冲奶粉的奶爸 关注 TA的个人主页 > 评论 原创 粉丝 喜欢 37 10 53 本文链接: https://blog.csdn.net/qq_32258777/article/details/89031479 等级: 博客 3 访问: 3万+ 积分: 646 排名: 11万+ 勋章: 📵 🎒

最新文章

利于iText实现html或者freemark等模板 生成PDF

读weixin-java-tools源码获取的开发小技 巧

spring boot 精华 在线生成二维码的api接口

工厂模式的概述及应用

分类专栏 1篇 **Memory Analyzer** 1篇 **Demo** 15篇 开发实用技巧 **SpringCloud** 5篇 9篇 开发工具类 多线程 JVM **SpringBoot**

展开

归档 2019年4月 2篇 3篇 2019年3月 2019年2月 9篇 2019年1月 2篇 2018年12月 1篇 2018年11月 2018年10月 1篇 2018年9月

热门文章

史上最全的Excel导入导出之easyexcel 阅读数 16885

SpringBoot 整合 Mybatis-Plus + Mysql 阅读数 3679

展开

mybatis-plus之代码生成器

阅读数 1715 JVM之栈和栈帧

阅读数 1048

策略模式+反射-解决多重if-else if问题 阅读数 786

最新评论

mybatis-plus之代码生成器

hyly_zhard: [reply]weixin_42812986[/reply] 因 为它用的是freemarker的模板引擎,所以需要: ...

史上最全的Excel导入导出之ea...

weixin_44248561: [reply]qq_32258777[/reply] [code=java] /** * 读取少于1000行的excle, ī ...

史上最全的Excel导入导出之ea... qq_32258777: [reply]weixin_44248561[/reply]

兄弟,,没太听懂你的意思 史上最全的Excel导入导出之ea...

qq_32258777: [reply]weixin_44248561[/reply] 蓝色应该是12

史上最全的Excel导入导出之ea... weixin_44248561: [reply]qq_32258777[/reply]

我在做导入的时候遇到了一个问题,不知道怎: ...



CSDN学院 CSDN企业招聘

■ QQ客服 ■ kefu@csdn.net **2** 400-660-0108 ● 客服论坛 工作时间 8:30-22:00

关于我们 招聘 广告服务 网站地图 協 百度提供站内搜索 京ICP备19004658号 ©1999-2019 北京创新乐知网络技术有限

网络110报警服务 经营性网站备案信息 北京互联网违法和不良信息举报中心 中国互联网举报中心 家长监护 版权申诉

更上最全的Excel导入导出之easyexcel

CSDN 首页 博客 学院 下载 论坛 APP 问答 商城 活动 VIP会员 (续费9折) 专题 招聘 ITeye GitChat 图文课

置顶 2019年04月04日 17:56:55 冲奶粉的奶爸 阅读数 17271 更多 版权声明:本文为博主原创文章,遵循 CC 4.0 by-sa 版权协议,转载请附上原文出处链接和本声明。

喝水不忘挖井人,感谢阿里巴巴项目组提供了easyexcel工具类,github地址: https://github.com/alibaba/easyexcel

Python工程师

凸

分享海报说明

45

=:

文章目录

环境搭建

读取excel文件

小于1000行数据

默认读取

指定读取

大于1000行数据

默认读取

指定读取

导出excle

单个Sheet导出 无模型映射导出

模型映射导出

多个Sheet导出

工具类 测试类

环境搭建

- easyexcel 依赖 (必须)
- springboot (不是必须)
- lombok (不是必须)
 - <dependency>
 - <groupId>com.alibaba
- <artifactId>easyexcel</artifactId>
- <version>1.1.2-beat1 </dependency>
- <groupId>org.projectlombok</groupId>
- <artifactId>lombok</artifactId> 10 <version>1.18.2
- 11 </dependency>

读取excel文件

小于1000行数据

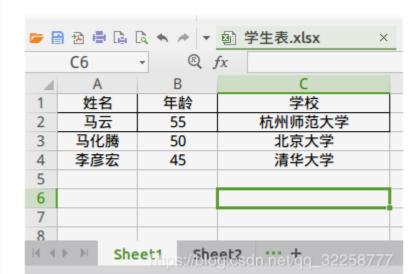
默认读取

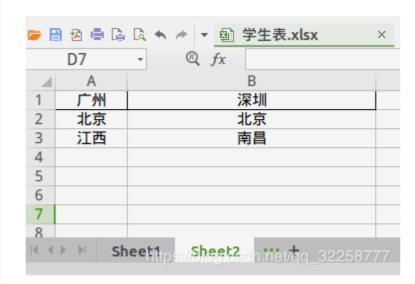
读取Sheet1的全部数据

- 1 String filePath = "/home/chenmingjian/Downloads/学生表.xlsx";
- 2 List<Object> objects = ExcelUtil.readLessThan1000Row(filePath);

