

# 任务二：拉勾教育后台管理系统（SSM）

广告模块

用户模块

## 广告模块

### 一 广告模块功能分析

在任务二中,首先先来完成拉勾教育后台管理系统的 广告管理模块, 广告模块包含了广告位列表查询、添加&修改广告位、回显广告位名称、广告分页查询、图片上传接口、新建&修改广告、回显广告信息、广告状态上下线等接口的编写

#### 1.1 课程管理

- 实现以下功能:
  - 广告位列表查询
  - 添加&修改广告位
  - 回显广告位名称
  - 广告分页查询
  - 图片上传接口
  - 新建&修改广告接口
  - 回显广告信息
  - 广告状态上下线

Edu Boss

Home / 广告位管理






添加广告位

spaceKey	广告位名称	创建时间	更新时间	操作
666	首页顶部推荐位	2020-07-14 13:03:31	2020-07-17 13:13:21	编辑
888	首页侧边广告位	2020-07-14 13:03:31	2020-07-17 11:53:02	编辑
999	首页顶部轮播	2020-07-14 13:03:31	2020-07-17 13:13:03	编辑
123	sadfa	2020-07-14 13:03:31	2020-07-17 11:21:22	编辑
456	ffff	2020-07-14 13:03:31	2020-07-17 11:21:26	编辑

Edu Boss

Home / 广告管理

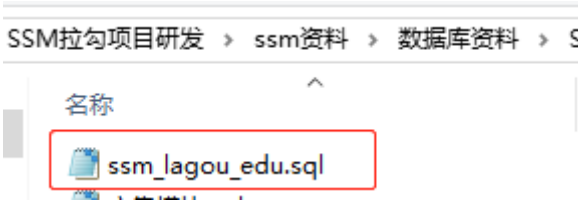
添加广告

id	广告名称	广告位置	广告图片	时间	上线/下线	操作
1074	java基础训练营	首页顶部推荐位		开始时间: 2020-06-23 17:03:27 到期时间: 2020-08-30 17:03:45	<input type="checkbox"/>	编辑
1075	精选课程	首页侧边广告位		开始时间: 2020-06-29 17:03:25 到期时间: 2020-08-30 17:03:45	<input checked="" type="checkbox"/>	编辑
1076	java训练营2	首页顶部轮播		开始时间: 2020-06-29 17:03:25 到期时间: 2020-08-30 17:03:45	<input type="checkbox"/>	编辑
1077	轮播广告2	首页顶部轮播		开始时间: 2020-06-29 17:03:25 到期时间: 2020-08-31 17:03:45	<input type="checkbox"/>	编辑
1078	广告3333	广告名称		开始时间: 2020-01-01 00:00:00 到期时间: 2020-09-30 00:00:00	<input type="checkbox"/>	编辑

## 二 广告管理模块表设计

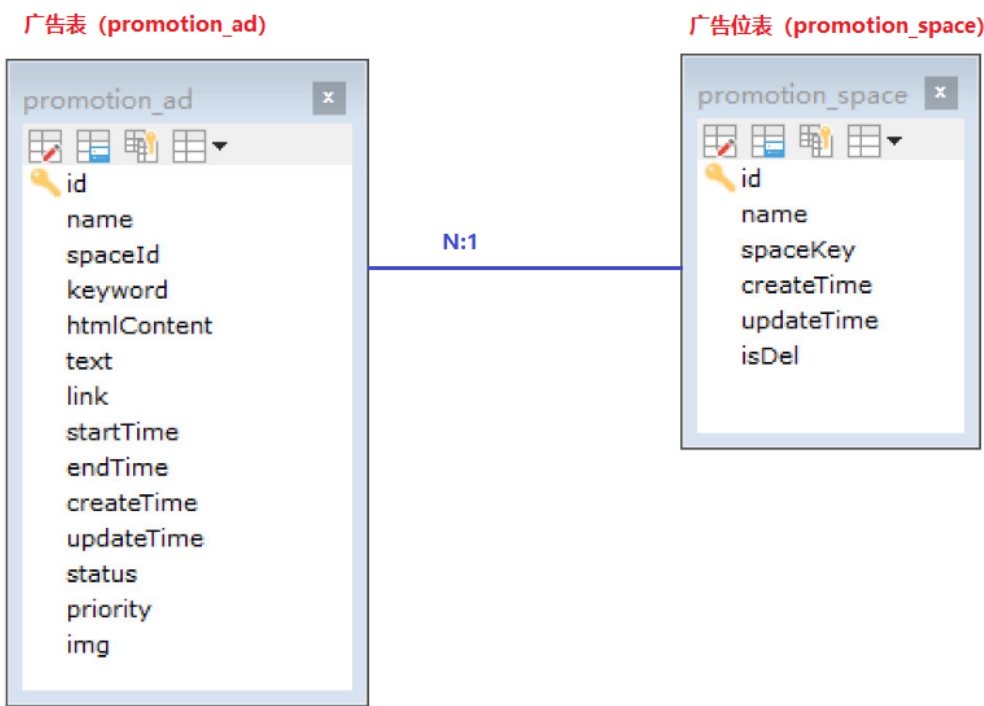
### 2.1 创建数据库及表

在资料中找到 ssm\_lagou\_edu.sql,使用SQLYog 执行SQL脚本 ,导入表结构及表信息



### 2.2 表关系介绍

#### 1.ER图



### 2.数据实体描述

#### 2.1 广告位表

字段名称	字段类型	约束	字段描述
id	int(11)	主键	广告位id
name	varchar(255)		广告位名称
spaceKey	varchar(255)		广告位标识
createTime	datetime		创建时间
updateTime	datetime		最后更新时间
isDel	int(2)		状态

