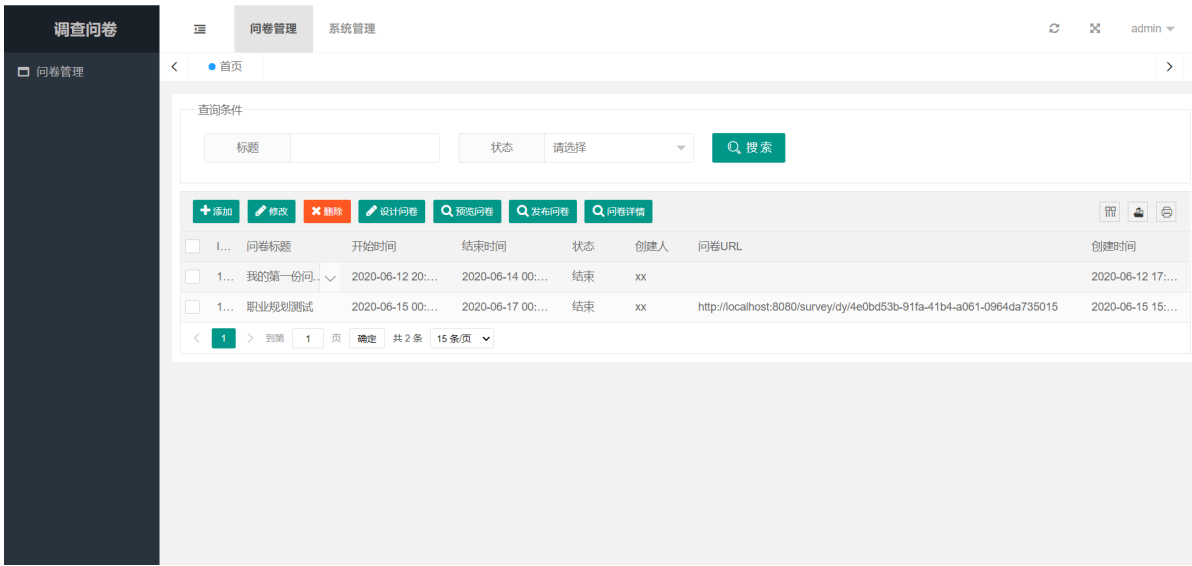
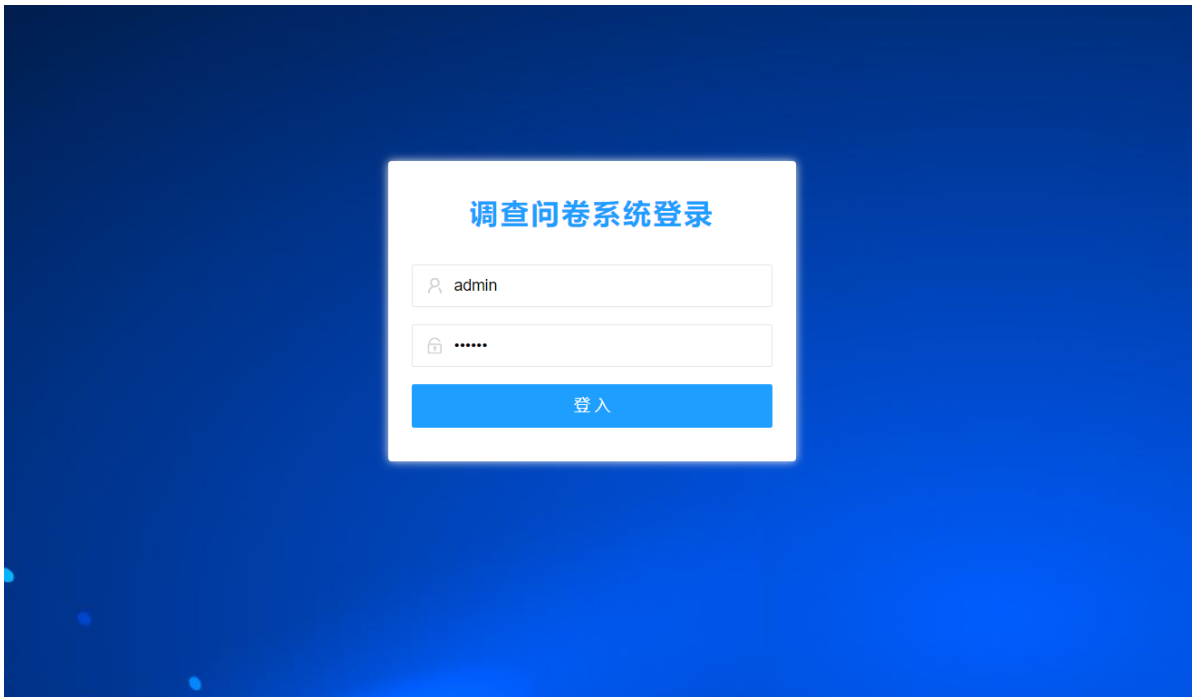
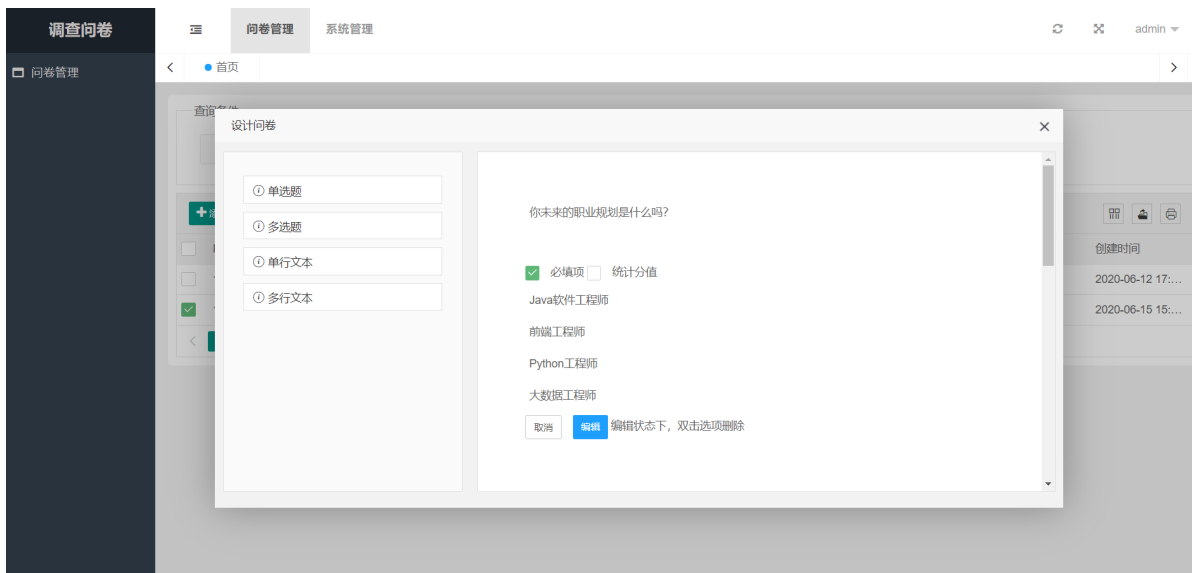


调查问卷系统

项目预览





优势:

- 项目搭建配置比较详细，很适合ssm入门同学
- Jquery使用较多，提高Jquery技术水平，提供前端技能。
- 在面试过程中，你可以将该项目作为一个自己开发的模块去讲给面试官。

1.项目技术

- 数据库: MySQL8.0
- 数据库设计软件: Power Designer16.5
- IDE: IDEA
- Web容器: Apache Tomcat 8.5
- 项目管理工具: Maven
- 后端技术: Spring+Spring MVC+MyBatis (SSM框架)
- 前端技术: **LayUI** (Vue/React)

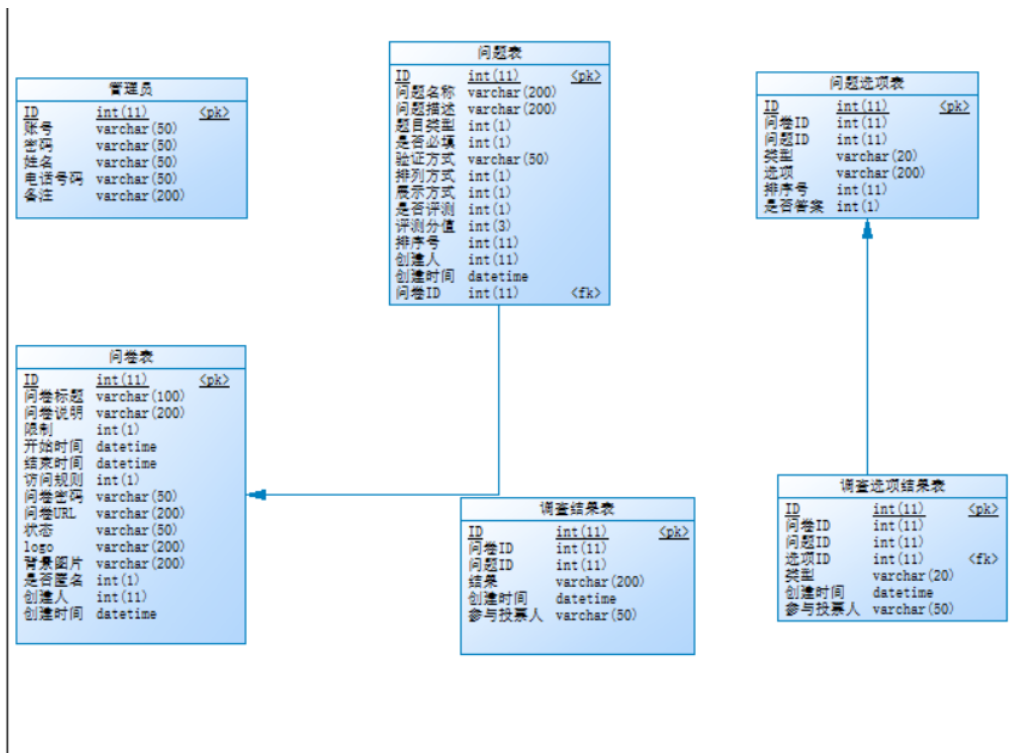
2.项目需求

- 问卷星
- 调查派
- 腾讯问卷

2.1系统角色

- 管理员: zhangsan、lisi (Admin表)
 - 管理员信息CURD
 - 制作调查问卷
 - 发布调查问卷
 - 统计调查结果
 - 图表展示
- 访客: (无需登录)
 - 参与调查

3.数据库设计



```

/*
Navicat Premium Data Transfer

Source Server          : localhost
Source Server Type     : MySQL
Source Server Version  : 80011
Source Host            : localhost:3306
Source Schema          : survey

Target Server Type     : MySQL
Target Server Version  : 80011
File Encoding          : 65001

Date: 22/06/2020 10:40:38
*/

SET NAMES utf8mb4;
SET FOREIGN_KEY_CHECKS = 0;

-- Table structure for tb_admin
-- -----
DROP TABLE IF EXISTS `tb_admin`;
CREATE TABLE `tb_admin` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `account` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,
  `password` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,
  `name` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,
  `phone` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,
  `remark` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,
  PRIMARY KEY (`id`) USING BTREE
)

```

```

) ENGINE = InnoDB AUTO_INCREMENT = 2081 CHARACTER SET = utf8 COLLATE =
utf8_general_ci ROW_FORMAT = Dynamic;

-- -----
-- Records of tb_admin
-- -----
INSERT INTO `tb_admin` VALUES (2073, 'admin',
'433db2fa87c8e880868dcf6ae9c2fca2', 'xx', '1853333333', '');
INSERT INTO `tb_admin` VALUES (2077, 'xx', '433db2fa87c8e880868dcf6ae9c2fca2',
'xxx', '1855555555', 'xxx');
INSERT INTO `tb_admin` VALUES (2079, 'asdfasdf',
'433db2fa87c8e880868dcf6ae9c2fca2', 'asdfasdfasdf', '1853333333', '');
INSERT INTO `tb_admin` VALUES (2080, '111111111111111111111111111111',
'433db2fa87c8e880868dcf6ae9c2fca2', '111111111111111111111111111111', '1853333333', '');

-- -----
-- Table structure for tb_answer_opt
-- -----
DROP TABLE IF EXISTS `tb_answer_opt`;
CREATE TABLE `tb_answer_opt` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `survey_id` int(11) NULL DEFAULT NULL,
  `question_id` int(11) NULL DEFAULT NULL,
  `opt_id` int(11) NULL DEFAULT NULL,
  `type` varchar(20) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL COMMENT '1radio|2checkbox',
  `create_time` datetime(0) NULL DEFAULT NULL,
  `voter` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL,
  PRIMARY KEY (`id`) USING BTREE,
  INDEX `FK_Reference_2` (`opt_id`) USING BTREE,
  CONSTRAINT `FK_Reference_2` FOREIGN KEY (`opt_id`) REFERENCES `tb_question_opt`
(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB AUTO_INCREMENT = 33 CHARACTER SET = utf8 COLLATE =
utf8_general_ci ROW_FORMAT = Dynamic;

-- -----
-- Records of tb_answer_opt
-- -----
INSERT INTO `tb_answer_opt` VALUES (23, 19, 33, 96, '1', '2020-06-16 09:39:38',
'ef539923-716d-4735-a286-81167a1f4e63');
INSERT INTO `tb_answer_opt` VALUES (24, 19, 34, 100, '2', '2020-06-16 09:39:38',
'ef539923-716d-4735-a286-81167a1f4e63');
INSERT INTO `tb_answer_opt` VALUES (25, 19, 33, 97, '1', '2020-06-16 09:39:45',
'de0be419-4847-4e00-8590-3b7ddfec1671');
INSERT INTO `tb_answer_opt` VALUES (26, 19, 34, 100, '2', '2020-06-16 09:39:45',
'de0be419-4847-4e00-8590-3b7ddfec1671');
INSERT INTO `tb_answer_opt` VALUES (27, 19, 34, 102, '2', '2020-06-16 09:39:45',
'de0be419-4847-4e00-8590-3b7ddfec1671');
INSERT INTO `tb_answer_opt` VALUES (28, 19, 33, 98, '1', '2020-06-16 09:53:17',
'6ac48f74-6794-406e-b354-ad8bcebc2ff7');
INSERT INTO `tb_answer_opt` VALUES (29, 19, 34, 103, '2', '2020-06-16 09:53:17',
'6ac48f74-6794-406e-b354-ad8bcebc2ff7');
INSERT INTO `tb_answer_opt` VALUES (30, 19, 33, 97, '1', '2020-06-16 09:54:59',
'd5ebbe66-39fd-476c-ba97-c28ef12a558b');
INSERT INTO `tb_answer_opt` VALUES (31, 19, 34, 101, '2', '2020-06-16 09:54:59',
'd5ebbe66-39fd-476c-ba97-c28ef12a558b');

```

```
INSERT INTO `tb_answer_opt` VALUES (32, 19, 34, 103, '2', '2020-06-16 09:54:59', 'd5ebbe66-39fd-476c-ba97-c28ef12a558b');
```

```
-- -----  
-- Table structure for tb_answer_txt  
-- -----
```

```
DROP TABLE IF EXISTS `tb_answer_txt`;  
CREATE TABLE `tb_answer_txt` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `survey_id` int(11) NULL DEFAULT NULL,  
  `question_id` int(11) NULL DEFAULT NULL,  
  `result` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,  
  `create_time` datetime(0) NULL DEFAULT NULL,  
  `voter` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,  
  PRIMARY KEY (`id`) USING BTREE  
) ENGINE = InnoDB AUTO_INCREMENT = 14 CHARACTER SET = utf8 COLLATE = utf8_general_ci ROW_FORMAT = Dynamic;
```

```
-- -----  
-- Records of tb_answer_txt  
-- -----
```

```
INSERT INTO `tb_answer_txt` VALUES (9, 19, 38, '111', '2020-06-16 09:39:38', NULL);  
INSERT INTO `tb_answer_txt` VALUES (10, 19, 38, '11222', '2020-06-16 09:39:45', NULL);  
INSERT INTO `tb_answer_txt` VALUES (11, 19, 38, '阿斯顿发斯蒂芬', '2020-06-16 09:41:59', NULL);  
INSERT INTO `tb_answer_txt` VALUES (12, 19, 38, '啊啊啊', '2020-06-16 09:53:17', '6ac48f74-6794-406e-b354-ad8bcebc2ff7');  
INSERT INTO `tb_answer_txt` VALUES (13, 19, 38, '啊啊啊', '2020-06-16 09:54:59', 'd5ebbe66-39fd-476c-ba97-c28ef12a558b');
```

```
-- -----  
-- Table structure for tb_question  
-- -----
```

```
DROP TABLE IF EXISTS `tb_question`;  
CREATE TABLE `tb_question` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `title` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,  
  `remark` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL,  
  `type` int(1) NULL DEFAULT NULL COMMENT '1radio|2checkbox|3text|4textarea',  
  `required` int(1) NULL DEFAULT NULL COMMENT '0非必填1必填',  
  `check_style` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT NULL COMMENT 'text;number;date',  
  `order_style` int(1) NULL DEFAULT NULL COMMENT '0顺序1随机',  
  `show_style` int(1) NULL DEFAULT NULL COMMENT '1;2;3;4',  
  `test` int(1) NULL DEFAULT NULL COMMENT '0不测评1测评',  
  `score` int(3) NULL DEFAULT NULL,  
  `orderby` int(11) NULL DEFAULT NULL,  
  `creator` int(11) NULL DEFAULT NULL,  
  `create_time` datetime(0) NULL DEFAULT NULL,  
  `survey_id` int(11) NULL DEFAULT NULL,  
  PRIMARY KEY (`id`) USING BTREE,  
  INDEX `FK_Reference_1`(`survey_id`) USING BTREE,
```

```

    CONSTRAINT `FK_Reference_1` FOREIGN KEY (`survey_id`) REFERENCES `tb_survey`
(`id`) ON DELETE RESTRICT ON UPDATE RESTRICT
) ENGINE = InnoDB AUTO_INCREMENT = 41 CHARACTER SET = utf8 COLLATE =
utf8_general_ci ROW_FORMAT = Dynamic;

-- -----
-- Records of tb_question
-- -----
INSERT INTO `tb_question` VALUES (27, '标题', '描述', 1, 1, NULL, NULL, NULL, 0,
0, NULL, NULL, '2020-06-13 10:22:00', 18);
INSERT INTO `tb_question` VALUES (28, '标题', '描述', 1, 1, NULL, NULL, NULL, 0,
0, NULL, NULL, '2020-06-15 13:42:43', 18);
INSERT INTO `tb_question` VALUES (29, '标题', '描述', 3, 1, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, '2020-06-15 13:42:45', 18);
INSERT INTO `tb_question` VALUES (30, '标题', '描述', 4, 1, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, '2020-06-15 13:42:46', 18);
INSERT INTO `tb_question` VALUES (31, '标题', '描述', 2, 1, NULL, NULL, NULL, 0,
0, NULL, NULL, '2020-06-15 13:43:15', 18);
INSERT INTO `tb_question` VALUES (32, '标题adsfgasdfgasdsfgsdfgsdfg', '', 1, 1,
NULL, NULL, NULL, 0, 0, NULL, NULL, '2020-06-15 13:46:03', 18);
INSERT INTO `tb_question` VALUES (33, '你未来的职业规划是什么吗?', '', 1, 1, NULL,
NULL, NULL, 0, 0, NULL, NULL, '2020-06-15 15:20:59', 19);
INSERT INTO `tb_question` VALUES (34, '你期望的薪水', '', 2, 1, NULL, NULL, NULL,
0, 0, NULL, NULL, '2020-06-15 15:21:25', 19);
INSERT INTO `tb_question` VALUES (38, '标题', '描述', 3, 1, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, '2020-06-16 09:14:05', 19);
INSERT INTO `tb_question` VALUES (40, '标题', '描述', 2, 1, NULL, NULL, NULL, 0,
0, NULL, NULL, '2020-06-16 10:55:44', 19);

-- -----
-- Table structure for tb_question_opt
-- -----
DROP TABLE IF EXISTS `tb_question_opt`;
CREATE TABLE `tb_question_opt` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `survey_id` int(11) NULL DEFAULT NULL,
  `question_id` int(11) NULL DEFAULT NULL,
  `type` varchar(20) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL COMMENT '1radio|2checkbox',
  `opt` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL,
  `orderby` int(11) NULL DEFAULT NULL,
  `answer` int(1) NULL DEFAULT NULL COMMENT '默认为NULL; 1答案',
  PRIMARY KEY (`id`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 120 CHARACTER SET = utf8 COLLATE =
utf8_general_ci ROW_FORMAT = Dynamic;

-- -----
-- Records of tb_question_opt
-- -----
INSERT INTO `tb_question_opt` VALUES (80, 18, 28, '1', '选项', 1, NULL);
INSERT INTO `tb_question_opt` VALUES (81, 18, 28, '1', '选项', 2, NULL);
INSERT INTO `tb_question_opt` VALUES (82, 18, 28, '1', '选项', 3, NULL);
INSERT INTO `tb_question_opt` VALUES (83, 18, 28, '1', '选项', 4, NULL);
INSERT INTO `tb_question_opt` VALUES (84, 18, 31, '2', '选项', 1, NULL);
INSERT INTO `tb_question_opt` VALUES (85, 18, 31, '2', '选项', 2, NULL);
INSERT INTO `tb_question_opt` VALUES (86, 18, 31, '2', '选项', 3, NULL);
INSERT INTO `tb_question_opt` VALUES (87, 18, 31, '2', '选项', 4, NULL);

```

```

INSERT INTO `tb_question_opt` VALUES (88, 18, 32, '1', '选项', 1, NULL);
INSERT INTO `tb_question_opt` VALUES (89, 18, 32, '1', '选项', 2, NULL);
INSERT INTO `tb_question_opt` VALUES (90, 18, 32, '1', '选项', 3, NULL);
INSERT INTO `tb_question_opt` VALUES (91, 18, 32, '1', '选项', 4, NULL);
INSERT INTO `tb_question_opt` VALUES (92, 18, 27, '1', '选项', 1, NULL);
INSERT INTO `tb_question_opt` VALUES (93, 18, 27, '1', '选项12qasdfasdf', 2,
NULL);
INSERT INTO `tb_question_opt` VALUES (94, 18, 27, '1', '选项asdfasdf', 3, NULL);
INSERT INTO `tb_question_opt` VALUES (95, 18, 27, '1', '选项
fffffffffasdddddddddddddddddddddddddddddddd', 4, NULL);
INSERT INTO `tb_question_opt` VALUES (96, 19, 33, '1', 'Java软件工程师', 1, NULL);
INSERT INTO `tb_question_opt` VALUES (97, 19, 33, '1', '前端工程师', 2, NULL);
INSERT INTO `tb_question_opt` VALUES (98, 19, 33, '1', 'Python工程师', 3, NULL);
INSERT INTO `tb_question_opt` VALUES (99, 19, 33, '1', '大数据工程师', 4, NULL);
INSERT INTO `tb_question_opt` VALUES (100, 19, 34, '2', '5000', 1, NULL);
INSERT INTO `tb_question_opt` VALUES (101, 19, 34, '2', '8000', 2, NULL);
INSERT INTO `tb_question_opt` VALUES (102, 19, 34, '2', '10000', 3, NULL);
INSERT INTO `tb_question_opt` VALUES (103, 19, 34, '2', '15000', 4, NULL);
INSERT INTO `tb_question_opt` VALUES (116, 19, 40, '2', '选项', 1, NULL);
INSERT INTO `tb_question_opt` VALUES (117, 19, 40, '2', '选项', 2, NULL);
INSERT INTO `tb_question_opt` VALUES (118, 19, 40, '2', '选项', 3, NULL);
INSERT INTO `tb_question_opt` VALUES (119, 19, 40, '2', '选项', 4, NULL);

```

```

-- -----
-- Table structure for tb_survey
-- -----

```

```

DROP TABLE IF EXISTS `tb_survey`;
CREATE TABLE `tb_survey` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `title` varchar(100) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL,
  `remark` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL,
  `bounds` int(1) NULL DEFAULT NULL COMMENT '0:不限制;1:限制',
  `start_time` datetime(0) NULL DEFAULT NULL,
  `end_time` datetime(0) NULL DEFAULT NULL,
  `rules` int(1) NULL DEFAULT NULL COMMENT '0公开;1密码',
  `password` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL,
  `url` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL,
  `state` varchar(50) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL COMMENT '创建、执行中、结束',
  `logo` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL,
  `bgimg` varchar(200) CHARACTER SET utf8 COLLATE utf8_general_ci NULL DEFAULT
NULL,
  `anon` int(1) NULL DEFAULT NULL COMMENT '0匿名; 1不匿名',
  `creator` int(11) NULL DEFAULT NULL,
  `create_time` datetime(0) NULL DEFAULT NULL,
  PRIMARY KEY (`id`) USING BTREE
) ENGINE = InnoDB AUTO_INCREMENT = 20 CHARACTER SET = utf8 COLLATE =
utf8_general_ci ROW_FORMAT = Dynamic;

```

```

-- -----
-- Records of tb_survey
-- -----

```

```

INSERT INTO `tb_survey` VALUES (18, '我的第一份问卷调查', '我的第一份问卷调查我的第一份问卷调查我的第一份问卷调查我的第一份问卷调查', 0, '2020-06-12 20:00:00', '2020-06-14 00:00:00', 0, '', '', '结束', NULL, '790f8992311349dda939ab97b11ecfae_timg.jpg', 0, 2073, '2020-06-12 17:03:51');
INSERT INTO `tb_survey` VALUES (19, '职业规划测试', '职业规划测试职业规划测试职业规划测试职业规划测试职业规划测试职业规划测试', 0, '2020-06-15 00:00:00', '2020-06-17 00:00:00', 0, '', 'http://localhost:8080/survey/dy/4e0bd53b-91fa-41b4-a061-0964da735015', '执行中', NULL, '76c973d99eed4ffb90edf806bc5470f4_timg.jpg', 0, 2073, '2020-06-15 15:20:15');

SET FOREIGN_KEY_CHECKS = 1;

```

4.搭建项目

4.1Maven搭建项目

1 创建maven-archtype-webapp的项目

2配置maven路径



3 提示build success构建成功

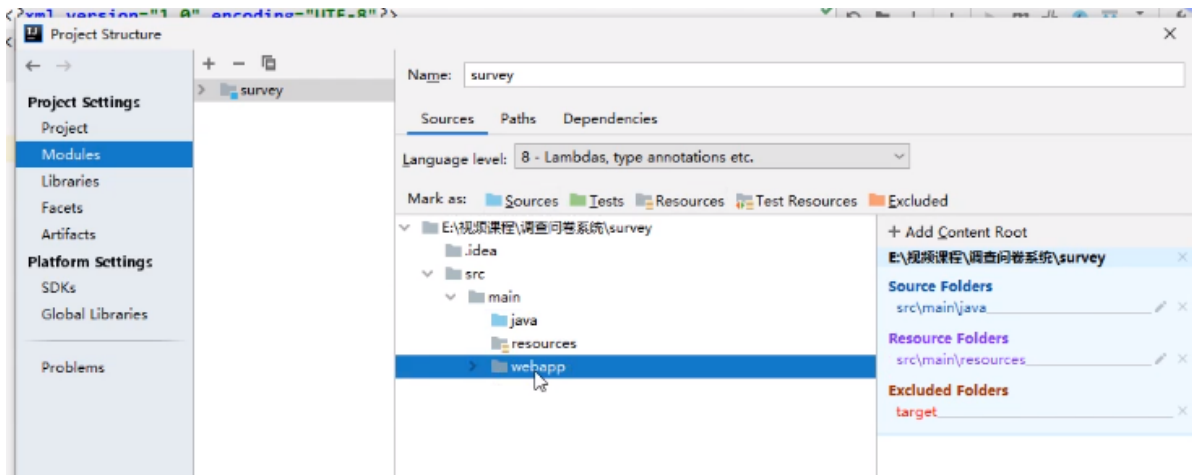
```

[INFO] Parameter: package, Value: com.yanzhen
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: groupId, Value: com.yanzhen
[INFO] Parameter: artifactId, Value: survey
[INFO] Project created from Archetype in dir: C:\Users\Admin\AppData
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 45.001 s
[INFO] Finished at: 2020-06-03T10:06:03+08:00
[INFO] -----

```

4 删除多余插件

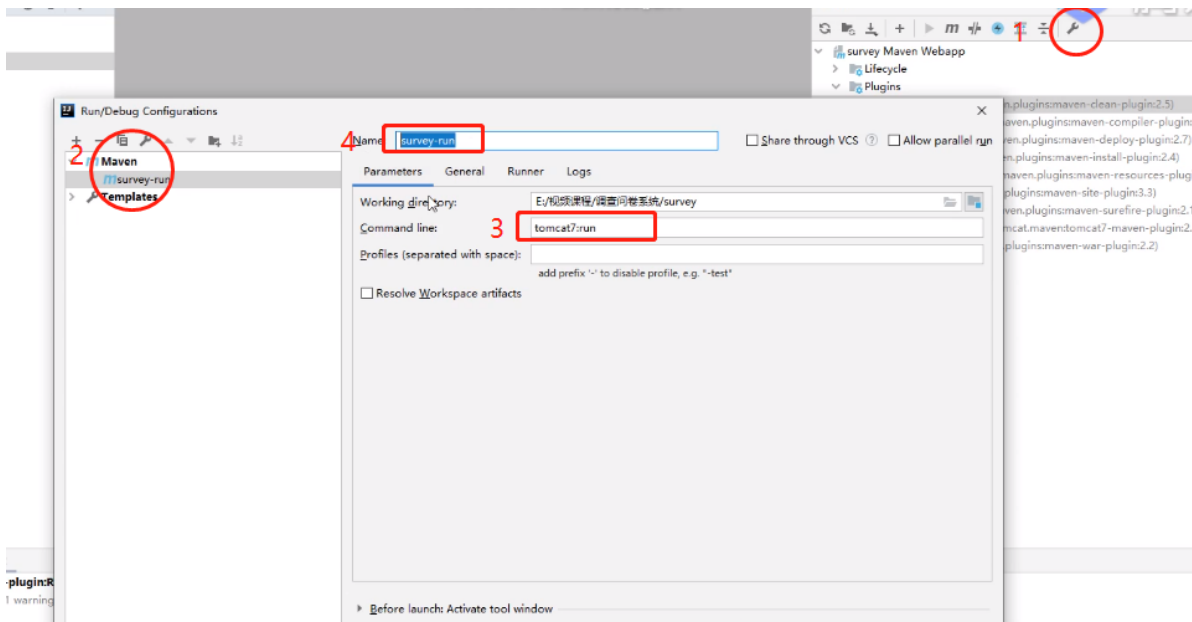
5 创建 java和resources文件夹

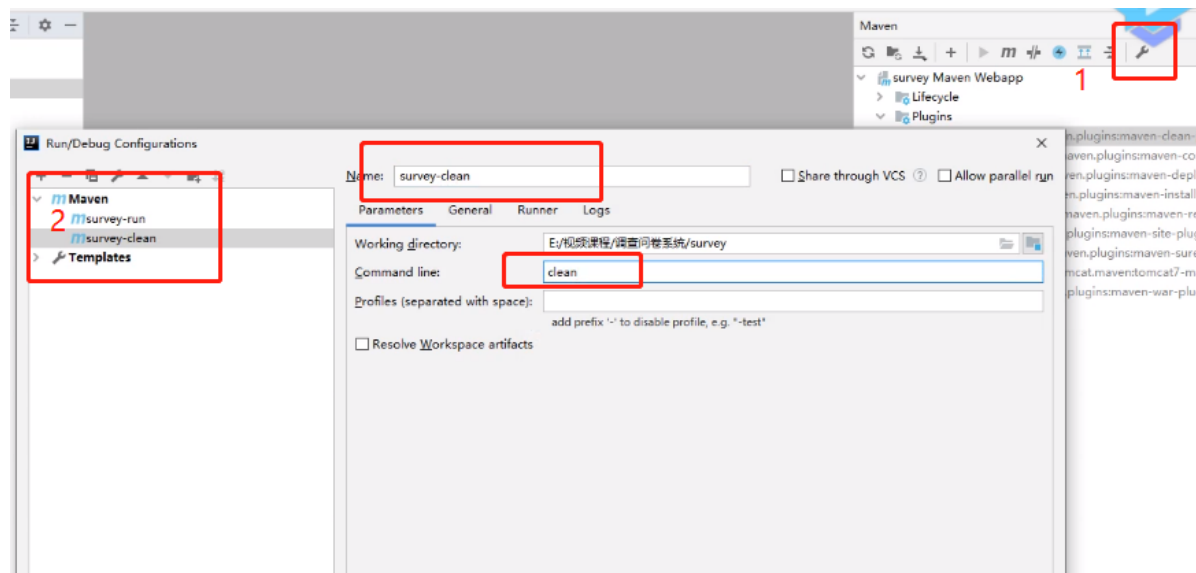


6 配置tomcat插件

```
<plugins>
  <plugin>
    <groupId>org.apache.tomcat.maven</groupId>
    <artifactId>tomcat7-maven-plugin</artifactId>
    <version>2.2</version>
    <configuration>
      <!-- 端口控制 -->
      <port>8080</port>
      <!-- 项目路径控制意味着 http://localhost:8080/abc -->
      <path>/survey</path>
      <!-- 编码 -->
      <uriEncoding>UTF-8</uriEncoding>
    </configuration>
  </plugin>
</plugins>
```

7 配置tomcat命令





参考视频 第四节.maven搭建

4.2引入Spring、Spring MVC框架

第一步：编写pom.xml文件

```
<?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>

  <groupId>com.yanzhen</groupId>
  <artifactId>survey</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>war</packaging>

  <name>survey Maven Webapp</name>
  <!-- FIXME change it to the project's website -->
  <url>http://www.example.com</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
    <spring.version>5.2.6.RELEASE</spring.version>
  </properties>

  <dependencies>
    <!--测试-->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
    <!--spring jar包-->
    <dependency>
```

```

        <groupId>org.springframework</groupId>
        <artifactId>spring-context</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-core</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-beans</artifactId>
        <version>${spring.version}</version>
    </dependency>
    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-aop</artifactId>
        <version>${spring.version}</version>
    </dependency>

    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-webmvc</artifactId>
        <version>${spring.version}</version>
    </dependency>

    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-jdbc</artifactId>
        <version>${spring.version}</version>
    </dependency>

    <dependency>
        <groupId>org.springframework</groupId>
        <artifactId>spring-tx</artifactId>
        <version>${spring.version}</version>
    </dependency>

</dependencies>

<build>
    <finalName>survey</finalName>
    <plugins>
        <plugin>
            <groupId>org.apache.tomcat.maven</groupId>
            <artifactId>tomcat7-maven-plugin</artifactId>
            <version>2.2</version>
            <configuration>
                <!-- 端口控制 -->
                <port>8080</port>
                <!-- 项目路径控制 http://localhost:8080/abc -->
                <path>/survey</path>
                <uriEncoding>UTF-8</uriEncoding>
            </configuration>
        </plugin>
    </plugins>
</build>

```

```
</project>
```

第二步：编写web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
        http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
    version="3.1">

    <!-- 配置前端控制器 -->
    <servlet>
        <servlet-name>survey</servlet-name>
        <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-
class>
        <init-param>
            <param-name>contextConfigLocation</param-name>
            <param-value>classpath:spring-mvc.xml</param-value>
        </init-param>
        <load-on-startup>1</load-on-startup>
    </servlet>

    <servlet-mapping>
        <servlet-name>survey</servlet-name>
        <url-pattern>/</url-pattern>
    </servlet-mapping>

    <!--加载Spring核心配置文件-->
    <context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:applicationContext.xml</param-value>
    </context-param>

    <listener>
        <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-class>
    </listener>

    <!--配置编码过滤器-->
    <filter>
        <filter-name>encodingFilter</filter-name>
        <filter-
class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
        <init-param>
            <param-name>encoding</param-name>
            <param-value>utf-8</param-value>
        </init-param>
    </filter>

    <filter-mapping>
        <filter-name>encodingFilter</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
```

```
</web-app>
```

第三步：编写sppllicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:p="http://www.springframework.org/schema/p"
       xmlns:util="http://www.springframework.org/schema/util"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/util
                           http://www.springframework.org/schema/util/spring-util.xsd
                           http://www.springframework.org/schema/context
                           http://www.springframework.org/schema/context/spring-context.xsd ">

</beans>
```

第四步：编写spring-mvc.xml文件

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xmlns:aop="http://www.springframework.org/schema/aop"
       xmlns:tx="http://www.springframework.org/schema/tx"
       xmlns:mvc="http://www.springframework.org/schema/mvc"
       xsi:schemaLocation="http://www.springframework.org/schema/beans
                           http://www.springframework.org/schema/beans/spring-beans.xsd
                           http://www.springframework.org/schema/context
                           http://www.springframework.org/schema/context/spring-context.xsd
                           http://www.springframework.org/schema/aop
                           http://www.springframework.org/schema/aop/spring-aop.xsd
                           http://www.springframework.org/schema/tx
                           http://www.springframework.org/schema/tx/spring-tx.xsd
                           http://www.springframework.org/schema/mvc
                           http://www.springframework.org/schema/mvc/spring-mvc.xsd">

