1. 为什么要使用模板引擎？

可以将数据与界面分开，提高代码的可维护性。

1. 页面渲染分为：前端渲染、后端渲染

前端渲染可以使用ajax技术

后端渲染可以使用ejs模板引擎

1. Ejs模板引擎使用：
2. 安装ejs: npm i ejs
3. 设置模板引擎： app.set(‘配置项’, ’值’)

App.set(‘view engine’,’ejs’);

App.set(‘views’,[‘模板文件存放目录1’,‘模板文件存放目录2’]);

App.engine(‘html’,require(‘ejs’).\_\_express);

1. 渲染页面：res.render(‘模板文件’[,{模板变量}]);
2. Ejs模板引擎的用法：

const express = require('express');

const path = require('path');

const app = express();

app.listen(4000, () => {

    console.log('server start at port 4000');

});

//设置模板引擎

app.set('view engine', 'ejs'); //设置模板引擎为ejs

app.set('views', [path.resolve(\_\_dirname, 'views'), path.resolve(\_\_dirname, 'moban')]); //设置模板文件存放位置

//告诉express html文件作为ejs模板文件来解析

app.engine('html', require('ejs').\_\_express);

//设计路由

app.get('/yesok', (req, res) => {

    res.send('Hello...');

});

app.get('/myejs', (req, res) => {

    let usr = '游客';

    let uArr = [{ "uname": "lisi", "age": 19, "email": "lisi@qq.com" }, { "uname": "zhangsan", "age": 20, "email": "zhangsan@163.com" }, { "uname": "demo", "age": 21, "email": "demo@sohu.com" }];

    let score = [80, 90, 70, 78, 81, 67];

    res.render('list', { usr, uArr, score });

});

app.get('/myhtml', (req, res) => {

    let score = [80, 90, 70, 78, 81, 67];

    res.render('detail.html', { score });

});

views/list.ejs模板文件内容如下：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <!-- html注释 -->

    <%# 引入公共模板文件 %>

        <%- include('header') %>

            <h2>ejs模板引擎</h2>

            <div>欢迎您

                <%=usr  %>回来</div>

            <div>

                <ul>

                    <% for(let i = 0;i<uArr.length;i++) { %>

                        <li>

                            <%=uArr[i].uname %> |

                                <%=uArr[i].age %> |

                                    <%=uArr[i].email %>

                        </li>

                        <% } %>

                            <!--

                    <li>李四 | 19 | lisi@qq.com</li>

                    -->

                </ul>

            </div>

            <div>

                <% for(let i = 0;i<score.length;i++) { %>

                    <p>

                        <%=score[i] %>

                    </p>

                    <% if(score[i] >= 80){ %>

                        <p>优秀</p>

                        <% }else if(score[i] >= 70){ %>

                            <p>良好</p>

                            <% }else{ %>

                                <p>一般</p>

                                <% } %>

                                    <% } %>

            </div>

</body>

</html>

moban/detail.html模板文件内容如下：

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <p>html文件作为ejs模板文件</p>

    <div>

        <% for(let i = 0;i<score.length;i++) { %>

            <p>

                <%=score[i] %>

            </p>

            <% if(score[i] >= 80){ %>

                <p>优秀</p>

                <% }else if(score[i] >= 70){ %>

                    <p>良好</p>

                    <% }else{ %>

                        <p>一般</p>

                        <% } %>

                            <% } %>

    </div>

</body>

</html>

2. 关于package.json文件中的”scripts”配置项的使用：

如果在package.json文件中的”scripts”配置项中增加了下面的配置:

    "scripts": {

        "testejs": "node  day09/1-ejs模板引擎.js",

        "start": "node  day09/1-ejs模板引擎2.js",

    }

那么我们可以使用npm run testejs来运行对应的.js文件，当配置项名称为”start”时则可以使用省略run，即使用npm start 来运行对应的.js文件。

1. Express的生成器:

1、express的生成器用途： 可以快速创建项目骨架。

2、express生成器用法：

1)、安装express生成器： npm i -g express-generator

-g参数要注意两点：

1. 使用npm -g 安装包时可以在任何目录中执行该命令，不一定要在当前项目根目录下；
2. 使用npm -g安装的包一台电脑只需要安装一次即可；
3. 使用npm -g安装的包都放在这个固定目录中：

C:\Users\ibm\AppData\Roaming\npm\node\_modules

2)、使用express生成器创建项目骨架：

express --view=模板引擎 项目名称

注意：创建项目骨架时要注意当前所在目录。

比如：

d:> express --view=ejs myoa

3)切换到myoa目录中并安装项目所有依赖包：

d:>cd myoa

d:>npm install

4)启动项目：

d:>npm start

1. express中router级别:
2. router级别路由语法：router.请求方式[get/post](‘路径’,回调方法)
3. 怎样定义、使用router级别路由?
4. 、在创建的router路由文件中添加下面的代码 :

const express = require(‘express’);

const router = express.Router();

//设计路由：

router.请求方式[get/post](‘路径’,回调方法)

module.exports = router;

1. 通常在入口文件(app.js)中引入对应的router路由文件并使用：

let routers = Require(‘./router路由文件’);

app.use(routers);

1. 配置热更新(nodemon)？
2. 安装nodemon: npm i -g nodemon
3. 将nodemon配置文件(nodemon.json)放到当前项目根目录中；
4. 使用nodemon启动项目： nodemon ./bin/www
5. 可以在package.json文件中”scripts”配置项中添加下面的配置：

    "scripts": {

        "start": "node ./bin/www",

        "dev": "nodemon  ./bin/www"

    },

最后可以使用命令：npm run dev //以热更新方式启动启动