瑞吉外卖

Day------1

搭建数据库

图形界面或则命令行

```
//创建数据库
create database reggie CHARACTER SET utf8mb4
```

```
导入sql文件命令
source d:/sql_file/db_reggie.sql;（路径不要右中文）
```

搭建maven

maven

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>2.4.5</version>
    </parent>
    <groupId>org.hbx.project</groupId>
    <artifactId>reggie</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <jaba.verson>1.8</jaba.verson>
    </properties>

    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
```

```xml
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
            <scope>compile</scope>
        </dependency>

        <dependency>
            <groupId>com.baomidou</groupId>
            <artifactId>mybatis-plus-boot-starter</artifactId>
            <version>3.4.2</version>
        </dependency>

        <dependency>
            <groupId>org.projectlombok</groupId>
            <artifactId>lombok</artifactId>
            <version>1.18.20</version>
        </dependency>

        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>fastjson</artifactId>
            <version>1.2.76</version>
        </dependency>

        <dependency>
            <groupId>commons-lang</groupId>
            <artifactId>commons-lang</artifactId>
            <version>2.6</version>
        </dependency>

        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <scope>runtime</scope>
        </dependency>

        <dependency>
            <groupId>com.alibaba</groupId>
            <artifactId>druid-spring-boot-starter</artifactId>
            <version>1.1.23</version>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
                <version>2.4.5</version>
            </plugin>
        </plugins>
    </build>


</project>
```

yml配置文件

```yaml
server:
  port: 8888
spring:
  application:
    name: reggie_take_out   #设置这个项目的名字（默认是项目文件的名字）
  datasource:
    url: jdbc:mysql://localhost:13306/reggie?
useSSL=false&useUnicode=true&characterEncoding=utf8
    username: root
    password: abc123
    driver-class-name: com.mysql.jdbc.Driver
mybatis-plus:
  configuration:
    #在映射实体或者属性时，将数据库中表名和字段名中的下划线去掉，按照驼峰命名法映射
    map-underscore-to-camel-case: true
    log-impl: org.apache.ibatis.logging.stdout.StdOutImpl
  global-config:
    db-config:
      id-type: ASSIGN_ID
```

主程序---springbootapplication

导入静态资源-----backed文件和front文件----放在static目录下

或者自己添加静态资源目录

```java
@Configuration
public class staticConfig implements WebMvcConfigurer {
    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {

 registry.addResourceHandler("/front/**").addResourceLocations("classpath:/front
/");

    }
}
```
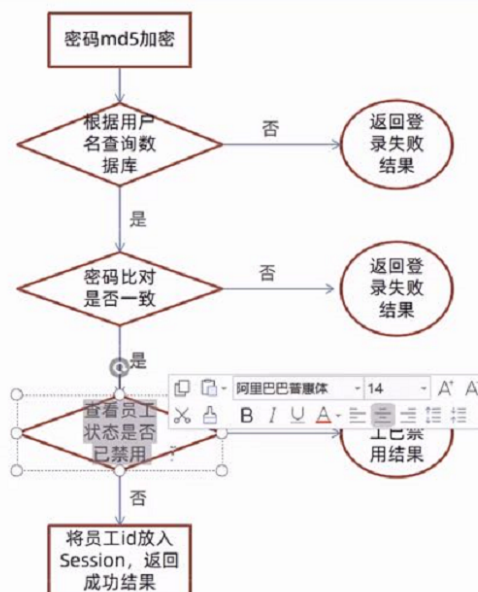
后台登陆功能

思路速记

controller - service-mapper-查询---返回dao（employee）

前端vue代码的method需要的变量值

规定服务器返回的封装为类型R（code、data、msg、map）---配合前端

登陆handler---

接受json

保存的session

处理逻辑

（数据库里面是）md5加密（DigetsUtils.M5D）----》查用户名-----》查密码

知识点

mybatisplus

mybatisplus----只需要接口继承父类接口就行

mapper接口继承BaseMapper

service接口继承IService

service实现类继承ServiceImpl<mapper,dao>并且实现service接口

函数介绍

lambdaQuery--eq里面写查询条件