    <bean
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/page/"></property>
        <property name="suffix" value=".jsp"></property>
    </bean>

    <mvc:default-servlet-handler/>

</beans>
```

- 需要spring核心配置文件、Spring MVC配置文件
 - 新建applicationContext.xml
 - 配置前端控制器等web.xml的相关配置

```

<!-- 配置前端控制器 -->
<servlet>
    <servlet-name>survey</servlet-name>
    <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:spring-mvc.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
    <servlet-name>survey</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
<!--加载Spring核心配置文件-->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext.xml</param-value>
</context-param>
<listener>
    <listener-
class>org.springframework.web.context.ContextLoaderListener</listener-
class>
</listener>
<!--配置编码过滤器-->
<filter>
    <filter-name>encodingFilter</filter-name>
    <filter-
class>org.springframework.web.filter.CharacterEncodingFilter</filter-
class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>utf-8</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>encodingFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

4.3引入Mybaitis框架

- mybatis相关的jar
- mybatis-spring.jar

```

<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>${mybatis.version}</version>
</dependency>

<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>2.0.4</version>
</dependency>

```

引入common-dbc2数据源相关的jar

```

<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-dbc2</artifactId>
  <version>2.7.0</version>
</dependency>

```

4.4SSM框架整合

```

<!--加载数据库配置文件 QQ:596183363-->
<context:property-placeholder location="classpath:db.properties" />

<context:component-scan base-package="com.yanzhen.service"/>

<!--配置数据源-->
<bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource">
  <property name="driverClassName" value="${jdbc.driver}"/>
  <property name="url" value="${jdbc.url}" />
  <property name="username" value="${jdbc.username}" />
  <property name="password" value="${jdbc.password}"/>
</bean>

<!--获取SqlSessionFactory工厂-->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
  <!--数据源-->
  <property name="dataSource" ref="dataSource"></property>
  <!--Mybatis主配置文件加载-->
  <property name="configLocation" value="classpath:mybatis-config.xml">
</property>
  <property name="mapperLocations">
    <list>
      <value>classpath:com/yanzhen/mapper/*.xml</value>
    </list>
  </property>
</bean>

<!--配置扫描Mapper文件-->
<bean id="mappersScanner"
class="org.mybatis.spring.mapper.MapperScannerConfigurer">
  <property name="basePackage" value="com.yanzhen.mapper"></property>
  <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>

```

```
</bean>
```

4.5 DAO层编码

1编写 AdminDao.class里的几个方法

```
import com.yanzhen.entity.Admin;

import java.util.List;
import java.util.Map;

public interface AdminDao {

    public int create(Admin admin);
    public int delete(Map<String,Object> paramMap);
    public int update(Map<String,Object> paramMap);
    public List<Admin> query(Map<String,Object> paramMap);
    public Admin detail(Map<String,Object> paramMap);
    public int count(Map<String,Object> paramMap);
}
```

2 编写AdminMapper.xml配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="com.yanzhen.mapper.AdminDao">

    <resultMap id="AdminMap" type="com.yanzhen.entity.Admin">
        <id column="id" property="id" />
        <result column="account" property="account" />
        <result column="password" property="password" />
        <result column="name" property="name" />
        <result column="phone" property="phone" />
        <result column="remark" property="remark" />
    </resultMap>

    <insert id="create" parameterType="com.yanzhen.entity.Admin"
keyProperty="id" useGeneratedKeys="true">
        insert into tb_admin(account,password,name,phone,remark) values (#{account},#{password},#{name},#{phone},#{remark})
    </insert>

    <sql id="AdminFindCirteria">
        <where>
            <if test="id!=null">and id=#{id}</if>
            <if test="account!=null">and account=#{account}</if>
            <if test="password!=null">and password=#{password}</if>
            <if test="name!=null">and name=#{name}</if>
            <if test="phone!=null">and phone=#{phone}</if>
        </where>
    </sql>

    <delete id="delete">
```



```

        delete from tb_admin
        <include refid="AdminFindCirteria"/>
    </delete>

    <select id="count" resultType="int">
        select count(1) from tb_admin
        <include refid="AdminFindCirteria"/>
    </select>

    <update id="update">
        update tb_admin set
        <include refid="AdminUpdateCirteria" />
        <include refid="AdminFindCirteria" />
    </update>

    <sql id="AdminUpdateCirteria">
        <set>
            <if test="account!=null">account=#{account},</if>
            <if test="password!=null">password=#{password},</if>
            <if test="name!=null">name=#{name},</if>
            <if test="phone!=null">phone=#{phone},</if>
            <if test="remark!=null">remark=#{remark},</if>
        </set>
    </sql>

    <select id="query" resultMap="AdminMap">
        select * from tb_admin
        <include refid="AdminFindCirteria" />
    </select>

    <select id="detail" resultMap="AdminMap">
        select * from tb_admin
        <include refid="AdminFindCirteria" />
        limit 1
    </select>

</mapper>

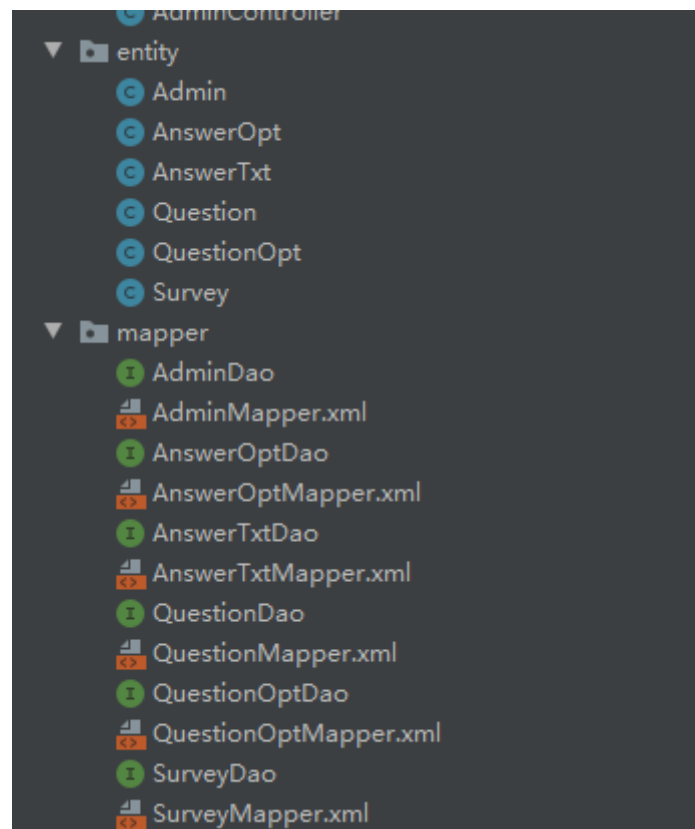
```

5.代码生成器

代码生成器项目：支持Spring Boot、SSM框架、Java基本JDBC技术

模块引擎采用Freemarker

5.1 生成Entity、DAO、DaoMapper



这里做了更改

用于区分更新和查询

```
<sql id="AdminFindCriteria">
    <where>
        <if test="account != null and account != ''">and account = #{account}</if>
        <if test="id != null and id != ''">and id = #{id}</if>
        <if test="name != null and name != ''">and name = #{name}</if>
        <if test="password != null and password != ''">and password = #{password}
    </if>
        <if test="phone != null and phone != ''">and phone = #{phone}</if>
        <if test="remark != null and remark != ''">and remark = #{remark}</if>
    </where>
</sql>
<sql id="AdminUpdateCriteria">
    <set>
        <if test="updateAccount != null and updateAccount != ''">account = #
{updateAccount},</if>
        <if test="updateId != null and updateId != ''">id = #{updateId},</if>
        <if test="updateName != null and updateName != ''">name = #{updateName},
    </if>
        <if test="updatePassword != null and updatePassword != ''">password = #
{updatePassword},</if>
        <if test="updatePhone != null and updatePhone != ''">phone = #
{updatePhone},</if>
        <if test="updateRemark != null and updateRemark != ''">remark = #
{updateRemark},</if>
    </set>
</sql>
```

6.Service层编码

6.1编写AdminService

```
@Service
public class AdminService {

    @Autowired
    private AdminDao adminDao;

    public int create(Admin pi){
        return adminDao.create(pi);
    }

    public int delete(Integer id){
        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("id",id);
        return adminDao.delete(paramMap);
    }

    public int update(Admin admin){
        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("id",admin.getId());
        paramMap.put("updateAccount",admin.getAccount());
        paramMap.put("updateName",admin.getName());
        paramMap.put("updatePassword",admin.getPassword());
        paramMap.put("updatePhone",admin.getPhone());
        paramMap.put("updateRemark",admin.getRemark());
        return adminDao.update(paramMap);
    }

    public List<Admin> query(Admin admin){
        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("id",admin.getId());
        paramMap.put("account",admin.getAccount());
        paramMap.put("name",admin.getName());
        paramMap.put("password",admin.getPassword());
        paramMap.put("phone",admin.getPhone());
        paramMap.put("remark",admin.getRemark());
        return adminDao.query(paramMap);
    }

    public Admin detail(Integer id){
        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("id",id);
        return adminDao.detail(paramMap);
    }

    public int count(Admin admin){
        Map<String, Object> paramMap = new HashMap<>();
        paramMap.put("id",admin.getId());
        paramMap.put("account",admin.getAccount());
        paramMap.put("name",admin.getName());
        paramMap.put("password",admin.getPassword());
        paramMap.put("phone",admin.getPhone());
        paramMap.put("remark",admin.getRemark());
```

```

        return adminDao.count(paramMap);
    }
}

```

6.2 pom.xml 中加入Guava工具类库

用于mybatis和spring、spring-mvc整合

```

<dependency>
    <groupId>com.google.guava</groupId>
    <artifactId>guava</artifactId>
    <version>29.0-jre</version>
</dependency>

```

6.3 编写BeanMapUtils.class 类 将entity对象 转换成Map

```

import com.google.common.base.CaseFormat;
import com.google.common.base.Joiner;
import com.google.common.base.Strings;
import com.google.common.collect.Lists;
import com.google.common.collect.Maps;
import com.yanzhen.entity.Admin;
import com.yanzhen.entity.Survey;
import org.springframework.cglib.beans.BeanMap;

import java.util.Date;
import java.util.List;
import java.util.Map;

public class BeanMapUtils {
    /**
     * 将对象转换为map
     * @param bean
     * @return
     */
    public static <T> Map<String, Object> beanToMap(T bean) {
        Map<String, Object> map = Maps.newHashMap();
        if (bean != null) {
            BeanMap beanMap = BeanMap.create(bean);
            for (Object key : beanMap.keySet()) {
                map.put(key+"", beanMap.get(key));
            }
        }
        return map;
    }

    public static <T> Map<String, Object> beanToMapForUpdate(T bean) {
        Map<String, Object> map = Maps.newHashMap();
        if (bean != null) {
            BeanMap beanMap = BeanMap.create(bean);
            for (Object key : beanMap.keySet()) {
                map.put("update"+upperFirstLatter(key+""), beanMap.get(key));
            }
        }
    }
}

```

```

        return map;
    }

    /**
     * 将map转换为javabean对象
     * @param map
     * @param bean
     * @return
     */
    public static <T> T mapToBean(Map<String, Object> map, T bean) {
        BeanMap beanMap = BeanMap.create(bean);
        beanMap.putAll(map);
        return bean;
    }

    /**
     * 将List<T>转换为List<Map<String, Object>>
     * @param objList
     * @return
     */
    public static <T> List<Map<String, Object>> objectsToMaps(List<T> objList) {
        List<Map<String, Object>> list = Lists.newArrayList();
        if (objList != null && objList.size() > 0) {
            Map<String, Object> map = null;
            T bean = null;
            for (int i = 0, size = objList.size(); i < size; i++) {
                bean = objList.get(i);
                map = beanToMap(bean);
                list.add(map);
            }
        }
        return list;
    }

    /**
     * 将List<Map<String, Object>>转换为List<T>
     * @param maps
     * @param clazz
     * @return
     * @throws InstantiationException
     * @throws IllegalAccessException
     */
    public static <T> List<T> mapsToObjects(List<Map<String, Object>>
maps, Class<T> clazz) throws InstantiationException, IllegalAccessException {
        List<T> list = Lists.newArrayList();
        if (maps != null && maps.size() > 0) {
            Map<String, Object> map = null;
            T bean = null;
            for (int i = 0, size = maps.size(); i < size; i++) {
                map = maps.get(i);
                bean = clazz.newInstance();
                mapToBean(map, bean);
                list.add(bean);
            }
        }
        return list;
    }
}

```

```

    public static String upperFirstLetter(String letter){
        char[] chars = letter.toCharArray();
        if(chars[0]>='a' && chars[0]<='z'){
            chars[0] = (char) (chars[0]-32);
        }
        return new String(chars);
    }

    public static void main(String[] args) {
        Admin admin = new Admin();
        admin.setId(1);
        admin.setAccount("1001");
        admin.setNow(new Date());
        Map<String, Object> map = beanToMap(admin);
        System.out.println(map);

        Survey survey = new Survey();
        survey.setEndTime(new Date());

        Map<String, Object> map2 = beanToMapForUpdate(survey);
        System.out.println(map2);

        //System.out.println(CaseFormat.LOWER_UNDERSCORE.to(CaseFormat.LOWER_CAMEL,"update_id"));

        //System.out.println(CaseFormat.LOWER_UNDERSCORE.to(CaseFormat.LOWER_CAMEL,"update_name"));
    }
}

```

6.4 编写MapperParameter.class文件

将new HashMap()改为链式结构

```

package com.yanzhen.utils;

import java.util.HashMap;
import java.util.Map;

public class MapParameter {

    private Map<String,Object> paramMap = new HashMap<>();

    private MapParameter(){

    }

    public static MapParameter getInstance(){
        return new MapParameter();
    }

    public Map<String,Object> getMap(){
        return paramMap;
    }
}

```

```

    }

    public MapParameter put(Map<String, Object> map){
        for (Map.Entry<String, Object> entry : map.entrySet()) {
            paramMap.put(entry.getKey(), entry.getValue());
        }
        return this;
    }

    public MapParameter put(String key, Object val){
        paramMap.put(key, val);
        return this;
    }

    public MapParameter add(String key, Object val){
        return this.put(key, val);
    }

    public MapParameter addId(Object val){
        return this.put("id", val);
    }

    public static void main(String[] args) {
        Map<String, Object> map = MapParameter.getInstance().put("name",
"hua").put("age", 30).getMap();
        System.out.println(map);
    }
}

```

6.5 修改AdminService.class

精简后的AdminService代码

```

package com.yanzhen.service;

import com.github.pagehelper.PageHelper;
import com.google.common.base.Joiner;
import com.google.common.base.Splitter;
import com.google.common.collect.Maps;
import com.yanzhen.entity.Admin;
import com.yanzhen.mapper.AdminDao;
import com.yanzhen.utils.BeanMapUtils;
import com.yanzhen.utils.MapParameter;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

@Service
@Transactional
public class AdminService {
    @Autowired
    private AdminDao adminDao;
}

```

```

public int create(Admin pi){
    return adminDao.create(pi);
}

public int delete(Integer id){
    return adminDao.delete(MapParameter.getInstance().addId(id).getMap());
}

public int update(Admin admin){
    Map<String, Object> map =
MapParameter.getInstance().put(BeanMapUtils.beanToMapForUpdate(admin)).addId(admin.getId()).getMap();
    return adminDao.update(map);
}

public List<Admin> query(Admin admin){
    PageHelper.startPage(admin.getPage(), admin.getLimit());
    Map<String, Object> map =
MapParameter.getInstance().put(BeanMapUtils.beanToMap(admin)).getMap();
    return adminDao.query(map);
}

public Admin detail(Integer id){
    return adminDao.detail(MapParameter.getInstance().addId(id).getMap());
}

public int count(Admin admin){
    Map<String, Object> map =
MapParameter.getInstance().put(BeanMapUtils.beanToMap(admin)).getMap();
    return adminDao.count(map);
}
}

```

7.Controller层编码

@ResponseBody 支持对象响应到客户端JSON

此时 return 对象 可以被解析为JSON格式

7.1 pom.xml注入相关的jar依赖

```

<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-core</artifactId>
    <version>${jackson.version}</version>
</dependency>
<dependency>
    <groupId>com.fasterxml.jackson.core</groupId>
    <artifactId>jackson-databind</artifactId>
    <version>${jackson.version}</version>
</dependency>

```


7.2 spring-mvc.xml 配置注解驱动

```
<mvc:annotation-driven>
    <mvc:message-converters>
        <bean
class="org.springframework.http.converter.json.MappingJackson2HttpMessageConvert
er" />
    </mvc:message-converters>
</mvc:annotation-driven>
```

7.3 配置字符串转换器

此时 return “success” 在页面上得到的值也是 “success”

需要spring-mvc.xml中加入字符串转换器

```
<bean id="utf8Charset" class="java.nio.charset.Charset"
    factory-method="forName">
    <constructor-arg value="UTF-8"/>
</bean>

<mvc:annotation-driven>
    <mvc:message-converters>
        <bean
class="org.springframework.http.converter.StringHttpMessageConverter">
            <constructor-arg ref="utf8Charset"/>
        </bean>
    </mvc:message-converters>
</mvc:annotation-driven>
```

7.4 配置时间格式

从页面输入2020-09-10，服务器只会把它当作字符串无法识别为Date类型

spring-mvc.xml中

```
<bean id="conversionService"
class="org.springframework.format.support.FormattingConversionServiceFactoryBean"
">
    <property name="converters">
        <set>
            <bean class="com.yanzhen.utils.DateConvert"></bean>
        </set>
    </property>
</bean>

<mvc:annotation-driven conversion-service="conversionService">
    <mvc:message-converters>
        <bean
class="org.springframework.http.converter.StringHttpMessageConverter">
            <constructor-arg ref="utf8Charset"/>
        </bean>
        <bean
class="org.springframework.http.converter.json.MappingJackson2HttpMessageConvert
er">
            <property name="objectMapper">
```

```

        <bean class="com.fasterxml.jackson.databind.ObjectMapper">
            <property name="dateFormat">
                <bean class="java.text.SimpleDateFormat">
                    <constructor-arg type="java.lang.String"
value="yyyy-MM-dd"/>
                </bean>
            </property>
        </bean>
    </mvc:message-converters>
</mvc:annotation-driven>

```

创建DateConvert.class类

```

package com.yanzhen.utils;

import org.springframework.core.convert.converter.Converter;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.logging.SimpleFormatter;

public class DateConvert implements Converter<String, Date> {

    @Override
    public Date convert(String source) {
        try {
            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
            return sdf.parse(source);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }
}

```

7.5 完整的spring-mvc.xml 文件为

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:context="http://www.springframework.org/schema/context"
xmlns:aop="http://www.springframework.org/schema/aop"
xmlns:tx="http://www.springframework.org/schema/tx"
xmlns:mvc="http://www.springframework.org/schema/mvc"
xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
http://www.springframework.org/schema/aop
http://www.springframework.org/schema/aop/spring-aop.xsd
http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx.xsd
http://www.springframework.org/schema/mvc

```

```

http://www.springframework.org/schema/mvc/spring-mvc.xsd">

<!--扫描controller-->
<context:component-scan base-package="com.yanzhen.controller"/>

<bean id="utf8Charset" class="java.nio.charset.Charset"
      factory-method="forName">
    <constructor-arg value="UTF-8"/>
</bean>

<bean id="conversionService"
class="org.springframework.format.support.FormattingConversionServiceFactoryBean"
">
    <property name="converters">
        <set>
            <bean class="com.yanzhen.utils.DateConvert"></bean>
        </set>
    </property>
</bean>

<mvc:annotation-driven conversion-service="conversionService">
    <mvc:message-converters>
        <bean
class="org.springframework.http.converter.StringHttpMessageConverter">
            <constructor-arg ref="utf8Charset"/>
        </bean>
        <bean
class="org.springframework.http.converter.json.MappingJackson2HttpMessageConvert
er">
            <property name="objectMapper">
                <bean class="com.fasterxml.jackson.databind.ObjectMapper">
                    <property name="dateFormat">
                        <bean class="java.text.SimpleDateFormat">
                            <constructor-arg type="java.lang.String"
value="yyyy-MM-dd"/>
                        </bean>
                    </property>
                </bean>
            </property>
        </bean>
    </property>
</mvc:message-converters>
</mvc:annotation-driven>

</beans>

```

8.Controller层与前端交互

8.1 前端采用普通的JSP技术（不含ajax）

举例：表格，在页面显示的时候，一般有来说有如下两种方式

8.2 采用页面跳转的方式进行

```
@PostMapping("/query")
public String query(Admin admin, ModelMap modelMap){
    List<Admin> list = adminService.query(admin);
    modelMap.addAttribute("list",list);
    return "../list.jsp";
}
```

```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>servlet-api</artifactId>
    <version>3.0-alpha-1</version>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>jsp-api</artifactId>
    <version>2.2</version>
    <scope>provided</scope>
</dependency>

<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
```

8.3 采用Ajax异步数据加载的方式进行

```
@GetMapping("/query2")
@ResponseBody
public List<Admin> query2(Admin admin, ModelMap modelMap){
    List<Admin> list = adminService.query(admin);
    return list;
}
```

目前主流的项目应用

- 前后端分离（异步请求），目前主流的互联网公司的产品。
- 异步请求（居多）+页面跳转（少部分）一直居多，未采用前后端分离的，基本都是采用该技术。
- 页面跳转（居多）+异步请求（少部分），早期的项目采用该技术的较多，

9.前端技术LayUI

前端框架采用layuimini

前端开发工具采用WebStorm\HBuilder\vscode

9.1 前端页面步骤

1 打开layui-mini2的项目

2 新建空工程

空工程命名为 survey-static

3 新建文件夹

在空工程中 新建css、js、images、lib、data文件夹

4 新建html

新建add.html、index.html、list.html、login.html

5 静态资源

将layui-mini中的css、js、lib中的内容复制过来，将init.json、table.json复制到data文件夹下,找一个1980x1080的图片放在images文件下作为背景图片，命名为login_bg.jpg。

6 html 编写

layuimini2 survey-static

index.html——>index.html

form.html——>add.html

table.html——>list.html

login.html——>login.html

7 源码

index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>layuimini-iframe版 v2 - 基于Layui的后台管理系统前端模板</title>
  <meta name="keywords" content="layuimini,layui,layui模板,layui后台,后台模板,admin,admin模板,layui mini">
  <meta name="description" content="layuimini基于layui的轻量级前端后台管理框架，最简洁、易用的后台框架模板，面向所有层次的前后端程序，只需提供一个接口就直接初始化整个框架，无需复杂操作。">
  <meta name="renderer" content="webkit">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta http-equiv="Access-Control-Allow-Origin" content="*>
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="format-detection" content="telephone=no">
  <link rel="icon" href="images/favicon.ico">
  <link rel="stylesheet" href="lib/layui-v2.5.5/css/layui.css" media="all">
  <link rel="stylesheet" href="css/layuimini.css?v=2.0.4.2" media="all">
  <link rel="stylesheet" href="css/themes/default.css" media="all">
```

```

<link rel="stylesheet" href="lib/font-awesome-4.7.0/css/font-
awesome.min.css" media="all">
<!--[if lt IE 9]>
<script src="js/html5.min.js"></script>
<script src="js/respond.min.js"></script>
<![endif]-->
<style id="layuimini-bg-color">
</style>
</head>
<body class="layui-layout-body layuimini-all">
<div class="layui-layout layui-layout-admin">

    <div class="layui-header header">
        <div class="layui-logo layuimini-logo"></div>

        <div class="layuimini-header-content">
            <a>
                <div class="layuimini-tool"><i title="展开" class="fa fa-outdent"
data-side-fold="1"></i></div>
            </a>

            <!--电脑端头部菜单-->
            <ul class="layui-nav layui-layout-left layuimini-header-menu
layuimini-menu-header-pc layuimini-pc-show">
                </ul>

            <!--手机端头部菜单-->
            <ul class="layui-nav layui-layout-left layuimini-header-menu
layuimini-mobile-show">
                <li class="layui-nav-item">
                    <a href="javascript:;"><i class="fa fa-list-ul"></i> 选择模块
</a>
                    <dl class="layui-nav-child layuimini-menu-header-mobile">
                        </dl>
                </li>
            </ul>

            <ul class="layui-nav layui-layout-right">

                <li class="layui-nav-item lay-unselect">
                    <a href="javascript:;" data-refresh="刷新"><i class="fa fa-
refresh"></i></a>
                </li>
                <li class="layui-nav-item lay-unselect">
                    <a href="javascript:;" data-clear="清理" class="layuimini-
clear"><i class="fa fa-trash-o"></i></a>
                </li>
                <li class="layui-nav-item mobile layui-hide-xs lay-unselect">
                    <a href="javascript:;" data-check-screen="full"><i class="fa
fa-arrows-alt"></i></a>
                </li>
                <li class="layui-nav-item layuimini-setting">
                    <a href="javascript:;">admin</a>
                    <dl class="layui-nav-child">
                        <dd>
                            <a href="javascript:;" layuimini-content-
href="page/user-setting.html" data-title="基本资料" data-icon="fa fa-gears">基本资
料<span class="layui-badge-dot"></span></a>

```

```

        </dd>
        <dd>
            <a href="javascript:;" layuimini-content-
href="page/user-password.html" data-title="修改密码" data-icon="fa fa-gears">修改密
码</a>
        </dd>
        <dd>
            <hr>
        </dd>
        <dd>
            <a href="javascript:;" class="login-out">退出登录</a>
        </dd>
    </dl>
</li>
<li class="layui-nav-item layuimini-select-bgcolor" lay-
unselect>
    <a href="javascript:;" data-bgcolor="配色方案"><i class="fa
fa-ellipsis-v"></i></a>
    </li>
</ul>
</div>
</div>

<!--无限极左侧菜单-->
<div class="layui-side layui-bg-black layuimini-menu-left">
</div>

<!--初始化加载层-->
<div class="layuimini-loader">
    <div class="layuimini-loader-inner"></div>
</div>

<!--手机端遮罩层-->
<div class="layuimini-make"></div>

<!-- 移动导航 -->
<div class="layuimini-site-mobile"><i class="layui-icon">☐</i></div>

<div class="layui-body">

    <div class="layuimini-tab layui-tab-rollTool layui-tab" lay-
filter="layuiminiTab" lay-allowclose="true">
        <ul class="layui-tab-title">
            <li class="layui-this" id="layuiminiHomeTabId" lay-id=""></li>
        </ul>
        <div class="layui-tab-control">
            <li class="layuimini-tab-roll-left layui-icon layui-icon-left">
</li>
            <li class="layuimini-tab-roll-right layui-icon layui-icon-
right"></li>
            <li class="layui-tab-tool layui-icon layui-icon-down">
                <ul class="layui-nav close-box">
                    <li class="layui-nav-item">
                        <a href="javascript:;"><span class="layui-nav-more">
</span></a>
                        <dl class="layui-nav-child">
                            <dd><a href="javascript:;" layuimini-tab-
close="current">关 闭 当 前</a></dd>

```

```

        <dd><a href="javascript:;" layuimini-tab-
close="other">关 闭 其 他</a></dd>
        <dd><a href="javascript:;" layuimini-tab-
close="all">关 闭 全 部</a></dd>
    </dl>
</li>
</ul>
</li>
</div>
<div class="layui-tab-content">
    <div id="layuiminiHomeTabIframe" class="layui-tab-item layui-
show"></div>
</div>
</div>

<script src="lib/layui-v2.5.5/layui.js" charset="utf-8"></script>
<script src="js/lay-config.js?v=2.0.0" charset="utf-8"></script>
<script>
    layui.use(['jquery', 'layer', 'miniAdmin', 'miniTongji'], function () {
        var $ = layui.jquery,
            layer = layui.layer,
            miniAdmin = layui.miniAdmin,
            miniTongji = layui.miniTongji;

        var options = {
            iniurl: "data/init.json",      // 初始化接口
            urlHashLocation: true,        // 是否打开hash定位
            bgColorDefault: false,        // 主题默认配置
            multiModule: true,            // 是否开启多模块
            menuChildOpen: false,         // 是否默认展开菜单
            loadingTime: 0,                // 初始化加载时间
            pageAnim: true,                // iframe窗口动画
            maxTabNum: 20,                 // 最大的tab打开数量
        };
        miniAdmin.render(options);

        $('.login-out').on("click", function () {
            layer.msg('退出登录成功', function () {
                window.location = 'page/login-1.html';
            });
        });
    });
</script>
</body>
</html>

```

add.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>layui</title>

```



```

<meta name="renderer" content="webkit">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
<link rel="stylesheet" href="lib/layui-v2.5.5/css/layui.css" media="all">
<link rel="stylesheet" href="css/public.css" media="all">
</head>
<body>
<div class="layuiimini-container">
  <div class="layuiimini-main">

    <blockquote class="layui-elem-quote layui-text">
      鉴于小伙伴的普遍反馈，先温馨提醒两个常见“问题”：1. <a
href="/doc/base/faq.html#form" target="_blank">为什么select/checkbox/radio没显示？
</a> 2. <a href="/doc/modules/form.html#render" target="_blank">动态添加的表单元素如
何更新？ </a>
    </blockquote>

    <fieldset class="layui-elem-field layui-field-title" style="margin-top:
20px;">
      <legend>表单集合演示</legend>
    </fieldset>

    <form class="layui-form" action="">
      <div class="layui-form-item">
        <label class="layui-form-label">单行输入框</label>
        <div class="layui-input-block">
          <input type="text" name="title" lay-verify="title"
autocomplete="off" placeholder="请输入标题" class="layui-input">
        </div>
      </div>
      <div class="layui-form-item">
        <label class="layui-form-label">验证必填项</label>
        <div class="layui-input-block">
          <input type="text" name="username" lay-verify="required"
lay-reqtext="用户名是必填项，岂能为空？" placeholder="请输入" autocomplete="off"
class="layui-input">
        </div>
      </div>

      <div class="layui-form-item">
        <div class="layui-inline">
          <label class="layui-form-label">验证手机</label>
          <div class="layui-input-inline">
            <input type="tel" name="phone" lay-
verify="required|phone" autocomplete="off" class="layui-input">
          </div>
        </div>
        <div class="layui-inline">
          <label class="layui-form-label">验证邮箱</label>
          <div class="layui-input-inline">
            <input type="text" name="email" lay-verify="email"
autocomplete="off" class="layui-input">
          </div>
        </div>
      </div>

      <div class="layui-form-item">

```

```
<div class="layui-inline">
  <label class="layui-form-label">多规则验证</label>
  <div class="layui-input-inline">
    <input type="text" name="number" lay-
verify="required|number" autocomplete="off" class="layui-input">
  </div>
</div>
<div class="layui-inline">
  <label class="layui-form-label">验证日期</label>
  <div class="layui-input-inline">
    <input type="text" name="date" id="date" lay-
verify="date" placeholder="yyyy-MM-dd" autocomplete="off" class="layui-input">
  </div>
</div>
<div class="layui-inline">
  <label class="layui-form-label">验证链接</label>
  <div class="layui-input-inline">
    <input type="tel" name="url" lay-verify="url"
autocomplete="off" class="layui-input">
  </div>
</div>
</div>

<div class="layui-form-item">
  <label class="layui-form-label">验证身份证</label>
  <div class="layui-input-block">
    <input type="text" name="identity" lay-verify="identity"
placeholder="" autocomplete="off" class="layui-input">
  </div>
</div>
<div class="layui-form-item">
  <label class="layui-form-label">自定义验证</label>
  <div class="layui-input-inline">
    <input type="password" name="password" lay-verify="pass"
placeholder="请输入密码" autocomplete="off" class="layui-input">
  </div>
  <div class="layui-form-mid layui-word-aux">请填写6到12位密码</div>
</div>

<div class="layui-form-item">
  <div class="layui-inline">
    <label class="layui-form-label">范围</label>
    <div class="layui-input-inline" style="width: 100px;">
      <input type="text" name="price_min" placeholder="¥"
autocomplete="off" class="layui-input">
    </div>
    <div class="layui-form-mid">-</div>
    <div class="layui-input-inline" style="width: 100px;">
      <input type="text" name="price_max" placeholder="¥"
autocomplete="off" class="layui-input">
    </div>
  </div>
</div>

<div class="layui-form-item">
  <label class="layui-form-label">单行选择框</label>
  <div class="layui-input-block">
    <select name="interest" lay-filter="aihao">
```

```

        <option value=""></option>
        <option value="0">写作</option>
        <option value="1" selected="">阅读</option>
        <option value="2">游戏</option>
        <option value="3">音乐</option>
        <option value="4">旅行</option>
    </select>
</div>
</div>

<div class="layui-form-item">
    <div class="layui-inline">
        <label class="layui-form-label">分组选择框</label>
        <div class="layui-input-inline">
            <select name="quiz">
                <option value="">请选择问题</option>
                <optgroup label="城市记忆">
                    <option value="你工作的第一个城市">你工作的第一个城市
</option>

                </optgroup>
                <optgroup label="学生时代">
                    <option value="你的工号">你的工号</option>
                    <option value="你最喜欢的老师">你最喜欢的老师
</option>

                </optgroup>
            </select>
        </div>
    </div>
    <div class="layui-inline">
        <label class="layui-form-label">搜索选择框</label>
        <div class="layui-input-inline">
            <select name="modules" lay-verify="required" lay-
search="">

                <option value="">直接选择或搜索选择</option>
                <option value="1">layer</option>
                <option value="2">form</option>
                <option value="3">layim</option>
                <option value="4">element</option>
                <option value="5">laytpl</option>
                <option value="6">upload</option>
                <option value="7">laydate</option>
                <option value="8">laypage</option>
                <option value="9">flow</option>
                <option value="10">util</option>
                <option value="11">code</option>
                <option value="12">tree</option>
                <option value="13">layedit</option>
                <option value="14">nav</option>
                <option value="15">tab</option>
                <option value="16">table</option>
                <option value="17">select</option>
                <option value="18">checkbox</option>
                <option value="19">switch</option>
                <option value="20">radio</option>
            </select>
        </div>
    </div>
</div>

```

```

</div>

<div class="layui-form-item">
  <label class="layui-form-label">联动选择框</label>
  <div class="layui-input-inline">
    <select name="quiz1">
      <option value="">请选择省</option>
      <option value="浙江" selected="">浙江省</option>
      <option value="你的工号">江西省</option>
      <option value="你最喜欢的老师">福建省</option>
    </select>
  </div>
  <div class="layui-input-inline">
    <select name="quiz2">
      <option value="">请选择市</option>
      <option value="杭州">杭州</option>
      <option value="宁波" disabled="">宁波</option>
      <option value="温州">温州</option>
      <option value="温州">台州</option>
      <option value="温州">绍兴</option>
    </select>
  </div>
  <div class="layui-input-inline">
    <select name="quiz3">
      <option value="">请选择县/区</option>
      <option value="西湖区">西湖区</option>
      <option value="余杭区">余杭区</option>
      <option value="拱墅区">临安市</option>
    </select>
  </div>
  <div class="layui-form-mid layui-word-aux">此处只是演示联动排版，并未
做联动交互</div>
</div>

<div class="layui-form-item">
  <label class="layui-form-label">复选框</label>
  <div class="layui-input-block">
    <input type="checkbox" name="like[write]" title="写作">
    <input type="checkbox" name="like[read]" title="阅读"
checked="">
    <input type="checkbox" name="like[game]" title="游戏">
  </div>
</div>

<div class="layui-form-item" pane="">
  <label class="layui-form-label">原始复选框</label>
  <div class="layui-input-block">
    <input type="checkbox" name="like1[write]" lay-
skin="primary" title="写作" checked="">
    <input type="checkbox" name="like1[read]" lay-skin="primary"
title="阅读">
    <input type="checkbox" name="like1[game]" lay-skin="primary"
title="游戏" disabled="">
  </div>
</div>

<div class="layui-form-item">
  <label class="layui-form-label">开关-默认关</label>