2.2 广告表

id	int(11)	主键	广告ID
name	varchar(255)		广告名称
spaceId	int(11)		广告位ID
keyword	varchar(255)		关键字
htmlContent	varchar(255)		文本内容
text	varchar(255)		备注
img	varchar(255)		图片链接
link	varchar(255)		跳转地址
startTime	datetime		开始时间
endTime	datetimedatetime		结束时间
createTime	datetime		创建时间
updateTime	datetime		最后更新时间
status	int(2)		状态(上架/下架)
priority	int(4)		优先级

三 广告管理模块接口实现

1. 广告位列表查询

1.1 需求分析

需求： 点击广告列表按钮进行广告列表展示

权限管理

课程管理

用户管理

广告管理

广告列表

添加广告位

spaceKey	广告位名称	创建时间	更新时间	操作
666	首页顶部推荐位	2020-07-14 13:03:31	2020-07-17 13:13:21	编辑
888	首页侧边广告位	2020-07-14 13:03:31	2020-07-17 11:53:02	编辑
999	首页顶部轮播	2020-07-14 13:03:31	2020-07-17 13:13:03	编辑
123	sadfa	2020-07-14 13:03:31	2020-07-17 11:21:22	编辑
456	ffff	2020-07-14 13:03:31	2020-07-17 11:21:26	编辑
789	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑
78911111	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑
78911111	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑
78911111	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑
78911111	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑
78911111	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑

1.2 查看接口文档，进行编码

查看接口文档

实体类：PromotionSpace

```
public class PromotionSpace {

    private Integer id;
    private String name;
    private String spaceKey;
    private Date createTime;
    private Date updateTime;
    private Integer isDel;

    @Override
    public String toString() {
        return "PromotionSpace{" +
            "id=" + id +
            ", name='" + name + '\'' +
            ", spaceKey='" + spaceKey + '\'' +
            ", createTime=" + createTime +
            ", updateTime=" + updateTime +
            ", isDel=" + isDel +
            '}';
    }

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getSpaceKey() {
        return spaceKey;
    }

    public void setSpaceKey(String spaceKey) {
        this.spaceKey = spaceKey;
    }

    public Date getCreateTime() {
        return createTime;
    }

    public void setCreateTime(Date createTime) {
        this.createTime = createTime;
    }

    public Date getUpdateTime() {
        return updateTime;
    }
}
```

```

    public void setUpdateTime(Date updateTime) {
        this.updateTime = updateTime;
    }

    public Integer getIsDel() {
        return isDel;
    }

    public void setIsDel(Integer isDel) {
        this.isDel = isDel;
    }
}

```

## Dao层: PromotionSpaceMapper

```

public interface PromotionSpaceMapper {

    /**
     * 获取所有的广告位
     */
    public List<PromotionSpace> findAllPromotionSpace();
}

```

```

<?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.lagou.dao.PromotionSpaceMapper">

    <select id="findAllPromotionSpace"
resultType="com.lagou.domain.PromotionSpace">
        select * from promotion_space
    </select>
</mapper>

```

## Service层: PromotionSpaceService

```

public interface PromotionSpaceService {

    /**
     * 获取所有的广告位
     */
    public List<PromotionSpace> findAllPromotionSpace();
}

```

```

@Service
public class PromotionSpaceServiceImpl implements PromotionSpaceService {

    @Autowired
    private PromotionSpaceMapper promotionSpaceMapper;

    @Override
    public List<PromotionSpace> findAllPromotionSpace() {
        List<PromotionSpace> allPromotionSpace =
        promotionSpaceMapper.findAllPromotionSpace();
        return allPromotionSpace;
    }
}

```

## Web层: PromotionSpaceController

```

@RestController
@RequestMapping("/PromotionSpace")
public class PromotionSpaceController {

    @Autowired
    private PromotionSpaceService promotionSpaceService;

    /**
     * 查询所有广告位列表
     */
    @RequestMapping("/findAllPromotionSpace")
    public ResponseResult findAllPromotionSpace(){

        List<PromotionSpace> allPromotionSpace =
        promotionSpaceService.findAllPromotionSpace();
        ResponseResult responseResult = new ResponseResult(true,200,"响应成功",allPromotionSpace);

        return responseResult;
    }
}

```

## Postman测试接口

## 2. 添加&修改广告位

### 1.1 需求分析

添加： 点击广告列表按钮进行广告列表展示

添加广告位

spaceKey	广告位名称	创建时间	更新时间	操作
666	首页顶部推荐位	2020-07-14 13:03:31	2020-07-17 13:13:21	编辑
888	首页侧边广告位	2020-07-14 13:03:31	2020-07-17 11:53:02	编辑
999	首页顶部轮播	2020-07-14 13:03:31	2020-07-17 13:13:03	编辑
123	sadfa	2020-07-14 13:03:31	2020-07-17 11:21:22	编辑

Home / 添加广告位

\* 广告位名称:

提交 重置

**修改：**页面回显基础上，点击提交按钮 真正进行数据修改

添加广告位

spaceKey	广告位名称	创建时间	更新时间	操作
666	首页顶部推荐位	2020-07-14 13:03:31	2020-07-17 13:13:21	编辑
888	首页侧边广告位	2020-07-14 13:03:31	2020-07-17 11:53:02	编辑
999	首页顶部轮播	2020-07-14 13:03:31	2020-07-17 13:13:03	编辑
123	sadfa	2020-07-14 13:03:31	2020-07-17 11:21:22	编辑
456	ffff	2020-07-14 13:03:31	2020-07-17 11:21:26	编辑
789	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑
78911111	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑
78911111	广告名称	2020-07-14 13:03:31	2020-07-14 13:03:11	编辑