指定读取

下面是学生表.xlsx中Sheet1, Sheet2的数据





获取Sheet1表头以下的信息

- 1 | String filePath = "/home/chenmingjian/Downloads/学生表.xlsx"; 2 // 第一个1代表sheet1, 第二个1代表从第几行开始读取数据, 行号最小值为0
- 3 | Sheet sheet = new Sheet(1, 1);
- 4 List<Object> objects = ExcelUtil.readLessThan1000Row(filePath, sheet);

获取Sheet2的所有信息

- 1 String filePath = "/home/chenmingjian/Downloads/学生表.xlsx"; 2 Sheet sheet = new Sheet(2, 0);
- 3 List<Object> objects = ExcelUtil.readLessThan1000Row(filePath, sheet);

大于1000行数据

默认读取

- 1 String filePath = "/home/chenmingjian/Downloads/学生表.xlsx"; 2 List<Object> objects = ExcelUtil.readMoreThan1000Row(filePath);

指定读取

- 1 String filePath = "/home/chenmingjian/Downloads/学生表.xlsx";
- 2 | Sheet sheet = new Sheet(1, 2);
- 3 List<Object> objects = ExcelUtil.readMoreThan1000Row(filePath, sheet);

导出excle

单个Sheet导出

无模型映射导出

- 1 | String filePath = "/home/chenmingjian/Downloads/测试.xlsx";
- 2 List<List<Object>> data = new ArrayList<>();



2019/8/27 史上最全的Excel导入导出之easyexcel-

```
史上最全的Excel导入导出之easyexcel - 点缀星空 - CSDN博客
   3 | data.add(Arrays.asList("111","222","333"));
   4 data.add(Arrays.asList("111","222","333"));
   5 data.add(Arrays.asList("111","222","333"));
   6 List<String> head = Arrays.asList("表头1", "表头2", "表头3");
   7 | ExcelUtil.writeBySimple(filePath,data,head);
结果
                              表头3
                 表头2
 111
              222
                          333
                          333
              222
  111
              222
                          333
 111
模型映射导出
1、定义好模型对象
   1 package com.springboot.utils.excel.test;
   3 import com.alibaba.excel.annotation.ExcelProperty;
   4 import com.alibaba.excel.metadata.BaseRowModel;
   5 import lombok.Data;
   6 import lombok.EqualsAndHashCode;
       * @description:
      * @author: chenmingjian
  11
       * @date: 19-4-3 14:44
  12
  13 @EqualsAndHashCode(callSuper = true)
  14 @Data
     public class TableHeaderExcelProperty extends BaseRowModel {
  15
  16
  17
          /**
  18
          * value: 表头名称
  19
          * index: 列的号, 0表示第一列
  20
          @ExcelProperty(value = "姓名", index = 0)
  21
  22
          private String name;
  23
  24
          @ExcelProperty(value = "年龄",index = 1)
  25
          private int age;
  26
  27
          @ExcelProperty(value = "学校",index = 2)
  28
          private String school;
  29 }
  30
2、调用方法
   1 | String filePath = "/home/chenmingjian/Downloads/测试.xlsx";
   2 | ArrayList<TableHeaderExcelProperty> data = new ArrayList<>();
       for(int i = 0; i < 4; i++){
           TableHeaderExcelProperty tableHeaderExcelProperty = new TableHeaderExcelProperty();
            tableHeaderExcelProperty.setName("cmj" + i);
            tableHeaderExcelProperty.setAge(22 + i);
           tableHeaderExcelProperty.setSchool("清华大学" + i);
            data.add(tableHeaderExcelProperty);
  10
       ExcelUtil.writeWithTemplate(filePath,data);
多个Sheet导出
1、定义好模型对象
   1 package com.springboot.utils.excel.test;
   3 | import com.alibaba.excel.annotation.ExcelProperty;
   4 import com.alibaba.excel.metadata.BaseRowModel;
   5 import lombok.Data;
   6 | import lombok.EqualsAndHashCode;
      /**
   9 * @description:
      * @author: chenmingjian
  11 * @date: 19-4-3 14:44
  12 */
  13 | @EqualsAndHashCode(callSuper = true)
  15 | public class TableHeaderExcelProperty extends BaseRowModel {
  16
  17
          /**
  18
          * value: 表头名称
  19
          * index: 列的号, 0表示第一列
  20
  21
          @ExcelProperty(value = "姓名", index = 0)
  22
          private String name;
  23
  24
          @ExcelProperty(value = "年龄",index = 1)
  25
          private int age;
  26
  27
          @ExcelProperty(value = "学校",index = 2)
  28
          private String school;
  29 }
  30
2、调用方法
   1  ArrayList<ExcelUtil.MultipleSheelPropety> list1 = new ArrayList<>();
      for(int j = 1; j < 4; j++){
            ArrayList<TableHeaderExcelProperty> list = new ArrayList<>();
            for(int i = 0; i < 4; i++){
               TableHeaderExcelProperty tableHeaderExcelProperty = new TableHeaderExcelProperty();
               tableHeaderExcelProperty.setName("cmj" + i);
               tableHeaderExcelProperty.setAge(22 + i);
               tableHeaderExcelProperty.setSchool("清华大学" + i);
               list.add(tableHeaderExcelProperty);
  10
  11
  12
            Sheet sheet = new Sheet(j, 0);
            sheet.setSheetName("sheet" + j);
  13
```