```

```

        <div class="layui-input-block">
            <input type="checkbox" name="close" lay-skin="switch" lay-
text="ON|OFF">
        </div>
    </div>
    <div class="layui-form-item">
        <label class="layui-form-label">开关-默认开</label>
        <div class="layui-input-block">
            <input type="checkbox" checked="" name="open" lay-
skin="switch" lay-filter="switchTest" lay-text="ON|OFF">
        </div>
    </div>
    <div class="layui-form-item">
        <label class="layui-form-label">单选框</label>
        <div class="layui-input-block">
            <input type="radio" name="sex" value="男" title="男"
checked="">
            <input type="radio" name="sex" value="女" title="女">
            <input type="radio" name="sex" value="禁" title="禁用"
disabled="">
        </div>
    </div>
    <div class="layui-form-item layui-form-text">
        <label class="layui-form-label">普通文本域</label>
        <div class="layui-input-block">
            <textarea placeholder="请输入内容" class="layui-textarea">
</textarea>
        </div>
    </div>
    <!--<div class="layui-form-item layui-form-text">
        <label class="layui-form-label">编辑器</label>
        <div class="layui-input-block">
            <textarea class="layui-textarea layui-hide" name="content" lay-
verify="content" id="LAY_demo_editor"></textarea>
        </div>
    </div>-->
    <div class="layui-form-item">
        <div class="layui-input-block">
            <button class="layui-btn" lay-submit="" lay-filter="demo1">立
即提交</button>
            <button type="reset" class="layui-btn layui-btn-primary">重置
</button>
        </div>
    </div>
</form>

<!-- 示例-970 -->
<ins class="adsbygoogle" style="display:inline-
block;width:970px;height:90px" data-ad-client="ca-pub-6111334333458862" data-ad-
slot="3820120620"></ins>

<fieldset class="layui-elem-field layui-field-title" style="margin-top:
50px;">
    <legend>初始赋值演示</legend>
</fieldset>

<form class="layui-form" action="" lay-filter="example">
    <div class="layui-form-item">

```

```

        <label class="layui-form-label">输入框</label>
        <div class="layui-input-block">
            <input type="text" name="username" lay-verify="title"
autocomplete="off" placeholder="请输入标题" class="layui-input">
        </div>
    </div>
    <div class="layui-form-item">
        <label class="layui-form-label">密码框</label>
        <div class="layui-input-block">
            <input type="password" name="password" placeholder="请输入密
码" autocomplete="off" class="layui-input">
        </div>
    </div>

    <div class="layui-form-item">
        <label class="layui-form-label">选择框</label>
        <div class="layui-input-block">
            <select name="interest" lay-filter="aihao">
                <option value=""></option>
                <option value="0">写作</option>
                <option value="1">阅读</option>
                <option value="2">游戏</option>
                <option value="3">音乐</option>
                <option value="4">旅行</option>
            </select>
        </div>
    </div>

    <div class="layui-form-item">
        <label class="layui-form-label">复选框</label>
        <div class="layui-input-block">
            <input type="checkbox" name="like[write]" title="写作">
            <input type="checkbox" name="like[read]" title="阅读">
            <input type="checkbox" name="like[daze]" title="发呆">
        </div>
    </div>

    <div class="layui-form-item">
        <label class="layui-form-label">开关</label>
        <div class="layui-input-block">
            <input type="checkbox" name="close" lay-skin="switch" lay-
text="ON|OFF">
        </div>
    </div>

    <div class="layui-form-item">
        <label class="layui-form-label">单选框</label>
        <div class="layui-input-block">
            <input type="radio" name="sex" value="男" title="男"
checked="">
            <input type="radio" name="sex" value="女" title="女">
        </div>
    </div>
    <div class="layui-form-item layui-form-text">
        <label class="layui-form-label">文本域</label>
        <div class="layui-input-block">
            <textarea placeholder="请输入内容" class="layui-textarea"
name="desc"></textarea>

```

```

        </div>
    </div>

    <div class="layui-form-item">
        <div class="layui-input-block">
            <button class="layui-btn" lay-submit="" lay-filter="demo1">立即提交</button>
        </div>
    </div>
</form>
<fieldset class="layui-elem-field layui-field-title" style="margin-top: 50px;">
    <legend>方框风格的表单集合</legend>
</fieldset>
<form class="layui-form layui-form-pane" action="">
    <div class="layui-form-item">
        <label class="layui-form-label">长输入框</label>
        <div class="layui-input-block">
            <input type="text" name="title" autocomplete="off"
placeholder="请输入标题" class="layui-input">
        </div>
    </div>
    <div class="layui-form-item">
        <label class="layui-form-label">短输入框</label>
        <div class="layui-input-inline">
            <input type="text" name="username" lay-verify="required"
placeholder="请输入" autocomplete="off" class="layui-input">
        </div>
    </div>

    <div class="layui-form-item">
        <div class="layui-inline">
            <label class="layui-form-label">日期选择</label>
            <div class="layui-input-block">
                <input type="text" name="date" id="date1"
autocomplete="off" class="layui-input">
            </div>
        </div>
        <div class="layui-inline">
            <label class="layui-form-label">行内表单</label>
            <div class="layui-input-inline">
                <input type="text" name="number" autocomplete="off"
class="layui-input">
            </div>
        </div>
    </div>
    <div class="layui-form-item">
        <label class="layui-form-label">密码</label>
        <div class="layui-input-inline">
            <input type="password" name="password" placeholder="请输入密码"
autocomplete="off" class="layui-input">
        </div>
        <div class="layui-form-mid layui-word-aux">请务必填写用户名</div>
    </div>

    <div class="layui-form-item">
        <div class="layui-inline">
            <label class="layui-form-label">范围</label>

```

```

        <div class="layui-input-inline" style="width: 100px;">
            <input type="text" name="price_min" placeholder="¥"
autocomplete="off" class="layui-input">
        </div>
        <div class="layui-form-mid"></div>
        <div class="layui-input-inline" style="width: 100px;">
            <input type="text" name="price_max" placeholder="¥"
autocomplete="off" class="layui-input">
        </div>
    </div>
</div>

<div class="layui-form-item">
    <label class="layui-form-label">单行选择框</label>
    <div class="layui-input-block">
        <select name="interest" lay-filter="aihao">
            <option value=""></option>
            <option value="0">写作</option>
            <option value="1" selected="">阅读</option>
            <option value="2">游戏</option>
            <option value="3">音乐</option>
            <option value="4">旅行</option>
        </select>
    </div>
</div>

<div class="layui-form-item">
    <label class="layui-form-label">行内选择框</label>
    <div class="layui-input-inline">
        <select name="quiz1">
            <option value="">请选择省</option>
            <option value="浙江" selected="">浙江省</option>
            <option value="你的工号">江西省</option>
            <option value="你最喜欢的老师">福建省</option>
        </select>
    </div>
    <div class="layui-input-inline">
        <select name="quiz2">
            <option value="">请选择市</option>
            <option value="杭州">杭州</option>
            <option value="宁波" disabled="">宁波</option>
            <option value="温州">温州</option>
            <option value="温州">台州</option>
            <option value="温州">绍兴</option>
        </select>
    </div>
    <div class="layui-input-inline">
        <select name="quiz3">
            <option value="">请选择县/区</option>
            <option value="西湖区">西湖区</option>
            <option value="余杭区">余杭区</option>
            <option value="拱墅区">临安市</option>
        </select>
    </div>
</div>

<div class="layui-form-item" pane="">
    <label class="layui-form-label">开关-开</label>
    <div class="layui-input-block">

```



```

        <input type="checkbox" checked="" name="open" lay-
skin="switch" lay-filter="switchTest" title="开关">
    </div>
</div>
<div class="layui-form-item" pane="">
    <label class="layui-form-label">单选框</label>
    <div class="layui-input-block">
        <input type="radio" name="sex" value="男" title="男"
checked="">
        <input type="radio" name="sex" value="女" title="女">
        <input type="radio" name="sex" value="禁" title="禁用"
disabled="">
    </div>
</div>
<div class="layui-form-item layui-form-text">
    <label class="layui-form-label">文本域</label>
    <div class="layui-input-block">
        <textarea placeholder="请输入内容" class="layui-textarea">
</textarea>
    </div>
</div>
<div class="layui-form-item">
    <button class="layui-btn" lay-submit="" lay-filter="demo2">跳转式
提交</button>
</div>
</form>
</div>
</div>

<script src="../../lib/layui-v2.5.5/layui.js" charset="utf-8"></script>
<!-- 注意：如果你直接复制所有代码到本地，上述js路径需要改成你本地的 -->
<script>
    layui.use(['form', 'layedit', 'laydate'], function () {
        var form = layui.form
            , layer = layui.layer
            , layedit = layui.layedit
            , laydate = layui.laydate;

        //日期
        laydate.render({
            elem: '#date'
        });
        laydate.render({
            elem: '#date1'
        });

        //创建一个编辑器
        var editIndex = layedit.build('LAY_demo_editor');

        //自定义验证规则
        form.verify({
            title: function (value) {
                if (value.length < 5) {
                    return '标题至少得5个字符啊';
                }
            }
        }, pass: [
            /^[\S]{6,12}$/

```

```

        , '密码必须6到12位，且不能出现空格'
    ]
    , content: function (value) {
        layeredit.sync(editIndex);
    }
});

//监听指定开关
form.on('switch(switchTest)', function (data) {
    layer.msg('开关checked: ' + (this.checked ? 'true' : 'false'), {
        offset: '6px'
    });
    layer.tips('温馨提示：请注意开关状态的文字可以随意定义，而不仅仅是ON|OFF',
data.othis)
});

//监听提交
form.on('submit(demo1)', function (data) {
    layer.alert(JSON.stringify(data.field), {
        title: '最终的提交信息'
    })
    return false;
});

//表单初始赋值
form.val('example', {
    "username": "贤心" // "name": "value"
    , "password": "123456"
    , "interest": 1
    , "like[write]": true //复选框选中状态
    , "close": true //开关状态
    , "sex": "女"
    , "desc": "我爱 layui"
});

});
</script>

</body>
</html>

```

list.html

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>layui</title>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
    <link rel="stylesheet" href="lib/layui-v2.5.5/css/layui.css" media="all">
    <link rel="stylesheet" href="css/public.css" media="all">
</head>
<body>

```

```

<div class="layuiimini-container">
  <div class="layuiimini-main">

    <fieldset class="table-search-fieldset">
      <legend>搜索信息</legend>
      <div style="margin: 10px 10px 10px 10px">
        <form class="layui-form layui-form-pane" action="">
          <div class="layui-form-item">
            <div class="layui-inline">
              <label class="layui-form-label">用户姓名</label>
              <div class="layui-input-inline">
                <input type="text" name="username"
autocomplete="off" class="layui-input">
              </div>
            </div>
            <div class="layui-inline">
              <label class="layui-form-label">用户性别</label>
              <div class="layui-input-inline">
                <input type="text" name="sex" autocomplete="off"
class="layui-input">
              </div>
            </div>
            <div class="layui-inline">
              <label class="layui-form-label">用户城市</label>
              <div class="layui-input-inline">
                <input type="text" name="city"
autocomplete="off" class="layui-input">
              </div>
            </div>
            <div class="layui-inline">
              <label class="layui-form-label">用户职业</label>
              <div class="layui-input-inline">
                <input type="text" name="classify"
autocomplete="off" class="layui-input">
              </div>
            </div>
            <div class="layui-inline">
              <button type="submit" class="layui-btn layui-btn-
primary" lay-submit lay-filter="data-search-btn"><i class="layui-icon">□</i> 搜
索</button>
            </div>
          </div>
        </form>
      </div>
    </fieldset>

    <script type="text/html" id="toolbarDemo">
      <div class="layui-btn-container">
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-
btn" lay-event="add"> 添加 </button>
        <button class="layui-btn layui-btn-sm layui-btn-danger data-
delete-btn" lay-event="delete"> 删除 </button>
      </div>
    </script>

    <table class="layui-hide" id="currentTableId" lay-
filter="currentTableFilter"></table>

```

```

        <script type="text/html" id="currentTableBar">
            <a class="layui-btn layui-btn-normal layui-btn-xs data-count-edit"
lay-event="edit">编辑</a>
            <a class="layui-btn layui-btn-xs layui-btn-danger data-count-delete"
lay-event="delete">删除</a>
        </script>

    </div>
</div>
<script src="lib/layui-v2.5.5/layui.js" charset="utf-8"></script>
<script>
    layui.use(['form', 'table'], function () {
        var $ = layui.jquery,
            form = layui.form,
            table = layui.table;

        table.render({
            elem: '#currentTableId',
            url: 'data/table.json',
            toolbar: '#toolbarDemo',
            defaultToolbar: ['filter', 'exports', 'print', {
                title: '提示',
                layEvent: 'LAYTABLE_TIPS',
                icon: 'layui-icon-tips'
            }],
            cols: [[
                {type: "checkbox", width: 50},
                {field: 'id', width: 80, title: 'ID', sort: true},
                {field: 'username', width: 80, title: '用户名'},
                {field: 'sex', width: 80, title: '性别', sort: true},
                {field: 'city', width: 80, title: '城市'},
                {field: 'sign', title: '签名', minwidth: 150},
                {field: 'experience', width: 80, title: '积分', sort: true},
                {field: 'score', width: 80, title: '评分', sort: true},
                {field: 'classify', width: 80, title: '职业'},
                {field: 'wealth', width: 135, title: '财富', sort: true},
                {title: '操作', minwidth: 150, toolbar: '#currentTableBar',
align: "center"}
            ]],
            limits: [10, 15, 20, 25, 50, 100],
            limit: 15,
            page: true,
            skin: 'line'
        });

        // 监听搜索操作
        form.on('submit(data-search-btn)', function (data) {
            var result = JSON.stringify(data.field);
            layer.alert(result, {
                title: '最终的搜索信息'
            });

            //执行搜索重载
            table.reload('currentTableId', {
                page: {
                    curr: 1
                }
            }, where: {

```

```

        searchParams: result
    }, 'data');

    return false;
});

/**
 * toolbar监听事件
 */
table.on('toolbar(currentTableFilter)', function (obj) {
    if (obj.event === 'add') { // 监听添加操作
        var index = layer.open({
            title: '添加用户',
            type: 2,
            shade: 0.2,
            maxmin: true,
            shadeClose: true,
            area: ['100%', '100%'],
            content: '../page/table/add.html',
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
    } else if (obj.event === 'delete') { // 监听删除操作
        var checkStatus = table.checkStatus('currentTableId')
            , data = checkStatus.data;
        layer.alert(JSON.stringify(data));
    }
});

//监听表格复选框选择
table.on('checkbox(currentTableFilter)', function (obj) {
    console.log(obj)
});

table.on('tool(currentTableFilter)', function (obj) {
    var data = obj.data;
    if (obj.event === 'edit') {

        var index = layer.open({
            title: '编辑用户',
            type: 2,
            shade: 0.2,
            maxmin: true,
            shadeClose: true,
            area: ['100%', '100%'],
            content: '../page/table/edit.html',
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
        return false;
    } else if (obj.event === 'delete') {
        layer.confirm('真的删除行么', function (index) {
            obj.del();
            layer.close(index);
        });
    }
});

```

```

    }
  });

});
</script>

</body>
</html>

```

login.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>后台管理-登陆</title>
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta http-equiv="Access-Control-Allow-Origin" content="*">
  <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="format-detection" content="telephone=no">
  <link rel="stylesheet" href="lib/layui-v2.5.5/css/layui.css" media="all">
  <!--[if lt IE 9]>
  <script src="js/html5.min.js"></script>
  <script src="js/respond.min.js"></script>
  <![endif]-->
  <style>
    html, body {width: 100%;height: 100%;overflow: hidden}
    body {background: #1E9FFF; background: url('images/login_bg.jpg')}
    body:after {content: '';background-repeat:no-repeat;background-
size:cover;-webkit-filter:blur(3px);-moz-filter:blur(3px);-o-filter:blur(3px);-
ms-
filter:blur(3px);filter:blur(3px);position:absolute;top:0;left:0;right:0;bottom:
0;z-index:-1;}
    .layui-container {width: 100%;height: 100%;overflow: hidden}
    .admin-login-background
{width:360px;height:300px;position:absolute;left:50%;top:40%;margin-
left:-180px;margin-top:-100px;}
    .logo-title {text-align:center;letter-spacing:2px;padding:14px 0;}
    .logo-title h1 {color:#1E9FFF;font-size:25px;font-weight:bold;}
    .login-form {background-color:#fff;border:1px solid #fff;border-
radius:3px;padding:14px 20px;box-shadow:0 0 8px #eeeeee;}
    .login-form .layui-form-item {position:relative;}
    .login-form .layui-form-item label
{position:absolute;left:1px;top:1px;width:38px;line-height:36px;text-
align:center;color:#d2d2d2;}
    .login-form .layui-form-item input {padding-left:36px;}
    .captcha {width:60%;display:inline-block;}
    .captcha-img {display:inline-block;width:34%;float:right;}
    .captcha-img img {height:34px;border:1px solid
#e6e6e6;height:36px;width:100%;}
  </style>
</head>
<body>
<div class="layui-container">

```

```

<div class="admin-login-background">
  <div class="layui-form login-form">
    <form class="layui-form" action="">
      <div class="layui-form-item logo-title">
        <h1>调查问卷系统登录</h1>
      </div>
      <div class="layui-form-item">
        <label class="layui-icon layui-icon-username"
for="username"></label>
        <input type="text" name="username" lay-
verify="required|account" placeholder="用户名或者邮箱" autocomplete="off"
class="layui-input" value="admin">
      </div>
      <div class="layui-form-item">
        <label class="layui-icon layui-icon-password"
for="password"></label>
        <input type="password" name="password" lay-
verify="required|password" placeholder="密码" autocomplete="off" class="layui-
input" value="123456">
      </div>
      <div class="layui-form-item">
        <button class="layui-btn layui-btn layui-btn-normal layui-
btn-fluid" lay-submit="" lay-filter="login">登 入</button>
      </div>
    </form>
  </div>
</div>
<script src="lib/jquery-3.4.1/jquery-3.4.1.min.js" charset="utf-8"></script>
<script src="lib/layui-v2.5.5/layui.js" charset="utf-8"></script>

<script>
  layui.use(['form'], function () {
    var form = layui.form,
        layer = layui.layer;

    // 登录过期的时候，跳出iframe框架
    if (top.location != self.location) top.location = self.location;

    // 进行登录操作
    form.on('submit(login)', function (data) {
      data = data.field;
      if (data.username == '') {
        layer.msg('用户名不能为空');
        return false;
      }
      if (data.password == '') {
        layer.msg('密码不能为空');
        return false;
      }
      if (data.captcha == '') {
        layer.msg('验证码不能为空');
        return false;
      }
      layer.msg('登录成功', function () {
        window.location = '../index.html';
      });
    });
  });

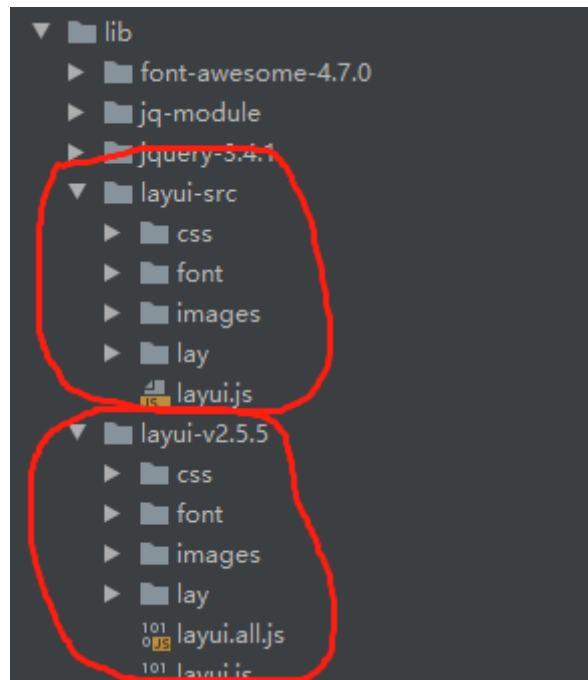
```

```

    });
    return false;
  });
});
</script>
</body>
</html>

```

因为登陆时有登录动画，layuimini的源代码无法查看，所以lib中新建layui-src文件夹，将layui的源代码lib下的layui内容复制进来



同时 修改 login、list、add、index页面的引用

```

<link rel="stylesheet" href="lib/layui-src/css/layui.css" media="all">

```

```

<script src="lib/layui-src/layui.js" charset="utf-8"></script>

```

8 页面信息整合

在survey项目中 webapp下新建static文件夹

css	2021/2/24 星期三 21:...	文件夹
data	2021/2/24 星期三 21:...	文件夹
images	2021/2/24 星期三 21:...	文件夹
js	2021/2/24 星期三 21:...	文件夹
lib	2021/3/1 星期一 11:02	文件夹
add.html	2020/6/10 星期三 10:...	360 se HTI
index.html	2020/6/10 星期三 10:...	360 se HTI
list.html	2020/6/10 星期三 10:...	360 se HTI
login.html	2020/6/10 星期三 11:...	360 se HTI

把以上文件复制进static文件夹中

10.前端与后端界面整合

1 放行静态资源

因为web.xml中前端控制器把静态资源请求拦截下来，所以需要对静态资源放行

在spring-mvc.xml中加上

```
<!--放行静态资源-->
<mvc:default-servlet-handler/>
```

2 同时web.xml中增加对静态资源的处理

```
<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>*.js</url-pattern>
  <url-pattern>*.css</url-pattern>
  <url-pattern>/assets/*</url-pattern>
  <url-pattern>/images/*</url-pattern>
</servlet-mapping>
```

3 创建LoginController.class

在controller文件夹下

```
package com.yanzhen.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class LoginController {

    @GetMapping("/login")
    public String login(){
        return "login";
    }

}
```

4 在spring-mvc.xml中配置视图解析器

即页面跳转的路径

```
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="prefix" value="/WEB-INF/page/"></property>
  <property name="suffix" value=".jsp"></property>
</bean>
```

5 SpringMVC 拦截器不拦截静态资源的三种处理方式方法

<https://www.cnblogs.com/mophy/p/8465598.html>

方案一、拦截器中增加针对静态资源不进行过滤(涉及spring-mvc.xml)



```
<mvc:resources location="/" mapping="/**/*.js"/>
<mvc:resources location="/" mapping="/**/*.css"/>
<mvc:resources location="/assets/" mapping="/assets/**/*.*/>
<mvc:resources location="/images/" mapping="/images/*" cache-period="360000"/>

<mvc:interceptors>
  <mvc:interceptor>
    <mvc:mapping path="/**/*.*/>
    <mvc:exclude-mapping path="/**/*.fonts/**/*.*/>
    <mvc:exclude-mapping path="/**/*.css"/>
    <mvc:exclude-mapping path="/**/*.js"/>
    <mvc:exclude-mapping path="/**/*.png"/>
    <mvc:exclude-mapping path="/**/*.gif"/>
    <mvc:exclude-mapping path="/**/*.jpg"/>
    <mvc:exclude-mapping path="/**/*.jpeg"/>
    <mvc:exclude-mapping path="/**/*.login**"/>
    <mvc:exclude-mapping path="/**/*.Login**"/>
    <bean class="com.luwei.console.mg.interceptor.visitInterceptor"></bean>
  </mvc:interceptor>
</mvc:interceptors>
```

方案二、使用默认的静态资源处理Servlet处理静态资源(涉及spring-mvc.xml, web.xml)

1、在spring-mvc.xml中启用默认Servlet

```
<mvc:default-servlet-handler/>
```

2、在web.xml中增加对静态资源的处理



```
<servlet-mapping>
  <servlet-name>default</servlet-name>
  <url-pattern>*.js</url-pattern>
  <url-pattern>*.css</url-pattern>
  <url-pattern>/assets/**</url-pattern>
  <url-pattern>/images/**</url-pattern>
</servlet-mapping>
```



但是当前的设置必须在Spring的Dispatcher的前面

方案三、修改Spring的全局拦截设置为*.do的拦截（涉及web.xml）

```
<servlet>
  <servlet-name>SpringMVC</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:spring-mvc.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
  <async-supported>true</async-supported>
</servlet>
<servlet-mapping>
  <servlet-name>SpringMVC</servlet-name>
  <url-pattern>*.do</url-pattern>
</servlet-mapping>
```

这样设置，Spring就会只针对以'.do'结尾的请求进行处理，不再维护静态资源

针对这三种方案的优劣分析：

第一种方案配置比较臃肿，多个拦截器时增加文件行数，不推荐使用；第二种方案使用默认的Servlet进行资源文件的访问，Spring拦截所有请求，然后再将资源文件交由默认的Servlet进行处理，性能上少有损耗；第三种方案Spring只是处理以'.do'结尾的访问，性能上更加高效，但是再访问路径上必须都以'.do'结尾，URL不太文雅；

综上所述，推荐使用第二和第三中方案

11.登录功能开发

11.1 登录

1 新建login.jsp

把login.html中的内容复制过来，保留jsp的第一行，修改src :static/....、href:static/....和background: url('static/images/login_bg.jpg'。jsp中url路径分割线为'/'！！

2 新建index.jsp

为了使项目启动后自动跳转到login.jsp页面，在webapp下创建一个index.jsp.如下：

重定向到login

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
response.sendRedirect("login");
%>
```

3 在web.xml中设置欢迎页为index.jsp

```
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

4 表单提交两种方式

form表单的形式提交:

POST、GET提交

采用Ajax的形式提交

方式一 application-www.默认提交方式form表单的

方式二 application/json: json字符串对象

前端提交的是json对象的字符串, 后端如何转换为对象或map对象呢?

```
@PostMapping("/login")
public String login(@RequestBody Map<String,Object> map){
    System.out.println(map);
    return "login";
}
```

方式三 mutip...文件上传的

5 编写loginController.class

```
package com.yanzhen.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;

import java.util.Map;

@Controller
public class LoginController {

    @GetMapping("/login")
    public String v_login(){
        return "login";
    }

    @PostMapping("/login")
    @ResponseBody
    public String login(@RequestBody Map<String,Object> map){
```

```

        System.out.println(map);
        return "login";
    }
}

```

6 编写ajax登录提交 login.jsp

data：从前端传送回来的数据

url：请求路径

type：请求方式

data：数据 json转string，请求参数

contentType：要求为String类型的参数，当发送信息至服务器时，内容编码类型默认为
"application/x-www- form-urlencoded"。该默认值适合大多数应用场合

dataType：设置接收到相应数据的格式

success：要求为Function类型的参数，请求成功后调用的回调函数，有两个参数。

(1)由服务器返回，并根据dataType参数进行处理后的数据。

(2)描述状态的字符串

```

$.ajax({
    url:'login',
    type:'POST',
    data:JSON.stringify(data),
    contentType:'application/json',
    dataType:'json',
    success:function(data){
        if(data.code == 0){
            location.href = "index";
        }else{
            layer.msg(data.msg);
        }
    }
})

```

login.jsp

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>后台管理-登陆</title>
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta http-equiv="Access-Control-Allow-Origin" content="*">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
    <meta name="apple-mobile-web-app-status-bar-style" content="black">
    <meta name="apple-mobile-web-app-capable" content="yes">

```

```

<meta name="format-detection" content="telephone=no">
<link rel="stylesheet" href="static/lib/layui-src/css/layui.css"
media="all">
<!--[if lt IE 9]>
<script src="js/html5.min.js"></script>
<script src="js/respond.min.js"></script>
<![endif]-->
<style>
    html, body {width: 100%;height: 100%;overflow: hidden}
    body {background: url("static/images/login_bg.jpg");}
    body:after {content: '';background-repeat:no-repeat;background-
size:cover;-webkit-filter:blur(3px);-moz-filter:blur(3px);-o-filter:blur(3px);-
ms-
filter:blur(3px);filter:blur(3px);position:absolute;top:0;left:0;right:0;bottom:
0;z-index:-1;}
    .layui-container {width: 100%;height: 100%;overflow: hidden}
    .admin-login-background
{width:360px;height:300px;position:absolute;left:50%;top:40%;margin-
left:-180px;margin-top:-100px;}
    .logo-title {text-align:center;letter-spacing:2px;padding:14px 0;}
    .logo-title h1 {color:#1E9FFF;font-size:25px;font-weight:bold;}
    .login-form {background-color:#fff;border:1px solid #fff;border-
radius:3px;padding:14px 20px;box-shadow:0 0 8px #eeeeee;}
    .login-form .layui-form-item {position:relative;}
    .login-form .layui-form-item label
{position:absolute;left:1px;top:1px;width:38px;line-height:36px;text-
align:center;color:#d2d2d2;}
    .login-form .layui-form-item input {padding-left:36px;}
    .captcha-img img {height:34px;border:1px solid
#e6e6e6;height:36px;width:100%;}
</style>
</head>
<body>
<div class="layui-container">
    <div class="admin-login-background">
        <div class="layui-form login-form">
            <form class="layui-form" action="">
                <div class="layui-form-item logo-title">
                    <h1>调查问卷系统登录</h1>
                </div>
                <div class="layui-form-item">
                    <label class="layui-icon layui-icon-username"></label>
                    <input type="text" name="account" lay-
verify="required|account" placeholder="请录入账号" autocomplete="off"
class="layui-input" value="admin">
                </div>
                <div class="layui-form-item">
                    <label class="layui-icon layui-icon-password"></label>
                    <input type="password" name="password" lay-
verify="required|password" placeholder="请录入密码" autocomplete="off"
class="layui-input" value="123456">
                </div>
                <div class="layui-form-item">
                    <button class="layui-btn layui-btn-normal layui-
btn-fluid" lay-submit="" lay-filter="login">登 入</button>
                </div>
            </form>
        </div>
    </div>

```

```

    </div>
</div>
<script src="static/lib/jquery-3.4.1/jquery-3.4.1.min.js" charset="utf-8">
</script>
<script src="static/lib/layui-src/layui.js" charset="utf-8"></script>
<script>
    layui.use(['form','jquery'], function () {
        var form = layui.form,
            layer = layui.layer,
            $ = layui.jquery;
        // 登录过期的时候, 跳出iframe框架
        if (top.location != self.location) top.location = self.location;
        // 进行登录操作
        form.on('submit(login)', function (data) {
            data = data.field;
            if (data.username == '') {
                layer.msg('用户名不能为空');
                return false;
            }
            if (data.password == '') {
                layer.msg('密码不能为空');
                return false;
            }
            $.ajax({
                url: 'login',
                type: 'POST',
                data: JSON.stringify(data),
                contentType: 'application/json',
                dataType: 'json',
                success: function (data) {
                    if (data.code == 0) {
                        location.href = "index";
                    } else {
                        layer.msg(data.msg);
                    }
                }
            })
            return false;
        });
    });
</script>
</body>
</html>

```

7 编写MD5Utils.class

用于密码加密,在utils文件夹下

```

package com.yanzhen.utils;

import org.springframework.util.DigestUtils;

public class MD5Utils {

    //盐

```

```

        private static final String salt = "Survey###$$@@";

        public static String getMD5(String string){
            String val = string+salt;
            return DigestUtils.md5DigestAsHex(val.getBytes()); //调用spring中的加密算法
        }

        public static void main(String[] args) {
            System.out.println(getMD5("123456"));
        }
    }
}

```

8 AdminService.class中加入登录方法

```

/**
 * 管理员登录
 * @param account
 * @param password
 * @return
 */
public Admin login(String account,String password){
    Map<String, Object> map =
    MapParameter.getInstance().add("account",account).add("password",password).getMap();
    return adminDao.detail(map);
}

```

9 login.jsp中 修改

用户名/邮箱的input标签 name='account'

```

</div>
<div class="layui-form-item">
    <label class="layui-icon layui-icon-username"></label>
    <input type="text" name="account" lay-verify="required|account" placeholder="用户名或者邮箱" autocomplete="off" class="layui-input">
</div>
<div class="layui-form-item">
    <label class="layui-icon layui-icon-password"></label>
    <input type="password" name="password" lay-verify="required|password" placeholder="密码" autocomplete="off" class="layui-input">
</div>

```

10 再次编写LoginService.class

```

@Controller
public class LoginController {

    @Autowired
    private AdminService adminService;

    @GetMapping("/login")
    public String v_login(){
        return "login";
    }

    @PostMapping("/login")

```

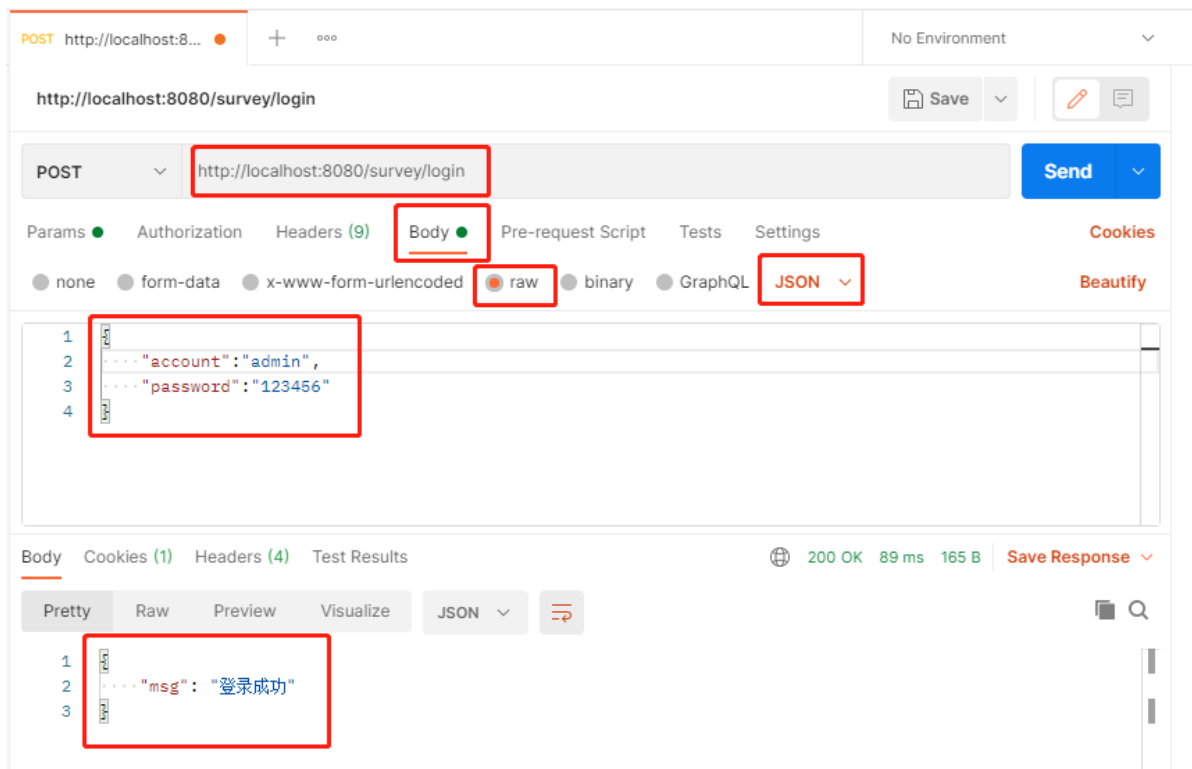


```

@ResponseBody
public Map<String, Object> login(@RequestBody Map<String, Object> map,
    HttpServletRequest request){
    String account=map.get("account")+"";
    String password=map.get("password")+"";
    if(Strings.isNullOrEmpty(account)||Strings.isNullOrEmpty(password)){
        return MapParameter.getInstance().put("error", "用户名或密码不能为空").getMap();
    }
    Admin admin =adminService.login(account, MD5Utils.getMD5(password));
    HttpSession session=request.getSession();
    session.setAttribute("admin",admin);
    return MapParameter.getInstance().put("msg", "登录成功").getMap();
}
}

```

11 打开postman 进行测试



POST http://localhost:8080/survey/login

Save

POST http://localhost:8080/survey/login Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "account": "",
3   "password": "123456"
4 }
```

Body Cookies (1) Headers (4) Test Results 200 OK 14 ms 185 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "error": "用户名或密码不能为空"
3 }
```

测试正常

12 用户名或密码输错 也报正常，发现后台校验出错

POST http://localhost:8080/survey/login

Save

POST http://localhost:8080/survey/login Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "account": "asdasdsa",
3   "password": "123456"
4 }
```

Body Cookies (1) Headers (4) Test Results 200 OK 26 ms 165 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "msg": "登录成功"
3 }
```

数据库信息

	id	account	password	name	phone	remark
1	admin	433db2fa87c8e880868dcf6ae9c2fca2	比克大魔王	18533333333		
*	NULL	NULL	NULL	NULL	NULL	NULL

13 修改后台校验 LoginController.class

```
@PostMapping("/login")
@ResponseBody
public Map<String, Object> login(@RequestBody Map<String, Object> map,
    HttpServletRequest request){
    String account=map.get("account")+"";
    String password=map.get("password")+"";
    if(Strings.isNullOrEmpty(account)||Strings.isNullOrEmpty(password)){
        return MapParameter.getInstance().put("error", "用户名或密码不能为
空").getMap();
    }
    Admin admin =adminService.login(account, MD5Utils.getMD5(password));
    if(admin!=null){
        HttpSession session=request.getSession();
        session.setAttribute("admin",admin);
        return MapParameter.getInstance().put("msg", "登录成功").getMap();
    }else {
        return MapParameter.getInstance().put("error", "用户名或密码错误").getMap();
    }
}
```

14 utils 文件夹下创建MapController.class