Home / 编辑广告位

\* 广告位名称:

首页顶部推荐位

提交

## 2.2 查看接口文档，进行编码

查看接口文档

Dao层：PromotionSpaceMapper

```

public interface PromotionSpaceMapper {

    /**
     * 添加广告位
     */
    public void savePromotionSpace(PromotionSpace promotionSpace);

    /**
     * 修改广告位
     */
}

```

```

        public void updatePromotionSpace(PromotionSpace promotionSpace);

    }

```

```

<?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.lagou.dao.PromotionSpaceMapper">

    <insert id="savePromotionSpace"
parameterType="com.lagou.domain.PromotionSpace">
        insert into promotion_space values(null,#{name},#{spaceKey},#
{createTime},#{updateTime},#{isDel})
    </insert>

    <update id="updatePromotionSpace"
parameterType="com.lagou.domain.PromotionSpace">
        UPDATE promotion_space SET NAME = #{name},updateTime = #{updateTime}
where id = #{id}
    </update>
</mapper>

```

## Service层: PromotionSpaceService

```

public interface PromotionSpaceService {

    void savePromotionSpace(PromotionSpace promotionSpace);

    void updatePromotionSpace(PromotionSpace promotionSpace);

}

```

```

@Service
public class PromotionSpaceServiceImpl implements PromotionSpaceService {

    @Autowired
    private PromotionSpaceMapper promotionSpaceMapper;

    @Override
    public void savePromotionSpace(PromotionSpace promotionSpace) {

        // 封装PromotionSpace
        UUID uuid = UUID.randomUUID();

        promotionSpace.setSpaceKey(uuid.toString());
        promotionSpace.setCreateTime(new Date());
        promotionSpace.setUpdateTime(new Date());
        promotionSpace.setIsDel(0);
        promotionSpaceMapper.savePromotionSpace(promotionSpace);

    }
}

```



```

@Override
public void updatePromotionSpace(PromotionSpace promotionSpace) {
    promotionSpace.setUpdateTime(new Date());
    promotionSpaceMapper.updatePromotionSpace(promotionSpace);
}

}

```

## Web层: PromotionSpaceController

```

@RestController
@RequestMapping("/PromotionSpace")
public class PromotionSpaceController {

    @Autowired
    private PromotionSpaceService promotionSpaceService;

    /**
     * 添加&修改广告位
     */
    @RequestMapping("/saveOrUpdatePromotionSpace")
    public ResponseResult savePromotionSpace(@RequestBody PromotionSpace
    promotionSpace){

        try {
            if(promotionSpace.getId() == null){
                //新增
                promotionSpaceService.savePromotionSpace(promotionSpace);
                ResponseResult responseResult = new ResponseResult(true,200,"响应
成功","");
                return responseResult;

            }else{
                //修改
                promotionSpaceService.updatePromotionSpace(promotionSpace);
                ResponseResult responseResult = new ResponseResult(true,200,"响应
成功","");
                return responseResult;
            }
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }

    }

}

```

## Postman测试接口

## 3.回显广告位名称

### 3.1 需求分析

需求：点击编辑按钮，进行广告位信息回显

spaceKey	广告位名称	创建时间	更新时间	操作
666	首页顶部推荐位	2020-07-14 13:03:31	2020-07-17 13:13:21	<a href="#">编辑</a>

Home / 编辑广告位

\* 广告位名称:

首页顶部推荐位

提交

### 2.2 查看接口文档，进行编码

查看接口文档

Dao层: PromotionSpaceMapper

```
public interface PromotionSpaceMapper {

    /**
     * 根据id 查询广告位信息
     */
    PromotionSpace findPromotionSpaceById(int id);

}
```

```
<?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.lagou.dao.PromotionSpaceMapper">

    <select id="findPromotionSpaceById" parameterType="int"
resultType="com.lagou.domain.PromotionSpace">
        SELECT id,NAME FROM promotion_space WHERE id = #{id};
    </select>
</mapper>
```

Service层: PromotionSpaceService

```
public interface PromotionSpaceService {

    PromotionSpace findPromotionSpaceById(int id);

}
```

```
@Service
public class PromotionSpaceServiceImpl implements PromotionSpaceService {

    @Autowired
```

```

private PromotionSpaceMapper promotionSpaceMapper;

@Override
public PromotionSpace findPromotionSpaceById(int id) {

    PromotionSpace promotionSpace =
promotionSpaceMapper.findPromotionSpaceById(id);
    return promotionSpace;
}

}

```

## Web层: PromotionSpaceController

```

@RestController
@RequestMapping("/PromotionSpace")
public class PromotionSpaceController {

    @Autowired
    private PromotionSpaceService promotionSpaceService;

    /**
     * 根据id查询 广告位信息
     * */
    @RequestMapping("/findPromotionSpaceById")
    public ResponseEntity findPromotionSpaceById(@RequestParam int id){

        PromotionSpace promotionSpace =
promotionSpaceService.findPromotionSpaceById(id);
        ResponseEntity result = new ResponseEntity(true,200,"响应成功",promotionSpace);
        return result;

    }

}

