课程目标:

1. 删除app信息

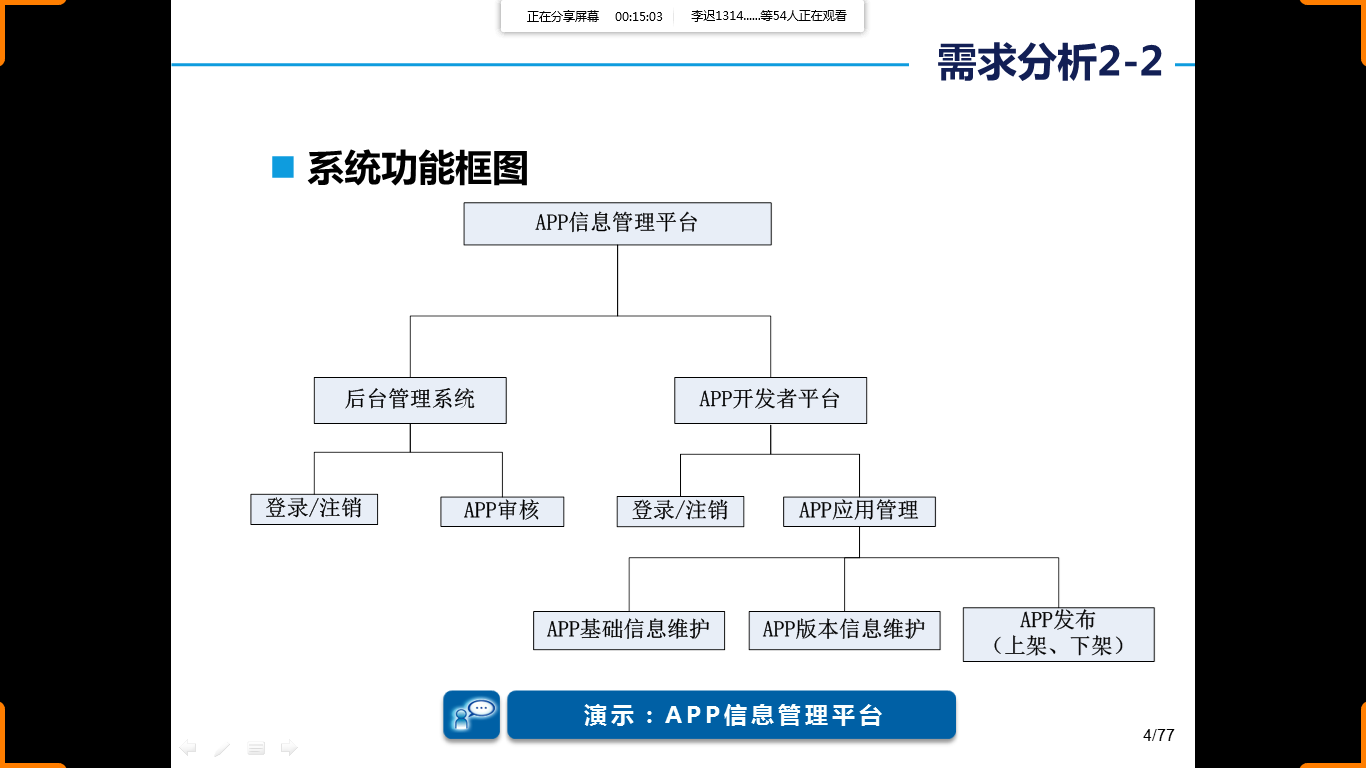
理解需要事务支持

1. 新增版本信息

>>2.1 显示当前app下所有版本信息

>>2.2 新增app版本信息

回顾:app信息平台的业务功能:



1. 删除app信息

理解:app信息表与app版本信息存在一对多的关系(一个app信息对应多个版本信息)

分析:当删除某个app的同时还要删除app的版本信息

-- 删除版本

DELETE FROM app\_version WHERE appid=?

-- 删除app

delete FROM `app\_info` WHERE id=?