```
package com.yanzhen.utils;

import java.util.HashMap;
import java.util.Map;

public class MapController {

    /**
     * private static final int SUCCESS_CODE=200;
     * private static final int ERROR_CODE=500;
     * 方便前端进行分析
     */

    private static final int SUCCESS_CODE=200;

    private static final int ERROR_CODE=500;

    private Map<String, Object> paramMap = new HashMap<>();

    private MapController(){

    }

    public static MapController getInstance(){
        return new MapController();
    }

    public Map<String, Object> getMap(){
        return paramMap;
    }
}
```

```

    }

    public MapController put(Map<String, Object> map){
        for (Map.Entry<String, Object> entry : map.entrySet()) {
            paramMap.put(entry.getKey(), entry.getValue());
        }
        return this;
    }

    public MapController put(String key, Object val){
        paramMap.put(key, val);
        return this;
    }

    public MapController add(String key, Object val){
        return this.put(key, val);
    }

    public MapController success(){
        return this.put("msg", "操作成功").put("code", SUCCESS_CODE);
    }

    public MapController error(){
        return this.put("msg", "操作失败").put("code", ERROR_CODE);
    }

    public MapController success(String str){
        return this.put("msg", str).put("code", SUCCESS_CODE);
    }

    public MapController error(String str){
        return this.put("msg", str).put("code", ERROR_CODE);
    }

    public static void main(String[] args) {
        Map<String, Object> map = MapController.getInstance().put("name",
        "hua").put("age", 30).getMap();
        System.out.println(map);
    }
}

```

15 重新修改LoginController.class

```

@PostMapping("/login")
@ResponseBody
public Map<String, Object> login(@RequestBody Map<String, Object> map,
HttpServletRequest request){
    String account=map.get("account")+"";
    String password=map.get("password")+"";
    if(Strings.isNullOrEmpty(account)||Strings.isNullOrEmpty(password)){
        return MapController.getInstance().error("用户名或密码不能为空").getMap();
    }
    Admin admin =adminService.login(account, MD5Utils.getMD5(password));
}

```

```

        if(admin!=null){
            HttpSession session=request.getSession();
            session.setAttribute("admin",admin);
            return MapController.getInstance().success().getMap();
        }else {
            return MapController.getInstance().error("用户名或密码错误").getMap();
        }
    }
}

```

16 修改login.jsp

```

    }
    $.ajax({
        url:'login',
        type:'POST',
        data:JSON.stringify(data),
        contentType:'application/json',
        dataType:'json',
        success:function(data){
            if(data.code == 200){
                location.href = "index";
            }else{
                layer.msg(data.msg);
            }
        }
    })
}

```

11.2 首页菜单

1 编写IndexController.class

controller文件夹下的IndexController.class

```

@Controller
public class IndexController {
    @GetMapping("/index")
    public String index() {
        return "index";
    }
}

```

2 复制index.html 到index.jsp

3 保留index.jsp的第一行，同时修改src、href、iniUrl等的路径

4 删除logo的img标签

该行代码位置 src/main/webapp/static/js/lay-module/layuimini/miniAdmin.js

5 更改index.jsp中的标题

修改data/init.json 修改完以后 为了防止别人查看网页源代码，localhost:8080/survey/static/data/init.json，

直接得到json数据。因此把init.json放在Rescourse下面。

修改后的init.json为

```
{
  "homeInfo": {
    "title": "问卷管理",
    "href": "survey/list"
  },
  "logoInfo": {
    "title": "调查问卷",
    "image": "",
    "href": ""
  },
  "menuInfo": [

    {
      "title": "问卷管理",
      "icon": "fa fa-address-book",
      "href": "",
      "target": "_self",
      "child": [
        {
          "title": "问卷管理",
          "href": "survey/list",
          "icon": "fa fa-window-maximize",
          "target": "_self"
        },
        {
          "title": "问卷管理",
          "href": "survey/list",
          "icon": "fa fa-window-maximize",
          "target": "_self"
        },
        {
          "title": "类型管理",
          "href": "survey/list",
          "icon": "fa fa-window-maximize",
          "target": "_self"
        },
        {
          "title": "查询统计",
          "href": "survey/list",
          "icon": "fa fa-window-maximize",
          "target": "_self"
        }
      ]
    },
    {

```

```

        "title": "系统管理",
        "icon": "fa fa-slideshare",
        "href": "",
        "target": "_self",
        "child": [
            {
                "title": "管理员",
                "href": "admin/list",
                "icon": "fa fa-superpowers",
                "target": "_self"
            },
            {
                "title": "用户管理",
                "href": "admin/list",
                "icon": "fa fa-superpowers",
                "target": "_self"
            }
        ]
    }
}
]
}
}

```

同时IndexController.class加入

```

@Controller
public class IndexController {

    @Value("classpath:init.json")
    private Resource resource;

    @GetMapping("/index")
    public String index(){
        return "index";
    }

    @GetMapping("/menu")
    @ResponseBody
    public void menu(HttpServletResponse response) {
        try {
            File file = resource.getFile();
            FileReader fileReader = new FileReader(file);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            String str;
            StringBuffer sb = new StringBuffer();
            while ((str = bufferedReader.readLine()) != null){
                sb.append(str);
            }
            bufferedReader.close();
            fileReader.close();
            response.setContentType("application/json;charset=utf-8");
            response.getWriter().print(sb.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

```



```

        <div class="layui-inline">
            <label class="layui-form-label">账号</label>
            <div class="layui-input-inline">
                <input type="text" name="account"
autocomplete="off" class="layui-input">
            </div>
        </div>
        <div class="layui-inline">
            <label class="layui-form-label">姓名</label>
            <div class="layui-input-inline">
                <input type="text" name="name"
autocomplete="off" class="layui-input">
            </div>
        </div>
        <div class="layui-inline">
            <label class="layui-form-label">手机号码</label>
            <div class="layui-input-inline">
                <input type="text" name="phone"
autocomplete="off" class="layui-input">
            </div>
        </div>
        <div class="layui-inline">
            <button type="submit" class="layui-btn layui-btn-
primary" lay-submit lay-filter="data-search-btn"><i class="layui-icon">□</i> 搜
索</button>
        </div>
    </div>
</form>
</div>
</fieldset>

<script type="text/html" id="toolbarDemo">
    <div class="layui-btn-container">
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-
btn" lay-event="add"> 添加 </button>
        <button class="layui-btn layui-btn-sm layui-btn-danger data-
delete-btn" lay-event="delete"> 删除 </button>
    </div>
</script>

<table class="layui-hide" id="currentTableId" lay-
filter="currentTableFilter"></table>

<script type="text/html" id="currentTableBar">
    <a class="layui-btn layui-btn-normal layui-btn-xs data-count-edit"
lay-event="edit">编辑</a>
    <a class="layui-btn layui-btn-xs layui-btn-danger data-count-delete"
lay-event="delete">删除</a>
</script>

</div>
</div>
<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<script>
    layui.use(['form', 'table'], function () {
        var $ = layui.jquery,
            form = layui.form,
            table = layui.table;
    });

```

```

table.render({
  elem: '#currentTableId',
  url: '../static/data/table.json',
  toolbar: '#toolbarDemo',
  defaultToolbar: ['filter', 'exports', 'print'],
  cols: [[
    {type: "checkbox", width: 50},
    {field: 'id', width: 80, title: 'ID', sort: true},
    {field: 'username', width: 80, title: '用户名'},
    {field: 'sex', width: 80, title: '性别', sort: true},
    {field: 'city', width: 80, title: '城市'},
    {field: 'sign', title: '签名', minWidth: 150},
    {field: 'experience', width: 80, title: '积分', sort: true},
    {field: 'score', width: 80, title: '评分', sort: true},
    {field: 'classify', width: 80, title: '职业'},
    {field: 'wealth', width: 135, title: '财富', sort: true},
    {title: '操作', minWidth: 150, toolbar: '#currentTableBar',
align: "center"}
  ]],
  limits: [10, 15, 20, 25, 50, 100],
  limit: 15,
  page: true,
  skin: 'line'
});

// 监听搜索操作
form.on('submit(data-search-btn)', function (data) {
  var result = JSON.stringify(data.field);
  layer.alert(result, {
    title: '最终的搜索信息'
  });

  //执行搜索重载
  table.reload('currentTableId', {
    page: {
      curr: 1
    }
    , where: {
      searchParams: result
    }
  }, 'data');

  return false;
});

/**
 * toolbar监听事件
 */
table.on('toolbar(currentTableFilter)', function (obj) {
  if (obj.event === 'add') { // 监听添加操作
    var index = layer.open({
      title: '添加用户',
      type: 2,
      shade: 0.2,
      maxmin: true,
      shadeClose: true,
      area: ['100%', '100%'],

```

```

        content: '../page/table/add.html',
    });
    $(window).on("resize", function () {
        layer.full(index);
    });
    } else if (obj.event === 'delete') { // 监听删除操作
        var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
        layer.alert(JSON.stringify(data));
    }
    });

//监听表格复选框选择
table.on('checkbox(currentTableFilter)', function (obj) {
    console.log(obj)
});

table.on('tool(currentTableFilter)', function (obj) {
    var data = obj.data;
    if (obj.event === 'edit') {

        var index = layer.open({
            title: '编辑用户',
            type: 2,
            shade: 0.2,
            maxmin: true,
            shadeClose: true,
            area: ['100%', '100%'],
            content: '../page/table/edit.html',
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
        return false;
    } else if (obj.event === 'delete') {
        layer.confirm('真的删除行么', function (index) {
            obj.del();
            layer.close(index);
        });
    }
});

});
</script>

</body>
</html>

```

2 AdminController.class中加入方法用从index.jsp的用户管理跳转

```

@Controller
@RequestMapping("/admin")
public class AdminController {

```

```

.....
@GetMapping("/list")
public String list(){
    return "admin/list"; //经过视图解析器跳转到webapp下page\admin\list.jsp
}

.....
}

```

同时修改init.json 的href为admin/list

```

{
  "title": "系统管理",
  "icon": "fa fa-slideshare",
  "href": "",
  "target": "_self",
  "child": [
    {
      "title": "用户管理",
      "href": "admin/list",
      "icon": "fa fa-superpowers",
      "target": "_self"
    },
    {
      "title": "日志管理",
      "href": "",
      "icon": "fa fa-superpowers",
      "target": "_self"
    }
  ]
}

```

3 修改MapController.class

```

package com.yanzhen.utils;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class MapControl {

    private static final int SUCCESS_CODE = 0; //修改为0
    private static final int ERROR_CODE = 500;

    private Map<String, Object> paramMap = new HashMap<>();

    private MapControl(){

    }
}

```

```

public static MapControl getInstance(){
    return new MapControl();
}

public Map<String, Object> getMap(){
    return paramMap;
}

public MapControl put(Map<String, Object> map){
    for (Map.Entry<String, Object> entry : map.entrySet()) {
        paramMap.put(entry.getKey(), entry.getValue());
    }
    return this;
}

public MapControl put(String key, Object val){
    paramMap.put(key, val);
    return this;
}

public MapControl add(String key, Object val){
    return this.put(key, val);
}

public MapControl success(){
    return this.put("msg", "操作成功").put("code", SUCCESS_CODE);
}

public MapControl error(){
    return this.put("msg", "操作失败").put("code", ERROR_CODE);
}

public MapControl success(String str){
    return this.put("msg", str).put("code", SUCCESS_CODE);
}

public MapControl error(String str){
    return this.put("msg", str).put("code", ERROR_CODE);
}

//新增方法
public MapControl page(List<?> data, Integer count){
    return this.put("msg", "返回成功").put("code", SUCCESS_CODE).put("count", count).put("data", data);
}

public MapControl page(List<?> data, Integer count, int code){
    return this.put("msg", "返回成功").put("code", code).put("count", count).put("data", data);
}

public static void main(String[] args) {
    Map<String, Object> map = MapControl.getInstance().put("name",
    "hua").put("age", 30).getMap();
    System.out.println(map);
}

```

```
}
```

因为SUCCESS_CODE 改变了 login.jsp 对应修改 data.code==0

```
$.ajax({
    url: 'login',
    type: 'POST',
    data: JSON.stringify(data),
    contentType: 'application/json',
    dataType: 'json',
    success: function(data){
        if(data.code == 0){
            location.href = "index";
        }else{
            layer.msg(data.msg);
        }
    }
})
})
```

4 修改AdminController.class

```
@PostMapping("/query")
@ResponseBody
public Map<String, Object> query(Admin admin, ModelMap modelMap){
    List<Admin> list=adminService.query(admin);
    Integer count=adminService.count(admin);
    return MapController.getInstance().page(list, count).getMap();
}
```

5 调试错误 修改list.jsp 的 table.render

原始

```
table.render({
    elem: '#currentTableId',
    url: '../static/data/table.json',
    toolbar: '#toolbarDemo',
    defaultToolbar: ['filter', 'exports', 'print'],
    cols: [[
        {type: "checkbox", width: 50},
        {field: 'id', width: 80, title: 'ID', sort: true},
        {field: 'username', width: 80, title: '用户名'},
        {field: 'sex', width: 80, title: '性别', sort: true},
        {field: 'city', width: 80, title: '城市'},
        {field: 'sign', title: '签名', minWidth: 150},
        {field: 'experience', width: 80, title: '积分', sort: true},
        {field: 'score', width: 80, title: '评分', sort: true},
        {field: 'classify', width: 80, title: '职业'},
        {field: 'wealth', width: 135, title: '财富', sort: true},
        {title: '操作', minWidth: 150, toolbar: '#currentTableBar', align: "center"}
    ]],
    limits: [10, 15, 20, 25, 50, 100],
    limit: 15,
    page: true,
    skin: 'line'
});
```

修改后

```
table.render({
  elem: '#currentTableId',
  url: 'query',
  toolbar: '#toolbarDemo',
  contentType: 'application/json',
  method: "post",
  defaultToolbar: ['filter', 'exports', 'print'],
  cols: [[
    {type: "checkbox", width: 50},
    {field: 'id', width: 80, title: 'ID', sort: true},
    {field: 'account', width: 180, title: '用户名'},
    {field: 'name', width: 280, title: '姓名', sort: true},
    {field: 'phone', width: 280, title: '手机'},
    {field: 'remark', title: '备注', minWidth: 250}
  ]],
  limits: [10, 15, 20, 25, 50, 100],
  limit: 15,
  page: true,
  skin: 'line'
});
```

现在出现问题了,

错误1 无法登录, 密码错误, 密码、账号实际无误

错误2 用户管理查询不到数据

重新理一下, 登录逻辑

项目启动 ----- 浏览器输入http://localhost:8080/survey/ -----
 ----- 跳转到webapp\index.jsp ----- 转发到 LoginController.java
 的 v_login()方法 ----- 经过视图解析器
 http://localhost:8080/survey/login ----- 跳转到WEB-INF/page/login.jsp ----- 输入 账号、密码，点击提交 -----
 ----- 通过Ajax (url : login type:POST) 将account、password 转变成map -----
 ----- 提交到 LoginController.java 的 login(map , request)方法 -----
 -----通过测试发现 **Admin admin =adminService.login(account, MD5Utils.getMD5(password));** 这里查不到信息 ----- 查找AdminService.java
 的login()方法 ----- 对该方法进行测试 ----- **return
 adminDao.detail(map)** 查到的信息为null----- 查看adminDao的detail () 相关程序
 AdminDao接口、AdminMapper.xml -----没有错误-----查看数据库-----数据
 库tb_admin表中没有管理员信息-----手动加入数据-----错误
 1 解决-----return adminDao.detail(map); 从数据库中查询到登录信息 -----
 ----- 将admin对象传到session中-----封装成map返回到前端WEB-INF/page/login.jsp页面-----通过Ajax的success()函数 location.href =
 "index" -----跳转到 IndexController.java 的index()方法-----
 -----return " index " ----- 跳转到
 http://localhost:8080/survey/index 页面-----读取options= iniUrl:"
 menu" -----跳转到IndexController.java的 menu()方法 -----将
 init.json中的内容读取出来 返回到前端 index.jsp页面 -----点击系统管理中的用户管理---
 -----init.jsp中的href : admin/list -----跳转到
 AdminController.java 中的list()方法-----return "admin/list";-----
 -----跳转到admin/list页面 (没有@ResponseBody 注解的 都跳转到相应页面) -----
 -----通过table.render的url: query , method: post -----跳转到
 AdminController.java的query()方法-----查询tb_admin表中所有信息 -----
 -----提交到list.jsp页面上-----错误2 解决

6 注销搜索按钮跳提示框的方法

修改list.jsp

```
// 监听搜索操作
form.on('submit(data-search-btn)', function (data) {
    var result = JSON.stringify(data.field);
    //layer.alert(result, {
    //    title: '最终的搜索信息'
    //});
});
```

注销搜索按钮跳提示框的方法

7 插入数据到tb_admin

为了查看list中分页数据，手动直插入数据到tb_admin中

为了使id从1开始，先

You are using safe **update** mode and you tried to **update** a **table** without a **WHERE** that uses a KEY column To disable safe mode, toggle the option **in** Preferences -> SQL Queries and reconnect.

先将数据库设置一下，这是因为MySQL运行在safe-updates模式下，该模式会导致非主键条件下无法执行update或者delete命令

```
show variables like 'SQL_SAFE_UPDATES';  
SET SQL_SAFE_UPDATES = 0;
```

//设置完在输入以下

```
TRUNCATE survey.tb_admin ; //清空表
```

```
SELECT * FROM survey.tb_admin; //查询所有数据
```

```
insert into tb_admin(account,password) select account,password from tb_admin;  
//向表里插入原表中的数据 执行几次
```

```
update tb_admin set account=concat('admin',id) ; //将account和id拼接在一起 最后执行这个
```

```
update tb_admin set name=concat('管理员',id)  
,phone=concat("1325667",id),remark=concat("管理员",id);
```

执行结果

	id	account	password	name	phone	remark
▶	1	admin1	433db2fa87c8e880868dcf6ae9c2fca2	比克大魔王	132556756327183	管理员
	2	admin2	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	3	admin3	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	4	admin4	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	6	admin6	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	7	admin7	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	8	admin8	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	9	admin9	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	13	admin13	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	14	admin14	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	15	admin15	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	16	admin16	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	17	admin17	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL
	18	admin18	433db2fa87c8e880868dcf6ae9c2fca2	NULL	NULL	NULL

8 前端显示结果

查询条件					
账号		姓名		手机号码	
Q 搜索					

添加	删除	
----	----	--

<input type="checkbox"/>	ID	用户名	姓名	手机	备注
<input type="checkbox"/>	1	admin1	管理员1	13256671	管理员1
<input type="checkbox"/>	2	admin2	管理员2	13256672	管理员2
<input type="checkbox"/>	3	admin3	管理员3	13256673	管理员3
<input type="checkbox"/>	4	admin4	管理员4	13256674	管理员4
<input type="checkbox"/>	6	admin6	管理员6	13256676	管理员6
<input type="checkbox"/>	7	admin7	管理员7	13256677	管理员7
<input type="checkbox"/>	8	admin8	管理员8	13256678	管理员8
<input type="checkbox"/>	9	admin9	管理员9	13256679	管理员9
<input type="checkbox"/>	13	admin13	管理员13	132566713	管理员13
<input type="checkbox"/>	14	admin14	管理员14	132566714	管理员14

没有做分页 接下来做分页处理

9 分页处理

The screenshot shows a web application with a user management interface. The interface includes a search form with fields for '账号' (Account), '姓名' (Name), and '手机号码' (Mobile Number), and a '搜索' (Search) button. Below the search form is a table with columns: ID, 用户名 (Username), 姓名 (Name), and 手机 (Mobile). The table contains 14 rows of user data. The network tab shows a GET request to /survey/admin/list with a query parameter 'page=1, limit: 15' highlighted in red.

前端传过来的参数 limit和page

```

@PostMapping("/query")
@ResponseBody
public Map<String, Object> query(Admin admin, ModelMap modelMap) {
    System.out.println(admin);
    System.out.println(admin.getLimit() + " " + admin.getPage());
    List<Admin> list = adminService.query(admin);
    Integer count = adminService.count(admin);
    return MapControl.getInstance().page(list, count).getMap();
}

```

查询不到

找错 少写了一个

```

@ResponseBody
public Map<String, Object> query(@RequestBody Admin admin, ModelMap modelMap){
    System.out.println(admin);
    System.out.println(admin.getLimit()+" "+admin.getPage());
    List<Admin> list=adminService.query(admin);
    Integer count=adminService.count(admin);
    return MapControl.getInstance().page(list, count).getMap();
}

```

再运行就收到了

```

Admin{account='null', id=null, name='null', password='null', phone='null', remark='null', now=null}
15 1

```

10 组合查询

修改list.jsp

```

//执行搜索重载
table.reload('currentTableId', {
    page: {
        curr: 1
    },
    where: {
        "account":data.field.account,
        "name":data.field.name,
        "phone":data.field.phone
    }
}, 'data');

return false;
});

```

这样后端AdminController.java就能接收到数据，此时就出现一个问题，搜索时不输入，传到后端的数据就是“”，空字符串，这样在进行数据库查询的时候就会出问题，所以，需要把数据库查询相关的配置改掉，如下：查询条件加上了非空判断

```

<sql id="AdminFindCriteria">
    <where>
        <if test="account != null and account != ''">and account = #{account}</if>
        <if test="id != null and id != ''">and id = #{id}</if>
        <if test="name != null and name != ''">and name = #{name}</if>
        <if test="password != null and password != ''">and password = #{password}</if>
        <if test="phone != null and phone != ''">and phone = #{phone}</if>
        <if test="remark != null and remark != ''">and remark = #{remark}</if>
    </where>
</sql>
<sql id="AdminUpdateCriteria">
    <set>

```

```

        <if test="updateAccount != null and updateAccount != ''">account = #
{updateAccount}</if>
        <if test="updateId != null and updateId != ' ' ">id = #{updateId}</if>
        <if test="updateName != null and updateName != ' ' ">name = #{updateName},
</if>
        <if test="updatePassword != null and updatePassword != ''">password = #
{updatePassword}</if>
        <if test="updatePhone != null and updatePhone != ''">phone = #
{updatePhone}</if>
        <if test="updateRemark != null and updateRemark != ''">remark = #
{updateRemark}</if>
    </set>
</sql>

```

同时为了使代码更简洁

list.jsp改为

```

//执行搜索重载
    table.reload('currentTableId', {
        page: {
            curr: 1
        },
        contentType: 'application/json',
        where: data.field
    }, 'data');

    return false;
});

```

向后端提交json数据

测试组合查询 代码无误

● 问卷管理

● 用户管理 ×

查询条件

账号admin4

姓名

手机号码

搜索

添加删除

	ID	用户名	姓名	手机	备注
<input type="checkbox"/>	4	admin4	管理员4	13256674	管理员4

< 1 >

到第1页

确定

共1条

15条页

11 查询分页

采用mybatis的一些插件pageheler

pom.xml中加入

```
<dependency>
    <groupId>com.github.pagehelper</groupId>
    <artifactId>pagehelper</artifactId>
    <version>5.1.11</version>
</dependency>
```

applicationContext.xml中加入

```
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <!--数据源-->
    <property name="dataSource" ref="dataSource"></property>
    <!--Mybatis主配置文件加载-->
    <property name="configLocation" value="classpath:mybatis-config.xml">
</property>
    <property name="mapperLocations">
        <list>
            <value>classpath:com/yanzhen/mapper/*.xml</value>
        </list>
    </property>
    <property name="plugins">
        <array>
            <bean class="com.github.pagehelper.PageInterceptor">
                <property name="properties">
                    <value>
                        helperDialect=mysql
                    </value>
                </property>
            </bean>
        </array>
    </property>
</bean>
```

修改AdminService

query () 方法加一句话

```
public List<Admin> query(Admin admin){
    PageHelper.startPage(admin.getPage(), admin.getLimit()); //加这一句
    Map<String, Object> map =
    MapParameter.getInstance().put(BeanMapUtils.beanToMap(admin)).getMap();
    return adminDao.query(map);
}
```

分页功能好了

<input type="checkbox"/>	3	admin3	管理员3	13256673	管理员3
<input type="checkbox"/>	4	admin4	管理员4	13256674	管理员4
<input type="checkbox"/>	6	admin6	管理员6	13256676	管理员6
<input type="checkbox"/>	7	admin7	管理员7	13256677	管理员7
<input type="checkbox"/>	8	admin8	管理员8	13256678	管理员8
<input type="checkbox"/>	9	admin9	管理员9	13256679	管理员9
<input type="checkbox"/>	13	admin13	管理员13	132566713	管理员13
<input type="checkbox"/>	14	admin14	管理员14	132566714	管理员14

< 1 2 3 ... 52 > 到第 1 页 确定 共 512 条 10 条/页 ▼

12 设置mybatis的日志

这样每次操作数据库就会打印日志信息

pom.xml中加入log4j

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
</dependency>
```

在resources中创建log4j.properties ,把以下内容复制进去

```
#定义LOG输出级别
log4j.rootLogger=DEBUG,INFO,Console,File
#定义日志输出目的地为控制台
log4j.appender.Console=org.apache.log4j.ConsoleAppender
log4j.appender.Console.Target=System.out
#可以灵活地指定日志输出格式，下面一行是指定具体的格式
log4j.appender.Console.layout = org.apache.log4j.PatternLayout
log4j.appender.Console.layout.ConversionPattern = [%p] [%d{yyyy-MM-dd HH\:mm\:ss}] [%c]%m %l%n

#文件大小到达指定尺寸的时候产生一个新的文件
log4j.appender.File = org.apache.log4j.RollingFileAppender
#指定输出目录
log4j.appender.File.File = logs/survey.log
#定义文件最大大小
log4j.appender.File.MaxFileSize = 10MB
#输出所以日志，如果换成DEBUG表示输出DEBUG以上级别日志
log4j.appender.File.Threshold = ALL
log4j.appender.File.layout = org.apache.log4j.PatternLayout
log4j.appender.File.layout.ConversionPattern = [%p] [%d{yyyy-MM-dd HH\:mm\:ss}] [%c]%m %l%n
#mybatis 日志配置 需要将日志的输出级别调为debug
log4j.logger.com.ibatis=DEBUG
log4j.logger.com.ibatis.common.jdbc.SimpleDataSource=DEBUG
log4j.logger.com.ibatis.common.jdbc.ScriptRunner=DEBUG
log4j.logger.com.ibatis.sqlmap.engine.impl.SqlMapClientDelegate=DEBUG
#与sql相关
log4j.logger.java.sql.Connection=DEBUG
log4j.logger.java.sql.Statement=DEBUG
log4j.logger.java.sql.PreparedStatement=DEBUG
```

修改mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <settings>
        <setting name="logImpl" value="STDOUT_LOGGING" />
    </settings>
</configuration>
```

11.4 新增页面

1 新建add.list

在admin文件夹下新建add.jsp，把 add.html 复制进来，保留jsp的第一行

2 修改list.jsp的添加标签的url

```
table.on('toolbar(currentTableFilter)', function (obj) {
    if (obj.event === 'add') { // 监听添加操作
        var index = layer.open({
            title: '添加用户',
            type: 2,
            shade: 0.2,
            maxmin: true,
            shadeClose: true,
            area: ['80%', '80%'],
            content: 'create',
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
    }
});
```

3 修改AdminController.java

添加一个新方法

```
@GetMapping("/create")
public String v_create(){
    return "admin/add";
}
```

让页面跳转到admin/add.jsp页面

4修改add.jsp的静态引用

href、src等 修改后运行



5 删除和修改add.jsp页面

修改前

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>layui</title>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge, chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/css/public.css" media="all">
</head>
<body>
<div class="layuiimini-container">
    <div class="layuiimini-main">

        <blockquote class="layui-elem-quote layui-text">
            鉴于小伙伴的普遍反馈，先温馨提醒两个常见问题：1. <a
href="/doc/base/faq.html#form" target="_blank">为什么select/checkbox/radio没显示？
</a> 2. <a href="/doc/modules/form.html#render" target="_blank">动态添加的表单元素如
何更新？ </a>
        </blockquote>

        <fieldset class="layui-elem-field layui-field-title" style="margin-top:
20px;">
            <legend>表单集合演示</legend>
        </fieldset>
```

```

<form class="layui-form" action="">
  <div class="layui-form-item">
    <label class="layui-form-label">单行输入框</label>
    <div class="layui-input-block">
      <input type="text" name="title" lay-verify="title"
autocomplete="off" placeholder="请输入标题" class="layui-input">
    </div>
  </div>
  <div class="layui-form-item">
    <label class="layui-form-label">验证必填项</label>
    <div class="layui-input-block">
      <input type="text" name="username" lay-verify="required"
lay-reqtext="用户名是必填项，岂能为空？" placeholder="请输入" autocomplete="off"
class="layui-input">
    </div>
  </div>

  <div class="layui-form-item">
    <div class="layui-inline">
      <label class="layui-form-label">验证手机</label>
      <div class="layui-input-inline">
        <input type="tel" name="phone" lay-
verify="required|phone" autocomplete="off" class="layui-input">
      </div>
    </div>
    <div class="layui-inline">
      <label class="layui-form-label">验证邮箱</label>
      <div class="layui-input-inline">
        <input type="text" name="email" lay-verify="email"
autocomplete="off" class="layui-input">
      </div>
    </div>
  </div>

  <div class="layui-form-item">
    <div class="layui-inline">
      <label class="layui-form-label">多规则验证</label>
      <div class="layui-input-inline">
        <input type="text" name="number" lay-
verify="required|number" autocomplete="off" class="layui-input">
      </div>
    </div>
    <div class="layui-inline">
      <label class="layui-form-label">验证日期</label>
      <div class="layui-input-inline">
        <input type="text" name="date" id="date" lay-
verify="date" placeholder="yyyy-MM-dd" autocomplete="off" class="layui-input">
      </div>
    </div>
    <div class="layui-inline">
      <label class="layui-form-label">验证链接</label>
      <div class="layui-input-inline">
        <input type="tel" name="url" lay-verify="url"
autocomplete="off" class="layui-input">
      </div>
    </div>
  </div>
</div>

```

```

<div class="layui-form-item">
  <label class="layui-form-label">验证身份证</label>
  <div class="layui-input-block">
    <input type="text" name="identity" lay-verify="identity"
placeholder="" autocomplete="off" class="layui-input">
  </div>
</div>
<div class="layui-form-item">
  <label class="layui-form-label">自定义验证</label>
  <div class="layui-input-inline">
    <input type="password" name="password" lay-verify="pass"
placeholder="请输入密码" autocomplete="off" class="layui-input">
  </div>
  <div class="layui-form-mid layui-word-aux">请填写6到12位密码</div>
</div>

<div class="layui-form-item">
  <div class="layui-inline">
    <label class="layui-form-label">范围</label>
    <div class="layui-input-inline" style="width: 100px;">
      <input type="text" name="price_min" placeholder="¥"
autocomplete="off" class="layui-input">
    </div>
    <div class="layui-form-mid">-</div>
    <div class="layui-input-inline" style="width: 100px;">
      <input type="text" name="price_max" placeholder="¥"
autocomplete="off" class="layui-input">
    </div>
  </div>
</div>

<div class="layui-form-item">
  <label class="layui-form-label">单行选择框</label>
  <div class="layui-input-block">
    <select name="interest" lay-filter="aihao">
      <option value=""></option>
      <option value="0">写作</option>
      <option value="1" selected="">阅读</option>
      <option value="2">游戏</option>
      <option value="3">音乐</option>
      <option value="4">旅行</option>
    </select>
  </div>
</div>

<div class="layui-form-item">
  <div class="layui-inline">
    <label class="layui-form-label">分组选择框</label>
    <div class="layui-input-inline">
      <select name="quiz">
        <option value="">请选择问题</option>
        <optgroup label="城市记忆">
          <option value="你工作的第一个城市">你工作的第一个城市
</option>
        </optgroup>
        <optgroup label="学生时代">

```

```

        <option value="你的工号">你的工号</option>
        <option value="你最喜欢的老师">你最喜欢的老师
</option>

        </optgroup>
    </select>
</div>
</div>
<div class="layui-inline">
    <label class="layui-form-label">搜索选择框</label>
    <div class="layui-input-inline">
        <select name="modules" lay-verify="required" lay-
search="">

            <option value="">直接选择或搜索选择</option>
            <option value="1">layer</option>
            <option value="2">form</option>
            <option value="3">layim</option>
            <option value="4">element</option>
            <option value="5">laytpl</option>
            <option value="6">upload</option>
            <option value="7">laydate</option>
            <option value="8">laypage</option>
            <option value="9">flow</option>
            <option value="10">util</option>
            <option value="11">code</option>
            <option value="12">tree</option>
            <option value="13">layedit</option>
            <option value="14">nav</option>
            <option value="15">tab</option>
            <option value="16">table</option>
            <option value="17">select</option>
            <option value="18">checkbox</option>
            <option value="19">switch</option>
            <option value="20">radio</option>
        </select>
    </div>
</div>
</div>
<div class="layui-form-item">
    <label class="layui-form-label">联动选择框</label>
    <div class="layui-input-inline">
        <select name="quiz1">
            <option value="">请选择省</option>
            <option value="浙江" selected="">浙江省</option>
            <option value="你的工号">江西省</option>
            <option value="你最喜欢的老师">福建省</option>
        </select>
    </div>
    <div class="layui-input-inline">
        <select name="quiz2">
            <option value="">请选择市</option>
            <option value="杭州">杭州</option>
            <option value="宁波" disabled="">宁波</option>
            <option value="温州">温州</option>
            <option value="温州">台州</option>
            <option value="温州">绍兴</option>
        </select>
    </div>
</div>

```

```

        <div class="layui-input-inline">
            <select name="quiz3">
                <option value="">请选择县/区</option>
                <option value="西湖区">西湖区</option>
                <option value="余杭区">余杭区</option>
                <option value="拱墅区">临安市</option>
            </select>
        </div>
        <div class="layui-form-mid layui-word-aux">此处只是演示联动排版，并未
做联动交互</div>
    </div>

    <div class="layui-form-item">
        <label class="layui-form-label">复选框</label>
        <div class="layui-input-block">
            <input type="checkbox" name="like[write]" title="写作">
            <input type="checkbox" name="like[read]" title="阅读"
checked="">
            <input type="checkbox" name="like[game]" title="游戏">
        </div>
    </div>

    <div class="layui-form-item" pane="">
        <label class="layui-form-label">原始复选框</label>
        <div class="layui-input-block">
            <input type="checkbox" name="like1[write]" lay-
skin="primary" title="写作" checked="">
            <input type="checkbox" name="like1[read]" lay-skin="primary"
title="阅读">
            <input type="checkbox" name="like1[game]" lay-skin="primary"
title="游戏" disabled="">
        </div>
    </div>

    <div class="layui-form-item">
        <label class="layui-form-label">开关-默认关</label>
        <div class="layui-input-block">
            <input type="checkbox" name="close" lay-skin="switch" lay-
text="ON|OFF">
        </div>
    </div>

    <div class="layui-form-item">
        <label class="layui-form-label">开关-默认开</label>
        <div class="layui-input-block">
            <input type="checkbox" checked="" name="open" lay-
skin="switch" lay-filter="switchTest" lay-text="ON|OFF">
        </div>
    </div>

    <div class="layui-form-item">
        <label class="layui-form-label">单选框</label>
        <div class="layui-input-block">
            <input type="radio" name="sex" value="男" title="男"
checked="">
            <input type="radio" name="sex" value="女" title="女">
            <input type="radio" name="sex" value="禁" title="禁用"
disabled="">
        </div>
    </div>

```

```

        <div class="layui-form-item layui-form-text">
            <label class="layui-form-label">普通文本域</label>
            <div class="layui-input-block">
                <textarea placeholder="请输入内容" class="layui-textarea">
</textarea>
            </div>
        </div>
        <!--<div class="layui-form-item layui-form-text">
            <label class="layui-form-label">编辑器</label>
            <div class="layui-input-block">
                <textarea class="layui-textarea layui-hide" name="content" lay-
verify="content" id="LAY_demo_editor"></textarea>
            </div>
        </div-->
        <div class="layui-form-item">
            <div class="layui-input-block">
                <button class="layui-btn" lay-submit="" lay-filter="demo1">立
即提交</button>
                <button type="reset" class="layui-btn layui-btn-primary">重置
</button>
            </div>
        </div>
    </form>

    <!-- 示例-970 -->
    <ins class="adsbygoogle" style="display:inline-
block;width:970px;height:90px" data-ad-client="ca-pub-6111334333458862" data-ad-
slot="3820120620"></ins>

    <fieldset class="layui-elem-field layui-field-title" style="margin-top:
50px;">
        <legend>初始赋值演示</legend>
    </fieldset>

    <form class="layui-form" action="" lay-filter="example">
        <div class="layui-form-item">
            <label class="layui-form-label">输入框</label>
            <div class="layui-input-block">
                <input type="text" name="username" lay-verify="title"
autocomplete="off" placeholder="请输入标题" class="layui-input">
            </div>
        </div>
        <div class="layui-form-item">
            <label class="layui-form-label">密码框</label>
            <div class="layui-input-block">
                <input type="password" name="password" placeholder="请输入密
码" autocomplete="off" class="layui-input">
            </div>
        </div>

        <div class="layui-form-item">
            <label class="layui-form-label">选择框</label>
            <div class="layui-input-block">
                <select name="interest" lay-filter="aihao">
                    <option value=""></option>
                    <option value="0">写作</option>
                    <option value="1">阅读</option>
                    <option value="2">游戏</option>