ExcelUtil.MultipleSheelPropety multipleSheelPropety = new ExcelUtil.MultipleSheelPropety();

ExcelUtil.writeWithMultipleSheel("/home/chenmingjian/Downloads/aaa.xlsx",list1);

凸

分享海报说明

45

0

https://blog.csdn.net/qq_32258777/article/details/89031479#excel_23

14 15

16

17 18 19

20

212223

工具类

multipleSheelPropety.setData(list);
multipleSheelPropety.setSheet(sheet);

list1.add(multipleSheelPropety);

1 package com.springboot.utils.excel;

4 import com.alibaba.excel.ExcelWriter;

8 import com.alibaba.excel.metadata.Sheet;

9 import lombok.Data;

3 import com.alibaba.excel.EasyExcelFactory;

5 | import com.alibaba.excel.context.AnalysisContext;

7 import com.alibaba.excel.metadata.BaseRowModel;

6 import com.alibaba.excel.event.AnalysisEventListener;

```
史上最全的Excel导入导出之easyexcel - 点缀星空 - CSDN博客
 10 import lombok.Getter;
 11 import lombok.Setter;
 12 import lombok.extern.slf4j.Slf4j;
 13 import org.springframework.util.CollectionUtils;
 14 import org.springframework.util.StringUtils;
 15
 16 | import java.io.*;
 17 import java.util.ArrayList;
 18 import java.util.Collections;
 19 import java.util.List;
 20
 21
     /**
      * @description:
 23
      * @author: chenmingjian
     * @date: 19-3-18 16:16
 25
     */
 26 @Slf4j
 27 | public class ExcelUtil {
 28
 29
        private static Sheet initSheet;
 30
 31
        static {
 32
          initSheet = new Sheet(1, 0);
 33
          initSheet.setSheetName("sheet");
 34
          // 设置自适应宽度
 35
          initSheet.setAutoWidth(Boolean.TRUE);
 36
 37
 38
 39
        * 读取少于1000行数据
 40
         * @param filePath 文件绝对路径
 41
         * @return
 42
         */
 43
        public static List<Object> readLessThan1000Row(String filePath){
 44
           return readLessThan1000RowBySheet(filePath,null);
 45
 46
 47
 48
        * 读小于1000行数据, 带样式
 49
         * filePath 文件绝对路径
 50
         * initSheet :
 51
               sheetNo: sheet页码,默认为1
 52
               headLineMun: 从第几行开始读取数据,默认为0, 表示从第一行开始读取
 53
               clazz: 返回数据List<0bject> 中0bject的类名
 54
 55
        public static List<Object> readLessThan1000RowBySheet(String filePath, Sheet sheet){
 56
          if(!StringUtils.hasText(filePath)){
 57
              return null;
 58
 59
          sheet = sheet != null ? sheet : initSheet;
 60
 61
           InputStream fileStream = null;
 63
          try {
 64
              fileStream = new FileInputStream(filePath);
 65
             return EasyExcelFactory.read(fileStream, sheet);
 66
          } catch (FileNotFoundException e) {
 67
              log.info("找不到文件或文件路径错误,文件:{}",filePath);
 68
          }finally {
 69
             try {
 70
                if(fileStream != null){
 71
                   fileStream.close();
 72
 73
             } catch (IOException e) {
 74
                log.info("excel文件读取失败, 失败原因: {}", e);
 75
 76
 77
          return null;
 78
 79
 80
        /**
 81
         * 读大于1000行数据
 82
         * @param filePath 文件觉得路径
 83
         * @return
 84
 85
        public static List<Object> readMoreThan1000Row(String filePath){
 86
          return readMoreThan1000RowBySheet(filePath,null);
 87
 88
 89
       /**
 90
         * 读大于1000行数据, 带样式
 91
         * @param filePath 文件觉得路径
 92
         * @return
 93
         */
 94
        public static List<Object> readMoreThan1000RowBySheet(String filePath, Sheet sheet){
 95
          if(!StringUtils.hasText(filePath)){
 96
              return null;
 97
 98
 99
          sheet = sheet != null ? sheet : initSheet;
100
101
          InputStream fileStream = null;
102
          try {
103
              fileStream = new FileInputStream(filePath);
104
             ExcelListener excelListener = new ExcelListener();
105
             EasyExcelFactory.readBySax(fileStream, sheet, excelListener);
106
             return excelListener.getDatas();
107
          } catch (FileNotFoundException e) {
108
             log.error("找不到文件或文件路径错误,文件:{}",filePath);
109
          }finally {
110
             try {
                if(fileStream != null){
111
                   fileStream.close();
112
113
             } catch (IOException e) {
114
                log.error("excel文件读取失败, 失败原因: {}", e);
115
116
117
118
          return null;
119
120
121
       /**
122
        * 生成excle
123
         * @param filePath 绝对路径, 如:/home/chenmingjian/Downloads/aaa.xlsx
124
         * @param data 数据源
125
         * @param head 表头
126
         */
        public static void writeBySimple(String filePath, List<List<Object>> data, List<String> head){
127
128
          writeSimpleBySheet(filePath,data,head,null);
129
130
131
        /**
132
        * 生成excle
133
         * @param filePath 绝对路径, 如:/home/chenmingjian/Downloads/aaa.xlsx
134
         * @param data 数据源
         * @param sheet excle页面样式
135
136
         * @param head 表头
137
138
        public static void writeSimpleBySheet(String filePath, List<List<Object>> data, List<String> head, Shee
139
          sheet = (sheet != null) ? sheet : initSheet;
140
          if(head != null){
141
142
             List<List<String>> list = new ArrayList<>();
143
             head.forEach(h -> list.add(Collections.singletonList(h)));
144
              sheet.setHead(list);
145
146
147
          OutputStream outputStream = null;
148
          ExcelWriter writer = null;
149
          try {
150
              outputStream = new FileOutputStream(filePath);
```