1. 编写dao相关的持久化操作
   1. 编写删除版本的持久化操作

接口:

|  |
| --- |
| *//删除app下的版本* **int** deleteByAppId(Long id); |

持久化操作:

|  |
| --- |
| *<!--持久化操作-->* <**delete id="deleteByAppId" parameterType="long"**>  DELETE FROM app\_version WHERE appid=#{appid} </**delete**> |

* 1. 删除app信息的持久化操作

略

1. 编写业务器

接口:

|  |
| --- |
| **public void** deleteAppInfo(Long id) ; |

实现:

|  |
| --- |
| //考虑事务支持 //注意:支持事务的方法中不能使用try..catch //因为事务是方法上添加切面支持,方法报错进行回滚，成功提交 @Transactional **public void** deleteAppInfo(Long id) {  //建议先删除外键信息(版本信息)、再删除主键(app信息)  //1.删除app信息对应的版本  **appVersionMapper**.deleteByAppId(id);  *//int in=0,j=11; j =j/in; //错误场景*  //2.删除app信息  **appInfoMapper**.deleteByPrimaryKey(id); } |

1. 编写控制器

|  |
| --- |
| *//删除app信息* @RequestMapping(**"/deleteAppInfo"**) **public** String updateAppInfo(Long id,String oldpath,HttpServletRequest request){  **try** {  **appInfoService**.deleteAppInfo(id);   **//删除原图  String delPath=request.getSession().getServletContext().getRealPath(oldpath);  File delfile=new File(delPath);  if(delfile.exists())//判断文件是否存在  delfile.delete(); //删除旧图   return "redirect:searchApp";  }catch(Exception e){  e.printStackTrace();  }**  **return "redirect:/jsp/error.jsp"**; } |

1. 前端删除的链接

<**a class="deleteApp" data-toggle="tooltip" data-placement="top" title="" data-original-title="删除APP基础信息以及全部版本信息"**>删除</**a**>

注意: app查询需要在查询app信息时，将app信息的图片路径查询出来才能进行传递图片

|  |
| --- |
| //给删除按钮添加绑加点击事件 $("a[class='deleteApp']").click(function(){  //在jquery中利用标签的属性进行传值  var appid=$(this).attr("appid");  var apppic=$(this).attr("apppic");  if(confirm("确定想删除吗？")){  location.href="${pageContext.request.contextPath}/deleteAppInfo?id="+appid+"&oldpath="+apppic;  } }); |

1. 升级删除功能使用异步实现(选做)

1. 修改控制器代码:返回json数据，而不是跳转页面

|  |
| --- |
| *//删除app信息* @RequestMapping(**"/deleteAppInfo"**) @ResponseBody **public** String deleteAppInfo(Long id,String oldpath,HttpServletRequest request){  **try**{  *//删除app对应的图片 upload/logo/xxx.jpg* String path=request.getServletContext().getRealPath(oldpath);  File delFile=**new** File(path);  **if**(delFile.exists())  delFile.delete();  *//删除app的版本对应的安装文件apk ?查询数据库得到对应的文件路径进行删除  //调用业务删除app信息* **appInfoService**.deleteAppInfo(id);  **return "{\"result\":1}"**; *//1表示成功* }**catch** (Exception e){  e.printStackTrace();  }  **return "{\"result\":0}"**; *//0表示失败* } |

2. 编写前端发送异步请求实现删除

|  |
| --- |
| *//异步删除* $(**"a[class='deleteApp']"**).click(**function**(){  *//在jquery中利用标签的属性进行传值* **var** appid=$(**this**).attr(**"appid"**);  **var** apppic=$(**this**).attr(**"apppic"**);  **if**(*confirm*(**"确定想删除吗？"**)){  *//发送异步请求实现删除* $.post(**"${**pageContext.request.contextPath**}/deleteAppInfo"**,{**"id"**:appid,**"oldpath"**:apppic},**function**(data){  **if**(data.**result**==1){ *//删除成功  alert*(**"删除成功"**);  *//刷新* ***location***.reload();  }**else**{  *alert*(**"删除失败"**);  }  },**"json"**);  }  }); |

1. 添加app新增版本的页面



1. 实现App版本信息的添加
2. 在新增app版本信息的页面查询所有版本信息

分析查询:

-- 查询

-- 软件名称 版本号 版本大小(单位:M) 发布状态 APK文件下载 最新更新时间

-- 采用联连查询

|  |
| --- |
| SELECT app\_info.`softwareName`,app\_version.`id`,app\_version.`versionNo`,app\_version.`versionSize`,  downloadlink,app\_version.modifyDate, apkfilename,  (SELECT valuename FROM data\_dictionary  WHERE typename='发布状态' AND valueid=app\_version.`publishStatus`) AS pstatusname  FROM app\_version LEFT JOIN app\_info  ON app\_version.`appId`=app\_info.`id`  WHERE app\_version.`appId`=58  或者  SELECT id,appid,versionno,downloadlink,versionsize,apkfilename,  (SELECT softwareName FROM app\_info WHERE id=appid) AS softwareName,  (SELECT valuename FROM `data\_dictionary` WHERE typename='发布状态' AND valueid=publishstatus)  AS p**statusname**  FROM app\_version WHERE appid=61 |