```

```

        <option value="3">音乐</option>
        <option value="4">旅行</option>
    </select>
</div>

<div class="layui-form-item">
    <label class="layui-form-label">复选框</label>
    <div class="layui-input-block">
        <input type="checkbox" name="like[write]" title="写作">
        <input type="checkbox" name="like[read]" title="阅读">
        <input type="checkbox" name="like[daze]" title="发呆">
    </div>
</div>

<div class="layui-form-item">
    <label class="layui-form-label">开关</label>
    <div class="layui-input-block">
        <input type="checkbox" name="close" lay-skin="switch" lay-
text="ON|OFF">
    </div>
</div>

<div class="layui-form-item">
    <label class="layui-form-label">单选框</label>
    <div class="layui-input-block">
        <input type="radio" name="sex" value="男" title="男"
checked="">
        <input type="radio" name="sex" value="女" title="女">
    </div>
</div>
<div class="layui-form-item layui-form-text">
    <label class="layui-form-label">文本域</label>
    <div class="layui-input-block">
        <textarea placeholder="请输入内容" class="layui-textarea"
name="desc"></textarea>
    </div>
</div>

<div class="layui-form-item">
    <div class="layui-input-block">
        <button class="layui-btn" lay-submit="" lay-filter="demo1">立
即提交</button>
    </div>
</div>
</form>
<fieldset class="layui-elem-field layui-field-title" style="margin-top:
50px;">
    <legend>方框风格的表单集合</legend>
</fieldset>
<form class="layui-form layui-form-pane" action="">
    <div class="layui-form-item">
        <label class="layui-form-label">长输入框</label>
        <div class="layui-input-block">
            <input type="text" name="title" autocomplete="off"
placeholder="请输入标题" class="layui-input">
        </div>
    </div>
</form>

```

```
<div class="layui-form-item">
  <label class="layui-form-label">短输入框</label>
  <div class="layui-input-inline">
    <input type="text" name="username" lay-verify="required"
placeholder="请输入" autocomplete="off" class="layui-input">
  </div>
</div>

<div class="layui-form-item">
  <div class="layui-inline">
    <label class="layui-form-label">日期选择</label>
    <div class="layui-input-block">
      <input type="text" name="date" id="date1"
autocomplete="off" class="layui-input">
    </div>
  </div>
  <div class="layui-inline">
    <label class="layui-form-label">行内表单</label>
    <div class="layui-input-inline">
      <input type="text" name="number" autocomplete="off"
class="layui-input">
    </div>
  </div>
</div>
<div class="layui-form-item">
  <label class="layui-form-label">密码</label>
  <div class="layui-input-inline">
    <input type="password" name="password" placeholder="请输入密
码" autocomplete="off" class="layui-input">
  </div>
  <div class="layui-form-mid layui-word-aux">请务必填写用户名</div>
</div>

<div class="layui-form-item">
  <div class="layui-inline">
    <label class="layui-form-label">范围</label>
    <div class="layui-input-inline" style="width: 100px;">
      <input type="text" name="price_min" placeholder="¥"
autocomplete="off" class="layui-input">
    </div>
    <div class="layui-form-mid">--</div>
    <div class="layui-input-inline" style="width: 100px;">
      <input type="text" name="price_max" placeholder="¥"
autocomplete="off" class="layui-input">
    </div>
  </div>
</div>

<div class="layui-form-item">
  <label class="layui-form-label">单行选择框</label>
  <div class="layui-input-block">
    <select name="interest" lay-filter="aihao">
      <option value=""></option>
      <option value="0">写作</option>
      <option value="1" selected="">阅读</option>
      <option value="2">游戏</option>
      <option value="3">音乐</option>
      <option value="4">旅行</option>
    </select>
  </div>
</div>
```



```

        </select>
    </div>
</div>

<div class="layui-form-item">
    <label class="layui-form-label">行内选择框</label>
    <div class="layui-input-inline">
        <select name="quiz1">
            <option value="">请选择省</option>
            <option value="浙江" selected="">浙江省</option>
            <option value="你的工号">江西省</option>
            <option value="你最喜欢的老师">福建省</option>
        </select>
    </div>
    <div class="layui-input-inline">
        <select name="quiz2">
            <option value="">请选择市</option>
            <option value="杭州">杭州</option>
            <option value="宁波" disabled="">宁波</option>
            <option value="温州">温州</option>
            <option value="温州">台州</option>
            <option value="温州">绍兴</option>
        </select>
    </div>
    <div class="layui-input-inline">
        <select name="quiz3">
            <option value="">请选择县/区</option>
            <option value="西湖区">西湖区</option>
            <option value="余杭区">余杭区</option>
            <option value="拱墅区">临安市</option>
        </select>
    </div>
</div>
<div class="layui-form-item" pane="">
    <label class="layui-form-label">开关-开</label>
    <div class="layui-input-block">
        <input type="checkbox" checked="" name="open" lay-
skin="switch" lay-filter="switchTest" title="开关">
    </div>
</div>
<div class="layui-form-item" pane="">
    <label class="layui-form-label">单选框</label>
    <div class="layui-input-block">
        <input type="radio" name="sex" value="男" title="男"
checked="">
        <input type="radio" name="sex" value="女" title="女">
        <input type="radio" name="sex" value="禁" title="禁用"
disabled="">
    </div>
</div>
<div class="layui-form-item layui-form-text">
    <label class="layui-form-label">文本域</label>
    <div class="layui-input-block">
        <textarea placeholder="请输入内容" class="layui-textarea">
</textarea>
    </div>
</div>
<div class="layui-form-item">

```

```

        <button class="layui-btn" lay-submit="" lay-filter="demo2">跳转式
提交</button>
    </div>
</form>
</div>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<!-- 注意：如果你直接复制所有代码到本地，上述js路径需要改成你本地的 -->
<script>
    layui.use(['form', 'layedit', 'laydate'], function () {
        var form = layui.form
            , layer = layui.layer
            , layedit = layui.layedit
            , laydate = layui.laydate;

        //日期
        laydate.render({
            elem: '#date'
        });
        laydate.render({
            elem: '#date1'
        });

        //创建一个编辑器
        var editIndex = layedit.build('LAY_demo_editor');

        //自定义验证规则
        form.verify({
            title: function (value) {
                if (value.length < 5) {
                    return '标题至少得5个字符啊';
                }
            },
            pass: [
                /^[\S]{6,12}$/
                , '密码必须6到12位，且不能出现空格'
            ],
            content: function (value) {
                layedit.sync(editIndex);
            }
        });

        //监听指定开关
        form.on('switch(switchTest)', function (data) {
            layer.msg('开关checked: ' + (this.checked ? 'true' : 'false'), {
                offset: '6px'
            });
            layer.tips('温馨提示：请注意开关状态的文字可以随意定义，而不仅仅是ON|OFF',
data.othis)
        });

        //监听提交
        form.on('submit(demo1)', function (data) {
            layer.alert(JSON.stringify(data.field), {
                title: '最终的提交信息'
            })
            return false;
        });
    });

```

```

});

//表单初始赋值
form.val('example', {
    "username": "贤心" // "name": "value"
    , "password": "123456"
    , "interest": 1
    , "like[write]": true //复选框选中状态
    , "close": true //开关状态
    , "sex": "女"
    , "desc": "我爱 layui"
});

});
</script>

</body>
</html>

```

修改后

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>layui</title>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/css/public.css" media="all">
</head>
<body>
<div class="layuimini-container">
    <div class="layuimini-main">
        <form class="layui-form" action="">
            <div class="layui-form-item">
                <label class="layui-form-label">账号</label>
                <div class="layui-input-block">
                    <input type="text" name="account" lay-verify="required"
autocomplete="off" placeholder="请输入账号" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">密码</label>
                <div class="layui-input-block">
                    <input type="text" name="password" lay-verify="required"
autocomplete="off" placeholder="请输入密码" value="123456" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">姓名</label>
                <div class="layui-input-block">

```

```

        <input type="text" name="name" lay-verify="required"
autocomplete="off" placeholder="请输入姓名" class="layui-input">
    </div>
</div>
<div class="layui-form-item">
    <label class="layui-form-label">手机</label>
    <div class="layui-input-block">
        <input type="text" name="phone" lay-verify="phone"
value="1111111111" autocomplete="off" placeholder="请输入手机" class="layui-
input">
    </div>
</div>
<div class="layui-form-item layui-form-text">
    <label class="layui-form-label">备注</label>
    <div class="layui-input-block">
        <textarea placeholder="请输入备注" name="remark" class="layui-
textarea"></textarea>
    </div>
</div>
<div class="layui-form-item">
    <div class="layui-input-block">
        <button type="reset" class="layui-btn layui-btn-primary">重置
</button>
        <button class="layui-btn" lay-submit="" lay-filter="demo1">立
即提交</button>
    </div>
</div>
</form>

</div>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<!-- 注意：如果你直接复制所有代码到本地，上述js路径需要改成你本地的 -->
<script>
    layui.use(['form','jquery','layer'], function () {
        var form = layui.form,$=layui.jquery,layer=layui.layer,layer =
layui.layer;
        //监听提交
        var index = parent.layer.getFrameIndex(window.name); //获取窗口索引
        form.on('submit(demo1)', function (data) {
            $.ajax({
                url:"create",
                type:"POST",
                contentType:'application/json',
                dataType:'json',
                data:JSON.stringify(data.field),
                success:function(data){
                    layer.msg(data.msg,{time:500},
                        function(){
                            parent.layer.close(index);
                        });
                }
            });
            return false;
        });
    });

```

```

</script>

</body>
</html>

```

通过Ajax，传送json数据给前端，----AdminController.java-----create()方法

```

@PostMapping("/create")
@ResponseBody
public Map<String, Object> create(@RequestBody Admin admin){
    int result = adminService.create(admin);

    if(result <= 0){
        return MapControl.getInstance().error().getMap();
    }
    return MapControl.getInstance().success().getMap();
}

```

6 注意Ajax

```

<script>
    layui.use(['form', 'jquery', 'layer'], function () {
        var form = layui.form, $ = layui.jquery, layer = layui.layer, layer =
        layui.layer;
        //监听提交
        var index = parent.layer.getFrameIndex(window.name); //获取窗口索引
        form.on('submit(demo1)', function (data) {
            $.ajax({
                url: "create", //请求路径
                type: "POST", //请求方式
                contentType: 'application/json', //响应数据格式
                dataType: 'json', //预期服务器返回的数据格式
                data: JSON.stringify(data.field), //发送到服务器的数据格式为json
                success: function (data) { //请求成功后的回调函数
                    layer.msg(data.msg, {time: 500}, //提交后 提示信息 延迟0.5s
                        function () { //定义一个关闭函数，自动关闭
                            parent.layer.close(index); //传入窗口索引值，关闭窗口
                        }
                    );
                }
            });
            return false; //表格提交失败后返回false
        });
    });
</script>

```

11.5 删除页面

1 修改list.jsp的监听删除操作

```

JSON.stringify(data) //将字符串转变为json数据
JSON.parse() //将json数据转变为字符串

```

加入删除数量判断

```

table.on('toolbar(currentTableFilter)', function (obj) {
    if (obj.event === 'add') { // 监听添加操作
        var index = layer.open({
            title: '添加用户',
            type: 2,
            shade: 0.2,
            maxmin: false,
            shadeClose: true,
            area: ['80%', '80%'],
            content: 'create',
            end: function () { //end方法，只要关闭窗口就会刷新
                table.reload('currentTableId');
            }
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
    } else if (obj.event === 'delete') { // 监听删除操作
        var checkStatus = table.checkStatus('currentTableId')
            , data = checkStatus.data;
        var arr=[];
        for(index in data){
            arr.push(data[index].id);
        }
        if(arr.length <1){
            layer.msg("请至少选择一行数据删除",{time:1000});
            return;
        }

        $.ajax({
            url:"delete",
            type:"POST",
            dataType:'json',
            data:'ids='+arr.join(","),
            success:function(data){
                layer.msg(data.msg,{time:500},
                    function(){
                        table.reload('currentTableId'); //刷新重新加载
                    });
            }
        });
    } else if (obj.event === 'edit') { // 监听编辑操作

        var checkStatus = table.checkStatus('currentTableId')
            , data = checkStatus.data;
        var arr=[];
        for(index in data){
            arr.push(data[index].id);
        }

        if(arr.length !=1){
            layer.msg("请选择一行数据修改",{time:1000});
            return;
        }

        var index = layer.open({

```

```

        title: '修改用户',
        type: 2,
        shade: 0.2,
        maxmin:false,
        shadeClose: true,
        area: ['80%', '80%'],
        content: 'detail?id='+arr[0],
        end:function(){
            table.reload('currentTableId');
        }
    });
    $(window).on("resize", function () {
        layer.full(index);
    });
}
});

```

2 批量删除

Adminservice.java 添加deleteBatch () 方法

```

public int deleteBatch(String ids){
    int flag=0;
    List<String> list= Splitter.on(",").splitToList(ids); //将字符串转变为list集合
    for(String s:list){

        adminDao.delete(MapParameter.getInstance().addId(Integer.parseInt(s)).getMap());
    }
    flag++;
    return flag;
}

```

3 修改AdminController.java

```

@PostMapping("/delete")
@ResponseBody
public Map<String,Object> delete(String ids){

    int result=adminService.deleteBatch(ids);
    if(result<=0){
        return MapControl.getInstance().error().getMap();
    }
    return MapControl.getInstance().success().getMap();
}

```

11.6 编辑页面

1 修改list.jsp

添加编辑按钮

```
<script type="text/html" id="toolbarDemo">
    <div class="layui-btn-container">
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn" lay-
event="add"> 添加 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-delete-btn"
lay-event="edit"> 编辑 </button>
        <button class="layui-btn layui-btn-sm layui-btn-danger data-delete-btn"
lay-event="delete"> 删除 </button>

    </div>
</script>
```

2 编写 监听编辑操作

```
else if (obj.event === 'edit') { // 监听编辑操作

    var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
    var arr=[];
    for(index in data){
        arr.push(data[index].id);
    }

    if(arr.length !=1){
        layer.msg("请选择一行数据修改",{time:1000});
        return;
    }

    var index = layer.open({
        title: '修改用户',
        type: 2,
        shade: 0.2,
        maxmin:false,
        shadeClose: true,
        area: ['80%', '80%'],
        content: 'detail?id='+arr[0],
        end:function(){
            table.reload('currentTableId');
        }
    });
    $(window).on("resize", function () {
        layer.full(index);
    });
}
});
```


3 编写update.jsp

复制add.jsp 修改名称update

update 与 add不同的是 update需要回显，先查出来修改内容显示到页面上

```
form.val('demo2', {
  "account": "${admin.account}",
  "name": "${admin.name}",
  "phone": "${admin.phone}",
  "remark": "${admin.remark}",
})
```

同时 提交的是时候 需要把id值也提交，用于数据库修改，因此创建一个隐藏于域标签

```
<input type="hidden" name="id" value="${admin.id}" lay-verify="required"
autocomplete="off" placeholder="请输入账号" class="layui-input">
```

修改ajax的url为 update

```
form.on('submit(demo1)', function (data) {
  $.ajax({
    url: "update",
    type: "POST",
    contentType: 'application/json',
    dataType: 'json',
    data: JSON.stringify(data.field),
    success: function(data){
      layer.msg(data.msg, {time: 500},
        function(){
          parent.layer.close(index);
        });
    }
  });
  return false;
});
```

4 修改图标

引入图标库

```
<link rel="stylesheet" href="../../static/lib/font-awesome-4.7.0/css/font-
awesome.css" media="all">
```

加入图标标签

```
<button class="layui-btn layui-btn-sm data-add-btn" lay-event="add"> <i
class="fa fa-plus"></i> 添加</button>
<button class="layui-btn layui-btn-sm layui-btn-sm data-delete-btn" lay-
event="edit"><i class="fa fa-pencil"></i> 修改 </button>
<button class="layui-btn layui-btn-sm layui-btn-danger data-delete-btn" lay-
event="delete"><i class="fa fa-remove"></i> 删除 </button>
```

5 完整的update.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>layui</title>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
    <link rel="stylesheet" href="../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../static/css/public.css" media="all">
</head>
<body>
<div class="layui-mini-container">
    <div class="layui-mini-main">
        <form class="layui-form" action="" lay-filter="demo2">
            <input type="hidden" name="id" value="{admin.id}" lay-
verify="required" autocomplete="off" placeholder="请输入账号" class="layui-input">
            <div class="layui-form-item">
                <label class="layui-form-label">账号</label>
                <div class="layui-input-block">
                    <input type="text" name="account" lay-verify="required"
autocomplete="off" placeholder="请输入账号" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">姓名</label>
                <div class="layui-input-block">
                    <input type="text" name="name" lay-verify="required"
autocomplete="off" placeholder="请输入姓名" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">手机</label>
                <div class="layui-input-block">
                    <input type="text" name="phone" lay-verify="phone"
value="18533333333" autocomplete="off" placeholder="请输入手机" class="layui-
input">
                </div>
            </div>
            <div class="layui-form-item layui-form-text">
                <label class="layui-form-label">备注</label>
                <div class="layui-input-block">
                    <textarea placeholder="请输入备注" name="remark" class="layui-
textarea"></textarea>
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-input-block">
                    <button type="reset" class="layui-btn layui-btn-primary">重置
</button>
                    <button class="layui-btn" lay-submit="" lay-filter="demo1">立
即提交</button>
```

```

        </div>
    </div>
</form>

</div>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<!-- 注意：如果你直接复制所有代码到本地，上述js路径需要改成你本地的 -->
<script>
    layui.use(['form', 'jquery', 'layer'], function () {
        var form = layui.form, $=layui.jquery, layer=layui.layer, layer =
        layui.layer;
        //监听提交
        var index = parent.layer.getFrameIndex(window.name); //获取窗口索引
        form.on('submit(demo1)', function (data) {
            $.ajax({
                url: "update",
                type: "POST",
                contentType: 'application/json',
                dataType: 'json',
                data: JSON.stringify(data.field),
                success: function (data) {
                    layer.msg(data.msg, {time: 500},
                        function () {
                            parent.layer.close(index);
                        });
                }
            });
            return false;
        });

        form.val('demo2', {
            "account": "${admin.account}",
            "name": "${admin.name}",
            "phone": "${admin.phone}",
            "remark": "${admin.remark}",
        })
    });
</script>

</body>
</html>

```

12 问卷管理

12.1 问卷管理

1 创建SurveyService

复制AdminService,改名SurveyService。

Ctrl+r 批量替换

Admin--->Survey

admin--->survey

AdminDao---->SurveyDao

adminDao---->surveyDao

2 创建SurveyController

复制AdminController 替换同上述

3 创建survey的页面

复制webapp/admin , 改名为survey

4 修改init.json

问卷管理: href: " survey/list "

5 修改survey/list.jsp

修改table.render

```
table.render({
    elem: '#currentTableId',
    url: 'query',
    toolbar: '#toolbarDemo',
    contentType: 'application/json',
    method: "post",
    defaultToolbar: ['filter', 'exports', 'print'],
    cols: [[
        {type: "checkbox", width: 50},
        {field: 'id', width: 80, title: 'ID', sort: true},
        {field: 'title', title: '问卷标题'},
        {field: 'startTime', width: 130, title: '开始时间', sort: true},
        {field: 'endTime', width: 130, title: '结束时间'},
        {field: 'state', width: 130, title: '状态'},
        {field: 'creator', width: 100, title: '创建人'},
        {field: 'createTime', width: 130, title: '创建时间'}
    ]],
    limits: [10, 15, 20, 25, 50, 100],
    limit: 15,
    page: true,
    skin: 'line'
});
```

12.2 问卷添加功能

1 add.jsp页面编写

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>layui</title>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/css/public.css" media="all">
</head>
<body>
<div class="layui-mini-container">
    <div class="layui-mini-main">
        <form class="layui-form" action="">
            <div class="layui-form-item">
                <label class="layui-form-label">标题</label>
                <div class="layui-input-block">
                    <input type="text" name="title" lay-verify="required"
autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item layui-form-text">
                <label class="layui-form-label">说明</label>
                <div class="layui-input-block">
                    <input type="text" name="remark" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item">
                <div class="layui-inline">
                    <label class="layui-form-label">开始时间</label>
                    <div class="layui-input-inline">
                        <input type="text" name="startTime" id="startTime" lay-
verify="datetime" placeholder="yyyy-MM-dd HH:mm:ss" autocomplete="off"
class="layui-input">
                    </div>
                </div>
                <div class="layui-inline">
                    <label class="layui-form-label">结束时间</label>
                    <div class="layui-input-inline">
                        <input type="text" name="endTime" id="endTime" lay-
verify="datetime" autocomplete="off" placeholder="yyyy-MM-dd HH:mm:ss"
class="layui-input">
                    </div>
                </div>
            </div>
            <div class="layui-form-item">
                <label class="layui-form-label">限制</label>
                <div class="layui-input-inline">
                    <select name="bounds" lay-filter="bounds">
```

```

        <option value="0">不限制</option>
        <option value="1">IP限制</option>
    </select>
</div>
</div>
<div class="layui-form-item">
    <label class="layui-form-label">访问规则</label>
    <div class="layui-input-inline">
        <select name="rules" lay-filter="rules">
            <option value="0">公开访问</option>
            <option value="1">密码访问</option>
        </select>
    </div>
    <div class="layui-input-inline">
        <input type="text" name="password" id="password"
placeholder="请设置访问密码" autocomplete="off" style="display: none;"
class="layui-input">
    </div>
</div>

    <div class="layui-form-item">
        <label class="layui-form-label">匿名访问</label>
        <div class="layui-input-block">
            <input type="checkbox" checked name="anon" lay-skin="switch"
lay-filter="switchTest" lay-text="ON|OFF" value="0">
        </div>
    </div>

    <div class="layui-form-item">
        <div class="layui-input-block">
            <button type="reset" class="layui-btn layui-btn-primary">重置
</button>
            <button class="layui-btn" lay-submit="" lay-filter="demo1">立
即提交</button>
        </div>
    </div>

</form>

</div>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<!-- 注意：如果你直接复制所有代码到本地，上述js路径需要改成你本地的 -->
<script>
    layui.use(['form', 'jquery', 'layer', 'laydate'], function () {
        var form =
layui.form, $=layui.jquery, layer=layui.layer, laydate=layui.laydate, layer =
layui.layer;

        //日期
        laydate.render({
            elem: "#startTime",
            type: 'datetime'
        })
        laydate.render({
            elem: "#endTime",

```

```

        type: 'datetime'
    })

    //监听提交
    var index = parent.layer.getFrameIndex(window.name); //获取窗口索引
    form.on('submit(demo1)', function (data) {
        $.ajax({
            url: "create",
            type: "POST",
            contentType: 'application/json',
            dataType: 'json',
            data: JSON.stringify(data.field),
            success: function (data) {
                layer.msg(data.msg, {time: 500},
                    function () {
                        parent.layer.close(index);
                    });
            }
        });
        return false;
    });

    //显示密码输入框
    form.on('select(rules)', function (data) {
        if (data.value == "1") {
            $('#password').show();
        } else {
            $('#password').hide();
        }
    });
});

</script>

</body>
</html>

```

2 创建人、状态信息、匿名信息获取

登录的时候有admin，所以从session中获取。

1创建一个SessionUtils.java类

```

package com.yanzhen.utils;

import com.yanzhen.entity.Admin;

import javax.servlet.http.HttpServletRequest;

public class SessionUtils {

    private static final String key = "admin";
    public static void setAdmin(HttpServletRequest request, Admin admin) {
        request.getSession().setAttribute(key, admin);
    }
    public static Admin getAdmin(HttpServletRequest request) {
        if (request.getSession().getAttribute(key) != null) {

```

```

        return (Admin)request.getSession().getAttribute(key);
    }else{
        return null;
    }
}
}
}

```

2 修改SurveyController.java 的create方法

Survey实体类中加入三个变量

```

public static final String state_create="创建";
public static final String state_exec="执行中";
public static final String state_over="结束";

```

修改SurveyController.java 的create方法

```

@PostMapping("/create")
@ResponseBody
public Map<String,Object> create(@RequestBody Survey survey, HttpServletRequest request){

    Admin currAdmin= SessionUtils.getAdmin(request);
    survey.setCreator(currAdmin.getId()); //设置创建者
    survey.setState(Survey.state_create); //设置问卷状态
    survey.setAnon(survey.getAnon() != null ? 0 : 1); //设置匿名信息
    int result =surveyService.create(survey);
    if(result<=0){
        return MapControl.getInstance().error().getMap();
    }
    return MapControl.getInstance().success().getMap();
}

```

3 修改LoginController.java 的login方法

```

@PostMapping("/login")
@ResponseBody
public Map<String, Object> login(@RequestBody Map<String,Object> map,
HttpServletRequest request){
    String account=map.get("account")+"";
    String password=map.get("password")+"";
    if(Strings.isNullOrEmpty(account)||Strings.isNullOrEmpty(password)){
        return MapControl.getInstance().error("用户名或密码不能为空").getMap();
    }
    Admin admin =adminService.login(account, MD5Utils.getMD5(password));
    if(admin!=null){
        SessionUtils.setAdmin(request,admin); //修改这一行
        return MapControl.getInstance().success().getMap();
    }else {
        return MapControl.getInstance().error("用户名或密码错误").getMap();
    }
}
}

```


4 更改密码保存

AdminController.java

```
@PostMapping("/create")
@ResponseBody
public Map<String, Object> create(@RequestBody Admin admin){
    admin.setPassword(MD5Utils.getMD5(admin.getPassword())); //对输入进去的密码进行加密
    int result =adminService.create(admin);
    if(result<=0){
        return MapControl.getInstance().error().getMap();
    }
    return MapControl.getInstance().success().getMap();
}
```

12.3 问卷修改功能

1 编写update.jsp

复制admin/update, 重命名为update

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>layui</title>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css" media="all">
    <link rel="stylesheet" href="../../static/css/public.css" media="all">
</head>
<body>
<div class="layuiimini-container">
    <div class="layuiimini-main">
        <form class="layui-form" action="" lay-filter="myform">
            <input type="hidden" name="id" value="{survey.id}" lay-verify="required" autocomplete="off" class="layui-input">
            <div class="layui-form-item">
                <label class="layui-form-label">标题</label>
                <div class="layui-input-block">
                    <input type="text" name="title" lay-verify="required" autocomplete="off" class="layui-input">
                </div>
            </div>
            <div class="layui-form-item layui-form-text">
                <label class="layui-form-label">说明</label>
                <div class="layui-input-block">
                    <input type="text" name="remark" class="layui-input">
                </div>
            </div>
        </form>
    </div>
</div>
```

```

<div class="layui-form-item">
  <div class="layui-inline">
    <label class="layui-form-label">开始时间</label>
    <div class="layui-input-inline">
      <input type="text" name="startTime" id="startTime" lay-
verify="datetime" placeholder="yyyy-MM-dd HH:mm:ss" autocomplete="off"
class="layui-input">
    </div>
  </div>
  <div class="layui-inline">
    <label class="layui-form-label">结束时间</label>
    <div class="layui-input-inline">
      <input type="tel" name="endTime" id="endTime" lay-
verify="datetime" autocomplete="off" placeholder="yyyy-MM-dd HH:mm:ss"
class="layui-input">
    </div>
  </div>
</div>
<div class="layui-form-item">
  <label class="layui-form-label">限制</label>
  <div class="layui-input-inline">
    <select name="bounds" lay-filter="bounds">
      <option value="0">不限制</option>
      <option value="1">IP限制</option>
    </select>
  </div>
</div>
<div class="layui-form-item">
  <label class="layui-form-label">访问规则</label>
  <div class="layui-input-inline">
    <select name="rules" lay-filter="rules">
      <option value="0">公开访问</option>
      <option value="1">密码访问</option>
    </select>
  </div>
  <div class="layui-input-inline">
    <input type="text" name="password" id="password"
placeholder="请设置访问密码" autocomplete="off" style="display: none;"
class="layui-input">
  </div>
</div>

<div class="layui-form-item">
  <label class="layui-form-label">匿名访问</label>
  <div class="layui-input-block">
    <input type="checkbox" checked name="anon" lay-skin="switch"
lay-filter="switchTest" lay-text="ON|OFF" value="0">
  </div>
</div>

<div class="layui-form-item">
  <div class="layui-input-block">
    <button type="reset" class="layui-btn layui-btn-primary">重置
</button>
    <button class="layui-btn" lay-submit="" lay-filter="demo1">立
即提交</button>
  </div>
</div>

```

```

        </form>

    </div>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<!-- 注意：如果你直接复制所有代码到本地，上述js路径需要改成你本地的 -->
<script>
    layui.use(['form', 'jquery', 'layer', 'laydate'], function () {
        var form =
layui.form, $=layui.jquery, layer=layui.layer, laydate=layui.laydate, layer =
layui.layer;

        //日期
        laydate.render({
            elem: "#startTime",
            type: 'datetime'
        })
        laydate.render({
            elem: "#endTime",
            type: 'datetime'
        })

        //监听提交
        var index = parent.layer.getFrameIndex(window.name); //获取窗口索引
        form.on('submit(demo1)', function (data) {
            $.ajax({
                url: "update",
                type: "POST",
                contentType: 'application/json',
                dataType: 'json',
                data: JSON.stringify(data.field),
                success: function (data) {
                    layer.msg(data.msg, {time: 500},
                        function () {
                            parent.layer.close(index);
                        });
                }
            });
            return false;
        });

        //显示密码输入框
        form.on('select(rules)', function (data) {
            if(data.value == "1"){
                $('#password').show();
            }else{
                $('#password').hide();
            }
        });

        if("${survey.rules}"==="1"){
            $('#password').show();
        }

        form.val('myform', {

```

```

        "title":"${survey.title}",
        "remark":"${survey.remark}",
        "bounds":"${survey.bounds}",
        "startTime":'<fmt:formatDate value="${survey.startTime}"
pattern="yyyy-MM-dd HH:mm:ss" />',
        "endTime":'<fmt:formatDate value="${survey.endTime}" pattern="yyyy-
MM-dd HH:mm:ss" />',
        "rules":"${survey.rules}",
        "password":"${survey.password}",
        <c:if test="${survey.anon eq 0}">
        "anon":"${survey.anon}",
        </c:if>
    })
});

</script>

</body>
</html>

```

2 修改ServiceController.java

```

@PostMapping("/create")
@ResponseBody
public Map<String,Object> create(@RequestBody Survey survey,
HttpServletRequest request){

    Admin currAdmin= SessionUtils.getAdmin(request);
    survey.setCreator(currAdmin.getId());
    survey.setState(Survey.state_create);
    survey.setAnon(survey.getAnon() !=null?0:1);
    int result =surveyService.create(survey);
    if(result<=0){
        return MapControl.getInstance().error().getMap();
    }
    return MapControl.getInstance().success().getMap();
}

@PostMapping("/update")
@ResponseBody
public Map<String,Object> update(@RequestBody Survey survey){
    survey.setAnon(survey.getAnon() !=null?0:1);
    int result=surveyService.update(survey);
    if(result<=0){
        return MapControl.getInstance().error().getMap();
    }
    return MapControl.getInstance().success().getMap();
}

```

12.4 问卷查询-搜索功能

1 编写list.jsp

更改搜索项目

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>管理员列表</title>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/lib/font-awesome-4.7.0/css/font-
awesome.css" media="all">
    <link rel="stylesheet" href="../../static/css/public.css" media="all">

</head>
<body>
<div class="layui-mini-container">
    <div class="layui-mini-main">

        <fieldset class="table-search-fieldset">
            <legend>查询条件</legend>
            <div style="margin: 10px 10px 10px 10px">
                <form class="layui-form layui-form-pane" action="">
                    <div class="layui-form-item">
                        <div class="layui-inline">
                            <label class="layui-form-label">标题</label>
                            <div class="layui-input-inline">
                                <input type="text" name="title"
autocomplete="off" class="layui-input">
                            </div>
                        </div>
                        <div class="layui-inline">
                            <label class="layui-form-label">状态</label>
                            <div class="layui-input-inline">
                                <select name="state" lay-filter="state">
                                    <option value="">请选择</option>
                                    <option value="创建">创建</option>
                                    <option value="执行中">执行中</option>
                                    <option value="结束">结束</option>
                                </select>
                            </div>
                        </div>
                    </div>
                    <div class="layui-inline">
                        <button type="submit" class="layui-btn " lay-submit
lay-filter="data-search-btn"><i class="layui-icon">□</i> 搜 索</button>
                    </div>
                </div>
            </form>
        </fieldset>
    </div>
</div>
```

```

        </div>
    </fieldset>

    <script type="text/html" id="toolbarDemo">
        <div class="layui-btn-container">
            <button class="layui-btn layui-btn-sm data-add-btn" lay-
event="add"> <i class="fa fa-plus"></i> 添加</button>
            <button class="layui-btn layui-btn-sm layui-btn-sm data-delete-
btn" lay-event="edit"><i class="fa fa-pencil"></i> 修改 </button>
            <button class="layui-btn layui-btn-sm layui-btn-danger data-
delete-btn" lay-event="delete"><i class="fa fa-remove"></i> 删除 </button>

        </div>
    </script>

    <table class="layui-hide" id="currentTableId" lay-
filter="currentTableFilter"></table>

</div>
</div>
<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<script>
    layui.use(['form', 'table'], function () {
        var $ = layui.jquery,
            form = layui.form,
            table = layui.table;

        table.render({
            elem: '#currentTableId',
            url: 'query',
            toolbar: '#toolbarDemo',
            contentType: 'application/json',
            method: "post",
            defaultToolbar: ['filter', 'exports', 'print'],
            cols: [[
                {type: "checkbox", width: 50},
                {field: 'id', width: 80, title: 'ID', sort: true},
                {field: 'title', title: '问卷标题'},
                {field: 'startTime', width: 130, title: '开始时间', sort: true},
                {field: 'endTime', width: 130, title: '结束时间'},
                {field: 'state', width: 130, title: '状态'},
                {field: 'creator', width: 100, title: '创建人'},
                {field: 'createTime', width: 130, title: '创建时间'}
            ]],
            limits: [10, 15, 20, 25, 50, 100],
            limit: 15,
            page: true,
            skin: 'line'
        });

        // 监听搜索操作
        form.on('submit(data-search-btn)', function (data) {
            var result = JSON.stringify(data.field);
            //layer.alert(result, {
            //    title: '最终的搜索信息'
            //});

```

```

//执行搜索重载
table.reload('currentTableId', {
    page: {
        curr: 1
    },
    contentType: 'application/json',
    where: data.field
}, 'data');

return false;
});

/**
 * toolbar监听事件
 */
table.on('toolbar(currentTableFilter)', function (obj) {
    if (obj.event === 'add') { // 监听添加操作
        var index = layer.open({
            title: '添加问卷',
            type: 2,
            shade: 0.2,
            maxmin: false,
            shadeClose: true,
            area: ['80%', '80%'],
            content: 'create',
            end: function () {
                table.reload('currentTableId');
            }
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
    } else if (obj.event === 'delete') { // 监听删除操作
        var checkStatus = table.checkStatus('currentTableId')
            , data = checkStatus.data;
        var arr=[];
        for(index in data){
            arr.push(data[index].id);
        }
        if(arr.length <1){
            layer.msg("请至少选择一行数据删除",{time:1000});
            return;
        }
        $.ajax({
            url: "delete",
            type: "POST",
            dataType: 'json',
            data: 'ids='+arr.join(","),
            success: function (data) {
                layer.msg(data.msg, {time: 500},
                    function () {
                        table.reload('currentTableId');
                    }
                );
            }
        });
    } else if (obj.event === 'edit') { // 监听编辑操作

```

```

        var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
        var arr=[];
        for(index in data){
            arr.push(data[index].id);
        }

        if(arr.length !=1){
            layer.msg("请选择一行数据修改",{time:1000});
            return;
        }

        var index = layer.open({
            title: '修改问卷',
            type: 2,
            shade: 0.2,
            maxmin:false,
            shadeClose: true,
            area: ['80%', '80%'],
            content: 'detail?id='+arr[0],
            end:function(){
                table.reload('currentTableId');
            }
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
    }
});