VIP

凸

分享海报说明

45

```
2019/8/27
```

```
史上最全的Excel导入导出之easyexcel - 点缀星空 - CSDN博客
151
             writer = EasyExcelFactory.getWriter(outputStream);
152
             writer.writel(data, sheet);
153
          } catch (FileNotFoundException e) {
154
             log.error("找不到文件或文件路径错误,文件:{}",filePath);
155
          }finally {
156
             try {
157
                if(writer != null){
158
                  writer.finish();
159
160
                if(outputStream != null){
161
162
                   outputStream.close();
163
164
165
             } catch (IOException e) {
166
                log.error("excel文件导出失败,失败原因: {}", e);
167
168
169
170
171
172
       /**
173
        * 生成excle
174
         * @param filePath 绝对路径, 如:/home/chenmingjian/Downloads/aaa.xlsx
175
         * @param data 数据源
176
177
       public static void writeWithTemplate(String filePath, List<? extends BaseRowModel> data){
178
          writeWithTemplateAndSheet(filePath,data,null);
179
       }
180
       /**
181
182
         * @param filePath 绝对路径, 如:/home/chenmingjian/Downloads/aaa.xlsx
183
184
         * @param data 数据源
185
         * @param sheet excle页面样式
186
187
        public static void writeWithTemplateAndSheet(String filePath, List<? extends BaseRowModel> data, Sheet
188
          if(CollectionUtils.isEmpty(data)){
189
             return;
190
191
192
          sheet = (sheet != null) ? sheet : initSheet;
193
          sheet.setClazz(data.get(0).getClass());
194
195
          OutputStream outputStream = null;
196
          ExcelWriter writer = null;
197
          try {
198
             outputStream = new FileOutputStream(filePath);
199
             writer = EasyExcelFactory.getWriter(outputStream);
             writer.write(data, sheet);
200
201
          } catch (FileNotFoundException e) {
202
             log.error("找不到文件或文件路径错误,文件:{}",filePath);
203
          }finally {
204
             try {
205
                if(writer != null){
206
                  writer.finish();
207
208
209
                if(outputStream != null){
210
                   outputStream.close();
211
212
             } catch (IOException e) {
213
                log.error("excel文件导出失败, 失败原因: {}", e);
214
215
216
217
218
219
       /**
220
        * 生成多Sheet的excle
221
         * @param filePath 绝对路径, 如:/home/chenmingjian/Downloads/aaa.xlsx
222
         * @param multipleSheelPropetys
223
224
        public static void writeWithMultipleSheel(String filePath,List<MultipleSheelPropety> multipleSheelPrope
225
          if(CollectionUtils.isEmpty(multipleSheelPropetys)){
226
             return;
227
228
229
          OutputStream outputStream = null;
230
          ExcelWriter writer = null;
231
          try {
232
             outputStream = new FileOutputStream(filePath);
233
             writer = EasyExcelFactory.getWriter(outputStream);
234
             for (MultipleSheelPropety multipleSheelPropetys) {
235
                Sheet sheet = multipleSheelPropety.getSheet() != null ? multipleSheelPropety.getSheet() : init
236
                if(!CollectionUtils.isEmpty(multipleSheelPropety.getData())){
237
                   sheet.setClazz(multipleSheelPropety.getData().get(0).getClass());
238
239
                writer.write(multipleSheelPropety.getData(), sheet);
240
241
242
          } catch (FileNotFoundException e) {
             log.error("找不到文件或文件路径错误,文件:{}",filePath);
243
244
          }finally {
245
             try {
246
                if(writer != null){
247
                  writer.finish();
248
249
250
                if(outputStream != null){
251
                   outputStream.close();
252
             } catch (IOException e) {
253
254
                log.error("excel文件导出失败, 失败原因: {}", e);
255
256
257
258
259
260
       261
262
263
       @Data
264
       public static class MultipleSheelPropety{
265
266
          private List<? extends BaseRowModel> data;
267
268
          private Sheet sheet;
269
270
271
       /**
272
        * 解析监听器,
273
        * 每解析一行会回调invoke()方法。
274
         * 整个excel解析结束会执行doAfterAllAnalysed()方法
275
276
         * @author: chenmingjian
277
         * @date: 19-4-3 14:11
278
        */
279
       @Getter
280
       @Setter
281
        public static class ExcelListener extends AnalysisEventListener {
282
283
          private List<Object> datas = new ArrayList<>();
284
285
          /**
286
           * 逐行解析
287
           * object : 当前行的数据
288
           */
289
          @Override
290
          public void invoke(Object object, AnalysisContext context) {
291
             // 当前行
```