```

## 4. 广告分页查询

### 3.1 需求分析

需求：点击广告列表，对广告信息进行分页列表展示

Edu Boss

Home / 广告管理

权限管理

角色列表

菜单列表

资源列表

课程管理




用户管理

广告管理

广告列表

广告位列表

添加广告

id	广告名称	广告位置	广告图片	时间	上线/下线	操作
1074	Java基础训练营	首页顶部推荐位		开始时间: 2020-06-23 17:03:27 到期时间: 2020-08-30 17:03:45	<input checked="" type="checkbox"/>	编辑
1075	精选课程	首页侧边广告位		开始时间: 2020-06-29 17:03:25 到期时间: 2020-08-30 17:03:45	<input checked="" type="checkbox"/>	编辑
1076	Java训练营2	首页顶部轮播		开始时间: 2020-06-29 17:03:25 到期时间: 2020-08-30 17:03:45	<input checked="" type="checkbox"/>	编辑
1092	测试广告555			开始时间: 2020-01-01 10:00:00 到期时间: 2020-01-01 10:00:00	<input checked="" type="checkbox"/>	编辑

共 18 条    6 条/页    < 1 2 3 4 >    前往 1 页

## 3.2 查看接口文档，进行编码

### 查看接口文档

#### 实体类：PromotionAd

```

public class PromotionAd {

    // 标识
    private Integer id;

    // 广告名
    private String name;

    // 广告位id
    private Integer spaceId;

    // 精确搜索关键词
    private String keyword;

    // 静态广告的内容
    private String htmlContent;

    // 文字一
    private String text;

    // 链接一
    private String link;

    // 开始时间
    private Date startTime;

    // 结束时间
    private Date endTime;

    private Integer status;

    private Date createTime;

    private Date updateTime;

    // 优先级

```

```

    private Integer priority;

    private String img;

    getter/setter..
}

```

## PromotionAdVo

```

public class PromotionAdVo {

    private Integer currentPage = 1;

    private Integer pageSize = 10;

    public Integer getCurrentPage() {
        return currentPage;
    }

    public void setCurrentPage(Integer currentPage) {
        this.currentPage = currentPage;
    }

    public Integer getPageSize() {
        return pageSize;
    }

    public void setPageSize(Integer pageSize) {
        this.pageSize = pageSize;
    }

}

```

## Dao层: PromotionAdMapper

```

public interface PromotionAdMapper {

    /**
     * 分页获取所有的广告列表
     */
    public List<PromotionAd> findAllAdByPage();

}

```

```

<mapper namespace="com.lagou.dao.PromotionAdMapper">

    <resultMap id="ad_space" type="com.lagou.domain.PromotionAd">

        <id property="id" column="id"></id>
        <result property="name" column="name"/>
        <result property="spaceId" column="spaceId"/>
        <result property="keyword" column="keyword"/>
        <result property="htmlContent" column="htmlContent"/>
        <result property="text" column="text"/>
        <result property="link" column="link"/>
        <result property="startTime" column="startTime"/>

    
```

```

        <result property="endTime" column="endTime"/>
        <result property="createTime" column="createTime"/>
        <result property="updateTime" column="updateTime"/>
        <result property="status" column="status"/>
        <result property="priority" column="priority"/>
        <result property="img" column="img"/>

        <association property="promotionSpace"
select="com.lagou.dao.PromotionSpaceMapper.findPromotionSpaceById"
column="spaceId" javaType="com.lagou.domain.PromotionSpace"></association>

    </resultMap>

    <!--分页查询广告信息-->
    <select id="findAllPromotionAdByPage" resultMap="ad_space">
        select * from promotion_ad
    </select>

</mapper>

```

## Service层: PromotionAdService

applicationContext.xml

```

    <!--2.sqlSessionFactory-->
    <bean id="sqlSessionFactory"
class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource"></property>
        <property name="typeAliasesPackage" value="com.lagou.domain"/>
        <property name="plugins">
            <array>
                <bean class="com.github.pagehelper.PageHelper">
                    <property name="properties">
                        <value>helperDialect=mysql</value>
                    </property>
                </bean>
            </array>
        </property>

        <!--引入加载mybatis核心配置文件-->
        <property name="configLocation" value="classpath:sqlMapConfig.xml">
</property>
    </bean>

```

```

public interface PromotionAdService {

    /**
     * 分页获取所有的广告列表
     */
    public PageInfo findAllAdByPage(PromotionAdVo adVo);
}

```

```

@Service
public class PromotionAdServiceImpl implements PromotionAdService {

    @Autowired
    private PromotionAdMapper adMapper;

    @Override
    public PageInfo findAllAdByPage(PromotionAdVo adVo) {

        PageHelper.startPage(adVo.getCurrentPage(), adVo.getPageSize());
        List<PromotionAd> allAd = adMapper.findAllAdByPage();
        PageInfo<PromotionAd> adPageInfo = new PageInfo<>(allAd);
        return adPageInfo;
    }

}

```

## Web层: PromotionAdController

```

@RestController
@RequestMapping("/PromotionAd")
public class PromotionAdController {

    @Autowired
    private PromotionAdService adService;

    /*
     * 分页查询所有广告信息
     */
    @RequestMapping("/findAllPromotionAd")
    public ResponseEntity findAllAdByPage(PromotionAdVo adVo) {
        PageInfo allAdByPage = adService.findAllAdByPage(adVo);

        ResponseEntity responseResult = new ResponseEntity(true, 200, "响应成功",
allAdByPage);
        return responseResult;
    }

}

