

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

☰ 目录视图

☰ 摘要视图

RSS 订阅

从创业到再就业，浅谈对程序员职业生涯的看法 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

Excel 报表导入导出

标签： excel 报表 导入 导出

2017-04-30 19:19 172人阅读 评论(1)

☰ 分类： web后台 (12) ▼

❗ 版权声明：本文为博主原创文章，未经博主允许不得转载。

使用 Excel 进行报表的导入导出，首先下载相关的 jar 和 excel util。

[Excel Util 下载地址](#)

把相应的 jar 和 util 类放到项目中就可以在需要打印出来的实体类的get方法加上注解@ExcelField

```

    @NotNull(message = "倍率不能为空")
    @ExcelField(title = "倍率", align = 1, sort = 30)
    public Double getRate() {
        return rate;
    }

    public void setRate(Double rate) {
        this.rate = rate;
    }

    @Length(min = 0, max = 4, message = "仪表类型长度必须介于 0 和 4 之间")
    @ExcelField(title = "仪表类型", align = 1, sort = 40)
    public String getMeterType() {
        return meterType;
    }

```

Excel 报表数据模板下载代码：

controller层：

```

[java]
01. <span style="font-size:14px;">
02. /**
03.  * 下载导入Excel表信息数据模板
04.  */
05. @RequiresPermissions("metermessage:meterMessage:view")
06. @RequestMapping(value = "import/template")
07. public String importFileTemplate(HttpServletResponse response, RedirectAttributes redirectAttributes, String houseType) {
08.     try {
09.         meterMessageService.importFileTemplate(response);
10.         return null;
11.     } catch (Exception e) {
12.         addMessage(redirectAttributes, "导入模板下载失败！失败信息: " + e.getMessage());
13.     }
14.     return "redirect:" + Global.getAdminPath() + "/metermessage/meterMessage/?repage&houseType=" + houseType;
15. }
16.
17. </span>

```

service层：

```
[java]
01. <span style="font-size:14px;">
02. //Excel表信息数据模板
03. public void importFileTemplate(HttpServletResponse response) throws IOException {
04.     String fileName = "Excel表信息数据导入模板.xlsx";
05.     List<MeterMessage> list = Lists.newArrayList();
06.     MeterMessage meter = new MeterMessage();
07.     meter.setOfficeName("*****");
08.     meter.setUnitNumber("1");
09.     meter.setFloor("1");
10.     meter.setRoomNumber("01");
11.     meter.setMeterNo("001");
12.     meter.setRate(new Double(10));
13.     meter.setMeterTypeLabel("水表");
14.     list.add(meter);
15.     new ExportExcel("Excel表信息数据", MeterMessage.class, 2).setDataList(list).write(response, fileName).dispose();
16. }
17.
18. </span>
```

Excel表信息导入代码：

controller层：

```
[java]
01. <span style="font-size:14px;">
02. /**
03.  * 导入Excel信息数据
04.  */
05. @RequiresPermissions("metermessage:meterMessage:edit")
06. @RequestMapping(value = "import", method = RequestMethod.POST)
07. public String importFile(MultipartFile file, RedirectAttributes redirectAttributes, String houseType) {
08.     if (Global.isDemoMode()) {
09.         addMessage(redirectAttributes, "演示模式，不允许操作！");
10.         return "redirect:" + Global.getAdminPath() + "/metermessage/meterMessage/?repage&houseType=" + houseType;
11.     }
12.     try {
13.         String message = meterMessageService.importFile(file);
14.         addMessage(redirectAttributes, message);
15.     } catch (Exception e) {
16.         addMessage(redirectAttributes, "导入Excel信息失败！失败信息: " + e.getMessage());
17.     }
18.     return "redirect:" + Global.getAdminPath() + "/metermessage/meterMessage/?repage&houseType=" + houseType;
19. }
20.
21.
22. </span>
```