凸

分享海报说明

45

2019/8/27

```
史上最全的Excel导入导出之easyexcel - 点缀星空 - CSDN博客
292
          // context.getCurrentRowNum()
293
          if (object != null) {
294
             datas.add(object);
                                                                                                   凸
295
296
297
298
299
        /**
                                                                                               分享海报说明
300
         * 解析完所有数据后会调用该方法
                                                                                                   45
         */
301
                                                                                                   302
        @Override
303
        public void doAfterAllAnalysed(AnalysisContext context) {
304
          //解析结束销毁不用的资源
305
                                                                                                   306
307
308
      309
310
311 }
312
```

```
测试类
   1 | package com.springboot.utils.excel;
   3 import com.alibaba.excel.annotation.ExcelProperty;
   4 import com.alibaba.excel.metadata.BaseRowModel;
   5 import com.alibaba.excel.metadata.Sheet;
   6 import lombok.Data;
   7 import lombok.EqualsAndHashCode;
   8 import org.junit.runner.RunWith;
   9 import org.springframework.boot.test.context.SpringBootTest;
  10 | import org.springframework.test.context.junit4.SpringRunner;
  11
  12 | import java.util.ArrayList;
  13 import java.util.Arrays;
  14 import java.util.List;
  15
  16
      /**
       * @description: 测试类
       * @author: chenmingjian
       * @date: 19-4-4 15:24
  19
  20
       */
  21 @SpringBootTest
  22 @RunWith(SpringRunner.class)
  23
      public class Test {
  24
  25
          /**
  26
           * 读取少于1000行的excle
  27
  28
          @org.junit.Test
  29
          public void readLessThan1000Row(){
  30
              String filePath = "/home/chenmingjian/Downloads/测试.xlsx";
  31
              List<Object> objects = ExcelUtil.readLessThan1000Row(filePath);
  32
              objects.forEach(System.out::println);
  33
  34
  35
          /**
  36
           * 读取少于1000行的excle,可以指定sheet和从几行读起
  37
  38
          @org.junit.Test
  39
          public void readLessThan1000RowBySheet(){
  40
              String filePath = "/home/chenmingjian/Downloads/测试.xlsx";
  41
              Sheet sheet = new Sheet(1, 1);
  42
             List<Object> objects = ExcelUtil.readLessThan1000RowBySheet(filePath,sheet);
  43
              objects.forEach(System.out::println);
  44
          }
  45
  46
  47
           * 读取大于1000 行的excle
  48
           * 带sheet参数的方法可参照测试方法readLessThan1000RowBySheet()
  49
           */
  50
          @org.junit.Test
  51
          public void readMoreThan1000Row(){
  52
              String filePath = "/home/chenmingjian/Downloads/测试.xlsx";
  53
             List<Object> objects = ExcelUtil.readMoreThan1000Row(filePath);
  54
              objects.forEach(System.out::println);
  55
  56
  57
  58
  59
           * 生成excle
  60
           * #sheet 参数的方法可参照测试方法readLessThan1000RowBySheet()
  61
           */
  62
          @org.junit.Test
  63
          public void writeBySimple(){
  64
              String filePath = "/home/chenmingjian/Downloads/测试.xlsx";
  65
              List<List<Object>> data = new ArrayList<>();