```

## Postman测试接口

## 5. 图片上传接口

### 1.1 需求分析

需求: 添加广告页面, 点击上传按钮, 需完成图片上传

Home / 添加广告

\* 广告名称:

广告位置:

请选择

\* 开始时间:

🕒 选择日期

\* 到期时间:

🕒 选择日期

上线/下线:

☒ 下线 ☐ 上线

广告图片:

拉勾教育

点击上传

建议尺寸: 230\*300px, JPG、PNG格式, 图片小于150K

\* 广告链接:

广告备注:

请输入内容

提交

重置

## 2.2 查看接口文档，进行编码

### 查看接口文档

#### Web层: PromotionAdController

```
@RestController
@RequestMapping("/PromotionSpace")
public class PromotionSpaceController {

    @Autowired
    private PromotionSpaceService promotionSpaceService;

    /**
     * 文件上传
     */
    @RequestMapping("/PromotionAdUpload")
    public ResponseResult fileupload(@RequestParam("file") MultipartFile file,
        HttpServletRequest request) throws IOException {
```



```

try {
    //1.判断文件是否为空
    if (file.isEmpty()) {
        throw new RuntimeException();
    }

    //2.获取项目部署路径
    String realPath = request.getServletContext().getRealPath("/");
    String webappsPath = realPath.substring(0,
realPath.indexOf("ssm_web"));

    //3.获取原文件名
    String fileName = file.getOriginalFilename();

    //4.新文件名
    String newFileName = System.currentTimeMillis() +
fileName.substring(fileName.lastIndexOf("."));

    //5.上传文件
    String uploadPath = webappsPath + "upload\\";
    File filePath = new File(uploadPath, newFileName);

    //如果目录不存在就创建目录
    if (!filePath.getParentFile().exists()) {
        filePath.getParentFile().mkdirs();
        System.out.println("创建目录: " + filePath);
    }
    file.transferTo(filePath);

    //6.将文件名和文件路径返回
    Map<String, String> map = new HashMap<>();
    map.put("fileName", newFileName);
    map.put("filePath", LOCAL_URL + "/upload/" + newFileName);
    ResponseResult result = new ResponseResult(true, 200, "响应成功",
map);

    return result;
} catch (Exception e) {
    e.printStackTrace();
}
return null;
}
}

```

## Postman测试接口

## 6. 新建&修改广告（自行完成）

### 1.1 需求分析

新建需求：点击提交按钮，将页面内容保存到数据库

\* 广告名称:

广告位置:

\* 开始时间:

\* 到期时间:

上线/下线: ☒ 下线 ☐ 上线

广告图片: 

点击上传

建议尺寸: 230\*300px, JPG、PNG格式, 图片小于150K

\* 广告链接:

广告备注:

修改需求: 点击编辑按钮, 由前端实现数据回显, 在回显页面进行数据修改, 将修改后值更新到数据库中

id	广告名称	广告位置	广告图片	时间	上线/下线	操作
1074	Java基础训练营	首页顶部推荐位		开始时间: 2020-06-23 17:03:27 到期时间: 2020-08-30 17:03:45	<input checked="" type="checkbox"/>	<input checked="" type="button" value="编辑"/>

\* 广告名称: java基础训练营

广告位置: 首页顶部推荐位

\* 开始时间: 2020-06-23 17:03:27

\* 到期时间: 2020-08-30 17:03:45

上线/下线: ☐ 下线 ☒ 上线

广告图片:



点击上传

建议尺寸: 230\*300px, JPG、PNG格式, 图片小于150K

\* 广告链接: https://edu.lagou.com/

广告备注: sdfasdf

提交

## 2.2 查看接口文档，进行编码

查看接口文档

Dao层: PromotionAdMapper

```
public interface PromotionAdMapper {  
  
    void savePromotionAd(PromotionAd promotionAd);  
  
    void updatePromotionAd(PromotionAd promotionAd);  
}
```

```
<?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.lagou.dao.PromotionAdMapper">
```

```

<!--添加广告-->
<insert id="savePromotionAd" parameterType="com.lagou.domain.PromotionAd" >
    INSERT INTO promotion_ad VALUES(NULL,#{name},#{spaceId},#{keyword},#{
htmlContent},#{text},#{link},#{startTime},#{endTime},#{createTime},#{
updateTime},#{status},#{priority},#{img});
</insert>

<!--更新广告-->
<update id="updatePromotionAd" parameterType="com.lagou.domain.PromotionAd"
>
    update promotion_ad
    <trim prefix="SET" suffixOverrides=",">
        <if test="name != null and name != ''">
            name = #{name},
        </if>

        <if test="spaceId != null and spaceId != ''">
            spaceId = #{spaceId},
        </if>

        <if test="link != null">
            link=#{link},
        </if>

        <if test="status != null and status != '' or status == 0">
            status=#{status},
        </if>

        <if test="img != null">
            img=#{img},
        </if>

        <if test="text != null">
            text=#{text},
        </if>

        <if test="startTime != null">
            startTime=#{startTime},
        </if>

        <if test="endTime != null">
            endTime=#{endTime},
        </if>

        <if test="updateTime != null">
            updateTime=#{updateTime},
        </if>

    </trim>

    <where>
        <if test="id != null and id != '' ">id = #{id}</if>
    </where>
</update>
</mapper>