//监听表格复选框选择
table.on('checkbox(currentTableFilter)', function (obj) {

});

table.on('tool(currentTableFilter)', function (obj) {
    var data = obj.data;
    if (obj.event === 'edit') {

        var index = layer.open({
            title: '修改问卷',
            type: 2,
            shade: 0.2,
            maxmin:true,
            shadeClose: true,
            area: ['100%', '100%'],
            content: '../page/table/edit.html',
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
        return false;
    } else if (obj.event === 'delete') {
        layer.confirm('真的删除行么', function (index) {
            obj.del();
            layer.close(index);
        });
    }
}

```



```

    });

    });
</script>

</body>
</html>

```

2 解决创建人是id值的问题

1 创建系统初始化类

获得tb_admin中所有admin的id和name信息

```

package com.yanzhen.utils;

import com.yanzhen.entity.Admin;
import com.yanzhen.mapper.AdminDao;
import org.springframework.beans.factory.annotation.Autowired;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

public class SystemInit {
    @Autowired
    private AdminDao adminDao;

    public static Map<Integer, Admin> adminMap = new HashMap<>();

    public void init(){
        List<Admin> list = adminDao.query(null);
        for (Admin admin : list) {
            adminMap.put(admin.getId(), admin);
        }
        System.out.println("初始化加载数据..." + adminMap);
    }
}

```

2 为该类配置applicationContext.xml的信息

```
<bean class="com.yanzhen.utils.SystemInit" init-method="init"></bean>
```

3 测试 运行项目

```

Closing non transactional SqlSession [org.apache.ibatis.session.defaults.DefaultSqlSession@280079c]
*****{1=com.yanzhen.entity.Admin@73a542c4, 2=com.yanzhen.entity.Admin@7402ba7, 3=com.yanzhen.entity
.Admin@1e0d31a9, 4=com.yanzhen.entity.Admin@6bee0d9f, 6=com.yanzhen.entity.Admin@3204c144, 7=com.yanzhen.entity.Admin@349fd9b0,
8=com.yanzhen.entity.Admin@43e826f3, 9=com.yanzhen.entity.Admin@58f149e4, 13=com.yanzhen.entity.Admin@4c6bad5f, 14=com.yanzhen
.entity.Admin@6f3bf3b, 15=com.yanzhen.entity.Admin@4aedeaa8, 16=com.yanzhen.entity.Admin@56c219b0, 17=com.yanzhen.entity
.Admin@47524762, 29=com.yanzhen.entity.Admin@5cc332d4, 30=com.yanzhen.entity.Admin@2e6afc7a, 31=com.yanzhen.entity.Admin@68237f9e
32=com.yanzhen.entity.Admin@3d1f5caf, 33=com.yanzhen.entity.Admin@780f8c4a, 34=com.yanzhen.entity.Admin@17f361d, 35=com.yanzhen
.entity.Admin@72536623, 36=com.yanzhen.entity.Admin@35eea6e2, 37=com.yanzhen.entity.Admin@310701c9, 38=com.yanzhen.entity
.Admin@4930162a, 39=com.yanzhen.entity.Admin@203b1351, 40=com.yanzhen.entity.Admin@60226802, 41=com.yanzhen.entity.Admin@69222486
42=com.yanzhen.entity.Admin@20eaece, 43=com.yanzhen.entity.Admin@5833d019, 59=com.yanzhen.entity.Admin@2b22c165, 60=com.yanzhen
.entity.Admin@535cd693, 61=com.yanzhen.entity.Admin@24b7dea8, 62=com.yanzhen.entity.Admin@62c2713f, 63=com.yanzhen.entity
.Admin@5fd6adeb, 64=com.yanzhen.entity.Admin@6a8b10cc, 65=com.yanzhen.entity.Admin@7053c1ab, 66=com.yanzhen.entity.Admin@1e91d17,
67=com.yanzhen.entity.Admin@6b3a5134, 68=com.yanzhen.entity.Admin@590521ef, 69=com.yanzhen.entity.Admin@3735edfd, 70=com.yanzhen
.entity.Admin@7e7d4fa9, 71=com.yanzhen.entity.Admin@477702a3, 72=com.yanzhen.entity.Admin@370d49fb, 73=com.yanzhen.entity

```

获取成功

4 将获取的信息发送出去

修改SurveyController.java 的query方法

```
@PostMapping("/query")
@ResponseBody
public Map<String, Object> query(@RequestBody Survey survey, ModelMap modelMap){
    List<Survey> list=surveyService.query(survey);
    //将创建者信息写入survey对象
    for(Survey entity:list){
        entity.setAdmin(SystemInit.adminMap.get(entity.getCreator()));
    }
    Integer count=surveyService.count(survey);
    return MapControl.getInstance().page(list, count).getMap();
}
```

5 list.jsp

```
{field: 'account', width: 100, title: '创建人',templet:'<div>{{d.admin.name}}
</div>'},
```

12.5 问卷状态改变

1 将问卷的有关时间的属性都设置成 年-月-日 时:分:秒

修改spring-mvc.xml 的时间格式转换器

```
<bean
class="org.springframework.http.converter.json.MappingJackson2HttpMessageConvert
er">
    <property name="objectMapper">
        <bean class="com.fasterxml.jackson.databind.ObjectMapper">
            <property name="dateFormat">
                <bean class="java.text.SimpleDateFormat">
                    <constructor-arg type="java.lang.String" value="yyyy-MM-dd
HH:mm:ss"/>
                </bean>
            </property>
        </bean>
    </property>
</bean>
```

add.jsp中加入时间约束

```
<div class="layui-form-item">
    <div class="layui-inline">
        <label class="layui-form-label">开始时间</label>
        <div class="layui-input-inline">
            <input type="text" name="startTime" id="startTime" lay-
verify="datetime" placeholder="yyyy-MM-dd HH:mm:ss" autocomplete="off"
class="layui-input">
        </div>
    </div>
    <div class="layui-inline">
```

```

        <label class="layui-form-label">结束时间</label>
        <div class="layui-input-inline">
            <input type="text" name="endTime" id="endTime" lay-
verify="datetime" autocomplete="off" placeholder="yyyy-MM-dd HH:mm:ss"
class="layui-input">
        </div>
    </div>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<!-- 注意：如果你直接复制所有代码到本地，上述js路径需要改成你本地的 -->
<script>
var form =
layui.form,$=layui.jquery,layer=layui.layer,laydate=layui.laydate,layer =
layui.layer;
laydate.render({
    elem:"#startTime",
    type: 'datetime'
})
laydate.render({
    elem:"#endTime",
    type: 'datetime'
})
</script>

```

此时发现，只要时间带上秒，就无法添加，于是寻找错误

重新复制一下spring-mvc.xml配置即可

2 SpringBoot时间监督

操作步骤：

- 1.在启动类上加入 @EnableScheduling 注解
- 2.在需求的方法上加入 @Scheduled(cron="0 0 0 ? * SUN") 注解
- 3.配置cron规则 (cron="0 0 0 ? * SUN")

Cron规则

Cron表达式是一个字符串，字符串以5或6个空格隔开，分为6或7个域，每一个域代表一个含义，Cron有如下两种语法格式：

- (1) Seconds Minutes Hours DayofMonth Month DayofWeek Year
- (2) Seconds Minutes Hours DayofMonth Month DayofWeek

一、结构

corn从左到右（用空格隔开）：秒 分 小时 月份中的日期 月份 星期中的日期 年份

二、各字段的含义

字段	允许值	允许的特殊字符
秒 (Seconds)	0~59的整数	, - * / 四个字符
分 (Minutes)	0~59的整数	, - * / 四个字符
小时 (Hours)	0~23的整数	, - * / 四个字符
日期 (DayofMonth)	1~31的整数 (但是你需要考虑你月的天数)	, - * ? / L W C 八个字符
月份 (Month)	1~12的整数或者 JAN-DEC	, - * / 四个字符
星期 (DayofWeek)	1~7的整数或者 SUN-SAT (1=SUN)	, - * ? / L C # 八个字符
年(可选, 留空) (Year)	1970~2099	, - * / 四个字符

注意事项:

每一个域都使用数字,但还可以出现如下特殊字符,它们的含义是:

- (1) *: 表示匹配该域的任意值。假如在Minutes域使用*,即表示每分钟都会触发事件。
- (2) ?: 只能用在DayOfMonth和DayOfWeek两个域。它也匹配域的任意值,但实际不会。因为DayOfMonth和DayOfWeek会相互影响。例如想在每月的20日触发调度,不管20日到底是星期几,则只能使用如下写法: 13 13 15 20 * ?,其中最后一位只能用?,而不能使用*,如果使用*表示不管星期几都会触发,实际上并不是这样。
- (3) -: 表示范围。例如在Minutes域使用5-20,表示从5分到20分钟每分钟触发一次
- (4) /: 表示起始时间开始触发,然后每隔固定时间触发一次。例如在Minutes域使用5/20,则意味着5分钟触发一次,而25,45等分别触发一次。
- (5) ,: 表示列出枚举值。例如:在Minutes域使用5,20,则意味着在5和20分每分钟触发一次。
- (6) L: 表示最后,只能出现在DayOfWeek和DayOfMonth域。如果在DayOfWeek域使用5L,意味着在最后的一个星期四触发。
- (7) W: 表示有效工作日(周一到周五),只能出现在DayOfMonth域,系统将在离指定日期的最近的有效工作日触发事件。例如:在DayOfMonth使用5W,如果5日是星期六,则将在最近的工作日:星期五,即4日触发。如果5日是星期天,则在6日(周一)触发;如果5日在星期一到星期五中的一天,则就在5日触发。另外一点,W的最近寻找不会跨过月份。

(8) LW:这两个字符可以连用,表示在某个月最后一个工作日,即最后一个星期五。

(9) #: 用于确定每个月第几个星期几,只能出现在DayOfMonth域。例如在4#2,表示某月的第二个星期三。

三、常用表达式例子

- (1) 0 0 2 1 * ? * 表示在每月的1日的凌晨2点调整任务
- (2) 0 15 10 ? * MON-FRI 表示周一到周五每天上午10:15执行作业
- (3) 0 15 10 ? 6L 2002-2006 表示2002-2006年的每个月的最后一个星期五上午10:15执行作
- (4) 0 0 10,14,16 * * ? 每天上午10点,下午2点,4点
- (5) 0 0/30 9-17 * * ? 朝九晚五工作时间内每半小时
- (6) 0 0 12 ? * WED 表示每个星期三中午12点
- (7) 0 0 12 * * ? 每天中午12点触发
- (8) 0 15 10 ? * * 每天上午10:15触发
- (9) 0 15 10 * * ? 每天上午10:15触发
- (10) 0 15 10 * * ? * 每天上午10:15触发
- (11) 0 15 10 * * ? 2005 2005年的每天上午10:15触发
- (12) 0 * 14 * * ? 在每天下午2点到下午2:59期间的每1分钟触发
- (13) 0 0/5 14 * * ? 在每天下午2点到下午2:55期间的每5分钟触发
- (14) 0 0/5 14,18 * * ? 在每天下午2点到2:55期间和下午6点到6:55期间的每5分钟触发
- (15) 0 0-5 14 * * ? 在每天下午2点到下午2:05期间的每1分钟触发
- (16) 0 10,44 14 ? 3 WED 每年三月的星期三的下午2:10和2:44触发
- (17) 0 15 10 ? * MON-FRI 周一至五的上午10:15触发
- (18) 0 15 10 15 * ? 每月15日上午10:15触发
- (19) 0 15 10 L * ? 每月最后一日的上午10:15触发
- (20) 0 15 10 ? * 6L 每月的最后一个星期五上午10:15触发
- (21) 0 15 10 ? * 6L 2002-2005 2002年至2005年的每月的最后一个星期五上午10:15触
- (22) 0 15 10 ? * 6#3 每月的第三个星期五上午10:15触发

3 做一个定时任务来修改问卷的状态

sql语句更新原理

```
select id from tb_survey where start_time <=now() and end_time>now() and state = '创建'

update tb_survey where state='执行中' where...

select * from tb_survey where end_time<now() and state = '执行中'

update tb_survey where state='结束' where...
```

1 SurveyDao中添加方法

```
package com.yanzhen.mapper;

import java.util.List;
import java.util.Map;

import com.yanzhen.entity.Survey;

public interface SurveyDao {
    public int create(Survey pi);
    public int delete(Map<String, Object> paramMap);
    public int update(Map<String, Object> paramMap);
    public List<Survey> query(Map<String, Object> paramMap);
    public Survey detail(Map<String, Object> paramMap);
    public int count(Map<String, Object> paramMap);
    public List<Integer> queryByState(String state); //新增方法
    public List<Integer> queryByStateForExec(String state); //新增方法
}
```

2 修改SurveyMapper.xml

SurveyMapper.xml中添加sql配置

<![CDATA[]]> 这个标记所包含的内容将表示为**纯文本**

```
<select id="queryByState" parameterType="string" resultType="int">
    <![CDATA[select id from tb_survey where start_time <=now() and end_time>now()
and state = #{_parameter}]]>
</select>
<select id="queryByStateForExec" parameterType="string" resultType="int">
    <![CDATA[select id from tb_survey where end_time<now() and state = #
#{_parameter}]]>
</select>
```

mybatis中<![CDATA[]]> 的作用

在使用mybatis 时我们sql是写在xml 映射文件中，如果写的sql中有一些特殊的字符的话，在解析xml文件的时候会被转义，但我们不希望他被转义，所以我们要使用<![CDATA[]]>来解决。

<![CDATA[]]>是什么，这是XML语法

如果文本包含了很多的"<"字符 "<="和"&"字符——就象程序代码一样，那么最好把他们都放到CDATA部件中

3 修改SurveyService.java

添加updateState () 方法

```
public void updateState(){
    List<Integer> list = surveyDao.queryByState(Survey.state_create);
    for (Integer id : list) {

        surveyDao.update(MapParameter.getInstance().add("updateState",Survey.state_exec)
        ).add("id",id).getMap());
    }
    List<Integer> list2 = surveyDao.queryByStateForExec(Survey.state_exec);
    for (Integer id : list2) {

        surveyDao.update(MapParameter.getInstance().add("updateState",Survey.state_over)
        ).add("id",id).getMap());
    }
}
```

4 新建一个类ScheduleTask.java

```
package com.yanzhen.utils;
import org.springframework.scheduling.annotation.EnableScheduling;
import org.springframework.scheduling.annotation.Scheduled;

@EnableScheduling
public class ScheduleTask {
    @Autowired
    private SurveyService surveyService;
    /**
     * 调查问卷状态的任务
     */
    @Scheduled(fixedRate=20000) //每20秒执行一次
    // @Scheduled(cron = "* * */1 * * ?") 每一分钟触发一次
    public void state(){
        System.out.println("执行任务....");
        surveyService.updateState();
    }
}
```

5 修改ApplicationContext.xml

创建ScheduleTask类

```
<bean class="com.yanzhen.utils.ScheduleTask"/>
```

4 解决log4j日志乱码问题

log4j中配置字符集

```
log4j.appender.A1.Encoding=UTF-8
```

13 问卷设计

13.1 question页面编写

老师已经写好的mypaper.html 拿来用

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <link rel="stylesheet" href="lib/layui-src/css/layui.css" media="all">
  <style>
    html,body{
      height: 100%;
      background-color: #F2F2F2;
    }
    .left{
      position: absolute;top: 0px;bottom: 10px;background-color:
red;width: 300px;background-color: #FAFAFA;border: 1px solid #e0e0e0;
    }
    .left ul{
      margin: 0px;
      padding:0px;
      width: 250px;
      margin:auto;
      margin-top: 30px;
    }
    .left ul li{
      height: 35px;
      line-height: 35px;
      margin-top: 10px;
      background-color: #FFFFFF;
      border: 1px solid #e0e0e0;
      padding-left: 10px;
      box-sizing: border-box;
    }
    .left ul li a{
      display: block;
    }
    .main{
      position: absolute;top: 0px;bottom: 10px;right: 0px;left:
320px;background-color: #ffffff;border: 1px solid #e0e0e0;
    }
    .main ul{
      width: 100%;
      margin:0px;
      padding: 0px;
    }
    .main ul li{
      margin-top: 50px;
      border-top: 1px solid #ffffff;
      border-bottom: 1px solid #ffffff;
    }
    .main ul li>div{
      padding-left: 50px;
```



```

}
.main ul li:hover{
    background-color: #FAFAFA;
    border-top: 1px solid #e0e0e0;
    border-bottom: 1px solid #e0e0e0;
}
.main label{
    font-size: 14px;
    padding: 10px 10px;
    color: #666666;
}
.edit{
    padding: 5px 5px;
    border: 1px solid #cccccc;
    min-height: 20px;
    width: 80%;
    margin-top: 10px;
    margin-left: 10px;
    border-radius: 2px;
    color: #666666;
    background-color: #ffffff;
    cursor: default;
}
.no_edit{
    padding: 5px 5px;
    min-height: 20px;
    width: 80%;
    margin-top: 10px;
    margin-left: 10px;
    border-radius: 2px;
    color: #666666;
}
.required{
    margin-left: 10px;
    margin-top: 10px;
    color: #666666;
}
.required select, input{
    border: 1px solid #cccccc;
    height: 30px;
    border-radius: 2px;
    margin-left: 10px;
}
.options{
}
.options .edit{
    width: 75%;
    float: left;
}
.options div{
    margin-top: 10px;
}
.que-btn{
    padding-top: 10px;
    padding-bottom: 10px;
    padding-left: 60px !important;
    clear: both;
}

```

```

        [contenteditable]:focus{outline: none;border: 1px solid #479de6;}
        .show{
            display: block;
        }
        .hide{
            display: none;
        }
    }
</style>
</head>
<body>
<div class="layuimini-container" style="position: absolute;bottom: 0px;top:
0px;background-color: #F2F2F2;left: 0px;right:0px;margin: 10px;">
    <div class="left">
        <ul>
            <li>
                <a href="javascript:void(0)" class="radio_type">
                    <i class="layui-icon layui-icon-about"></i>
                    单选题
                </a>
            </li>
            <li>
                <a href="javascript:void(0)" class="checkbox_type">
                    <i class="layui-icon layui-icon-about"></i>
                    多选题
                </a>
            </li>
            <li>
                <a href="javascript:void(0)" class="input_type">
                    <i class="layui-icon layui-icon-about"></i>
                    单行文本
                </a>
            </li>
            <li>
                <a href="javascript:void(0)" class="text_type">
                    <i class="layui-icon layui-icon-about"></i>
                    多行文本
                </a>
            </li>
        </ul>
    </div>
    <div class="main" id="main" style="overflow: auto">
        <ul class="content">
        </ul>
    </div>
</div>

<div class="radioTemplate" style="display: none">
    <li class="using">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>
            <div class="edit" contenteditable="true">描述</div>
        </div>
        <div class="required">
            <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            必填项
            <div class="layui-unselect layui-form-checkbox checkscore" lay-
skin="primary"><i class="layui-icon layui-icon-ok"></i></div>

```

```

        统计分值
        <span class="edit hide scorevalue" contenteditable="true"
style="width: 60px;padding: 7px 20px;">5</span>
    </div>
    <div class="options">
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
    </div>
    <div class="que-btn">
        <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysaveitem" lay-event="addItem"> 新增选项 </button>
        <span class="layui-text">编辑状态下，双击选项删除</span>
    </div>
</li>
</div>

<div class="checkboxTemplate" style="display: none">
    <li class="using">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>
            <div class="edit" contenteditable="true">描述</div>
        </div>
        <div class="required">
            <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            必填项
            <div class="layui-unselect layui-form-checkbox checkscore" lay-
skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            统计分值
            <span class="edit hide scorevalue" contenteditable="true"
style="width: 60px;padding: 7px 20px;">5</span>
        </div>
        <div class="options">
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
        </div>
        <div class="que-btn">
            <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysaveitem" lay-event="addItem"> 新增选项 </button>
            <span class="layui-text">编辑状态下，双击选项删除</span>
        </div>
    </li>

```

```

</div>

<div class="inputTemplate" style="display: none">
  <li class="using">
    <div class="title">
      <div class="edit" contenteditable="true">标题</div>
      <div class="edit" contenteditable="true">描述</div>
    </div>
    <div class="required">
      <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
      必填项
    </div>
    <div class="que-btn">
      <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
      <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
      <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
    </div>
  </li>
</div>

<div class="textTemplate" style="display: none">
  <li class="using">
    <div class="title">
      <div class="edit" contenteditable="true">标题</div>
      <div class="edit" contenteditable="true">描述</div>
    </div>
    <div class="required">
      <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
      必填项
    </div>
    <div class="que-btn">
      <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
      <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
      <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
    </div>
  </li>
</div>

<script src="lib/layui-src/layui.js" charset="utf-8"></script>
<script src="js/lay-config.js" charset="utf-8"></script>
<script>
  layui.use(['jquery', 'layer'], function () {
    let $ = layui.jquery, layer = layui.layer;

    $('.radio_type, .checkbox_type, .input_type, .text_type').on('click', function(){
      if(isEditing()){
        layer.msg("正在编辑，请先保存");
        return;
      }
    });
  });

```

```

    }
    $('.content').find(".using").removeClass("using");
    let template = loadTemplation($(this).prop("className"));
    $('.content').append($(template));
    $('#main').animate({scrollTop: $(".content").height()}, 300);
});

```

//删除操作

```

$(".content").on('click', '.mydel', function(){
    //判断是否是当前编辑项
    var flag = $(this).parent().parent().hasClass("using");
    if(flag){
        $(this).parent().parent().remove();
    }else{
        if(isEditing()){
            layer.msg("正在编辑，请先保存");
            return;
        }else{
            $(this).parent().parent().remove();
        }
    }
    //清除using状态
    clearUsing();
});

```

//判断是否有正在编辑的项目

```

function isEditing(){
    var length = $('.content').find(".using").length;
    if(length == 0){
        return false;
    }else{
        return true;
    }
}

```

//保存之后，清除保存状态

```

function clearUsing(){
    var obj = $('.content').find(".using");
    obj.removeClass("using");
}

```

//保存操作

```

$(".content").on('click', '.mysave', function(){
    var obj = $(this).parents(".using");
    obj.find(".edit").prop("contenteditable", false);
    obj.find(".edit").removeClass("edit").addClass("no_edit");
    //隐藏
    obj.find('.mysave').hide();
    obj.find('.mysaveitem').hide();
    obj.find('.myedit').show();
    clearUsing();
});

```

//新增选项操作

```

$(".content").on('click', '.mysaveitem', function(){
    var obj = $(this).parents(".using");

```

```

        obj.find(".options").append($('
```

```

        case "radio_type":
            return $('radioTemplate').html();
        case "checkbox_type":
            return $('checkboxTemplate').html();
        case "input_type":
            return $('inputTemplate').html();
        case "text_type":
            return $('textTemplate').html();
    }
}
});
</script>
</body>
</html>

```

13.2 修改List.jsp页面

加入按钮 设计问卷、预览问卷、发布问卷、问卷详情

添加相应的监听事件

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>管理员列表</title>
    <meta name="renderer" content="webkit">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1">
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/lib/font-awesome-4.7.0/css/font-
awesome.css" media="all">
    <link rel="stylesheet" href="../../static/css/public.css" media="all">
</head>
<body>
<div class="layuiimini-container">
    <div class="layuiimini-main">
        <fieldset class="table-search-fieldset">
            <legend>查询条件</legend>
            <div style="margin: 10px 10px 10px 10px">
                <form class="layui-form layui-form-pane" action="">
                    <div class="layui-form-item">
                        <div class="layui-inline">
                            <label class="layui-form-label">标题</label>
                            <div class="layui-input-inline">
                                <input type="text" name="title"
autocomplete="off" class="layui-input">
                            </div>
                        </div>
                        <div class="layui-inline">
                            <label class="layui-form-label">状态</label>
                            <div class="layui-input-inline">
                                <select name="state" lay-filter="state">
                                    <option value="">请选择</option>

```

```

        <option value="创建">创建</option>
        <option value="执行中">执行中</option>
        <option value="结束">结束</option>
    </select>
</div>
</div>
<div class="layui-inline">
    <button type="submit" class="layui-btn " lay-submit
lay-filter="data-search-btn"><i class="layui-icon">□</i> 搜 索</button>
</div>
</div>
</form>
</div>
</fieldset>

<script type="text/html" id="toolbarDemo">
    <div class="layui-btn-container">
        <button class="layui-btn layui-btn-sm data-add-btn" lay-
event="add">
            <i class="fa fa-plus"></i>
            添加 </button>
        <button class="layui-btn layui-btn-sm layui-btn-sm data-delete-
btn" lay-event="edit">
            <i class="fa fa-pencil"></i>
            修改 </button>
        <button class="layui-btn layui-btn-sm layui-btn-danger data-
delete-btn" lay-event="delete">
            <i class="fa fa-remove"></i>
            删除 </button>
        <button class="layui-btn layui-btn-sm layui-btn-sm data-delete-
btn" lay-event="editque">
            <i class="fa fa-pencil"></i>
            设计问卷 </button>
        <button class="layui-btn layui-btn-sm layui-btn-sm data-delete-
btn" lay-event="preview">
            <i class="fa fa-search"></i>
            预览问卷 </button>
        <button class="layui-btn layui-btn-sm layui-btn-sm data-delete-
btn" lay-event="publish">
            <i class="fa fa-search"></i>
            发布问卷 </button>
        <button class="layui-btn layui-btn-sm layui-btn-sm data-delete-
btn" lay-event="query_detail">
            <i class="fa fa-search"></i>
            问卷详情 </button>

    </div>
</script>
<table class="layui-hide" id="currentTableId" lay-
filter="currentTableFilter"></table>

</div>
</div>
<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<script>
    layui.use(['form', 'table'], function () {
        var $ = layui.jquery,

```



```

        form = layui.form,
        table = layui.table;

    table.render({
        elem: '#currentTableId',
        url: 'query',
        toolbar: '#toolbarDemo',
        contentType: 'application/json',
        method: "post",
        defaultToolbar: ['filter', 'exports', 'print'],
        cols: [[
            {type: "checkbox", width: 50},
            {field: 'id', width: 80, title: 'ID', sort: true},
            {field: 'title', title: '问卷标题'},
            {field: 'url', width: 500, title: '问卷URL'},
            {field: 'startTime', width: 200, title: '开始时间'},
            {field: 'endTime', width: 200, title: '结束时间'},
            {field: 'state', width: 130, title: '状态'},
            {field: 'name', width: 100, title: '创建人', templet: '<div>
{{d.admin.name}}</div>'},
            {field: 'createTime', width: 200, title: '创建时间'}
        ]],
        limits: [10, 15, 20, 25, 50, 100],
        limit: 15,
        page: true,
        skin: 'line'
    });

    // 监听搜索操作
    form.on('submit(data-search-btn)', function (data) {
        var result = JSON.stringify(data.field);
        /*layer.alert(result, {
            title: '最终的搜索信息'
        });*/
        //执行搜索重载
        table.reload('currentTableId', {
            page: {
                curr: 1
            },
            contentType: 'application/json',
            where: data.field
        }, 'data');

        return false;
    });

    /**
     * toolbar监听事件
     */
    table.on('toolbar(currentTableFilter)', function (obj) {
        if (obj.event === 'add') { // 监听添加操作
            var index = layer.open({
                title: '添加用户',
                type: 2,
                shade: 0.2,
                maxmin: false,
                shadeClose: true,
                area: ['80%', '80%'],

```

```

        content: 'create',
        end:function(){
            table.reload('currentTableId');
        }
    });
    $(window).on("resize", function () {
        layer.full(index);
    });
} else if (obj.event === 'delete') { // 监听删除操作
    var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
    var arr=[];
    for(index in data){
        arr.push(data[index].id);
    }
    if(arr.length <=0){
        layer.msg("请选择至少一行数据删除",{time:1000});
        return;
    }
    $.ajax({
        url:"delete",
        type:"POST",
        dataType:'json',
        data:'ids='+arr.join(","),
        success:function(data){
            layer.msg(data.msg,{time:500},
                function(){
                    table.reload('currentTableId');
                });
        }
    });
} else if (obj.event === 'edit') { // 监听编辑操作

    var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
    var arr=[];
    for(index in data){
        arr.push(data[index].id);
    }

    if(arr.length !=1){
        layer.msg("请选择一行数据修改",{time:1000});
        return;
    }
    var index = layer.open({
        title: '修改问卷',
        type: 2,
        shade: 0.2,
        maxmin:false,
        shadeClose: true,
        area: ['80%', '80%'],
        content: 'detail?id='+arr[0],
        end:function(){
            table.reload('currentTableId');
        }
    });
    $(window).on("resize", function () {
        layer.full(index);
    });

```

```

});
}else if (obj.event === 'editque') { // 监听编辑操作

    var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
    var arr=[];
    for(index in data){
        arr.push(data[index].id);
    }

    if(arr.length !=1){
        layer.msg("请选择一行数据修改",{time:1000});
        return;
    }
    var index = layer.open({
        title: '设计问卷',
        type: 2,
        shade: 0.2,
        maxmin:false,
        shadeClose: true,
        area: ['80%', '80%'],
        content: 'question?id='+arr[0],
        end:function(){
            table.reload('currentTableId');
        }
    });
    $(window).on("resize", function () {
        layer.full(index);
    });
}else if(obj.event === 'preview'){
    var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
    var arr=[];
    for(index in data){
        arr.push(data[index].id);
    }

    if(arr.length !=1){
        layer.msg("请选择一行数据修改",{time:1000});
        return;
    }
    window.open("preview/"+arr[0]);
}else if(obj.event === 'publish'){
    var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
    var arr=[];
    for(index in data){
        arr.push(data[index].id);
    }

    if(arr.length !=1){
        layer.msg("请选择一行数据修改",{time:1000});
        return;
    }

    $.ajax({
        url:"publish",
        type:"POST",

```

```

        dataType: 'json',
        data: 'id='+arr[0],
        success: function(data){
            layer.msg(data.msg, {time: 2000},
                function(){
                    table.reload('currentTableId');
                });
        }
    });

    }else if(obj.event === 'query_detail'){
        var checkStatus = table.checkStatus('currentTableId')
        , data = checkStatus.data;
        var arr=[];
        for(index in data){
            arr.push(data[index].id);
        }
        if(arr.length !=1){
            layer.msg("请选择一行数据查看", {time: 1000});
            return;
        }
        if(data[0].url == ""){
            layer.msg("请先发布再查看详情", {time: 1000});
            return;
        }
        window.open("query_detail/"+arr[0]);
    }

});

//监听表格复选框选择
table.on('checkbox(currentTableFilter)', function (obj) {

});

table.on('tool(currentTableFilter)', function (obj) {
    var data = obj.data;
    if (obj.event === 'edit') {
        var index = layer.open({
            title: '编辑用户',
            type: 2,
            shade: 0.2,
            maxmin: true,
            shadeClose: true,
            area: ['100%', '100%'],
            content: '../page/table/edit.html',
        });
        $(window).on("resize", function () {
            layer.full(index);
        });
        return false;
    } else if (obj.event === 'delete') {
        layer.confirm('真的删除行么', function (index) {
            obj.del();
            layer.close(index);
        });
    }
});

```

```

});

});
</script>

</body>
</html>

```

修改下src、href等引用

13.3 将CSS样式引用分离出去

新建question.css

```
<link rel="stylesheet" href="../static/lib/css/question.css" media="all">
```

13.4 编写保存提交方法

console.log () 在前端控制台打印

html ()显示所选标签的内容

question.jsp

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../static/css/question.css" media="all">
</head>
<body>
<div class="layui-mini-container" style="position: absolute; bottom: 0px; top:
0px; background-color: #F2F2F2; left: 0px; right: 0px; margin: 10px;">
    <div class="left">
        <ul>
            <li>
                <a href="javascript:void(0)" class="radio_type">
                    <i class="layui-icon layui-icon-about"></i>
                    单选题
                </a>
            </li>
            <li>
                <a href="javascript:void(0)" class="checkbox_type">
                    <i class="layui-icon layui-icon-about"></i>
                    多选题
                </a>
            </li>
            <li>
                <a href="javascript:void(0)" class="input_type">
                    <i class="layui-icon layui-icon-about"></i>
                    单行文本
                </a>
            </li>

```

```

        <li>
            <a href="javascript:void(0)" class="text_type">
                <i class="layui-icon layui-icon-about"></i>
                多行文本
            </a>
        </li>
    </ul>
</div>
<div class="main" id="main" style="overflow: auto">
    <ul class="content">
    </ul>
</div>
</div>

<div class="radioTemplate" style="display: none">
    <li class="using radio">
        <input type="hidden">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>
            <div class="edit" contenteditable="true">描述</div>
        </div>
        <div class="required">
            <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            必填项
            <div class="layui-unselect layui-form-checkbox checkscore" lay-
skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            统计分值
            <span class="edit hide scorevalue" contenteditable="true"
style="width: 60px;padding: 7px 20px;">5</span>
            <span class="edit hide answer" title="答案序号" contenteditable="true"
style="width: 60px;padding: 7px 20px;"></span>
        </div>
        <div class="options">
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
        </div>
        <div class="que-btn">
            <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysaveitem" lay-event="addItem"> 新增选项 </button>
            <span class="layui-text">编辑状态下，双击选项删除</span>
        </div>
    </li>
</div>

<div class="checkboxTemplate" style="display: none">
    <li class="using checkbox">
        <input type="hidden">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>

```

```

        <div class="edit" contenteditable="true">描述</div>
    </div>
    <div class="required">
        <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
        必填项
        <div class="layui-unselect layui-form-checkbox checkscore" lay-
skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
        统计分值
        <span class="edit hide scorevalue" contenteditable="true"
style="width: 60px;padding: 7px 20px;">5</span>
        <span class="edit hide answer" title="答案序号" contenteditable="true"
style="width: 60px;padding: 7px 20px;"></span>
    </div>
    <div class="options">
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
    </div>
    <div class="que-btn">
        <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysaveitem" lay-event="addItem"> 新增选项 </button>
        <span class="layui-text">编辑状态下，双击选项删除</span>
    </div>
</li>
</div>

<div class="inputTemplate" style="display: none">
    <li class="using inputx">
        <input type="hidden">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>
            <div class="edit" contenteditable="true">描述</div>
        </div>
        <div class="required">
            <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            必填项
        </div>
        <div class="que-btn">
            <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
        </div>
    </li>
</div>

```

```

<div class="textTemplate" style="display: none">
  <li class="using">
    <input type="hidden">
    <div class="title">
      <div class="edit" contenteditable="true">标题</div>
      <div class="edit" contenteditable="true">描述</div>
    </div>
    <div class="required">
      <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
      必填项
    </div>
    <div class="que-btn">
      <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
      <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
      <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
    </div>
  </li>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<script src="../../static/js/lay-config.js" charset="utf-8"></script>
<script>
  layui.use(['jquery','layer'],function () {
    let $ = layui.jquery, layer = layui.layer;

    $('.radio_type,.checkbox_type,.input_type,.text_type').on('click',function(){
      if(isEditing()){
        layer.msg("正在编辑，请先保存");
        return;
      }
      $('.content').find(".using").removeClass("using");
      let template = loadTemplation($(this).prop("className"));
      $('.content').append($(template));
      $('#main').animate({scrollTop:$('.content').height()}, 300);
    });

    //删除操作
    $(".content").on('click', '.mydel',function(){
      //判断是否是当前编辑项
      var flag = $(this).parent().parent().hasClass("using");
      if(flag){
        $(this).parent().parent().remove();
      }else{
        if(isEditing()){
          layer.msg("正在编辑，请先保存");
          return;
        }else{
          $(this).parent().parent().remove();
        }
      }
    }
    //清除using状态
    clearUsing();
  });

```



```

        var id = $(this).parent().parent().find("input").val();
        if(id){
            $.ajax({
                url:"../question/delete",
                type:"POST",
                data:"ids="+id,
                dataType:'json',
                success:function (data) {
                    layer.msg("删除成功");
                }
            })
        }
    });

    //判断是否有正在编辑的项目
    function isEditing(){
        var length = $('.content').find(".using").length;
        if(length == 0){
            return false;
        }else{
            return true;
        }
    }

    //保存之后，清除保存状态
    function clearUsing(){
        var obj = $('.content').find(".using");
        obj.removeClass("using");
    }

    //保存操作
    $(".content").on('click', '.mysave', function(){
        var obj = $(this).parents(".using");
        obj.find(".edit").prop("contenteditable", false);
        obj.find(".edit").removeClass("edit").addClass("no_edit");
        //隐藏
        obj.find('.mysave').hide();
        obj.find('.mysaveitem').hide();
        obj.find('.myedit').show();
        clearUsing();

        //写入数据库
        save(obj);
    });

    init();
    function init(){
        var param = {};
        param.surveyId = ${survey.id};
        $.ajax({
            url:"../question/query",
            type:"POST",
            contentType:"application/json",
            data:JSON.stringify(param),
            dataType:'json',
            success:function (data) {