  66
              data.add(Arrays.asList("111","222","333"));
  67
              data.add(Arrays.asList("111","222","333"));
  68
              data.add(Arrays.asList("111","222","333"));
  69
              List<String> head = Arrays.asList("表头1", "表头2", "表头3");
  70
              ExcelUtil.writeBySimple(filePath,data,head);
  71
  72
  73
  74
          /**
  75
           * 生成excle, 帶用模型
  76
           * 带sheet 参数的方法可参照测试方法readLessThan1000RowBySheet()
  77
           */
  78
          @org.junit.Test
  79
          public void writeWithTemplate(){
  80
              String filePath = "/home/chenmingjian/Downloads/测试.xlsx";
  81
              ArrayList<TableHeaderExcelProperty> data = new ArrayList<>();
              for(int i = 0; i < 4; i++){
  83
                  TableHeaderExcelProperty tableHeaderExcelProperty = new TableHeaderExcelProperty();
  84
                  tableHeaderExcelProperty.setName("cmj" + i);
  85
                  tableHeaderExcelProperty.setAge(22 + i);
  86
                  tableHeaderExcelProperty.setSchool("清华大学" + i);
  87
                  data.add(tableHeaderExcelProperty);
  88
  89
              ExcelUtil.writeWithTemplate(filePath,data);
  90
  91
  92
  93
          /**
  94
           * 生成excle, 带用模型,带多个sheet
  95
  96
          @org.junit.Test
  97
          public void writeWithMultipleSheel(){
  98
              ArrayList<ExcelUtil.MultipleSheelPropety> list1 = new ArrayList<>();
  99
              for(int j = 1; j < 4; j++){
                  ArrayList<TableHeaderExcelProperty> list = new ArrayList<>();
 100
 101
                  for(int i = 0; i < 4; i++){
 102
                      TableHeaderExcelProperty tableHeaderExcelProperty = new TableHeaderExcelProperty();
                      tableHeaderExcelProperty.setName("cmj" + i);
 103
                      tableHeaderExcelProperty.setAge(22 + i);
 104
 105
                      tableHeaderExcelProperty.setSchool("清华大学" + i);
 106
                     list.add(tableHeaderExcelProperty);
 107
 108
 109
                  Sheet sheet = new Sheet(j, 0);
                  sheet.setSheetName("sheet" + j);
 110
 111
 112
                  ExcelUtil.MultipleSheelPropety multipleSheelPropety = new ExcelUtil.MultipleSheelPropety();
 113
                  multipleSheelPropety.setData(list);
 114
                  multipleSheelPropety.setSheet(sheet);
```





```
史上最全的Excel导入导出之easyexcel - 点缀星空 - CSDN博客
116
              list1.add(multipleSheelPropety);
117
118
119
          ExcelUtil.writeWithMultipleSheel("/home/chenmingjian/Downloads/aaa.xlsx",list1);
120
121
122
123
124
125
       126
127
128
       /**
129
        * @description:
130
        * @author: chenmingjian
131
        * @date: 19-4-3 14:44
132
133
       @EqualsAndHashCode(callSuper = true)
134
135
       public static class TableHeaderExcelProperty extends BaseRowModel {
136
137
           /**
138
           * value: 表头名称
139
           * index: 列的号, 0表示第一列
140
141
           @ExcelProperty(value = "姓名", index = 0)
142
           private String name;
143
144
           @ExcelProperty(value = "年龄",index = 1)
145
           private int age;
146
147
           @ExcelProperty(value = "学校",index = 2)
148
           private String school;
149
150
151
       152
153
154
155
```