```

## Service层: PromotionAdService

```
public interface PromotionAdService {  
  
    void savePromotionAd(PromotionAd promotionAd);  
  
    void updatePromotionAd(PromotionAd promotionAd);  
  
}
```

```
@Service  
public class PromotionAdServiceImpl implements PromotionAdService {  
  
    @Autowired  
    private PromotionAdMapper adMapper;  
  
    @Override  
    public void savePromotionAd(PromotionAd promotionAd) {  
  
        adMapper.savePromotionAd(promotionAd);  
    }  
  
    @Override  
    public void updatePromotionAd(PromotionAd promotionAd) {  
  
        adMapper.updatePromotionAd(promotionAd);  
    }  
  
}
```

## Web层: PromotionAdController

```
@RestController  
@RequestMapping("/PromotionAd")  
public class PromotionAdController {  
  
    @Autowired  
    private PromotionAdService adService;  
  
    /*  
        新增或更新广告位置  
    */  
    @RequestMapping("/saveOrUpdatePromotionAd")  
    public ResponseEntity saveOrUpdatePromotionAd(@RequestBody PromotionAd  
promotionAd) {  
  
        try {  
  
            if (promotionAd.getId() == null) {  
                Date date = new Date();  
                promotionAd.setCreateTime(date);  
                promotionAd.setUpdateTime(date);  
            }  
        }  
    }  
}
```

```

        adService.savePromotionAd(promotionAd);
        ResponseResult result = new ResponseResult(true, 200, "响应成功",
null);

        return result;

    } else {
        Date date = new Date();
        promotionAd.setUpdateTime(date);

        adService.updatePromotionAd(promotionAd);
        ResponseResult result = new ResponseResult(true, 200, "响应成功",
null);

        return result;
    }

    } catch (Exception e) {
        e.printStackTrace();
    }
    return null;
}

}

```

## Postman测试接口

## 7. 回显广告信息（自行完成）

### 1.1 需求分析

需求：点击编辑按钮，进行广告位名称回显

The screenshot shows a web application interface. At the top, there is a button labeled "添加广告位" (Add Ad Position). Below it is a table with the following data:

spaceKey	广告位名称	创建时间	更新时间	操作
666	首页顶部推荐位	2020-07-14 13:03:31	2020-07-17 13:13:21	编辑
888	首页侧边广告位	2020-07-14 13:03:31	2020-07-17 11:53:02	编辑
999	首页顶部轮播	2020-07-14 13:03:31	2020-07-17 13:13:03	编辑
123	sadfa	2020-07-14 13:03:31	2020-07-17 11:21:22	编辑

Below the table, there is a breadcrumb navigation: "Home / 添加广告位". Underneath is a form for editing an ad position. It has a label "广告位名称:" followed by a text input field. Below the input field are two buttons: "提交" (Submit) and "重置" (Reset).

### 2.2 查看接口文档，进行编码

查看接口文档

## Dao层: PromotionAdMapper

```
public interface PromotionAdMapper {

    /**
     * 根据id查询广告信息
     * */
    PromotionAd findPromotionAdById(int id);
}

<?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.lagou.dao.PromotionAdMapper">

    <!-- 根据id查询广告信息 -->
    <select id="findPromotionAdById" parameterType="int"
resultType="com.lagou.domain.PromotionAd">
        SELECT
            id,
            NAME,
            spaceId,
            startTime,
            endTime,
            STATUS,
            img,
            link,
            TEXT
        FROM promotion_ad WHERE id = #{id}
    </select>
</mapper>
```

## Service层: PromotionAdService

```
public interface PromotionAdService {

    /**
     * 回显广告信息
     * */
    PromotionAd findPromotionAdById(int id);
}

@Service
public class PromotionSpaceServiceImpl implements PromotionSpaceService {

    @Autowired
    private PromotionAdMapper adMapper;

    @Override
    public PromotionAd findPromotionAdById(int id) {
        PromotionAd promotionAd = adMapper.findPromotionAdById(id);
    }
}
```

```

        return promotionAd;
    }
}

```

## Web层: PromotionAdController

```

@RestController
@RequestMapping("/PromotionAd")
public class PromotionAdController {

    @Autowired
    private PromotionAdService adService;

    /**
     * 根据id回显 广告数据
     * */
    @RequestMapping("/findPromotionAdById")
    public ResponseResult findPromotionAdById(@RequestParam int id){

        try {
            PromotionAd promotionAd = adService.findPromotionAdById(id);
            ResponseResult result = new ResponseResult(true,200,"响应成功",promotionAd);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}

```

## Postman测试接口

## 8. 广告状态上下线

### 1.1 需求分析

需求：点击按钮，实现状态的动态上下线

id	广告名称	广告位置	广告图片	时间	上线/下线	操作
1074	java基础训练营	首页顶部推荐位		开始时间: 2020-06-23 17:03:27 到期时间: 2020-08-30 17:03:45	<input checked="" type="checkbox"/>	编辑

### 2.2 查看接口文档，进行编码

查看接口文档

Dao层: PromotionSpaceMapper



```
public interface PromotionAdMapper {

    void updatePromotionAdStatus(PromotionAd promotionAd);

}
```

```
<?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.lagou.dao.PromotionAdMapper">

    <!--void updatePromotionAdStatus(PromotionAd promotionAd);-->
    <select id="updatePromotionAdStatus"
parameterType="com.lagou.domain.PromotionAd">
        UPDATE promotion_ad SET STATUS = #{status} ,updatetime = #{updateTime}
WHERE id = #{id}
    </select>
</mapper>
```

