Workbook workbook = **null**;  
 File importFile = **null**;  
 InputStream is = **null**;  
 List<LinkedList<String>> totalDataList = **new** LinkedList<>();  
  
 **if** (file.equals(**""**) || file.getSize() <= 0) {  
 file = **null**;  
 } **else** {  
 is = file.getInputStream();  
 importFile = **new** File(file.getOriginalFilename());  
 *inputStreamToFile*(is, importFile);  
 is.close();  
 }  
 List<?> pictures = **new** LinkedList<>();  
  
 **try** {  
 is = **new** FileInputStream(importFile);  
 **if** (importFile.getAbsolutePath().endsWith(**".xls"**)) {  
 workbook = **new** HSSFWorkbook(is);  
 } **else if** (importFile.getAbsolutePath().endsWith(**".xlsx"**)) {  
 workbook = **new** XSSFWorkbook(is);  
 }  
 **int** sheetCount = workbook.getNumberOfSheets();  
 **if** (sheetCount > 0) {  
 *// 文本内容* **for** (**int** i = 0; i < sheetCount; i++) {  
 Sheet sheet = workbook.getSheetAt(0);  
 LinkedList<String> midList = **new** LinkedList<>();  
 **for** (**int** rownum = sheet.getFirstRowNum() + 1; rownum <= sheet.getLastRowNum(); rownum++) {  
 Row row = sheet.getRow(rownum);  
 **if** (row == **null** || row.getFirstCellNum() < 0) {  
 **break**;  
 }  
  
 **for** (**int** columnnum = row.getFirstCellNum(); columnnum <= row  
 .getLastCellNum(); columnnum++) {  
 String cellValue = getCellValue(row.getCell(columnnum));  
 **if** (cellValue.trim().equals(**"无"**)) {  
 cellValue = **null**;  
 }  
 midList.add(cellValue);  
 *//将数据按照行，存储到list中* }  
 *//所有列的信息已经解析放在list中* totalDataList.add(midList);  
 midList = **new** LinkedList<>();  
 }  
 }  
  
 *// 图片内容处理* pictures = workbook.getAllPictures();  
 }  
 } **catch** (IOException ignored) {  
 } **finally** {  
 **try** {  
 **if** (workbook != **null**) {  
 workbook.close();  
 }  
 **if** (is != **null**) {  
 is.close();  
 }  
 } **catch** (IOException e) {  
 }  
 }  
 *//针对拿到的文本数据和图片数据进行整理* **for** (**int** i = 0; i < totalDataList.size(); i++) {  
  
 *//时间处理* SimpleDateFormat simpleDateFormat = **new** SimpleDateFormat(**"yyyy-MM-dd"**);*//注意月份是MM* BusinessMapProjectDTO businessMapProjectDTO = **new** BusinessMapProjectDTO();  
 List<String> stringList = totalDataList.get(i);  
 *//省份不用处理* String province = stringList.get(0);  
 String city = stringList.get(1);  
 String county = stringList.get(2);  
 String beginTime = stringList.get(3).trim();  
 String endTime = stringList.get(4);  
  
 **if** (endTime != **null**) {  
 endTime = endTime.trim();  
 Date endTimeDate = simpleDateFormat.parse(endTime);  
 businessMapProjectDTO.setEndtime(endTimeDate);  
 } **else** {  
 businessMapProjectDTO.setEndtime(DateUtils.*getNowDate*());  
 }  
 String projectName = stringList.get(5);  
 String projectAbbreviation = stringList.get(6);  
 String projectIntroduction = stringList.get(7);  
 String projectProgress = stringList.get(8);  
 String prprojectCategory = stringList.get(9);  
 String pcprojectCategory = stringList.get(10);  
 String customer = stringList.get(11);  
 String salary = stringList.get(12);  
  