微软人工智能教育与学习共建社区

我们将在此提供人工智能应用开发的真实案例,与广大师生、开发者一起学习、一起贡献!

```
想对作者说点什么
weixin_44248561: 工具类报错啊
    2 for (MultipleSheelPropety multipleSheelPropety: multipleSheelPropetys)
               Sheet sheet = multipleSheelPropety.getSheet() != null ? multipleSheelPropety.getSheet() : initSheet;
               if(!CollectionUtils.isEmpty(multipleSheelPropety.getData())) {
                 sheet.\ setClazz\ (multipleSheelPropety.\ getData\ ().\ get\ (\textbf{0}).\ getClass\ ())\ ;
               writer.write(multipleSheelPropety.getData(), sheet);
    9
     (2周前 #12楼) 查看回复(12)
🜇 渣渣it男: 这里面读取大于1000行是异步进行的,怎么知道什么时候已经读取完了? (2周前 #11楼) 查看回复(1)
                                                                                                     凸
weixin_41175630: Exception in thread "main" java.lang.NoClassDefFoundError: org/apache/poi/ss/usermodel/Table at
                                                                                                     凸
   java.lang.ClassLoader.defineClass1(Native Method) at java.lang.ClassLoader.defineClass(ClassLoader.java:763) at
   java.security.SecureClassLoader.defineClass(SecureClassLoader.java:142) at
   iava net LIRI ClassI nader defineClass(LIRI ClassI nader iava·468) 据设个锆有没有 人 遇到讨? (1个日前 #10迷) 杏素同复(1)
                                            登录 查看 45 条热评 💙
Java解析excel工具easyexcel 助你快速简单避免OOM
                                                                                              阅读数 4万+
Java解析、生成Excel比较有名的框架有Apachepoi、jxl。但他们都存在一个严重的问题就是非常的耗内存,poi有一… 博文 来自:成长中的巨人
如何优雅地实现 Excel 文件导出功能?(阿里出品的 EasyExcel,安利一波)
                                                                                               阅读数 646
EasyExcel前言导出是后台管理系统的常用功能,当数据量特别大的时候会内存溢出和卡顿页面,曾经自己封装过一....博文来自:Mr_chen的博客
easyexcel 导入excel内存溢出
                                                                                               阅读数 449
之前使用的是阿里开源的easyexcel,可以避免大部分的OOM情况,之前测试的时候也确实是,100Mexcel文件单sh...博文 来自: hello world
easyexcel 读取大excel 十万级 java
                                                                                              阅读数 2290
转发请注明: https://blog.csdn.net/somdip/article/details/85243005<dependency&gt; &lt;groupId&gt;co... 博文 来自: somdip的博客
EasyExcel阿里开源excel导入导出
                                                                                              阅读数 1058
一、为什么使用easyexcelJava解析、生成Excel比较有名的框架有Apachepoi、jxl。但他们都存在一个严重的问题就… 博文 来自: HOME的博客
easyExcel
                                                                                              阅读数 185
https://blog.csdn.net/jiangjiandecsd/article/details/81115622https://github.com/alibaba/easyexcel
                                                                               博文 来自: u011243684的博客
阿里easyExcel进行导出遇到问题
                                                                                                  03-07
我用的是阿里的easyExcel 进行导出,导出的Excel中的数据是以.zip 结尾的一个文件,哪位大神帮忙看一下,这是我的代码...
                                                                                                   论坛
基于EasyExcel的excel解析工具类。
                                                                                                  02-25
基于EasyExcel框架的excel解析工具类。EasyExcel是阿里的excel解析工具,使用快捷方便,轻量。封装了常用的方法。使用前需要了解easyEx...
                                                                                                   下载
谁用过easyexcel,怎么用它实现锁定单元格和下拉表头的功能
                                                                                                  12-05
                                                                                                   问答
史上最全的Excel导入导出之easyexcel - 我的猪夫跑了 - CSDN博客
                                                                                                  6-24
史上最全的Excel导入导出之easyexcel - 很多时候犯错都..._CSDN博客
                                                                                                  7-20
easyexcel-好用的java Excel工具
                                                                                                  11-22
好用的java Excel工具,代替poi easyexcel重写了poi对07版Excel的解析,能够原本一个3M的excel用POI sax依然需要100M左右内存降低到KB..
                                                                                                   下载
史上最全的Excel导入导出之easyexcel - weixin_3921448..._CSDN博客
                                                                                                  6-20
使用EasyExcel对excel进行数据处理(导入导出) - wmx199..._CSDN博客
                                                                                                  7-3
使用阿里开源easyexcel工具遇坑案例
                                                                                              阅读数 1万+
项目地址: https://github.com/alibaba/easyexcel需要实现: 上传excel文件, 第一行需要按照模板固定一行文字: ... 博文 来自: daisy_apk的博客
                                                         laixiaoxing
                                                                                  somdip
                               132篇文章
                                                         225篇文章
                                                                                  72篇文章
                          关注 排名:千里之外
关注 排名:千里之外
                                                   关注 排名:千里之外
                                                                             关注 排名:千里之外
使用EasyExcel导入导出Excel - 很多时候犯错都是在不知..._CSDN博客
                                                                                                 7-30
使用阿里开源easyexcel工具遇坑案例 - daisy_apk的博客 - CSDN博客
                                                                                                  8-20
EasyExcel使用的正确姿势,工具类封装
                                                                                              阅读数 3448
项目中很可能用到导出excel文件的需求。easyexcel代码量较小,使用简单,而且性能较佳,是一个非常好的选择。.... 博文 来自: 明明如月的专栏
使用easy-Excel实现Excel的读取与导出
                                                                                              阅读数 1256
一.Excel数据的读取1.java代码packagecom.immo.framework.utils;importjava.io.FileInputStream;importjava.io.I... 博文 来自: qq_34542964的博客
```

