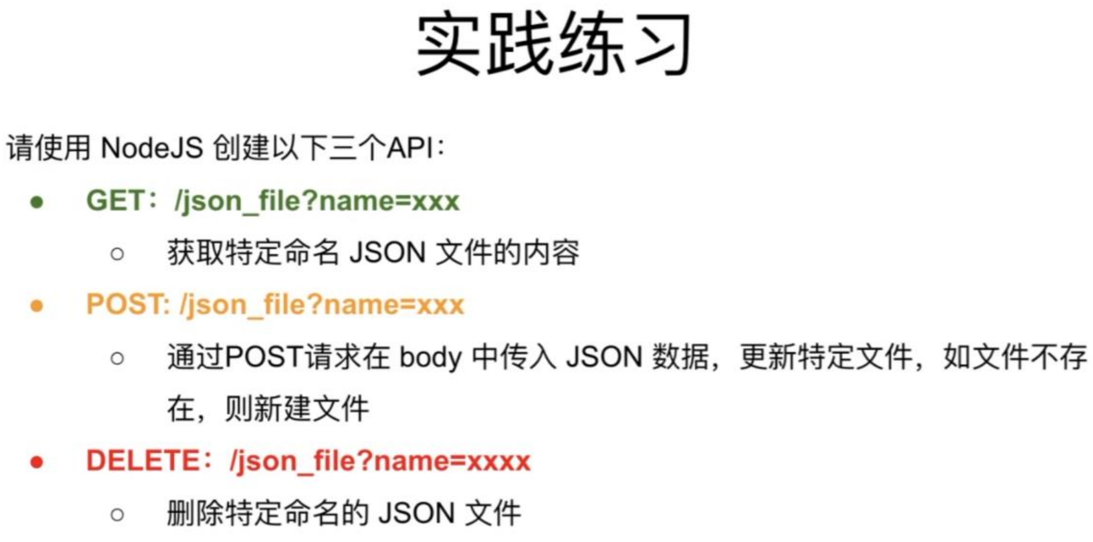
https://jsonplaceholder.typicode.com/posts?userId=1&id=2

```
app.get('/parameters', function(req, res) {
  const head_info = req.query.head;
  const para_info = req.query.para;
  head_html = '<h1>' + head_info + '</h1>'
  paragraph_html = '<p>' + para_info + '</p>'
  res.send(head_html + paragraph_html)
})
```

http://localhost:8080/parameters?head=Head&para=paragraph

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

app.post('/handle', function(req, res) {
  console.log(req.body);
  res.json(req.body);
})
```

Homework

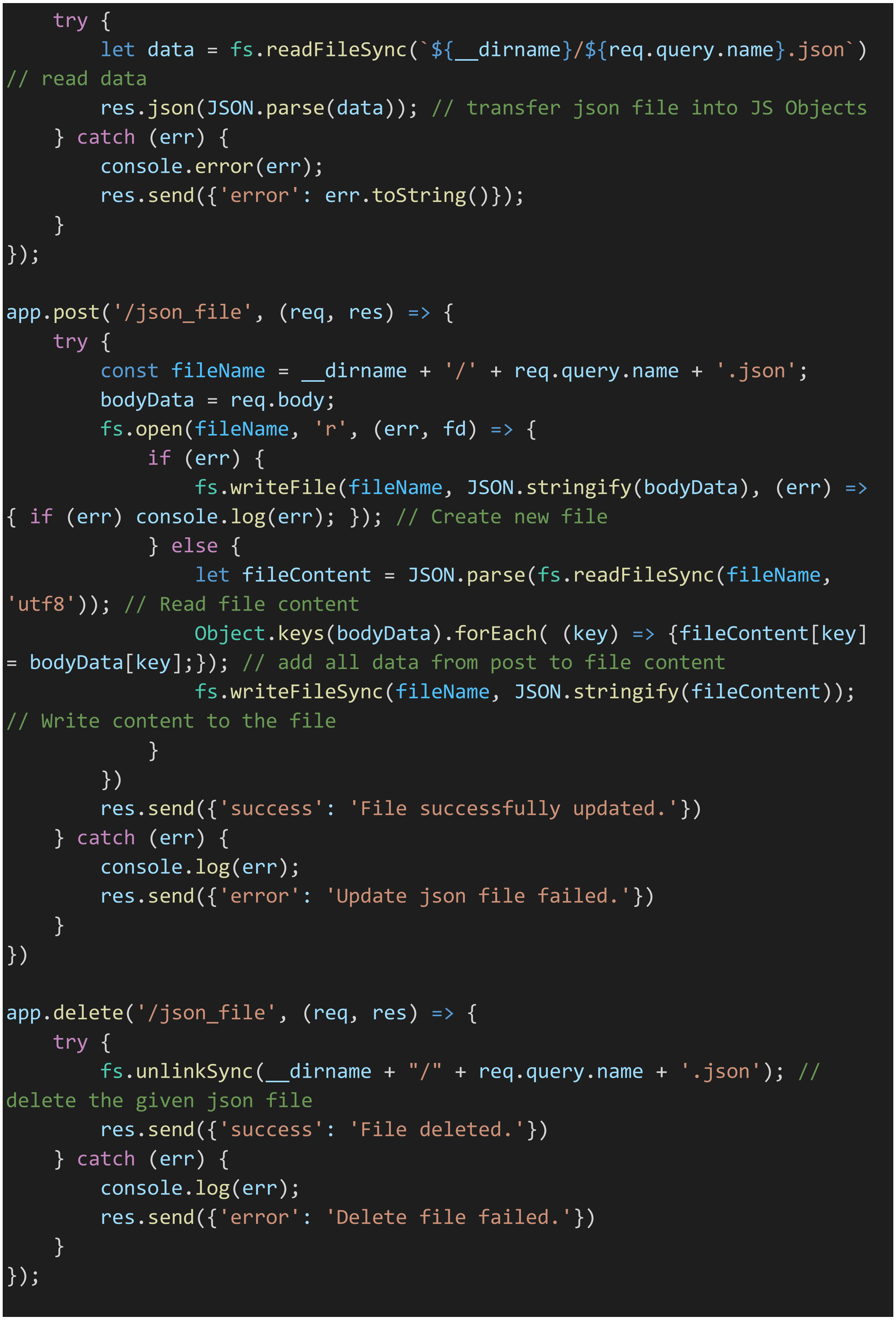


实践练习

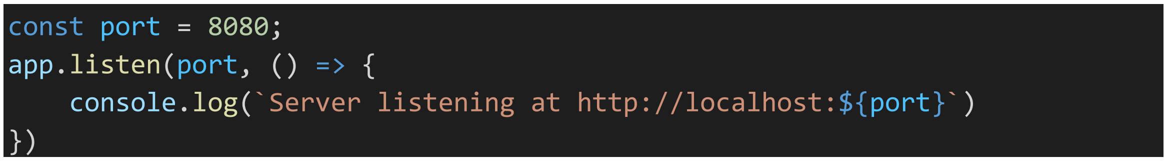
请使用 NodeJS 创建以下三个API：

- **GET：/json_file?name=xxx**
  - 获取特定命名 JSON 文件的内容
- **POST: /json_file?name=xxx**
  - 通过POST请求在 body 中传入 JSON 数据，更新特定文件，如文件不存在，则新建文件
- **DELETE：/json_file?name=xxxx**
  - 删除特定命名的 JSON 文件

```
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}`)
})
```