 String officerName = stringList.get(13);  
 String participants = stringList.get(14);  
 String status = stringList.get(15);  
 String deptName = stringList.get(16);  
 String fillInTime = stringList.get(17).trim();  
 String submitterName = stringList.get(18);  
  
  
 businessMapProjectDTO.setUpdateTime(DateUtils.*getNowDate*());  
 businessMapProjectDTO.setProvince(province);  
 businessMapProjectDTO.setCity(city);  
 businessMapProjectDTO.setCounty(county);  
 businessMapProjectDTO.setProjectName(projectName);  
 businessMapProjectDTO.setProjectAbbreviation(projectAbbreviation);  
 businessMapProjectDTO.setProjectIntroduction(projectIntroduction);  
 businessMapProjectDTO.setProjectProgress(projectProgress);  
 businessMapProjectDTO.setSubmitterName(submitterName);  
 **if** (prprojectCategory == **null**) {  
 businessMapProjectDTO.setProjectCategory(pcprojectCategory);  
 } **else** {  
 businessMapProjectDTO.setProjectCategory(prprojectCategory);  
 }  
 businessMapProjectDTO.setCustomer(customer);  
 *//收入，类型为BigDecimal* **if** (salary != **null**) {  
 businessMapProjectDTO.setSalary(**new** BigDecimal(salary));  
 }  
 *//是否重点：默认非重点* businessMapProjectDTO.setKeynote((**byte**) 0);  
 businessMapProjectDTO.setOfficerName(officerName);  
 businessMapProjectDTO.setParticipants(participants);  
 *//项目状态* **if** (status.equals(**"已完成"**)) {  
 businessMapProjectDTO.setStatus((**byte**) 1);  
 } **else** {  
 businessMapProjectDTO.setStatus((**byte**) 0);  
 }  
 businessMapProjectDTO.setDeptName(deptName);  
  
 Date beginTimeDate = simpleDateFormat.parse(beginTime);  
 businessMapProjectDTO.setBegintime(beginTimeDate);  
  
 Date fillInTimeDate = simpleDateFormat.parse(fillInTime);  
 businessMapProjectDTO.setFillInTime(fillInTimeDate);  
  
  
 *//接着处理表中另外11个字段* BusinessMapProvinceLocation businessMapProvinceLocation = **businessMapProvinceLocationService**.selectByProvinceAndCity(province, city);  
 String provinceCode;  
 *//经纬度* String longitude;  
 String latitude;  
 **if** (businessMapProvinceLocation == **null**) {  
 **return** AjaxResult.*error*(**"省份信息填写有误"**);  
 } **else** {  
 provinceCode = businessMapProvinceLocation.getCode();  
 longitude = businessMapProvinceLocation.getLongitude();  
 latitude = businessMapProvinceLocation.getLatitude();  
 businessMapProjectDTO.setProvinceCode(provinceCode);  
 businessMapProjectDTO.setLongitude(longitude);  
 businessMapProjectDTO.setLatitude(latitude);  
 }  
 *//市code* List<CityCodeVo> cityList = LocationUtil.*getCityByAdCode*(provinceCode);  
 cityList.forEach(cityCodeVo -> {  
 **if** (cityCodeVo.getName().equals(city)) {  
 businessMapProjectDTO.setCityCode(cityCodeVo.getCode());  
 }  
 });  
 *//县code* List<CityCodeVo> countyList = LocationUtil.*getCountryByAdCode*(provinceCode, businessMapProjectDTO.getCityCode());  
 countyList.forEach(cityCodeVo -> {  
 **if** (cityCodeVo.getName().equals(county)) {  
 businessMapProjectDTO.setCountyCode(cityCodeVo.getCode());  
 }  
 });  
 *//负责人id* SysUser sysUser = **iSysUserService**.selectUserByNickName(officerName);  
 **if** (sysUser == **null**) {  
 **return** AjaxResult.*error*(**"负责人信息有误"**);  
 } **else** {  
 businessMapProjectDTO.setOfficerId(sysUser.getUserId());  
 }  
 *//填报人* sysUser = **iSysUserService**.selectUserByNickName(submitterName);  
 **if** (sysUser == **null**) {  
 **return** AjaxResult.*error*(**"填报人信息有误"**);  
 } **else** {  
 businessMapProjectDTO.setSubmitterId(sysUser.getUserId());  
 businessMapProjectDTO.setDeptId(sysUser.getDeptId());  
 }  
  