## Service层: PromotionAdService

```
public interface PromotionSpaceService {

    void updatePromotionAdStatus(int id, int status);

}
```

```
@Service
public class PromotionSpaceServiceImpl implements PromotionSpaceService {

    @Autowired
    private PromotionAdMapper adMapper;

    @Override
    public void updatePromotionAdStatus(int id, int status) {

        PromotionAd promotionAd = new PromotionAd();
        promotionAd.setId(id);
        promotionAd.setStatus(status);
        promotionAd.setUpdateTime(new Date());

        adMapper.updatePromotionAdStatus(promotionAd);
    }

}
```

## Web层: PromotionAdController

```
@RestController
@RequestMapping("/PromotionAd")
```

```

public class PromotionAdController {

    @Autowired
    private PromotionAdService adService;

    /**
     * 广告位置上下线
     */

    @RequestMapping("/updatePromotionAdStatus")
    public ResponseResult updateCourseStatus(@RequestParam int id, @RequestParam
int status) {

        try {
            //执行修改操作
            if (status == 1) {
                adService.updatePromotionAdStatus(id, status);
            } else {
                adService.updatePromotionAdStatus(id, 0);
            }
            //保存修改后的状态,并返回
            Map<String, Integer> map = new HashMap<>();
            map.put("status", status);
            ResponseResult result = new ResponseResult(true, 200, "响应成功",
map);

            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

}

```

## Postman测试接口

# 用户模块

## 一 用户模块功能分析

在任务二中,再来完成拉勾教育后台管理系统的用户管理模块,广告管理模块包含了用户分页&条件查询、用户状态设置,（登陆、权限控制）等接口的编写

### 1.1 用户模块

- 实现以下功能:
  - 登陆（权限模块）
  - 权限控制（权限模块）

- 用户分页&条件查询
- 用户状态设置
- 分配角色（权限模块）

# Edu boss管理系统

## 登录

\* 手机号

请输入手机号

手机号必填

\* 密码

请输入密码

密码必填

登录

← 回到 用户端

Edu Boss

权限管理

课程管理

用户管理

广告管理

Home / 课程管理

课程名称

课程名称

状态

全部

查询

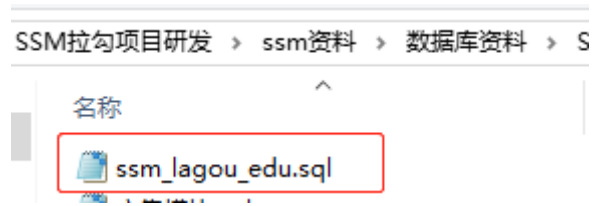
+ 新建课程

ID	课程名称	价格	排座	状态	操作		
13	111111	¥null	111111	●	下架	编辑	内容管理
11	大数据	¥199	20	●	上架	编辑	内容管理
10	React 入门	¥164	12	●	上架	编辑	内容管理
8	Vue.js 3.0 核心源码解析123	¥99	7	●	上架	编辑	内容管理
7	文案高手的18项修炼	¥263	1	●	下架	编辑	内容管理
9	秒杀11	¥200	1	●	下架	编辑	内容管理
12	111222	¥1	1	●	下架	编辑	内容管理
14	测试新增课程	¥null	1	●	上架	编辑	内容管理

## 二 用户管理模块表设计

## 2.1 创建数据库及表

在资料中找到 ssm\_lagou\_edu.sql,使用SQLYog 执行SQL脚本 ,导入表结构及表信息



## 2.2 表关系介绍

### 1.ER图

用户表

id
name
portrait
phone
password
reg_ip
account_non_expired
credentials_non_expired
account_non_locked
status
is_del
create_time
update_time

用户绑定微信表

id
user_id
union_id
open_id
nick_name
portrait
city
sex
create_time
update_time
is_del

用户发送验证码表

id
phone
verification_code
create_time
isCheck
check_times

## 三 用户管理模块接口实现

### 1. 用户分页&条件查询

#### 1.1 需求分析

需求：实现多条件分页组合查询

手机号		注册时间				
请输入手机号		自	开始日期	至	结束日期	查询
用户ID	头像	用户名	手机号	注册时间	状态	操作
100030023		昵称	18631142257	2020-08-14 21:51:51	●	禁用 分配角色
100030022		用户8666	18201288666	2020-07-13 17:43:52	●	禁用 分配角色
100030021		15811111111	15811111111	2020-07-13 11:35:20	●	禁用 分配角色
100030020		18211111111	18211111111	2020-07-10 13:57:41	●	禁用 分配角色
100030019		18201288771	18201288771	2020-07-10 12:20:16	●	禁用 分配角色
100030018		15813795039	15813795039	2020-07-10 12:17:35	●	禁用 分配角色
100030017		15811111111	15811111111	2020-07-10 11:25:45	●	分配角色
100030016		15510792995	15510792995	2020-07-10 11:23:56	●	禁用 分配角色
100030015		15801363195	15801363195	2020-07-10 11:21:50	●	禁用 分配角色
100030014		15321919571	15321919571	2020-07-10 11:10:26	●	禁用 分配角色
100030013		15801363456	15801363456	2020-07-10 11:09:11	●	禁用 分配角色
100030012		15510792994	15510792994	2020-07-10 10:43:01	●	禁用 分配角色
100030011		15321919577	15321919577	2020-07-10 10:19:15	●	分配角色