```
//查询数据库是否右这个用户名
        LambdaQueryWrapper<Employee> queryWrapper = new LambdaQueryWrapper<>();
        queryWrapper.eq(Employee::getUsername,"admin");
        Employee one = empService.getOne(queryWrapper);
```

==注意@Service这些注解是标在实现类上而不是接口上的==

html知识

这个方法绑定了登陆按钮

```
methods: {
        async handleLogin() {
          this.$refs.loginForm.validate(async (valid) => {
            if (valid) {
              this.loading = true
              let res = await loginApi(this.loginForm)
              if (String(res.code) === '1') {
                localStorage.setItem('userInfo',JSON.stringify(res.data))
                window.location.href= '../../index.html'
              } else {
                this.$message.error(res.msg)
                this.loading = false
              }
            }
          })
        }
    }


对应的js
function loginApi(data) {
  return $axios({
    'url': '/employee/login',     control的url
    'method': 'post',
    data
  })
}

发给服务器的是
data() {
        return {
          loginForm:{
            username: 'admin',
            password: '123456'
          }
        }

是一个json
```

debug的时候改一下前端的超时时间

Md5工具类--DigestUtils

```
String md5 = DigestUtils.md5DigestAsHex(password.getBytes());
```

后台退出功能

思路速记

右上角展示当前用户

退出按钮的前端代码分析

handler

  清除缓存

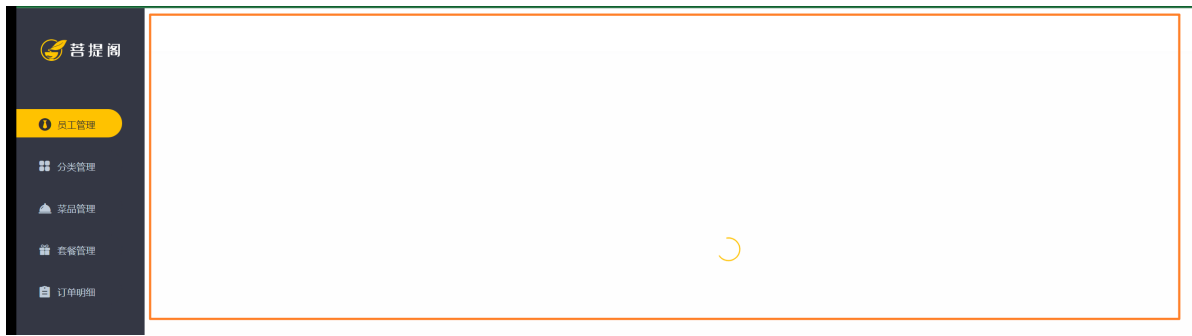## HTML知识点

```
        <iframe
          id="cIframe"
          class="c_iframe"
          name="cIframe"
          :src="iframeUrl"
          width="100%"
          height="auto"
          frameborder="0"
          v-show="!loading"
        ></iframe>
相当于留了一片空间用来动态展示src:src="iframeUrl"
```

利用v-for，v-if, v-else

```
                <el-menu-item v-else :index="item.id"
@click="menuHandle(item,false)">
                <i class="iconfont" :class="item.icon"></i>
                <span slot="title">{{item.name}}</span>
            </el-menu-item>
```

```
menuList: [
          // {
          //   id: '1',
          //   name: '门店管理',
          //   children: [
              {
                id: '2',
                name: '员工管理',
                url: 'page/member/list.html',
                icon: 'icon-member'
              },
```


绑定的方法

```
methods: {
        logout() {
          logoutApi().then((res)=>{
            if(res.code === 1){
              localStorage.removeItem('userInfo')
              window.location.href = 'page/login/login.html'
            }
          })
        },


      对应的js
      function logoutApi(){
  return $axios({
```

```
    'url': '/employee/logout', ----handlerurl
    'method': 'post',
  })
}
```