 *//开始处理图片* PictureData picture = (PictureData) pictures.get(i);  
 **byte**[] data = picture.getData();  
 *//图片扩展名* String pictureExtension = **"."** + picture.suggestFileExtension();  
 *//创建临时文件* File file1 = File.*createTempFile*(**"smart"**, **"map"**);  
 FileOutputStream out = **new** FileOutputStream(file1);  
 out.write(data);  
 out.close();  
  
 AjaxResult ajaxResult = uploadImageFromStream(file1, pictureExtension);  
 **if** (ajaxResult.get(**"code"**).equals(200)) {  
 *//上传成功了，保存attachId和attachurl* Attach attach = (Attach) ajaxResult.get(**"data"**);  
 businessMapProjectDTO.setAttachid(attach.getAttachId());  
 businessMapProjectDTO.setAttachUrl(attach.getAttachUrl());  
 }  
 *//插入数据* **int** insertFlag = **businessMapProjectService**.insert(businessMapProjectDTO);  
 **if** (insertFlag < 0) {  
 **return** AjaxResult.*error*(**"插入失败"**);  
 }  
 }  
 **return** AjaxResult.*success*(**"成功"**);  
}  
  
**private** AjaxResult uploadImageFromStream(File file, String fileExtension) **throws** IOException {  
  
 InputStream inputStream = **new** FileInputStream(file);  
 String fileName = file.getName() + fileExtension;  
  
 BusinessMapAttachVo businessMapAttach;  
  
 businessMapAttach = **businessMapAttachService**.selectByAttachName(fileName);  
 *//存在同名的附件信息* **if** (businessMapAttach != **null**) {  
 *//重命名该文件* String[] fileNameStr = fileName.split(**"\\."**);  
 String randomStr = RandomStringUtils.*randomAlphanumeric*(5);  
 fileName = fileNameStr[0] + randomStr + **"."** + fileNameStr[1];  
 }  
  
 **boolean** flag = FtpFileUtil.*uploadFile*(fileName, inputStream, **ftpHostName**, **ftpPort**, **ftpUserName**, **ftpPassWord**);  
  
 **int** addToAttach = 0;  
  
 *//文件上传成功则在数据库中写入附件库* **if** (flag) {  
  
 businessMapAttach = **new** BusinessMapAttachVo();  
 businessMapAttach.setAttachName(fileName);  
 businessMapAttach.setRemoteDirPath(**basepath**);  
 businessMapAttach.setAttachUrl(**downloadPath** + **"/"** + fileName);  
 businessMapAttach.setCreateTime(DateUtils.*getNowDate*());  
 businessMapAttach.setUpdateTime(DateUtils.*getNowDate*());  
 businessMapAttach.setAttachSize(file.length());  
  
 *//此处返回的是附件表的主键ID* addToAttach = **businessMapAttachService**.insert(businessMapAttach);  
  
 }  
  
 **if** (addToAttach > 0) {  
  
 *//再查一遍* BusinessMapAttachVo businessMapAttachVo = **businessMapAttachService**.selectByAttachName(fileName);  
  
 Attach attach = Attach.*builder*()  
 .attachId(businessMapAttachVo.getAttachId())  
 .attachUrl(businessMapAttach.getAttachUrl())  
 .build();  
  
 **return** AjaxResult.*success*(attach);  
 } **else** {  
 **return** AjaxResult.*error*(**"文件上传失败"**);  
 }  
}  
  
  
**private** String getCellValue(Cell cell) {  
 **if** (cell == **null**) {  
 **return ""**;  
 }  
 *// 都按文本格式读取* cell.setCellType(CellType.***STRING***);  
 **return** cell.getStringCellValue();  
}  
  
*//获取流文件***private static void** inputStreamToFile(InputStream ins, File file) {  
 **try** {  
 OutputStream os = **new** FileOutputStream(file);  
 **int** bytesRead = 0;  
 **byte**[] buffer = **new byte**[8192];  
 **while** ((bytesRead = ins.read(buffer, 0, 8192)) != -1) {  
 os.write(buffer, 0, bytesRead);  
 }  
 os.close();  
 ins.close();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
}  
  
*/\*\*  
 \* 删除本地临时文件  
 \*  
 \** ***@param file*** *\*/***public static void** delteTempFile(File file) {  
 **if** (file != **null**) {  
 File del = **new** File(file.toURI());  
 del.delete();  
 }  
  
  
}