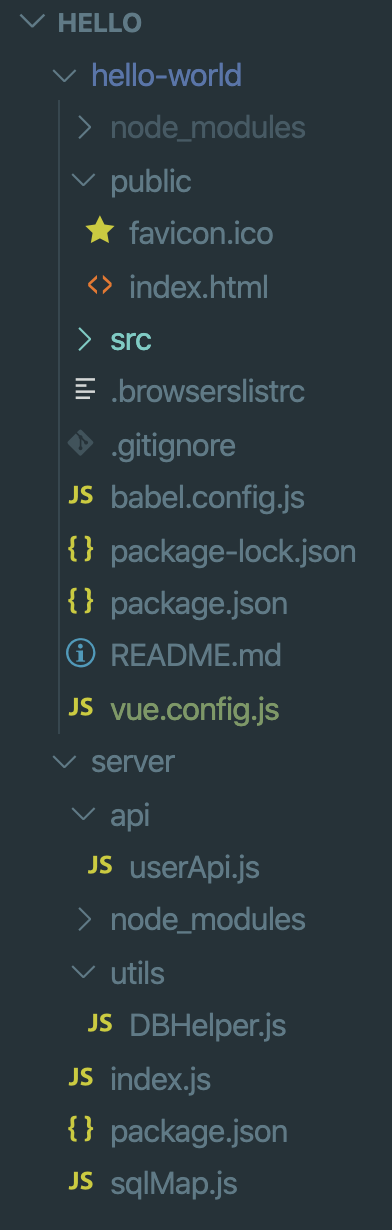
Express + MYSQL + Vuecli3

参考地址 <https://blog.csdn.net/qq_41555552/article/details/100111656>



cmd执行（以后不用再执行了 -g代表在全局安装）

cnpm install -g @vue/cli

cnpm install -g nodemon

新建一个 hello 文件夹

在hello目录下执行创建前端项目

vue create hello-world

新建表

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (

`id` int primary key auto\_increment,

`name` varchar(255) DEFAULT NULL,

`age` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

#### userApi.js

const express = require('express');

const router = express.Router();

const DBHelper = require('../utils/DBHelper');

const sql = require('../sqlMap');

// 增加用户

router.post('/addUser', (req, res) => {

let sqlStr = sql.user.add;

let params = req.body;

let conn = new DBHelper().getConn();

conn.query(sqlStr, [params.name, params.age], (err, result) => {

if (err) {

res.json(err);

} else {

res.json(result);

}

});

conn.end();

});

// 查询用户

router.post('/selectUser', (req, res) => {

let sqlStr = sql.user.select;

let params = req.body;

let conn = new DBHelper().getConn();

conn.query(sqlStr, [params.name], (err, result) => {

if (err) {

res.json(err);

} else {

res.json(result)

}

});

conn.end();

});

module.exports = router;

#### DBHelper.js

// 数据库连接助手

const mysql = require('mysql');

class DBHelper{

// 获取数据库连接

getConn(){

let conn = mysql.createConnection({

// 数据库连接配置

host:'127.0.0.1',

port:'3306',

user:'root',

password:'123456',

database:'hello' // 数据库名

});

conn.connect();

return conn;

}

}

module.exports = DBHelper;

#### index.js

// node后端服务器

const http = require('http');

const badyParser = require('body-parser');

const express = require('express');

const userApi = require('./api/userApi');

let app = express();

let server = http.createServer(app);

app.use(badyParser.json());

app.use(badyParser.urlencoded({

extended: false

}));

// 后端api路由

app.use('/api/user', userApi);

// 启动监听

server.listen(8088, () => {

console.log(‘success!! port:8088’)

})

#### sqlMap.js

// sql语句

var sqlMap = {

user: {

// 添加用户

add: 'insert into user(name, age) values ( ?, ?)',

// 查询用户

select: 'select \* from user where name like "%"?"%"?"%"'

}

}

module.exports = sqlMap;

在server目录下依次执行

npm init –y // package.json

cnpm install express -S

cnpm install mysql -S

nodemon index.js

#### HelloWorld.vue

<template>

<div class="hello">

<form action="">

姓名：<input type="text" name="username" v-model="userName"><br>

年龄：<input type="text" name="age" v-model="age"><br>

<button type="button" @click="addUser">提交信息</button>

</form>

<form action="">

<input type="text" v-model="keywords" placeholder="输入姓名查询">

<button type="button" @click="selectUser">查询</button>

</form>

</div>

</template>

<script>

import axios from 'axios'

export default {

name: 'upload',

props: {

msg: String

},

data(){

return{

userName:'',

age:'',

keywords:''

}

},

methods:{

addUser(){ //

let name = this.userName;

let age = this.age;

axios.post('/api/user/addUser',{

name,age

}).then(res=>{

console.log(res);

alert('信息添加成功');

}).catch(err=>{

console.log(err)

})

},

selectUser(){

let name = this.keywords;

axios.post('api/user/selectUser',{

name

}).then(res=>{

let data = res.data[0];

this.userName = data.name;

this.age = data.age

}).catch(err=>{

console.log(err)

})

}

}

}

</script>

## 设置代理处理跨域

在hello-world目录新建文件 vue.config.js

module.exports = {

devServer: {

port: ‘9099’,

proxy: { // 设置代理

'/api': {

target: 'http://127.0.0.1:8088', // 请求的目标服务器接口

changeOrigin: true, // 在本地会创建一个虚拟服务端，然后发送请求的数据，并同时接收请求的数据，这样服务端和服务端进行数据的交互就不会有跨域问题

pathRewrite: { // 重写请求

'^/api': '/api' // 替换target中的请求地址，也就是说在请求http://127.0.0.1:8888/XXXXX这个地址的时候直接写成/api即可。

}

}

}

}

}

//hello-world 目录下

执行

cnpm i axios -S

npm run serve