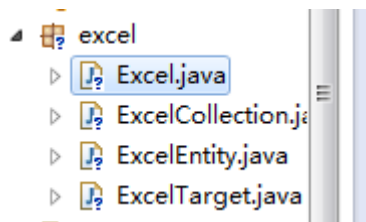


Excel 导出

使用到的注解类:



注解介绍:

ExcelTarget(导出类上面)

Id 设置导出 excel 的 ID,可以与字段做对应选择合适的字段

Excel (应用到字段上面)

exportName 导出到 excel 对应的名称如 name—“姓名”

exportFieldWidth 字段的 cell 宽度 默认为 10

exportFieldHeight 字段的高度(暂时仅对图片有效) 默认为 10

exportConvertSign 设置是否使用转换函数 1-使用,0 不使用默认为 0

exportType 导出类型 1 是文本 2 是图片 默认是文本

orderNum 导出顺序

isWrap 是否换行

例子

只是用名称

```
// 导出
@Excel(exportName="订单备注")
private java.lang.String content;
```

多个字段设置

```
// 导出
@Excel(exportName="订单备注",exportFieldWidth=10,exportConvertSign=1)
private java.lang.String content;
```

因为设置 covert 为 1,所以要在实体里面写一个 covert 函数

```
public String convertGetContent(){
    return "";
}
```

导出图片

```
private java.lang.String picUrl;
/**图片地址*/
@Excel(exportName="产品图片",exportType=2,exportFieldWidth=20,exportFieldHeight=20)
private java.lang.String picUrl;
```

对地址做映射

ExcelEntity (字段注解 标记为导出类,程序继续往下寻找字段)

(ExcelEntity 可以多层)

exportName 导出名称,这个不会显示到 excel 里面,主要是和 target 的 ID 做关联的

默认空即和所以导出都关联

```
/**
 * 客户信息
 */
@Entity
private CustomerEntity customer;
```

这样来做关联,实体类里里面做字段标记()

```
public class CustomerEntity implements
    /**id*/
    private java.lang.String id;
    /**业务员*/
    private TSUser user;
    /**名称*/
    @Excel(exportName="客户姓名")
    private java.lang.String name;
```

ExcelCollection(导出集合 支持 Collection 实现类,用来实现一对多的导出)

(list 里面不能再嵌套 list 了,暂时只支持一层 list)

exportName 显示到 excel 里面

```
/**
 * 订单详情
 */
@ExcelCollection(exportName="订单详情")
private List<OrderDetailEntity> list;
```

然后在这个 list 里面的实体来做标记

```
public class OrderDetailEntity implements java
    /**id*/
    private java.lang.String id;
    /**订单*/
    private OrderInfoEntity orderInfo;
    /**产品*/
    @ExcelEntity
    private ProductEntity product;
    /**产品风格*/
    @ExcelEntity
    private ProductStyleEntity productStyle;
    /**产品单价*/
    private java.lang.Double price;
    /**备注*/
    private java.lang.String content;
    /**数量*/
    @Excel(exportName="订单数量")
    private java.lang.String count;
```

进行属性关联

ExcelTitle 表头

```
public ExcelTitle(){
}

public ExcelTitle(String title,String time,String userName,String sheetName){
    this.title = title;
    this.time = time;
    this.userName = userName;
    this.sheetName = sheetName;
}
```

Title 这个表格名称 time 制表时间,username 制表人,sheetName 这个 sheet 的 Name

效果

准备发货列表	
	制表人: 管理员 制表时间: 2013-08-25

ExportExcelUtil 工具类

多个 sheet 个表格

```
/**
 * 一个excel 创建多个sheet
 *
 * @param list
 *      多个Map key title 对应表格Title key entity 对应表格对应实体 key data
 *      Collection 数据
 * @return
 */
public static HSSFWorkbook exportExcel(List<Map<String, Object>> list) {
    HSSFWorkbook workbook = new HSSFWorkbook();
    for (Map<String, Object> map : list) {
        createSheetInUserModel2File(workbook,
            (ExcelTitle) map.get("title"),
            (Class<?>) map.get("entity"),
            (Collection<?>) map.get("data"));
    }
    return workbook;
}
```

单个 sheet 的表格

```
/**
 *
 * @param entity
 *      表格标题属性
 * @param pojoClass
 *      Excel对象Class
 * @param dataSet
 *      Excel对象数据List
 */
public static HSSFWorkbook exportExcel(ExcelTitle entity,
    Class<?> pojoClass, Collection<?> dataSet) {
    HSSFWorkbook workbook = new HSSFWorkbook();
    createSheetInUserModel2File(workbook, entity, pojoClass, dataSet);
    return workbook;
}
```

使用方法,先给要导出的实体进行注解之后和以前差不多

```
List<OrderInfoEntity> orders = orderInfoService
    .getListByCriteriaQuery(cq, false);
workbook = ExportExcelUtil.exportExcel(getExcelTitle(),
    OrderInfoEntity.class, orders);
fOut = response.getOutputStream();
workbook.write(fOut);
```

这样就可以了

最后附一张效果图

准备发货列表										
制表人: 管理员 制表时间: 2013-08-25										
客户姓名	客户地址	客户电话	订单备注	订单时间	产品名称	产品图片	产品尺寸	产品颜色	订单数量	
小明	东华小学	1879740028		2013-08-23 23:15:24.0	010床头柜		650*450*620	柚木色	1	
小李	2133	2131		2013-08-08 20:32:12.0	006妆凳		450*400*460	黑色描银	1	