ajax项目

一、项目流程

1.管理员模块

（1）登陆

（2）查询所有用户信息

（3）删除用户信息

（4）查询单个用户信息（扩展）

2.用户模块

（1）用户注册

（2）用户修改个人资料

3.项目流程图

图示

描述已自动生成

二、数据库

1. 开启xampp

2. 创建数据表及插入数据 见素材《mydb.sql》

|  |
| --- |
| -- mydb sql  -- 创建一个-- mydb数据库  /\* SET NAMES UTF8; \*/  /\* DROP DATABASE IF EXISTS mydb; \*/  /\* CREATE DATABASE mydb CHARSET = UTF8; \*/  -- 进入 mydb 数据库  /\* use mydb; \*/  -- 创建一张用户表userinfo  -- u\_member会员  -- 电话号码不能重复  create table userinfo(  u\_id int auto\_increment primary key,  u\_names varchar(255) not null,  u\_phone varchar(255) not null unique,  u\_member INT not null  ) ENGINE = InnoDB charset = utf8;  INSERT INTO  userinfo(u\_names, u\_phone, u\_member)  VALUES  ('tom','15201076723',1),  ('nancy','15201197440',0),  ('jack','15101075644',1),  ('bob','15910257181',0),  ('lily','13522162834',1),  ('Kate','13661217464',0),  ('emma','13522162843',0),  ('bale','13671357146',0),  ('coy','13522172419',0),  ('ball','15910779501',1),  ('dan','15910780895',1)  -- 创建管理员表和数据  create table admin(  a\_id int auto\_increment primary key,  a\_names varchar(255) not null unique,  a\_pwd varchar(255) not null  ) ENGINE = InnoDB charset = utf8;  INSERT INTO  admin(a\_names, a\_pwd)  VALUES  ('zhangsan', '123'),  ('lisi', '123'),  ('wangwu', '123') |

三、搭建后台

1.创建项目目录

（1）创建目录ajaxpro

（2）在ajaxpro中创建views文件用于放置html文件

（3）安装依赖 npm install express

（4）安装mysql包 npm insatll mysql

2.创建链接池对象

连接池的作用就是用来管理连接，提升连接的利用效率

|  |
| --- |
| //引入mysql模块  const mysql = require('mysql');  //创建连接池对象  const pool = mysql.createPool({  host: '127.0.0.1',  port: '3306',  user: 'root',  password: '0000',  database: 'mydb',  //连接池可创建的最大连接数  connectionLimit: 15  });  //导出对象  module.exports = pool; |

3. 创建routes子路由文件夹

子路由文件夹是为了分模块开发，分别把用户模块和管理员模块分开，也就是逻辑分流。

（1）admin.js 管理员模块

（2）user.js 用户模块

4. 创建入口文件app.js

（1）引入第三方模块express

（2）创建WEB服务器

|  |
| --- |
| //创建WEB服务器  var app = express();  //设置端口  app.listen(8080); |

（3）引入子路由，挂载子路由

|  |
| --- |
| //引入用户路由器  var userRouter = require('./routes/user.js');  //引入管理员路由  var adminRouter = require('./routes/admin.js');  //路由配置后接口地址要以这两个资源目录开始再加后面的资源目录  //如/user/abc  //挂载用户路由器，/user  app.use('/user', userRouter);  //挂载用户路由器，/admin  app.use('/admin', adminRouter); |

（4）静态资源托管views放置html文件

|  |
| --- |
| app.use(express.static('./views')); |

（5）解析表单中的 url-encoded 格式的数据

|  |
| --- |
| app.use(express.urlencoded({extended: false})); |

（6）错误处理中间件

|  |
| --- |
| //错误处理中间件  app.use(function (err, req, res, next) {  //查看得到的错误信息  console.log(err);  //响应500,返回值500和信息  res.status(500).send({  code: 500,  msg: '服务器端错误'  });  }); |

四、客户端请求

1.首页分流

文件名：在views文件目录里创建index.html文件

作用：用户端和管理端分开

访问地址：<http://127.0.0.1:8080/index.html>

2.管理员登陆【admin 管理员模块】

文件名：在views文件目录里创建login.html文件

作用：管理员登陆（查询）

请求方式：POST

请求接口：http://127.0.0.1:8080/admin/login

访问地址：<http://127.0.0.1:8080/login.html>

3.用户管理界面【admin 管理员模块】

文件名：在views文件目录里创建list.html文件

作用：所有用户展示（查询、删除）

请求方式：GET,DELETE

请求接口：<http://127.0.0.1:8080/admin/login> 带参数查询

请求接口：<http://127.0.0.1:8080/admin/del> 带参数删除

访问地址：<http://127.0.0.1:8080/list.html>

4.单独查询用户【admin 管理员模块】

文件名：在views文件目录里创建query.html文件

作用：一个用户展示（查询带参数）

请求方式：GET

请求接口：<http://127.0.0.1:8080/admin/query> 带参数查询

访问地址：<http://127.0.0.1:8080/query.html>

5. 用户注册【user 用户模块】

文件名：在views文件目录里创建register.html文件

作用：用户自己注册（查询带参数）

请求方式：POST

请求接口：<http://127.0.0.1:8080/user/reg> 带参数查询

访问地址：<http://127.0.0.1:8080/register.html>

6. 用户修改信息【user 用户模块】

文件名：在views文件目录里创建edit.html文件

作用：用户自己注册（查询带参数）

请求方式：PUT

请求接口：<http://127.0.0.1:8080/user/edit> 带参数修改

访问地址：<http://127.0.0.1:8080/edit.html>