```

```

        var arr = data.data;
        for(var index=0;index<arr.length;index++){
            switch (arr[index].type) {
                case 1:
                    loadRadio(1,arr,index);
                    break;
                case 2:
                    loadRadio(2,arr,index);
                    break;
                case 3:
                    loadText(3,arr,index);
                    break;
                case 4:
                    loadText(4,arr,index);
                    break;
            }
        }
    }
}

function loadRadio(type,arr,index){
    let template;
    if(type == 1){
        template = $($('.radioTemplate').html());
    }else if(type == 2){
        template = $($('.checkboxTemplate').html());
    }
    //设置为div为非编辑状态
    template.find("div .edit").prop("contenteditable", false);
    template.removeClass("using");
    template.find("div .edit").removeClass("edit").addClass("no_edit");
    template.find("input").val(arr[index].id);
    //标题与描述
    template.find(".title").find("div").each(function(cnt){
        if(cnt == 0){
            $(this).html(arr[index].title);
        }else{
            $(this).html(arr[index].remark);
        }
    });
    //必填项
    template.find(".required").find("div").each(function(cnt){
        if(cnt == 0){
            if(arr[index].required==1){
                $(this).addClass("layui-form-checked");
            }else{
                $(this).removeClass("layui-form-checked");
            }
        }else{
            if(arr[index].test==1){
                $(this).addClass("layui-form-checked");
                template.find(".scorevalue").show();
                template.find(".answer").show();
                template.find(".scorevalue").html(arr[index].score);
            }else{
                $(this).removeClass("layui-form-checked");
            }
        }
    });
}

```

```

    }
  });

  //清空原有的预置选项
  template.find(".options").empty();
  var options = arr[index].options;
  var answerArray = [];
  for(var j=0;j<options.length;j++){
    template.find(".options").append($('
```

```

        template.find('.mysaveitem').hide();
        template.find('.myedit').show();
        $('.content').append(template);
    }

    function save(obj){
        var question = {};
        if(obj.hasClass("radio")){
            question.type = 1;
        }else if(obj.hasClass("checkbox")){
            question.type = 2;
        }else if(obj.hasClass("inputx")){
            question.type = 3;
        }else{
            question.type = 4;
        }
        //title+remark
        obj.find('.title').find('div').each(function(index){
            if(index == 0){
                question.title=$(this).html();
            }
            if(index == 1){
                question.remark=$(this).html();
            }
        });
        var _answer="";
        //是否必填
        obj.find('.required').find('div').each(function(index){
            if(index == 0){
                if($(this).hasClass('layui-form-checked')){
                    question.required=1;
                }else{
                    question.required=0;
                }
            }
            if(index == 1){
                if($(this).hasClass('layui-form-checked')){
                    question.test=1;
                    question.score=
obj.find('.required').find('.scorevalue').html();
                    _answer= obj.find('.required').find('.answer').html();
                }else{
                    question.test=0;
                    question.score=0;
                }
            }
        });
        if(question.type == 1 || question.type == 2){
            //选项
            var arr = [];
            var anArray = _answer.split(",");
            obj.find('.options').find('div').each(function(index){
                var option ={};
                option.opt = $(this).html();
                option.type = question.type;
                for(var i=0;i<anArray.length;i++){
                    if(anArray[i]==(index+1)){
                        option.answer = 1;
                    }
                }
            });
        }
    }

```

```

        }
    }
    arr.push(option);
});
question.options = arr;
}
question.surveyId = ${survey.id};
if(obj.find('input').val()){
    question.id = obj.find('input').val()
}
console.log(question);
$.ajax({
    url:"../question/create",
    type:"POST",
    contentType:"application/json",
    data:JSON.stringify(question),
    dataType:'json',
    success:function (data) {
        obj.find('input').val(data.id);
    }
})
}

//新增选项操作
$(".content").on('click', '.mysaveitem', function(){
    var obj = $(this).parents(".using");
    obj.find(".options").append($('
```

```

        if($(this).hasClass("layui-form-checked")){
            $(this).removeClass("layui-form-checked")
        }else{
            $(this).addClass("layui-form-checked")
        }
    }else{
        layer.msg("请切换编辑状态");
    }
});

//统计分值
$(".content").on('click', '.checkscore', function(){
    var flag = $(this).parent().parent().hasClass("using");
    if(flag){
        if($(this).hasClass("layui-form-checked")){
            $(this).addClass("layui-form-checked")
            $(this).siblings('.scorevalue').show();
            $(this).siblings('.answer').show();
        }else{
            $(this).removeClass("layui-form-checked")
            $(this).siblings('.scorevalue').hide();
            $(this).siblings('.answer').hide();
        }
    }else{
        layer.msg("请切换编辑状态");
    }
});

function loadTemplation(type){
    switch (type) {
        case "radio_type":
            return $('radioTemplate').html();
        case "checkbox_type":
            return $('checkboxTemplate').html();
        case "input_type":
            return $('inputTemplate').html();
        case "text_type":
            return $('textTemplate').html();
    }
}

});
</script>
</body>
</html>

```

13.5 编写QuestionController.java

```

package com.yanzhen.controller;

import com.yanzhen.entity.Question;
import com.yanzhen.service.QuestionService;
import com.yanzhen.utils.MapControl;
import org.springframework.beans.factory.annotation.Autowired;

```

```

import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.*;

import javax.servlet.http.HttpServletRequest;
import java.util.List;
import java.util.Map;

@Controller
@RequestMapping("/question")
public class QuestionController {

    @Autowired
    private QuestionService questionService;

    @PostMapping("/create")
    @ResponseBody
    public Map<String, Object> create(@RequestBody Question question){
        int result = questionService.create(question);
        if(result<=0){
            //失败的情况下
            return MapControl.getInstance().error().getMap();
        }
        return MapControl.getInstance().success().add("id", result).getMap();
    }

    @PostMapping("/delete")
    @ResponseBody
    public Map<String, Object> delete(String ids){
        int result = questionService.deleteBatch(ids);
        if(result<=0){
            //失败的情况下
            return MapControl.getInstance().error().getMap();
        }
        return MapControl.getInstance().success().getMap();
    }

    @PostMapping("/update")
    @ResponseBody
    public Map<String, Object> update(@RequestBody Question question){
        int result = questionService.update(question);
        if(result<=0){
            //失败的情况下
            return MapControl.getInstance().error().getMap();
        }
        return MapControl.getInstance().success().getMap();
    }

    @GetMapping("/list")
    public String list(){
        return "question/list";
    }

    @PostMapping("/query")
    @ResponseBody
    public Map<String, Object> query(@RequestBody Question question, ModelMap
modelMap){

```

```

        List<Question> list = questionService.query(question);
        Integer count = questionService.count(question);
        return MapControl.getInstance().page(list, count).getMap();
    }

    @GetMapping("/detail")
    public String detail(Integer id, ModelMap modelMap) {
        Question question = questionService.detail(id);
        modelMap.addAttribute("question", question);
        return "question/update";
    }

    @GetMapping("/question")
    public String question(Integer id, ModelMap modelMap) {
        Question question = questionService.detail(id);
        modelMap.addAttribute("question", question);
        return "question/question";
    }
}

```

13.6 编写SurveyService.java

```

package com.yanzhen.service;

import com.github.pagehelper.PageHelper;
import com.google.common.base.Splitter;
import com.yanzhen.entity.Survey;
import com.yanzhen.mapper.SurveyDao;
import com.yanzhen.utils.BeanMapUtils;
import com.yanzhen.utils.MapParameter;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.List;
import java.util.Map;

@Service
public class SurveyService {
    @Autowired
    private SurveyDao surveyDao;

    public int create(Survey pi) {
        return surveyDao.create(pi);
    }

    public int delete(Integer id) {
        return surveyDao.delete(MapParameter.getInstance().addId(id).getMap());
    }

    public int deleteBatch(String ids) {
        int flag = 0;
        List<String> list = Splitter.on(",").splitToList(ids);
    }
}

```



```

        for(String s:list){

surveyDao.delete(MapParameter.getInstance().addId(Integer.parseInt(s)).getMap()
);
            flag++;
        }
        return flag;

    }

    public int update(Survey survey){
        Map<String, Object> map =
MapParameter.getInstance().put(BeansMapUtils.beanToMapForUpdate(survey)).addId(su
rvey.getId()).getMap();
        return surveyDao.update(map);
    }

    public List<Survey> query(Survey survey){
        PageHelper.startPage(survey.getPage(),survey.getLimit());
        Map<String, Object> map =
MapParameter.getInstance().put(BeansMapUtils.beanToMap(survey)).getMap();
        return surveyDao.query(map);
    }

    public Survey detail(Integer id){
        return surveyDao.detail(MapParameter.getInstance().addId(id).getMap());
    }

    public int count(Survey survey){
        Map<String, Object> map =
MapParameter.getInstance().put(BeansMapUtils.beanToMap(survey)).getMap();
        return surveyDao.count(map);
    }

    public void updateState(){
        List<Integer> list = surveyDao.queryByState(Survey.state_create);
        for (Integer id : list) {

surveyDao.update(MapParameter.getInstance().add("updateState",Survey.state_exec
).add("id",id).getMap());
        }
        List<Integer> list2 = surveyDao.queryByStateForExec(Survey.state_exec);
        for (Integer id : list2) {

surveyDao.update(MapParameter.getInstance().add("updateState",Survey.state_over
).add("id",id).getMap());
        }
    }

}

```

13.7 question.jsp 代码注释

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/css/question.css" media="all">
</head>
<body>
<div class="layui-mini-container" style="position: absolute; bottom: 0px; top:
0px; background-color: #F2F2F2; left: 0px; right: 0px; margin: 10px;">

    <!-- 左侧的四种问题按钮 -->
    <div class="left">
        <ul>
            <li>
                <a href="javascript:void(0)" class="radio_type">
                    <i class="layui-icon layui-icon-about"></i>
                    单选题
                </a>
            </li>
            <li>
                <a href="javascript:void(0)" class="checkbox_type">
                    <i class="layui-icon layui-icon-about"></i>
                    多选题
                </a>
            </li>
            <li>
                <a href="javascript:void(0)" class="input_type">
                    <i class="layui-icon layui-icon-about"></i>
                    单行文本
                </a>
            </li>
            <li>
                <a href="javascript:void(0)" class="text_type">
                    <i class="layui-icon layui-icon-about"></i>
                    多行文本
                </a>
            </li>
        </ul>
    </div>
    <!-- 右侧的main区域 -->
    <div class="main" id="main" style="overflow: auto">
        <ul class="content">
        </ul>
    </div>
</div>

<div class="radioTemplate" style="display: none">
    <li class="using radio">
        <input type="hidden">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>
        </div>
    </li>
</div>
```

```

        <div class="edit" contenteditable="true">描述</div>
    </div>
    <div class="required">
        <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
        必填项
        <div class="layui-unselect layui-form-checkbox checkscore" lay-
skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
        统计分值
        <span class="edit hide scorevalue" contenteditable="true"
style="width: 60px;padding: 7px 20px;">5</span>
        <span class="edit hide answer" title="答案序号" contenteditable="true"
style="width: 60px;padding: 7px 20px;"></span>
    </div>
    <div class="options">
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
        <div class="edit" contenteditable="true">选项</div>
    </div>
    <div class="que-btn">
        <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysaveitem" lay-event="addItem"> 新增选项 </button>
        <span class="layui-text">编辑状态下，双击选项删除</span>
    </div>
</li>
</div>

<div class="checkboxTemplate" style="display: none">
    <li class="using checkbox">
        <input type="hidden">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>
            <div class="edit" contenteditable="true">描述</div>
        </div>
        <div class="required">
            <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            必填项
            <div class="layui-unselect layui-form-checkbox checkscore" lay-
skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            统计分值
            <span class="edit hide scorevalue" contenteditable="true"
style="width: 60px;padding: 7px 20px;">5</span>
            <span class="edit hide answer" title="答案序号" contenteditable="true"
style="width: 60px;padding: 7px 20px;"></span>
        </div>
        <div class="options">
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
            <div class="edit" contenteditable="true">选项</div>
        </div>
    </li>
</div>

```

```

        </div>
        <div class="que-btn">
            <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysaveitem" lay-event="addItem"> 新增选项 </button>
            <span class="layui-text">编辑状态下，双击选项删除</span>
        </div>
    </li>
</div>

```

```

<div class="inputTemplate" style="display: none">
    <li class="using inputx">
        <input type="hidden">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>
            <div class="edit" contenteditable="true">描述</div>
        </div>
        <div class="required">
            <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            必填项
        </div>
        <div class="que-btn">
            <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
        </div>
    </li>
</div>

```

```

<div class="textTemplate" style="display: none">
    <li class="using">
        <input type="hidden">
        <div class="title">
            <div class="edit" contenteditable="true">标题</div>
            <div class="edit" contenteditable="true">描述</div>
        </div>
        <div class="required">
            <div class="layui-unselect layui-form-checkbox layui-form-checked"
lay-skin="primary"><i class="layui-icon layui-icon-ok"></i></div>
            必填项
        </div>
        <div class="que-btn">
            <button class="layui-btn layui-btn-primary layui-btn-sm data-add-btn
mydel" lay-event="add"> 取消 </button>
            <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
myedit hide" lay-event="add"> 编辑 </button>

```

```

        <button class="layui-btn layui-btn-normal layui-btn-sm data-add-btn
mysave" lay-event="add"> 保存 </button>
    </div>
</li>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<script src="../../static/js/lay-config.js" charset="utf-8"></script>
<script>
    layui.use(['jquery', 'layer'], function () {
        let $ = layui.jquery, layer = layui.layer;

        //点击左侧四种问题按钮
        //选择四种问题类型，对这四种类型进行判断，如果正处于编辑状态，此时修改会提示！
        //如果不是处于编辑状态，通过移除 using 使其恢复保存状态

        $('.radio_type, .checkbox_type, .input_type, .text_type').on('click', function () {
            if (isEditing()) {
                layer.msg("正在编辑，请先保存");
                return;
            }
            $('.content').find(".using").removeClass("using");

            //这里没看懂 应该是在页面追加什么东西
            let template = loadTemplation($(this).prop("className"));
            $('.content').append($(template));
            $('#main').animate({scrollTop: $(".content").height()}, 300);
        });

        //删除操作 点击取消按钮
        //判断是否正在编辑，如果是先保存，如果不是移除该组件
        //调用clearusing函数 清除所有using状态
        //获得该问题的id值，上面设置了一个input标签用于存放input值
        //将该id值通过Ajax传递给后端
        $(".content").on('click', '.mydel', function () {
            //判断是否是当前编辑项
            var flag = $(this).parent().parent().hasClass("using");
            if (flag) {
                $(this).parent().parent().remove();
            } else {
                if (isEditing()) {
                    layer.msg("正在编辑，请先保存");
                    return;
                } else {
                    $(this).parent().parent().remove();
                }
            }
        });
        //清除using状态
        clearUsing();
        //获得该问题的id值，上面设置了一个input标签用于存放input值
        var id = $(this).parent().parent().find("input").val();
        if (id) {
            $.ajax({
                url: "../question/delete",
                type: "POST",
                data: "ids=" + id, //发送到服务器的数据
                dataType: 'json', //预期服务器返回的数据格式
                success: function (data) { //成功函数

```

```

        layer.msg("删除成功"); //弹出消息删除成功
    }
    })
}

});

//判断是否有正在编辑的项目
//判断所选内容的class中是否含有using 来判断是否在使用该组件
function isEditing(){
    var length = $('.content').find(".using").length;
    if(length == 0){
        return false;
    }else{
        return true;
    }
}

//保存之后，清除保存状态
//获取class中包含using的组件，移除using
function clearUsing(){
    var obj = $('.content').find(".using");
    obj.removeClass("using");
}

//保存操作
//保存按钮点击操作
$(".content").on('click', '.mysave', function(){
    //获取保存标签所在的正在使用的问题组件
    var obj = $(this).parents(".using");
    //设置可编辑状态为false
    obj.find(".edit").prop("contenteditable", false);
    //移除class中 edit 添加 no_edit
    obj.find(".edit").removeClass("edit").addClass("no_edit");
    //隐藏保存按钮
    obj.find('.mysave').hide();
    //隐藏新增选项按钮
    obj.find('.mysaveitem').hide();
    //显示编辑按钮
    obj.find('.myedit').show();
    //清除使用状态
    clearUsing();

    //调用写入数据库方法
    save(obj);
});

//初始化函数
init();
//定义初始化函数
function init(){
    var param = {};
    //获取调查问卷的id值
    param.surveyId = ${survey.id};

    $.ajax({

```

url:"../question/query", //应为question在survey文件夹下，所以请求路径需要向前返回一级

```
type:"POST",
contentType:"application/json", //响应数据格式
data:JSON.stringify(param), //向后端传输的数据格式json
dataType:'json', //预期从服务器返回的数据格式
success:function (data) {
    console.log(data);
    var arr = data.data; //获取服务器返回的json数据
    //question 中type字段 代表四种不同的试题类型
    // type指 1radio|2checkbox|3text|4textarea
    for(var index=0;index<arr.length;index++){
        switch (arr[index].type) {
            case 1:
                loadRadio(1,arr,index);
                break;
            case 2:
                loadRadio(2,arr,index);
                break;
            case 3:
                loadText(3,arr,index);
                break;
            case 4:
                loadText(4,arr,index);
                break;
        }
    }
}
})
}
```

//加载选择题方法（类型、数组、索引）

//同时读取数据库中已经保存的问题数据

//type 1radio|2checkbox|3text|4textarea

function loadRadio(type,arr,index){

//let数据类型 和var一样，var作用区域为函数范围，let作用区域当前代码块中

let template;

//判断是单选还是多选

//根据type值 调用不同的试题模板

// \$('#radioTemplate').html() 里面是一整套单选试题的代码

if(type == 1){

template = \$('#radioTemplate').html();

}else if(type == 2){

template = \$('#checkboxTemplate').html();

}

//设置为该问题为非编辑状态

template.find("div .edit").prop("contenteditable",false);

template.removeClass("using");

template.find("div .edit").removeClass("edit").addClass("no_edit");

//val() 方法返回或设置被选元素的值 设置id=arr[index].id

template.find("input").val(arr[index].id);

//标题与描述赋值 html (" ") 对选中标签赋值

template.find(".title").find("div").each(function(cnt){

if(cnt == 0){

//设置title=arr[index].title

\$(this).html(arr[index].title);

}else{

//设置remark=arr[index].remark

```

        $(this).html(arr[index].remark);
    }
});
//判断是否必填项
template.find(".required").find("div").each(function(cnt){
    //cnt没看懂
    if(cnt == 0){
        if(arr[index].required==1){
            //选中必选单选框
            $(this).addClass("layui-form-checked");
        }else{
            //移除必选单选框
            $(this).removeClass("layui-form-checked");
        }
    }else{
        //判断是否评测  1 评测  0 不评测
        if(arr[index].test==1){
            $(this).addClass("layui-form-checked");
            //显示分值框
            template.find(".scorevalue").show();
            //显示答案框
            template.find(".answer").show();
            //赋值score=arr[index].score
            template.find(".scorevalue").html(arr[index].score);
        }else{
            $(this).removeClass("layui-form-checked");
        }
    }
});

//清空原有的预置选项内容
template.find(".options").empty();
//获取数据库中这个问题的选项
var options = arr[index].options;
//创建答案数组
var answerArray = [];
//遍历 添加选项
for(var j=0;j<options.length;j++){
    //依次把选项代码放进去
    template.find(".options").append($('<div class="no_edit"
contenteditable="false">'+options[j].opt+'</div>'));
    //如果该选项的answer字段为1 代表 该选项是正确答案 把序号放进answerArray
    if(options[j].answer == 1){
        answerArray.push(j+1);
    }
}
//如果判断存在设置答案，则拼接一下
if(answerArray.length>0){
    //给答案框赋值，并用，分割
    template.find(".answer").html(answerArray.join(","));
}
//隐藏保存、新增选项按钮
template.find('.mysave').hide();
template.find('.mysaveitem').hide();
template.find('.myedit').show();
//在右侧区域追加问题
$('.content').append(template);

```



```

}

//加载文本问题方法 同上
function loadText(type,arr,index){
    let template;
    if(type == 3){
        //获取单行文本代码
        template = $($('.inputTemplate')).html();
    }else if(type == 4){
        //获取多行文本代码
        template = $($('.textTemplate')).html();
    }
    //设置为div为非编辑状态
    template.find("div .edit").prop("contenteditable",false);
    template.removeClass("using");
    template.find("div .edit").removeClass("edit").addClass("no_edit");
    template.find("input").val(arr[index].id);
    //标题与描述
    template.find(".title").find("div").each(function(cnt){
        if(cnt == 0){
            $(this).html(arr[index].title);
        }else{
            $(this).html(arr[index].remark);
        }
    });
    //必填项
    template.find(".required").find("div").each(function(cnt){
        if(cnt == 0){
            if(arr[index].required==1){
                $(this).addClass("layui-form-checked");
            }else{
                $(this).removeClass("layui-form-checked");
            }
        }
    });
    //隐藏按钮
    template.find('.mysave').hide();
    template.find('.mysaveitem').hide();
    template.find('.myedit').show();
    $('.content').append(template);
}

```

```

//保存方法，向数据库写操作
function save(obj){
    var question = {};
    //根据class 内容 为question.type 赋值
    if(obj.hasClass("radio")){
        question.type = 1;
    }else if(obj.hasClass("checkbox")){
        question.type = 2;
    }else if(obj.hasClass("inputx")){
        question.type = 3;
    }else{
        question.type = 4;
    }
    //为question.title、question.remark赋值

```

```

obj.find('.title').find('div').each(function(index){
    if(index == 0){
        question.title=$(this).html();
    }
    if(index == 1){
        question.remark=$(this).html();
    }
});
var _answer="";
//根据class判断是否必填 为question.required 、question.test、
question.score 赋值
obj.find('.required').find('div').each(function(index){
    //0非必填1必填
    if(index == 0){
        if($(this).hasClass('layui-form-checked')){
            question.required=1;
        }else{
            question.required=0;
        }
    }
    if(index == 1){
        //0 不测评 1 测评
        if($(this).hasClass('layui-form-checked')){
            question.test=1;
            question.score=
obj.find('.required').find('.scorevalue').html();
            //获得选项的内容
            _answer= obj.find('.required').find('.answer').html();
//获取选项内容
        }else{
            question.test=0;
            question.score=0;
        }
    }
});
//获取问题选项
//单选或多选的时候传问题的答案
if(question.type == 1 || question.type == 2){
    //选项
    var arr = [];
    var anArray = _answer.split(","); //把答案放在anArray中
    obj.find('.options').find('div').each(function(index){
        var option = {};
        option.opt = $(this).html();
        option.type = question.type;

        for(var i=0;i<anArray.length;i++){
            if(anArray[i]==(index+1)){
                option.answer = 1;
            }
        }
        arr.push(option);
    });
    question.options = arr;
}

//question.surveyId 赋值
question.surveyId = ${survey.id};

```

```

        //question.id 赋值
        if(obj.find('input').val()){
            question.id = obj.find('input').val()
        }

        console.log(question);
        //ajax向后端传数据
        $.ajax({
            url:"../question/create",
            type:"POST",
            contentType:"application/json",
            data:JSON.stringify(question),
            dataType:'json',
            success:function (data) {
                obj.find('input').val(data.id); //给问题id赋值
            }
        })
    }

    //新增选项操作
    $(".content").on('click','mysaveitem',function(){
        var obj = $(this).parents(".using");
        obj.find(".options").append($('
```

```

        $(this).removeClass("layui-form-checked")
    }else{
        $(this).addClass("layui-form-checked")
    }
}
}else{
    //否则提示信息
    layer.msg("请切换编辑状态");
}
});

//统计分值
$(".content").on('click', '.checkscore', function(){
    var flag = $(this).parent().parent().hasClass("using");
    if(flag){
        if($(this).hasClass("layui-form-checked")){
            $(this).addClass("layui-form-checked")
            $(this).siblings('.scorevalue').show();
            $(this).siblings('.answer').show();
        }else{
            $(this).removeClass("layui-form-checked")
            $(this).siblings('.scorevalue').hide();
            $(this).siblings('.answer').hide();
        }
    }else{
        layer.msg("请切换编辑状态");
    }
});

//加载问题模板方法
function loadTemplation(type){
    switch (type) {
        case "radio_type":
            return $('radioTemplate').html();
        case "checkbox_type":
            return $('checkboxTemplate').html();
        case "input_type":
            return $('inputTemplate').html();
        case "text_type":
            return $('textTemplate').html();
    }
}

```

```

});
</script>
</body>
</html>

```

14 问卷预览

14.1 SurveyController 添加preview方法

```
@Autowired
private QuestionService questionService;

@GetMapping("/preview/{id}")
public String preview(@PathVariable("id") Integer id, ModelMap modelMap) {
    Survey survey = surveyService.detail(id);
    Question question = new Question();
    question.setSurveyId(survey.getId());
    //查询一个问卷中的所有问题及选项
    List<Question> questions = questionService.query(question);
    //将问题设置为survey的属性
    survey.setQuestions(questions);
    modelMap.addAttribute("survey", survey);
    return "survey/preview";
}
```

14.2 写preview页面

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>${survey.title}预览</title>
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/css/question.css" media="all">
    <style>
        .bg{
            background-image: url("../../upload/${survey.bgimg}");
            background-size: 100% 100%;
        }
    </style>
</head>
<body>
<div style="margin: 5px; float: right; margin-right: 300px;">
    <form action="../../upload" enctype="multipart/form-data" method="post">
        <input type="file" name="file">
        <input type="hidden" name="id" value="${survey.id}">
        <input type="submit" style="background-color: #009688; width:
100px; color: #ffffff; border: 1px solid #cccccc;" value="上传背景图">
    </form>
</div>
<div class="layui-mini-container bg" style="position: absolute; bottom: 0px; top:
40px; left: 0px; right: 0px; margin: 10px;">
    <div class="preview" id="preview" style="overflow: auto;">
        <ul class="content">
            <c:forEach items="${survey.questions}" var="question"
varStatus="status">
                <c:if test="${question.type == 1 or question.type == 2}">
                    <div class="radioTemplate splitter">
```

```

        <li class="using radio">
            <input type="hidden">
            <div class="title">
                <div class="no_edit"><strong>${status.index+1}、
${question.title}</strong></div>
                <c:if test="${question.remark != null &&
question.remark != ''}">
                    <div class="no_edit">${question.remark}
</div>
                </c:if>
            </div>
            <div class="options">
                <ul>
                    <c:forEach items="${question.options}"
var="option">
                        <li>
                            <table>
                                <tr>
                                    <td>
                                        <c:if
test="${question.type == 1}"><input type="radio" name="r${question.id}"></c:if>
                                        <c:if
test="${question.type == 2}"><input type="checkbox" name="r${question.id}">
                                        </c:if>
                                    </td>
                                    <td>${option.opt}</td>
                                </tr>
                            </table>
                        </li>
                    </c:forEach>
                </ul>
            </div>
        </li>
    </div>
</c:if>
<c:if test="${question.type == 3 or question.type == 4 }">
    <div class="inputTemplate splitter">
        <li class="using inputx">
            <input type="hidden">
            <div class="title">
                <div class="no_edit"><strong>${status.index+1}、
${question.title}</strong></div>
                <c:if test="${question.remark != null &&
question.remark != ''}">
                    <div class="no_edit">${question.remark}
</div>
                </c:if>
            </div>
            <div>
                <c:if test="${question.type == 3 }">
                    <input type="text" class="inputx">
                </c:if>
                <c:if test="${question.type == 4 }">
                    <textarea class="mytext"></textarea>
                </c:if>
            </div>
        </li>
    </div>
</div>

```

```

        </c:if>
    </c:forEach>
</ul>
</div>
</div>

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<script src="../../static/js/lay-config.js" charset="utf-8"></script>
<script>
    layui.use(['jquery', 'layer'], function () {
        let $ = layui.jquery, layer = layui.layer;

        $('.radio_type, .checkbox_type, .input_type, .text_type').on('click', function () {
            if (isEditing()) {
                layer.msg("正在编辑，请先保存");
                return;
            }
            $('.content').find(".using").removeClass("using");
            let template = loadTemplation($(this).prop("className"));
            $('.content').append($(template));
            $('#main').animate({scrollTop: $(".content").height()}, 300);
        });

        // 删除操作
        $(".content").on('click', '.mydel', function () {
            // 判断是否是当前编辑项
            var flag = $(this).parent().parent().hasClass("using");
            if (flag) {
                $(this).parent().parent().remove();
            } else {
                if (isEditing()) {
                    layer.msg("正在编辑，请先保存");
                    return;
                } else {
                    $(this).parent().parent().remove();
                }
            }
        });
        // 清除using状态
        clearUsing();

        var id = $(this).parent().parent().find("input").val();
        if (id) {
            $.ajax({
                url: "../question/delete",
                type: "POST",
                data: "ids=" + id,
                dataType: 'json',
                success: function (data) {
                    layer.msg("删除成功");
                }
            });
        }
    });

```

```

//判断是否有正在编辑的项目
function isEditing(){
    var length = $(''.content').find(".using").length;
    if(length == 0){
        return false;
    }else{
        return true;
    }
}

//保存之后，清除保存状态
function clearUsing(){
    var obj = $(''.content').find(".using");
    obj.removeClass("using");
}

//保存操作
$($(".content").on('click', '.mysave', function(){
    var obj = $(this).parents(".using");
    obj.find(".edit").prop("contenteditable", false);
    obj.find(".edit").removeClass("edit").addClass("no_edit");
    //隐藏
    obj.find('.mysave').hide();
    obj.find('.mysaveitem').hide();
    obj.find('.myedit').show();
    clearUsing();

    //写入数据库
    save(obj);
}));

init();
function init(){
    var param = {};
    param.surveyId = ${survey.id};
    $.ajax({
        url:"../question/query",
        type:"POST",
        contentType:"application/json",
        data:JSON.stringify(param),
        dataType:'json',
        success:function (data) {
            var arr = data.data;
            for(var index=0;index<arr.length;index++){
                switch (arr[index].type) {
                    case 1:
                        loadRadio(1,arr,index);
                        break;
                    case 2:
                        loadRadio(2,arr,index);
                        break;
                    case 3:
                        loadText(3,arr,index);
                        break;
                    case 4:
                        loadText(4,arr,index);
                        break;
                }
            }
        }
    });
}

```



```

    }
  }
})
}

function loadRadio(type,arr,index){
  let template;
  if(type == 1){
    template = $($('.radioTemplate').html());
  }else if(type == 2){
    template = $($('.checkboxTemplate').html());
  }
  //设置为div为非编辑状态
  template.find("div .edit").prop("contenteditable",false);
  template.removeClass("using");
  template.find("div .edit").removeClass("edit").addClass("no_edit");
  template.find("input").val(arr[index].id);
  //标题与描述
  template.find(".title").find("div").each(function(cnt){
    if(cnt == 0){
      $(this).html(arr[index].title);
    }else{
      $(this).html(arr[index].remark);
    }
  });
  //必填项
  template.find(".required").find("div").each(function(cnt){
    if(cnt == 0){
      if(arr[index].required==1){
        $(this).addClass("layui-form-checked");
      }else{
        $(this).removeClass("layui-form-checked");
      }
    }else{
      if(arr[index].test==1){
        $(this).addClass("layui-form-checked");
        template.find(".scorevalue").show();
        template.find(".answer").show();
        template.find(".scorevalue").html(arr[index].score);
      }else{
        $(this).removeClass("layui-form-checked");
      }
    }
  });

  //清空原有的预置选项
  template.find(".options").empty();
  var options = arr[index].options;
  var answerArray = [];
  for(var j=0;j<options.length;j++){
    template.find(".options").append($('<div class="no_edit" contenteditable="false">' + options[j].opt + '</div>'));
    if(options[j].answer == 1){
      answerArray.push(j+1);
    }
  }
  //如果判断存在设置答案，则拼接一下
  if(answerArray.length>0){

```

```

        template.find(".answer").html(answerArray.join(", "));
    }
    //隐藏按钮
    template.find('.mysave').hide();
    template.find('.mysaveitem').hide();
    template.find('.myedit').show();
    $('#.content').append(template);
}

function loadText(type, arr, index){
    let template;
    if(type == 3){
        template = $('#.inputTemplate').html();
    }else if(type == 4){
        template = $('#.textTemplate').html();
    }
    //设置为div为非编辑状态
    template.find("div .edit").prop("contenteditable", false);
    template.removeClass("using");
    template.find("div .edit").removeClass("edit").addClass("no_edit");
    template.find("input").val(arr[index].id);
    //标题与描述
    template.find(".title").find("div").each(function(cnt){
        if(cnt == 0){
            $(this).html(arr[index].title);
        }else{
            $(this).html(arr[index].remark);
        }
    });
    //必填项
    template.find(".required").find("div").each(function(cnt){
        if(cnt == 0){
            if(arr[index].required==1){
                $(this).addClass("layui-form-checked");
            }else{
                $(this).removeClass("layui-form-checked");
            }
        }
    });
    //隐藏按钮
    template.find('.mysave').hide();
    template.find('.mysaveitem').hide();
    template.find('.myedit').show();
    $('#.content').append(template);
}

function save(obj){
    var question = {};
    if(obj.hasClass("radio")){
        question.type = 1;
    }else if(obj.hasClass("checkbox")){
        question.type = 2;
    }else if(obj.hasClass("inputx")){
        question.type = 3;
    }else{
        question.type = 4;
    }
}

```

```

//title+remark
obj.find('.title').find('div').each(function(index){
    if(index == 0){
        question.title=$(this).html();
    }
    if(index == 1){
        question.remark=$(this).html();
    }
});
var _answer="";
//是否必填
obj.find('.required').find('div').each(function(index){
    if(index == 0){
        if($(this).hasClass('layui-form-checked')){
            question.required=1;
        }else{
            question.required=0;
        }
    }
    if(index == 1){
        if($(this).hasClass('layui-form-checked')){
            question.test=1;
            question.score=
obj.find('.required').find('.scorevalue').html();
            _answer= obj.find('.required').find('.answer').html();
        }else{
            question.test=0;
            question.score=0;
        }
    }
});
if(question.type == 1 || question.type == 2){
    //选项
    var arr = [];
    var anArray = _answer.split(",");
    obj.find('.options').find('div').each(function(index){
        var option ={};
        option.opt = $(this).html();
        option.type = question.type;
        for(var i=0;i<anArray.length;i++){
            if(anArray[i]==(index+1)){
                option.answer = 1;
            }
        }
        arr.push(option);
    });
    question.options = arr;
}
question.surveyId = ${survey.id};
if(obj.find('input').val()){
    question.id = obj.find('input').val()
}
console.log(question);
$.ajax({
    url:"../question/create",
    type:"POST",
    contentType:"application/json",
    data:JSON.stringify(question),

```

```

        dataType: 'json',
        success: function (data) {
            obj.find('input').val(data.id);
        }
    })

}

//新增选项操作
$(".content").on('click', '.mysaveitem', function(){
    var obj = $(this).parents(".using");
    obj.find(".options").append($('<div class="edit"
contenteditable="true">选项</div>'));
});

//双击删除选项操作
$(".content").on('dblclick', '.edit', function(){
    $(this).remove();
});

//编辑操作
$(".content").on('click', '.myedit', function(){
    if(isEditing()){
        layer.msg("正在编辑，请先保存");
        return;
    }
    var obj = $(this).parent().parent().addClass('using');
    obj.find(".no_edit").prop("contenteditable", true);
    obj.find(".no_edit").removeClass("no_edit").addClass("edit");
    //隐藏
    obj.find('.mysave').show();
    obj.find('.mysaveitem').show();
    obj.find('.myedit').hide();
});

//是否必填项
$(".content").on('click', '.layui-form-checkbox', function(){
    //判断当前是否可编辑，如果不可编辑，则不可点击
    var flag = $(this).parent().parent().hasClass("using");
    if(flag){
        if($(this).hasClass("layui-form-checked")){
            $(this).removeClass("layui-form-checked")
        }else{
            $(this).addClass("layui-form-checked")
        }
    }else{
        layer.msg("请切换编辑状态");
    }
});

//统计分值
$(".content").on('click', '.checkscore', function(){
    var flag = $(this).parent().parent().hasClass("using");
    if(flag){
        if($(this).hasClass("layui-form-checked")){
            $(this).addClass("layui-form-checked")
        }
    }
});

```

```

        $(this).siblings('.scorevalue').show();
        $(this).siblings('.answer').show();
    }else{
        $(this).removeClass("layui-form-checked")
        $(this).siblings('.scorevalue').hide();
        $(this).siblings('.answer').hide();
    }
}
}
}
});

function loadTemplation(type){
    switch (type) {
        case "radio_type":
            return $('#radioTemplate').html();
        case "checkbox_type":
            return $('#checkboxTemplate').html();
        case "input_type":
            return $('#inputTemplate').html();
        case "text_type":
            return $('#textTemplate').html();
    }
}

});
</script>
</body>
</html>