共13条 20条/页 < 1 > 前往 1 页

## 2.2 查看接口文档，进行编码

### 查看接口文档

#### UserVo

```
public class UserVo {
    private Integer currentPage;
    private Integer pageSize;
    private String username;
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date startCreateTime;
    @DateTimeFormat(pattern = "yyyy-MM-dd")
    private Date endCreateTime;

    public Integer getCurrentPage() {
        return currentPage;
    }

    public void setCurrentPage(Integer currentPage) {
        this.currentPage = currentPage;
    }

    public Integer getPageSize() {
        return pageSize;
    }

    public void setPageSize(Integer pageSize) {
        this.pageSize = pageSize;
    }

    public String getUsername() {
        return username;
    }

    public void setUsername(String username) {
        this.username = username;
    }
}
```

```

    }

    public Date getStartCreateTime() {
        return startCreateTime;
    }

    public void setStartCreateTime(Date startCreateTime) {
        this.startCreateTime = startCreateTime;
    }

    public Date getEndCreateTime() {
        return endCreateTime;
    }

    public void setEndCreateTime(Date endCreateTime) {
        this.endCreateTime = endCreateTime;
    }
}

```

## Dao层: UserMapper

```

public interface UserMapper {

    /**
     查询所有用户
    */
    public List<User> findAllUserByPage(UserVo userVo);
}

```

```

<?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.lagou.dao.UserMapper">

    <!--查询所有用户-->
    <select id="findAllUserByPage" resultType="com.lagou.domain.User">
        SELECT
            id,
            NAME,
            portrait,
            phone,
            PASSWORD,
            STATUS,
            create_time
        FROM USER
        <where>
            <if test="true">
                and is_del != 1
            </if>
            <if test="username != null">
                and name = #{username}
            </if>
            <if test="startCreateTime != null and endCreateTime != null">
                and create_time BETWEEN #{startCreateTime} AND #{endCreateTime}
            </if>
        </where>
    </select>

```

```

        </if>
    </where>

</select>
</mapper>

```

## Service层: UserService

```

public interface UserService {

    /**
     * 查询所有用户
     */
    public PageInfo findAllUserByPage(UserVo userVo);
}

```

```

@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserMapper userMapper;

    @Override
    public PageInfo findAllUserByPage(UserVo userVo) {

        // 使用pageHelper
        PageHelper.startPage(userVo.getCurrentPage(),userVo.getPageSize());
        List<User> allUser = userMapper.findAllUser(userVo);
        PageInfo<User> pageInfo = new PageInfo<User>(allUser);

        System.out.println("总条数: "+pageInfo.getTotal());
        System.out.println("总页数: "+pageInfo.getPages());
        System.out.println("当前页: "+pageInfo.getPageNum());
        System.out.println("每页显示长度: "+pageInfo.getPageSize());
        System.out.println("是否第一页: "+pageInfo.isIsFirstPage());
        System.out.println("是否最后一页: "+pageInfo.isIsLastPage());

        return pageInfo;
    }

}

```

## Web层: UserController

```

@RestController
@RequestMapping("/user")
public class UserController {

    @Autowired
    private UserService userService;
}

```

```

@RequestMapping("/findAllUserByPage")
public ResponseResult findAllUserByPage(@RequestBody UserVo userVo){

    PageInfo pageInfo = userService.findAllUserByPage(userVo);

    ResponseResult responseResult = new ResponseResult(true,200,"响应成功",pageInfo);

    List<User> list = pageInfo.getList();
    System.out.println(list);
    return responseResult;
}
}

```

## Postman测试接口

## 2. 用户状态设置（自行完成）

### 1.1 需求分析

点击禁用，实现用户的状态变更

用户ID	头像	用户名	手机号	注册时间	状态	操作
100030023		昵称	18631142257	2020-08-14 21:51:51	<span style="color: green;">●</span>	<span style="border: 1px solid red; padding: 2px;">禁用</span> 分配角色
100030022		用户8666	18201288666	2020-07-13 17:43:52	<span style="color: green;">●</span>	禁用 分配角色
100030021		15811111111	15811111111	2020-07-13 11:35:20	<span style="color: green;">●</span>	禁用 分配角色

用户状态：ENABLE能登录，DISABLE不能登录

### 2.2 查看接口文档，进行编码

查看接口文档

Dao层：UserMapper

```

public interface UserMapper {

    /**
     * 修改用户状态
     * */
    public void updateUserStatus(@Param("id") int id,@Param("status") String status);
}

```



```
<?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.lagou.dao.UserMapper">

    <!-- 修改用户状态 -->
    <update id="updateUserStatus">
        UPDATE USER SET STATUS = #{status} where id = #{id};
    </update>
</mapper>
```

## Service层: UserService

```
public interface UserService {

    /**
     * 修改用户状态
     */
    public void updateUserStatus(int id, String status);
}
```

```
@Service
public class UserServiceImpl implements UserService {

    @Autowired
    private UserMapper userMapper;

    @Override
    public void updateUserStatus(int id, String status) {

        userMapper.updateUserStatus(id,status);
    }

}
```

## Web层: UserController

```
@RestController
@RequestMapping("/user")
public class UserController {

    @Autowired
    private UserService userService;

    /**
     * 修改用户状态
     *      ENABLE能登录，DISABLE不能登录
     */
    @RequestMapping("/updateUserStatus")
```

```
public ResponseResult updateUserStatus(@RequestParam int id ,@RequestParam
String status){

    if("ENABLE".equalsIgnoreCase(status)){
        status = "DISABLE";
    }else{
        status = "ENABLE";
    }

    userService.updateUserStatus(id,status);

    ResponseResult responseResult = new ResponseResult(true,200,"响应成
功",status);
    return responseResult;
}

}
```

## Postman测试接口