* 1. 编写Dao层

修改版本实体:添加软件名称属性，和发布状态名称属性

|  |
| --- |
| **private** String **softwarename**; *//软件名称* **private** String p**statusname**; *//发布状态名称*  setter和getter方法 略 |

接口:

|  |
| --- |
| *//查询某app下的版本信息* List<AppVersion> getVersionInfoByAppId(Long id); |

持久化操作:

|  |
| --- |
| *<!--查询当前app下的版本信息-->* <**select id="getVersionInfoByAppId"  parameterType="long"  resultType="AppVersion"**> SELECT id,appid,versionno,downloadlink,versionsize, apkfilename,modifydate, (SELECT softwareName FROM app\_info WHERE id=appid) AS softwareName, (SELECT valuename FROM `data\_dictionary` WHERE typename='发布状态' AND valueid=publishstatus) AS pstatusname  FROM app\_version WHERE appid=#{id}  </**select**> |

* 1. 编写业务层

接口:

|  |
| --- |
| **public interface** AppVersionService {  *//查询某app下的版本信息* List<AppVersion> getAppVersionByAppId(Long id); } |

实现:

|  |
| --- |
| @Service *//语义化 业务组件 //@Component //定义组件，业务组件，控制器组件* **public class** AppVersionServiceImpl **implements** AppVersionService {  @Autowired  **private** AppVersionMapper **appVersionMapper**;  @Override  **public** List<AppVersion> getAppVersionByAppId(Long id) {  **return appVersionMapper**.getVersionInfoByAppId(id);  } } |

* 1. 编写控制层

|  |
| --- |
| *//显示添加app版本的页面* @RequestMapping(**"/goAddVersion"**)  **public** String goAddVersion(Long id, Model model){  *//调用业务* List<AppVersion> list=**appVersionService**.getAppVersionByAppId(id);  *//填充数据* model.addAttribute(**"appVersions"**,list);  **return "developer/appversionadd"**; } |

* 1. 编写前端

略，根据查询的结果显示对应的信息

Appinfo页的跳转

|  |
| --- |
| <**li**><**a href="${**pageContext.request.contextPath**}/goAddVersion?id=${**appInfo.id**}" class="addVersion" data-toggle="tooltip" data-placement="top" title="" data-original-title="新增APP版本信息"**>新增版本</**a**> |

1. 实现添加版本的页面业务
   1. 编写dao层

略

* 1. 编写业务层

接口:

|  |
| --- |
| *//添加app版本* **int** addAppVersion(AppVersion appVersion); |

实现:

|  |
| --- |
| @Override **public int** addAppVersion(AppVersion appVersion) {  *//添加版本* **return appVersionMapper**.insertSelective(appVersion); } |

* 1. 编写控制器层

**<input type="text" name="appid" name="appId" value="${param.id}">**

|  |
| --- |
| *//添加app版本信息 //注意:给添加app版本的表单，需要添加获取appid的表单对象(隐藏域) //显示添加app版本信息* @RequestMapping(**"/addAppVersion"**) **public** String addVersion(AppVersion appVersion,  @RequestParam(value = **"a\_downloadLink"**,required = **false**)CommonsMultipartFile file,  HttpServletRequest request){  **try** {  *//1.文件上传  //上传图片* String path=request.getSession().getServletContext().getRealPath(**relativePath**);  String fileName=FileUploadUtil.*uploadFile*(path,file); *//上传  //2.信息入库  //设置保存的apk路径* appVersion.setDownloadlink(**relativePath**+**"/"**+fileName); *//相对路径  //设置上传文件的名称* appVersion.setApkfilename(file.getOriginalFilename());  *//重点 通过表单隐藏域传递app的编号  //调用业务入库* **appVersionService**.addAppVersion(appVersion);  **return "redirect:goAddVersion?id="**+appVersion.getAppid();  }**catch** (Exception e){  e.printStackTrace();  }  **return "redirect:/jsp/error.jsp"**; } |

注意:添加时传递appid的思路:

* 1. 编写前端页面

注意:表单对象的名称与实体属性名对应即可