目标

1. 实现app审核功能

>>实现app审核

1. 文件下载 重点
2. 登入添加验证码 会
3. 实现App审核的显示页面🡪进行审核操作

实现app信息的审核功能

分析:



1. 编写dao层

略 (自动生成修改的业务)

1. 编写业务层

接口:

|  |
| --- |
| */\*\*  \* //修改app的状态  \** ***@param appid*** *表示app的id  \** ***@param status*** *\* 1表示待审核 2.审核通过 3.审核不通过 4.上架 5.下架  \** ***@return*** *影响行数  \*/* **public int** updateAppInfoState(Long appid,Long status); |

实现:

|  |
| --- |
| @Override **public int** updateAppInfoState(Long appid, Long status) {  AppInfo appInfo=**new** AppInfo();  appInfo.setId(appid); *//设置主键* appInfo.setStatus(status); *//设置状态* **return appInfoMapper**.updateByPrimaryKeySelective(appInfo); } |

1. 编写控制器层

|  |
| --- |
| *//审核app信息 app编号id, status状态* @RequestMapping(**"/checkAppInfo"**) **public** String checkAppInfo(Long id,Long status){  **int** temp=**appInfoService**.updateAppInfoState(id,status);  **if**(temp>0)  **return "redirect:noPassAppInfo"**;  **else  return "redirect:jsp/error.jsp"**; } |

1. 编写前端表示层(页面) appcheck.jsp

注意:1.正常情况下，表单出现一个提交按钮

2.当表单中出现多个提交按钮(正常)、当点击哪个提交按钮时，当前提交按钮的值会传递到服务器，则其它没有点击的提交按钮，则不会发送到服务器

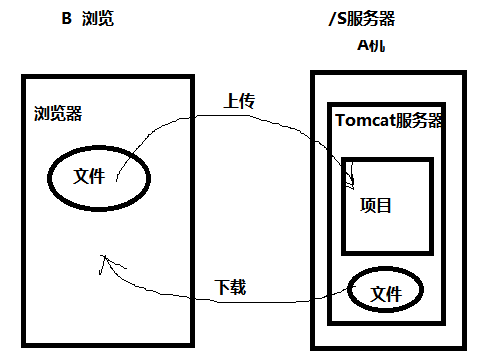
表单代码:

|  |
| --- |
| <**form class="form-horizontal form-label-left" action="${**pageContext.request.contextPath**}/checkAppInfo?appid=${**appinfo.id**}" method="post"**>  *<%--<input type="hidden" name="appid" value="${appinfo.id}">--%>*  显示的表单  对象……..  **<button type="submit" name="status" value="2" class="btn btn-success"**>审核通过</**button**> <**button type="submit" name="status" value="3" class="btn btn-success"**>审核不通过</**button**> </form> |

1. Springmvc中实现文件下载
2. 理解文件上传和下载

上传:用户在浏览器中将本地文件上传到服务器端

下载:将服务器上的文件下载到客户端(浏览器)本地



1. Springmvc实现文件下载的步骤

|  |
| --- |
| @Controller @RequestMapping(value = **"/"**) **public class** DownLoadController {   *//测试文件下载(示例)* @RequestMapping(**"download"**)  **public void** download(HttpServletResponse response) **throws** Exception{  *//读取下载文件的内容* FileInputStream fis=**new** FileInputStream(**"d:\\北大青鸟.mp3"**);  **byte** [] bs=**new byte**[fis.available()]; *//fis.available()表求文件大小* fis.read(bs);  fis.close();   *//实现文件下载  //1.设置下载的文件头 注意:如果不指定下载报头，则将数据写入响应流中  //response.setHeader("Content-Disposition","attachment;fileName=下载文件名称");* response.setHeader(**"Content-Disposition"**,**"attachment;fileName=old.mp3"**);  *//2.设置文件的类型* response.setContentType(**"audio/mp3"**);  *//3.将数据写入到响应流中即可* ServletOutputStream sos=response.getOutputStream();  sos.write(bs);  } } |

1. 实现app版本的下载

1.后台控制器下载 apk文件的代码

|  |
| --- |
| @RequestMapping(**"download"**) **public void** download(String file,  HttpServletRequest request,  HttpServletResponse response) **throws** Exception{  *//file文件的路径 fname表示原始文件名*  *//获取文件名称下载的文件名称(设置下载的文件名称)*  String fname=file.substring(file.lastIndexOf("/")+1);  *//读取下载文件的内容* String absolutePath=request.getSession().getServletContext().getRealPath(file);  FileInputStream fis=**new** FileInputStream(absolutePath);  **byte** [] bs=**new byte**[fis.available()]; *//fis.available()表求文件大小* fis.read(bs);  fis.close();   *//实现文件下载  //1.设置下载的文件头 注意:如果不指定下载报头，则将数据写入响应流中  //response.setHeader("Content-Disposition","attachment;fileName=下载文件名称");* response.setHeader(**"Content-Disposition"**,**"attachment;fileName="**+fname);  *//2.设置文件的类型* response.setContentType(**"application/vnd.android.package-archive"**);  *//3.将数据写入到响应流中即可* ServletOutputStream sos=response.getOutputStream();  sos.write(bs);  sos.flush(); sos.close(); } |

1. 前端页面的超链接

<**a href="${**pageContext.request.contextPath**}/download?file=${**appVersion.downloadlink**}"**>下载</**a**>

1. 验证码
2. 为什么需要验证码:

为了安全、防止循环提交

1. 生成验证码（验证码其实是java代码生成的图片）

略

1. 在网页中使用img标签引用验证码图片

<**img src="/code"**/>

1. 基于验证码实现登入

思路:先判断验证码是不是合法的，如果合法就去实现登入的业务

|  |
| --- |
| *//inputCode表示输入的验证码* @RequestMapping(**"/login"**) **public** String login(String inputCode,String username, String password, HttpSession session,Model model){  *//1.判断验证码  //获取生成的验证码* String code=session.getAttribute(**"codeValidate"**).toString();  **if**(inputCode.equalsIgnoreCase(code)) {  *//调用业务* DevUser devUser = **devUserService**.login(username, password);  **if** (devUser == **null**) {  model.addAttribute(**"error"**,**"用户名密码不正确"**);  *//登入失败* **return "devlogin"**;  } **else** {  *//登入成功,使用session保存用户的信息* session.setAttribute(**"devUser"**, devUser);  session.setMaxInactiveInterval(600); *//10分钟有效* **return "developer/main"**;  }  }**else** {  model.addAttribute(**"error"**,**"验证码输入不正确"**);  **return "devlogin"**;  } } |

1. 看不清、换一张

<**img src="/code" onclick="this**.**src**=**'/code?b='**+***Math***.random()**"**/>