service层：

```
[java]
01. <span style="font-size:14px;">
02.
03. //导入excel信息数据
04. @Transactional(readOnly = false)
05. public String importFile(MultipartFile file) throws Exception {
06.     int successNum = 0;
07.     int failureNum = 0;
08.     int row = 3;
09.     StringBuilder failureMsg = new StringBuilder();
10.     ImportExcel ei = new ImportExcel(file, 1, 0);
11.     List<MeterMessage> list = ei.getDataList(MeterMessage.class);
12.     Map<String, Object> map = new HashMap<String, Object>(); // 去重复查询用
13.     for (int i = 0; i < list.size(); i++) {
14.         MeterMessage meter = list.get(i);
15.
16.         String officeName = meter.getOfficeName();
17.         String unitNumber = meter.getUnitNumber();
18.         String floor = meter.getFloor();
19.         String roomNumber = meter.getRoomNumber();
```

```

20. String meterTypeLabel = meter.getMeterTypeLabel();
21. String meterNo = meter.getMeterNo();
22. Double rate = meter.getRate();
23.
24. if (StringUtils.isNotBlank(officerName)) {
25.     if (!map.containsKey(officerName)) {
26.         List<Office> officerList = officerDao.findOfficeByName(officerName);
27.         if (officerList != null && officerList.size() > 0) {
28.             meter.setOffice(officerList.get(0));
29.             map.put(officerName, officerList.get(0));
30.         } else {
31.             failureMsg.append("<br/>第" + row + "行数据: 小区' " + officerName
32.                 + "' 不存在,请先去'平台设置-->单位管理'创建该小区 ");
33.             failureNum++;
34.         }
35.     } else {
36.         meter.setOffice((Office) (map.get(officerName)));
37.     }
38. } else {
39.     failureMsg.append("<br/>第" + row + "行数据: 小区 不可为空 ");
40.     failureNum++;
41. }
42.
43. String officeId = meter.getOffice().getId();
44. if (StringUtils.isNotBlank(officeId) && StringUtils.isNotBlank(unitNumber) && StringUtils.isNotBlank(floor) && String
45.     if (!map.containsKey(officeId+unitNumber+floor+roomNumber)) {
46.         HouseInformation house = houseInformationDao.findHouseIdByCondition(officeId, unitNumber, floor, roomNumber);
47.         if (house != null) {
48.             meter.setHouse(house);
49.             map.put(officeId+unitNumber+floor+roomNumber, house);
50.         } else {
51.             failureMsg.append("<br/>第" + row + "行数据: 房屋' " + officerName
52.                 + "' 不存在;");
53.             failureNum++;
54.         }
55.     } else {
56.         meter.setHouse((HouseInformation) (map.get(officeId+unitNumber+floor+roomNumber)));
57.     }
58. } else {
59.     failureMsg.append("<br/>第" + row + "行数据: 房屋 不可为空 ");
60.     failureNum++;
61. }
62.
63. if (StringUtils.isNotBlank(meterTypeLabel)) {
64.     if (!map.containsKey(meterTypeLabel)) {
65.         String meterType = DictUtils.getDictValue(meterTypeLabel, "meter_type", "1");
66.         if (StringUtils.isNotBlank(meterType)) {
67.             meter.setMeterType(meterType);
68.             map.put(meterTypeLabel, meterType);
69.         } else {
70.             failureMsg.append("<br/>第" + row + "行数据: 仪表类型 '" + meterTypeLabel
71.                 + "' 不存在,请先去'平台设置-->系统设置-->字典管理'创建该仪表类型 ");
72.             failureNum++;
73.         }
74.     } else {
75.         meter.setMeterType((String)(map.get(meterTypeLabel)));
76.     }
77. } else {
78.     failureMsg.append("<br/>第" + row + "行数据: 仪表类型 不可为空 ");
79.     failureNum++;
80. }
81.
82. if (StringUtils.isNotBlank(meterNo)) {
83.     meter.setMeterNo(meterNo);
84. } else {
85.     failureMsg.append("<br/>第" + row + "行数据: 仪表编号 不可为空 ");
86.     failureNum++;
87. }
88.
89. if (rate != null) {
90.     meter.setRate(rate);
91. } else {
92.     failureMsg.append("<br/>第" + row + "行数据: 倍率 不可为空 ");
93.     failureNum++;
94. }
95.
96. meter.preInsert();
97. row++;
98. }

```

```
99.
100.         if (failureNum > 0) {
101.             failureMsg.insert(0, "失败 " + failureNum + " 条Excel信息, 导入信息如下: ");
102.         } else {
103.             try {
104.                 if (list != null && list.size() > 0) {
105.                     dao.insertMeterList(list);
106.                     successNum = list.size();
107.                 }
108.             } catch (ConstraintViolationException ex) {
109.                 failureMsg.append("<br/>excel信息 导入失败: ");
110.                 List<String> messageList = BeanValidators.extractPropertyAndMessageAsList(ex, ": ");
111.                 for (String message : messageList) {
112.                     failureMsg.append(message + "; ");
113.                     failureNum++;
114.                 }
115.             } catch (Exception ex) {
116.                 failureNum++;
117.                 failureMsg.append("<br/>excel信息 导入失败: " + ex.getMessage());
118.             }
119.         }
120.         return "已成功导入 " + successNum + " 条excel信息" + failureMsg;
121.     }
122.
123.
124. </span>
```

顶 踩
0 0

- 上一篇 [fmt:formatNumber use locale display negative currency in -\\$xxx.xx format in JSTL](#)
- 下一篇 [JPA映射持久化对象 \(Entity \)](#)

相关文章推荐

- 导入导出封装工具类 (二) jXLS Excel报表生成工...
- JXL制作属于自己的excel导入导出报表工具(二)xm...
- java jxl excel 导入导出的 总结 (建立超链接, 以...
- 人员卡片信息报表导入导出
- Spring MVC环境下用poi技术实现Excel的导入导出
- JXL制作属于自己的excel导入导出报表工具(三)do...
- J2EE项目系列 (四) --SSM框架构建积分系统和基...
- JXL制作属于自己的excel导入导出报表工具(一)背...
- C#导入导出Excel通用类(SamWang)
- 解密java实现Excel导入导出



定量控制器



代缴社保公司



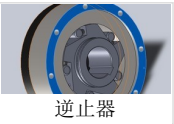
断路器测试仪



电磁加热设备



财务经理培训



逆止器



会计做账软件

猜你在找

- 机器学习之概率与统计推断
- 机器学习之凸优化
- 响应式布局全新探索
- 深度学习基础与TensorFlow实践
- 前端开发在线峰会
- 机器学习之数学基础
- 机器学习之矩阵
- 探究Linux的总线、设备、驱动模型
- 深度学习之神经网络原理与实战技巧
- TensorFlow实战进阶：手把手教你做图像识别应用

查看评论