```

14.3 编写preview跳转方法

list.jsp中

```

else if(obj.event === 'preview'){ //监听预览
    var checkStatus = table.checkStatus('currentTableId')
    , data = checkStatus.data;
    var arr=[];
    for(index in data){
        arr.push(data[index].id);
    }

    if(arr.length !=1){
        layer.msg("请选择一行数据修改",{time:1000});
        return;
    }
    window.open("preview/"+arr[0]); //向后端请求
}

```

14.4 添加问卷上传背景功能

1 上传相关配置

spring-mvc配置 配置上传大小 和 上传编码

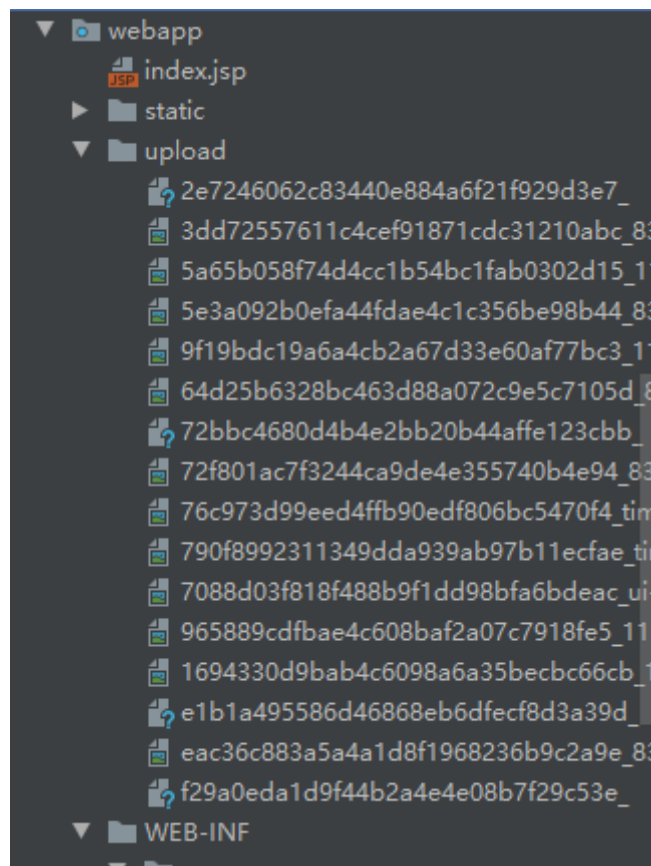
注意 文件大小 要手输入，粘贴可能有格式问题

```
<bean id="multipartResolver"

class="org.springframework.web.multipart.commons.CommonsMultipartResolver">
    <!--上传文件的最大大小，单位为字节 1024 x 1024 x 10-->
    <property name="maxUploadSize" value="10485760"></property>
    <!-- 上传文件的编码 -->
    <property name="defaultEncoding" value="UTF-8"></property>
</bean>
```

2 新建upload目录

webapp/upload



3 preview.jsp上传背景功能

```

<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>${survey.title}预览</title>
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css" media="all">
    <link rel="stylesheet" href="../../static/css/question.css" media="all">
    <style>
        .bg{
            background-image: url("../../upload/${survey.bgimg}");
            background-size: 100% 100%;
        }
    </style>
</head>
<body>
<div style="margin: 5px;float: right;margin-right: 300px;">
    <form action="../../upload" enctype="multipart/form-data" method="post">
        <input type="file" name="file">
        <input type="hidden" name="id" value="${survey.id}">
        <input type="submit" style="background-color: #009688;width: 100px;color: #ffffff;border:1px solid #cccccc;" value="上传背景图">
    </form>
</div>
<div class="layuimini-container bg" style="position: absolute;bottom: 0px;top: 40px;left: 0px;right:0px;margin: 10px;">

```

4 编写SurveyController.jsp 的上传功能

```

@RequestMapping(value = "/upload",method = RequestMethod.POST)
public String upload(Integer id, @RequestParam("file") MultipartFile
multipartFile, HttpServletRequest request){
    //上传的位置
    String path =
request.getSession().getServletContext().getRealPath("/upload/");
    //判断该路径是否存在
    File file = new File(path);
    if (!file.exists()) {
        file.mkdirs();
    }
    //上传文件项
    String filename = multipartFile.getOriginalFilename();
    String uuid = UUID.randomUUID().toString().replace("-", "");
    String saveName = uuid + "_" +
filename.substring(filename.lastIndexOf(File.separator) + 1);
    try {
        multipartFile.transferTo(new File(path, saveName));
        Survey survey = new Survey();
        survey.setId(id);
        survey.setBgimg(saveName);
        surveyService.update(survey);
    } catch (IOException e) {
        e.printStackTrace();
    }
    return "redirect:preview/"+id;
}

```

15 发布问卷

15.1 发布问卷按钮

list.jsp

```
<button class="layui-btn layui-btn-sm layui-btn-sm data-delete-btn" lay-
event="publish">
  <i class="fa fa-search"></i>
  发布问卷 </button>
```

15.2 发布问卷ajax方法

list.jsp

```
if(obj.event === 'publish'){ //发布问卷
    var checkStatus = table.checkStatus('currentTableId')
    , data = checkStatus.data;
    var arr=[];
    for(index in data){
        arr.push(data[index].id);
    }

    if(arr.length !=1){
        layer.msg("请选择一行数据修改",{time:1000});
        return;
    }

    $.ajax({
        url:"publish", //请求url http://localhost:8080/survey/publish
        type:"POST",
        dataType:'json',
        data:'id='+arr[0],
        success:function(data){
            layer.msg(data.msg,{time:2000},
                function(){
                    table.reload('currentTableId');
                });
        }
    });
}
```

15.3 编写SurveyController的publish方法

给问卷的url赋值

```
@PostMapping("/publish")
@ResponseBody
//接受从前端返回的 id数据 id=1:
public Map<String,Object> publish(Integer id,HttpServletRequest request){
    //调用surveyService的detail方法 查询survey对象 该对象url为空 或者 为了更新url
    Survey param = surveyService.detail(id);
    //对survey对象 即问卷状态进行判断,结束状态不能操作
    if(!survey.state_exec.equals(param.getState())){
        return MapControl.getInstance().error("操作失败,当前问卷未在执行
中!").getMap();
    }
    //调用UUID类 获取一个随机的字符串,保证链接安全
    //加dy是为了配置拦截器 精确拦截,精确放行
    String uuid = "/dy/"+UUID.randomUUID().toString();
    //创建一个空的survey 用于传递更新数据
```



```

//id 和 url 更具id查询到内容 根据url更新
Survey survey = new Survey();
survey.setId(id);
//http://localhost:8080/survey/ieieas-asdf-asdf-3-asd-f-asdf
// 获取本机名、获取端口号、获取项目路径+和uuid进行拼接
String url = "http://" + request.getServerName() + ":" +
request.getServerPort() + request.getContextPath() + uuid;
//设置url
survey.setUrl(url);
//进行数据库更新
surveyService.update(survey);
//返回成功消息
return MapControl.getInstance().success().getMap();
}

```

15.4 编写DanamicController.java

该类 用于问卷url响应

直接在浏览器中输入url，是什么请求方法？

get 因为你访问的URL里肯定带有请求方法 所以结合get请求的方法 就是get

```

package com.yanzhen.controller;

import com.yanzhen.entity.Question;
import com.yanzhen.entity.Survey;
import com.yanzhen.service.QuestionService;
import com.yanzhen.service.SurveyService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.ModelMap;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;

import java.util.List;

@Controller
public class DynamicController {

    @Autowired
    private SurveyService surveyService;

    @Autowired
    private QuestionService questionService;
    //浏览器输入问卷的url 将会跳转到该方法
    @GetMapping("/dy/{uuid}")
    public String preview(@PathVariable("uuid") String uuid, ModelMap modelMap){
        //新建一个空survey对象，用于查询
        Survey param = new Survey();
        //设置survey的state属性为执行中
        param.setState(Survey.state_exec);
        //调用方法，查询所有正在执行中的方法
        List<Survey> list = surveyService.queryAll(param);
        //定义一个空的survey对象
    }
}

```

```

        Survey entity = null;
        //遍历list, url不为空同时url中包括uuid的内容, 找到survey赋值给entity
        for (Survey survey : list) {
            if(survey.getUrl() !=null && survey.getUrl().contains(uuid)){
                entity = survey;
            }
        }
        //如果没找到, 提示失败信息, 跳转到error页面
        if(entity == null){
            modelMap.addAttribute("msg","你要访问的问卷已过期或不存在");
            return "error";
        }else{
            //如果找到, 新建一个空question对象
            Question question = new Question();
            //设置question的问卷id
            question.setSurveyId(entity.getId());
            //查询一个问卷中的所有问题及选项,
            //根据问卷id, 查询question中的所有问题
            List<Question> questions = questionService.query(question);
            //将问题设置为survey的属性
            entity.setQuestions(questions);
            //把entity封装为对象, 传递给前端
            modelMap.addAttribute("survey",entity);
            //跳转到exec页面
            return "survey/exec";
        }
    }

    @GetMapping("/dy/success")
    public String success(){
        return "survey/success";
    }
}

```

一 后端向前端传递数据的三种方式!!!

1 Ajax

后端

```

@ResponseBody
@RequestMapping(value = "/user/getUserAmount", method = RequestMethod.POST)
public JsonResult<Map<String, Integer>> getUserAmount() {
    JsonResult<Map<String, Integer>> result = new JsonResult<>();
    try {
        //已签到人数
        Integer signedAmount = userService.getSignedAmount();
        //总人数
        Integer totalUserAmount = userService.getTotalUserAmount();
        Map<String, Integer> amountMap = new HashMap<>();
        amountMap.put("signed", signedAmount);
        amountMap.put("sum", totalUserAmount);
        result.setData(amountMap);
        //这里举个例子, 如果觉得setData穿的信息不够, 还可以用setMessage方法多传一个字符串过
    } catch (Exception e) {
        //这里举个例子, 如果觉得setData穿的信息不够, 还可以用setMessage方法多传一个字符串过
    }
}

```

去

```

        result.setMessage("这是签到界面");
    } catch (Exception e) {
        log.warn(ExceptionUtils.getStackTrace(e));
    }
    return result;
}

```

前端

```

function getSignedAmount(){
    $.ajax({
        type: "post",
        url : getContextPath() + "/user/getUserAmount",
        dataType:'json',
        data: {
        },
        success: function(data){
            var amount = data.data;
            //取出总人数
            var totalAmount = amount.sum;
            //总人数减去已经签到的人数，就是未签到的人
            var unsignedNumber = amount.sum-amount.signed;
            //取出后端controller中setMessage方法传过来的字符串
            var logMessage = data.message;
        }
    });
}

```

2 HttpServletRequest

```

ResponseModel responseModel = new ResponseModel();
responseModel.setCode(0);
responseModel.setMessage("登录成功");
responseModel.setData(publicUsersModel);//不能删
httpServletRequest.getSession().setAttribute( s: "nameandpaw", publicUsersModel);//传到下一个页面index

```

3 Model

```

PublicUsersInfoModel publicUsersInfoModel = userInfoService.getUserInfo(userId);
model.addAttribute( s: "userInfo", publicUsersInfoModel);

```

二 前端向后端传递数据的!!!

1 form表单提交

from表单把所有属于表单中的内容提交给后台，例如输入框，单选框，多选框，文本域，文件域等。

1. 在后台可通过对应的name属性获取相应的值。
2. from表单中的action属性标识提交数据的地址。
3. method属性指明表单提交的方式。

```

1 <form action="demo.do" method="post">
2     用户名: <br>
3     <input type="text" name="username"><br>
4     密码:<br>
5     <input type="password" name="password" ><br><br>
6     <input type="submit" value="提交">
7 </form>

```

2 jquery中的ajax提交（常用）

JavaScript中也有ajax提交，但是代码太多，所以jQuery对JS中的ajax进行了简化。引入jQuery相应的包即可使用。一般格式为：

```
1 $.ajax({
2     url: "TestJsonServlet", // 提价的路径
3     type: "post",           // 提交方式
4     data: {
5         // 向后台提交的数据
6     },
7     dataType: "JSON",       // 规定请求成功后返回的数据
8     success: function (data) {
9         // 请求成功之后进入该方法，data为成功后返回的数据
10    },
11    error: function (errorMsg) {
12        // 请求失败之后进入该方法，errorMsg为失败后返回的错误信息
13    }
14 });
```

总结：以上两种方式如果不显示的指定post提交方式，则默认的提交方式为get方式提交。此外，ajax中的url也可以直接通过字符串拼接，然后向后台提交数据，这种方式为get方式提交。下面详细说明

3 url字符串拼接提交

1. 直接在ajax中url拼接数据

```
1 $.ajax({
2     url: "TestJsonServlet?id="+id+"&gender="+男, // 提价的路径
3     type: "get",           // 提交方式
4     dataType: "JSON",       // 规定请求成功后返回的数据
5     success: function (data) {
6         // 请求成功之后进入该方法，data为成功后返回的数据
7     },
8     error: function (errorMsg) {
9         // 请求失败之后进入该方法，errorMsg为失败后返回的错误信息
10    }
11 });
```

2. JS提交数据，通过window.location.href指定路径提交数据。

```
1 var deleteUser = function (deleteId) {
2     if (confirm("确认删除编号是【"+deleteId+"】的成员吗?")){
3         window.location.href="DeleteUserServlet?deleteId="+deleteId;
4     }
5 }
```

3. 通过a标签提交数据，通过a标签的href属性提交数据，和js提交数据类似。

```
1 <a href="DeleteUserServlet?id='3'&gender='男'"></a>
```

4. 在后台中也可以互相访问相应的Servlet

```
1 request.setAttribute("id","3");
2 request.setAttribute("gender","男");
3 request.getRequestDispatcher("updateUser.jsp").forward(request,response);
```

然后在对应的Servlet中接受参数，进行相应的处理

```
1 String id = request.getAttribute("id");
2 String gender = request.getAttribute("gender");
```

15.5 SurveyService 中添加queryAll方法

```
public List<Survey> queryAll(Survey survey){
    Map<String, Object> map =
    MapParameter.getInstance().put(BeanMapUtils.beanToMap(survey)).getMap();
    return surveyDao.query(map);
}
```

15.6 编写exec.jsp页面

url页面

```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>${survey.title}</title>
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/css/question.css" media="all">
    <style>
        .bg{
            background-image: url("../upload/${survey.bgimg}");
            background-size: 100% 100%;
            margin: 0px;
            padding: 0px;
        }
    </style>
</head>
<body class="bg">
<div style="text-align: center; font-size: 24px; padding-top: 20px; box-sizing:
border-box;">
    ${survey.title}
</div>
<div style="text-align: center; font-size: 16px; padding-top: 20px; box-sizing:
border-box; color: #999;">
    ${survey.remark}
</div>
<div class="layui-mini-container" style="position: absolute; bottom:
0px; top: 100px; left: 0px; right: 0px; margin: 10px;">
    <div class="preview" id="preview" style="overflow: auto;">
        <ul class="content">
            //使用jstl遍历问题 items 要遍历的对象 var临时变量
            //varState 封装了当前遍历的状态 state.index 从0开始
            //state.count从1开始
            <c:forEach items="${survey.questions}" var="question"
varStatus="status">
                //根据问题的类型 显示问题
                <c:if test="${question.type == 1 or question.type == 2 }">
                    <div class="radioTemplate splitter">
                        <li class="using radio">
                            //input 隐藏域 用于保存问题id 类型
                            <input type="hidden" name="questionId"
xtype="${question.type}" value="${question.id}">
```

```

        <div class="title">
            //加粗标题
            <div class="no_edit"><strong>${status.index+1}、
${question.title}</strong></div>
            //判断是否有备注内容 没有就不显示
            <c:if test="${question.remark != null &&
question.remark != ''}">
                <div class="no_edit">${question.remark}
</div>
            </c:if>
        </div>
        //遍历显示选项
        <div class="options">
            <ul>
                <c:forEach items="${question.options}"
var="option">
                    <li>
                        <table>
                            <tr>
                                <td>
                                    //根据问题类型 显示问题的id
                                    <c:if
test="${question.type == 1}">
                                        <input class="myselect" type="radio" name="r${question.id}"
value="${option.id}">
                                    </c:if>
                                    <c:if test="${question.type == 2}">
                                        <input class="myselect" type="checkbox" name="r${question.id}"
value="${option.id}">
                                    </c:if>
                                </td>
                                //显示选项内容
                                <td>${option.opt}</td>
                            </tr>
                        </table>
                    </li>
                </c:forEach>
            </ul>
        </div>
    </li>
</div>
</c:if>
<c:if test="${question.type == 3 or question.type == 4 }">
    <div class="inputTemplate splitter">
        <li class="using inputx">
            <input type="hidden" name="questionId"
xtype="${question.type}" value="${question.id}">
            <div class="title">
                <div class="no_edit"><strong>${status.index+1}、
${question.title}</strong></div>
                <c:if test="${question.remark != null &&
question.remark != ''}">
                    <div class="no_edit">${question.remark}
</div>
                </c:if>
            </div>
            <div>
                <c:if test="${question.type == 3 }">

```

```

        <input type="text" class="inputx"
id="r${question.id}">
        </c:if>
        <c:if test="${question.type == 4 }">
            <textarea class="mytext"
id="r${question.id}"></textarea>
        </c:if>
    </div>
</li>
</div>
</c:if>
</c:forEach>
<li style="text-align: center">
    <button class="layui-btn mysave">立即提交</button>
</li>
</ul>
</div>
<div>
</div>
</div>
<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<script src="../../static/js/lay-config.js" charset="utf-8"></script>
<script>
    layui.use(['jquery', 'layer'], function () {
        let $ = layui.jquery, layer = layui.layer;
        //提交按钮绑定的函数
        $('.mysave').click(function(){
            var optionArray = $('.myselect'); //选择所有选项以及答案文本框
            optionArray.each(function(index){
                //console.log($(this).prop("name"));
            })
            var list = []; //待提交的数组
            //一组问题数组
            var questionArr = $('input[name="questionId"]'); //选中隐藏域 封装为数组
            //对问题数组遍历
            questionArr.each(function(index){
                //选中该问题
                var that = $(this);
                //创建一个元组 this指调用方法的html元素
                //$(this) 指jquery对象
                //数组必须放相同的东西 元组可以放不同的东西
                var obj = {};
                obj.questionId = $(this).val();
                obj.surveyId = "${survey.id}";
                obj.type = $(this).attr("xtype");
                //一个问题产生一个数组结果
                //根据问题类型 产生不同的答案
                //给obj.opts赋值
                if(obj.type == 1 || obj.type == 2){
                    var tmp = [];
                    //选择选中的选项放在tmp中
                    optionArray.each(function(index){
                        if($(this).prop("name") == ("r"+that.val())){
                            if($(this).prop("checked")){
                                tmp.push($(this).val());
                            }
                        }
                    })
                }
            })
        })
    })

```

```

        obj.opts = tmp;
    }else {
        //设置答案
        obj.result = $("#r"+that.val()).val();
    }
    list.push(obj);
})

$.ajax({
    url:"../survey/submit",
    type:"POST",
    contentType:'application/json',
    dataType:'json',
    data:JSON.stringify(list),
    success:function(data){
        layer.msg(data.msg,{time:500},
            function(){
                window.location.href="success";
            });
    }
});
});
});
</script>
</body>
</html>

```

15.7 编写error、success页面

error

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
    ${msg}
</body>
</html>

```

success


```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
投票已完成，感谢您的参与！
</body>
</html>
```

16 问卷提交

先看懂exec.jsp

16.1 编写SurveyController 的submit方法

```
@PostMapping("/submit")
@ResponseBody
public Map<String, Object> submit(@RequestBody List<Map<String, Object>> list)
{

    List<AnswerOpt> optList = new ArrayList();
    List<AnswerTxt> txtList = new ArrayList();

    String uuid = UUID.randomUUID().toString();
    //遍历list集合 里面放的 questionId type surveyId result opts信息
    tb_question_opt表
    for (Map<String, Object> stringObjectMap : list) {
        //筛选选择题
        if(object2String(stringObjectMap.get("type")).equals("1") ||
object2String(stringObjectMap.get("type")).equals("2")){
            //获取选项集合
            List<Object> opts = (List<Object>) stringObjectMap.get("opts");
            //取出里面的内容进行封装
            for (Object opt : opts) {
                AnswerOpt answerOpt = new AnswerOpt();

                answerOpt.setQuestionId(object2Integer(stringObjectMap.get("questionId")));

                answerOpt.setSurveyId(object2Integer(stringObjectMap.get("surveyId")));

                answerOpt.setType(object2String(stringObjectMap.get("type")));
                answerOpt.setOptId(object2Integer(opt));
                answerOpt.setCreateTime(new Date());
                //设置投票者 用于统计信息
                answerOpt.setVoter(uuid);
                optList.add(answerOpt);
            }
        }
        //筛选简答题 向简答题答案库中提交
        if(object2String(stringObjectMap.get("type")).equals("3") ||
object2String(stringObjectMap.get("type")).equals("4")){
            AnswerTxt answerTxt = new AnswerTxt();
```

```

answerTxt.setQuestionId(object2Integer(stringObjectMap.get("questionId")));

answerTxt.setSurveyId(object2Integer(stringObjectMap.get("surveyId")));

answerTxt.setResult(object2String(stringObjectMap.get("result")));
        answerTxt.setCreateTime(new Date());
        answerTxt.setVoter(uuid);
        txtList.add(answerTxt);
    }
}
//向数据库提交
surveyService.submit(optList,txtList);
return MapControl.getInstance().success().getMap();
}

```

16.2 编写surveyService的submit () 方法

该方法用于同时提交选项数据和文本数据

```

@Autowired
private AnswerTxtDao answerTxtDao;

@Autowired
private AnswerOptDao answerOptDao;

public Integer submit(List<AnswerOpt> opts, List<AnswerTxt> txts){
    int flag = 0;
    for (AnswerOpt opt : opts) {
        flag = answerOptDao.create(opt);
    }
    for (AnswerTxt txt : txts) {
        flag = answerTxtDao.create(txt);
    }
    return flag;
}

```

17 问卷查询统计

17.1 编写问卷详情的按钮 和 方法

```

<button class="layui-btn layui-btn-sm layui-btn-sm data-delete-btn" lay-
event="query_detail">
    <i class="fa fa-search"></i>
    问卷详情 </button>

```

```

if(obj.event === 'query_detail'){
    var checkStatus = table.checkStatus('currentTableId')
    , data = checkStatus.data;
    var arr=[];
    for(index in data){

```

```

        arr.push(data[index].id);
    }
    if(arr.length !=1){
        layer.msg("请选择一行数据查看",{time:1000});
        return;
    }
    if(data[0].url == ""){
        layer.msg("请先发布再查看详情",{time:1000});
        return;
    }
    window.open("query_detail/"+arr[0]);
}

});

```

17.2 编写SurveyController的query_detail方法

```

@GetMapping("/query_detail/{id}")
public String query_detail(@PathVariable("id") Integer id,ModelMap modelMap){
    Survey survey = surveyService.detail(id);
    Question question = new Question();
    question.setSurveyId(survey.getId());
    //查询一个问卷中的所有问题及选项
    List<Question> questions = questionService.query(question);
    //将问题设置为survey的属性
    survey.setQuestions(questions);

    //总投票人数
    AnswerOpt answerOpt = new AnswerOpt();
    answerOpt.setSurveyId(id);
    //根据问卷id查询所有选择题答案
    List<AnswerOpt> answerOpts = surveyService.queryAnswerOpt(answerOpt);
    //利用hashset集合去重
    Set<String> set = new HashSet<String>();
    //把所有答案的voter加入hashset集合 去重后就是人数
    for (AnswerOpt opt : answerOpts) {
        set.add(opt.getVoter());
    }
    //遍历所有问题
    for (Question question1 : questions) {
        //遍历所有问题的各自的选项
        for (QuestionOpt questionOpt : question1.getOptions()) {
            int num = 0;
            //用查到的答案的选项与问卷中的选项的id进行比较
            //如果相同就num+1，num初始值为0，这样就获得了每个选项被选择的个数
            for (AnswerOpt opt : answerOpts) {
                if(questionOpt.getId() == opt.getOptId()){
                    num++;
                }
            }
            //设置该问题的num值
            questionOpt.setNum(num);
        }
    }
}

```

```

    }
    //向前端传送数据
    modelMap.addAttribute("survey",survey);
    modelMap.addAttribute("total",set.size());
    //跳转页面
    return "survey/query_detail";
}

```

17.3 编写SurveyService的queryAnswerOpt方法

查询所有选择题

```

public List<AnswerOpt> queryAnswerOpt(AnswerOpt answerOpt){
    Map<String, Object> map =
    MapParameter.getInstance().put(BeanMapUtils.beanToMap(answerOpt)).getMap();
    return answerOptDao.query(map);
}

```

17.4 编写query_detail页面

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>${survey.title}预览</title>
    <link rel="stylesheet" href="../../static/lib/layui-src/css/layui.css"
media="all">
    <link rel="stylesheet" href="../../static/css/question.css" media="all">
    <style>
        .bg{
            background-image: url("../../upload/${survey.bgimg}");
            background-size: 100% 100%;
        }
    </style>
</head>
<body>
<div class="layui-mini-container bg" style="position: absolute;bottom:
0px;top:0px;left: 0px;right:0px;margin: 0px;">
    <div class="preview" id="preview" style="overflow: auto;">
        <ul class="content">
            <c:forEach items="${survey.questions}" var="question"
varStatus="status">
                <c:if test="${question.type == 1 or question.type == 2 }">
                    <div class="radioTemplate splitter">
                        <li class="using radio">
                            <input type="hidden">
                            <div class="title">
                                <div class="no_edit"><strong>${status.index+1}、
${question.title}</strong></div>
                                <c:if test="${question.remark != null &&
question.remark != ''}">
                                    <div class="no_edit">${question.remark}
                                </div>
                            </c:if>

```

```

        </div>
        <div class="options">
            <ul>
                <c:forEach items="${question.options}"
var="option">
                    <li>
                        <table>
                            <tr>
                                <td>
                                    <c:if
test="${question.type == 1}"><input type="radio" name="r${question.id}"></c:if>
                                    <c:if
test="${question.type == 2}"><input type="checkbox" name="r${question.id}">
</c:if>
                                </td>
                                <td>
width="900px">${option.opt}</td>
                                <td>
                                    ${option.num}/${total}
                                </td>
                            </tr>
                        </table>
                    </li>
                </c:forEach>
            </ul>
        </div>
    </li>
</div>
</c:if>
<c:if test="${question.type == 3 or question.type == 4 }">
    <div class="inputTemplate splitter">
        <li class="using inputx">
            <input type="hidden">
            <div class="title">
                <div class="no_edit"><strong>${status.index+1}、
${question.title}</strong></div>
                <c:if test="${question.remark != null &&
question.remark != ''}">
                    <div class="no_edit">${question.remark}
                </div>
                </c:if>
            </div>
            <div>
                <c:if test="${question.type == 3 }">
                    <input type="text" class="inputx">
                </c:if>
                <c:if test="${question.type == 4 }">
                    <textarea class="mytext"></textarea>
                </c:if>
            </div>
        </li>
    </div>
</c:if>
</c:forEach>
</ul>
</div>
</div>

```

```

<script src="../../static/lib/layui-src/layui.js" charset="utf-8"></script>
<script src="../../static/js/lay-config.js" charset="utf-8"></script>
<script>
    layui.use(['jquery', 'layer'], function () {
        let $ = layui.jquery, layer = layui.layer;

        $('.radio_type, .checkbox_type, .input_type, .text_type').on('click', function () {
            if (isEditing()) {
                layer.msg("正在编辑，请先保存");
                return;
            }
            $('.content').find(".using").removeClass("using");
            let template = loadTemplation($(this).prop("className"));
            $('.content').append($(template));
            $('#main').animate({ scrollTop: $(".content").height() }, 300);
        });

        // 删除操作
        $(".content").on('click', '.mydel', function () {
            // 判断是否是当前编辑项
            var flag = $(this).parent().parent().hasClass("using");
            if (flag) {
                $(this).parent().parent().remove();
            } else {
                if (isEditing()) {
                    layer.msg("正在编辑，请先保存");
                    return;
                } else {
                    $(this).parent().parent().remove();
                }
            }
        });
        // 清除using状态
        clearUsing();

        var id = $(this).parent().parent().find("input").val();
        if (id) {
            $.ajax({
                url: "../question/delete",
                type: "POST",
                data: "ids=" + id,
                dataType: 'json',
                success: function (data) {
                    layer.msg("删除成功");
                }
            });
        }
    });

    // 判断是否有正在编辑的项目
    function isEditing() {
        var length = $('.content').find(".using").length;
        if (length == 0) {
            return false;
        } else {
            return true;
        }
    }

```

```

        return true;
    }
}

//保存之后，清除保存状态
function clearUsing(){
    var obj = $('.content').find(".using");
    obj.removeClass("using");
}

//保存操作
$($(".content").on('click', '.mysave', function(){
    var obj = $(this).parents(".using");
    obj.find(".edit").prop("contenteditable", false);
    obj.find(".edit").removeClass("edit").addClass("no_edit");
    //隐藏
    obj.find('.mysave').hide();
    obj.find('.mysaveitem').hide();
    obj.find('.myedit').show();
    clearUsing();

    //写入数据库
    save(obj);
}));

init();
function init(){
    var param = {};
    param.surveyId = ${survey.id};
    $.ajax({
        url: "../question/query",
        type: "POST",
        contentType: "application/json",
        data: JSON.stringify(param),
        dataType: 'json',
        success: function (data) {
            var arr = data.data;
            for(var index=0; index<arr.length; index++){
                switch (arr[index].type) {
                    case 1:
                        loadRadio(1, arr, index);
                        break;
                    case 2:
                        loadRadio(2, arr, index);
                        break;
                    case 3:
                        loadText(3, arr, index);
                        break;
                    case 4:
                        loadText(4, arr, index);
                        break;
                }
            }
        }
    })
}

function loadRadio(type, arr, index){

```



```

        $('#.content').append(template);
    }

    function loadText(type,arr,index){
        let template;
        if(type == 3){
            template = $($('.inputTemplate')).html();
        }else if(type == 4){
            template = $($('.textTemplate')).html();
        }
        //设置为div为非编辑状态
        template.find("div .edit").prop("contenteditable",false);
        template.removeClass("using");
        template.find("div .edit").removeClass("edit").addClass("no_edit");
        template.find("input").val(arr[index].id);
        //标题与描述
        template.find(".title").find("div").each(function(cnt){
            if(cnt == 0){
                $(this).html(arr[index].title);
            }else{
                $(this).html(arr[index].remark);
            }
        });
        //必填项
        template.find(".required").find("div").each(function(cnt){
            if(cnt == 0){
                if(arr[index].required==1){
                    $(this).addClass("layui-form-checked");
                }else{
                    $(this).removeClass("layui-form-checked");
                }
            }
        });
        //隐藏按钮
        template.find('.mysave').hide();
        template.find('.mysaveitem').hide();
        template.find('.myedit').show();
        $('#.content').append(template);
    }

    function save(obj){
        var question = {};
        if(obj.hasClass("radio")){
            question.type = 1;
        }else if(obj.hasClass("checkbox")){
            question.type = 2;
        }else if(obj.hasClass("inputx")){
            question.type = 3;
        }else{
            question.type = 4;
        }
        //title+remark
        obj.find('.title').find('div').each(function(index){
            if(index == 0){
                question.title=$(this).html();
            }
            if(index == 1){

```

```

        question.remark=$(this).html();
    }
});
var _answer="";
//是否必填
obj.find('.required').find('div').each(function(index){
    if(index == 0){
        if($(this).hasClass('layui-form-checked')){
            question.required=1;
        }else{
            question.required=0;
        }
    }
    if(index == 1){
        if($(this).hasClass('layui-form-checked')){
            question.test=1;
            question.score=
obj.find('.required').find('.scorevalue').html();
            _answer = obj.find('.required').find('.answer').html();
        }else{
            question.test=0;
            question.score=0;
        }
    }
});
if(question.type == 1 || question.type == 2){
    //选项
    var arr = [];
    var anArray = _answer.split(",");
    obj.find('.options').find('div').each(function(index){
        var option = {};
        option.opt = $(this).html();
        option.type = question.type;
        for(var i=0;i<anArray.length;i++){
            if(anArray[i]==(index+1)){
                option.answer = 1;
            }
        }
        arr.push(option);
    });
    question.options = arr;
}
question.surveyId = ${survey.id};
if(obj.find('input').val()){
    question.id = obj.find('input').val()
}
console.log(question);
$.ajax({
    url:"../question/create",
    type:"POST",
    contentType:"application/json",
    data:JSON.stringify(question),
    dataType:'json',
    success:function (data) {
        obj.find('input').val(data.id);
    }
})

```

```

}

//新增选项操作
$(".content").on('click', '.mysaveitem', function(){
    var obj = $(this).parents(".using");
    obj.find(".options").append($('<div class="edit"
contenteditable="true">选项</div>'));
});

//双击删除选项操作
$(".content").on('dblclick', '.edit', function(){
    $(this).remove();
});

//编辑操作
$(".content").on('click', '.myedit', function(){
    if(isEditing()){
        layer.msg("正在编辑，请先保存");
        return;
    }
    var obj = $(this).parent().parent().addClass('using');
    obj.find(".no_edit").prop("contenteditable", true);
    obj.find(".no_edit").removeClass("no_edit").addClass("edit");
    //隐藏
    obj.find('.mysave').show();
    obj.find('.mysaveitem').show();
    obj.find('.myedit').hide();
});

//是否必填项
$(".content").on('click', '.layui-form-checkbox', function(){
    //判断当前是否可编辑，如果不可编辑，则不可点击
    var flag = $(this).parent().parent().hasClass("using");
    if(flag){
        if($(this).hasClass("layui-form-checked")){
            $(this).removeClass("layui-form-checked")
        }else{
            $(this).addClass("layui-form-checked")
        }
    }else{
        layer.msg("请切换编辑状态");
    }
});

//统计分值
$(".content").on('click', '.checkscore', function(){
    var flag = $(this).parent().parent().hasClass("using");
    if(flag){
        if($(this).hasClass("layui-form-checked")){
            $(this).addClass("layui-form-checked")
            $(this).siblings('.scorevalue').show();
            $(this).siblings('.answer').show();
        }else{
            $(this).removeClass("layui-form-checked")
            $(this).siblings('.scorevalue').hide();
            $(this).siblings('.answer').hide();
        }
    }
});

```

```

        }
    }else{
        layer.msg("请切换编辑状态");
    }
});

function loadTemplation(type){
    switch (type) {
        case "radio_type":
            return $('#radioTemplate').html();
        case "checkbox_type":
            return $('#checkboxTemplate').html();
        case "input_type":
            return $('#inputTemplate').html();
        case "text_type":
            return $('#textTemplate').html();
    }
}

});
</script>
</body>
</html>

```

18配置拦截器

只有先登录才能访问

18.1 SessionInterceptor拦截器

```

package com.yanzhen.utils;

import com.yanzhen.entity.Admin;
import org.springframework.web.servlet.HandlerInterceptor;
import org.springframework.web.servlet.ModelAndView;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class SessionInterceptor implements HandlerInterceptor {

    @Override
    public boolean preHandle(HttpServletRequest request, HttpServletResponse
response, Object handler) throws Exception {
        Admin admin = SessionUtils.getAdmin(request);
        if(admin == null){
            response.setContentType("text/html;charset=utf-8");
            response.getWriter().print("您没有权限, 请<a href='login'>登录</a>");
            return false;
        }
        return true;
    }

    @Override

```

```

    public void postHandle(HttpServletRequest request, HttpServletResponse
response, Object handler, ModelAndView modelAndView) throws Exception {

    }

    @Override
    public void afterCompletion(HttpServletRequest request, HttpServletResponse
response, Object handler, Exception ex) throws Exception {

    }
}

```

18.2 配置spring-mvc的拦截器

```

<mvc:interceptors>
    <mvc:interceptor>
        <mvc:mapping path="/**/*"/>
        <mvc:exclude-mapping path="/login"/>
        <mvc:exclude-mapping path="/dy/*"/>
        <mvc:exclude-mapping path="/static/**"/>
        <bean class="com.yanzhen.utils.SessionInterceptor" />
    </mvc:interceptor>
</mvc:interceptors>

```

19 事务

1 applicationContext配置

```

<!--配置切面类-->
    <bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
        <property name="dataSource" ref="dataSource"/></property>
    </bean>
    <!--开启注解事务的支持-->
    <tx:annotation-driven transaction-manager="transactionManager">
</tx:annotation-driven>

```

2 每个service注解配置

```

@Service
@Transactional
public class AdminService {
    @Autowired
    private AdminDao adminDao;

    public int create(Admin pi) { retu

```

11.7 IDEA热更新插件(JRebel)