java导出excel表格 使用alibaba easyexcel - 如果有一..._CSDN博客



7-6

凸

分享海报说明

45

史上最全的Excel导入导出之easyexcel - 点缀星空 - CSDN博客 百万级数据EXCEL导出easyexcel - Java Mrsun的博客 - CSDN博客			8-12
阿里开源(EasyExcel			
这里写自定义目录标题欢迎使用Markdown编辑器新的改变功能快捷键合理的创建标题,有助于目录的生成如何改变	博文	来自:	阅读数 19 weixin_44934295
easyExcel导出excel - 傲宇苍穹 - CSDN博客			6-4
使用easyexcel导出excel文件 这里是导出一个excel文件,里面有两个sheet分别是税赋预测表和发票汇总表controller层@requestMapping(valu	博文	来自:	阅读数 5588 hello world
使用easyexcel读写Excel title: "使用easyexcel读写Excel" tags:Java学习笔记JAVA解析Excel工具easyexcelJava解析、生成Excel比较有名的…	博文	来自:	阅读数 5104 wsk1103
easyexcel一个很棒的Excel解析工具 easyexcel一个很棒的Excel解析工具前言Excel格式核心原理对比理解与使用总结前言最近手头上有一个项目,其中有…	博文	来自:	阅读数 1360 小林哥的博客
使用easyexcel导出多表头excel文件 多表头的modelpackagecom.souche.financial.report.model.eo;importcom.alibaba.excel.annotation.ExcelProp	博文	来自:	阅读数 1323 hello world
ALiBaBa的easyexcel (一) EasyExcel是阿里巴巴对POI封装的一个库,号称解决了POI的OOM问题,并且在使用上也更方便一些然而我在使用…	博文	来自:	阅读数 223 weixin_39214481
EasyExcel 目录一、前言 二、Apachepoi、jxl的缺陷 三、阿里出品的EasyExcel,安利一波 四、EasyExcel解决了什么 五、快速…	博文	来自:	阅读数 336 深圳不归客
百万级数据EXCEL导出easyexcel easyexcel项目git地址为: https://github.com/alibaba/easyexcel官方使用指南见: https://github.com/alibaba/ea	博文	来自:	阅读数 1918 Java_Mrsun的博客
阿里开源(EasyExcel)导出EXCEL 一.简介导出是后台管理系统的常用功能,当数据量特别大的时候会内存溢出和卡顿页面,曾经自己封装过一个导出,…	博文	来自:	阅读数 8250 请叫我猿叔叔的博客
JAVA使用easyexcel操作Excel 之前写过一篇《JAVA操作Excel》,介绍了jxl和poi读写Excel的实现,今天为大家介绍一下使用easyexcel对Excel进…	博文	来自:	阅读数 4万+ 蒋固金(jiangguji
可以解析大文件的easyexcel版本 1.1.2版本在堆内存400mb的时候解析10mb的excel 堆内存溢出。 但是1.0.1版本支持,因此将1.0.1里面的部分代码和1.1.2版本进行了整合 打	J		04-19 下载
easyexcel下载工具 优化excel下载功能,采用新的工具类降低内存:https://github.com/alibaba/easyexcelpom: easyex</td <td>博文</td> <td>来自:</td> <td>阅读数 358 技术成就人生</td>	博文	来自:	阅读数 358 技术成就人生
EasyExcel读xIsx文件的理解 1.首先要获取一个文件,这个文件获取之后最好是能用BufferInputStreamInputStreaminputStream=newFileInput	博文	来自:	阅读数 1024 Pruett的博客
2018年阿里巴巴关于java重要开源项目汇总 来源地址:segmentfault.com/a/1190000017346799作者:阿里云云栖社区(点击阅读原文前往)1.分布式应用服务开	博文	来自:	阅读数 453 冷夜轩的博客
使用easypoi或者easyexcel多表头导出 1、使用easypoi实现多表头导出。其实使用easypoi实现单表头的导出很简单,百度一下到处都是。但是对于多表头…	博文	来自:	阅读数 5553 qq_37025445的博客
easyexcel 简单使用 <dependency><groupid>com.alibaba</groupid><arti< td=""><td>博文</td><td>来自:</td><td>阅读数 1599 SakeOfWhat的博客</td></arti<></dependency>	博文	来自:	阅读数 1599 SakeOfWhat的博客
阿里出品的 EasyExcel —— 7 行代码实现 Excel 文件导出 点击进入阿里巴巴EasyExcel项目GitHub仓库地址阿里出品的EasyExcel,安利一波什么是EasyExcel?见名知意,就是…	博文	来自:	阅读数 159 Hern (宋兆恒)
Java使用easyExcel操作Excel案例 这两天一直在玩些小工具,今天整了下阿里巴巴的easyExcel,下面是案例:importcom.alibaba.excel.ExcelReader;imp	博文	来自:	阅读数 6641 IT人利哥的博客
excel导入导出 【springboot+easypoi】一行代码搞定excel导入导出开发中经常会遇到excel的处理,导入导出解析等等,java中比…	博文	来自:	阅读数 361 c0987k的博客
Excel解析easyexcel工具类 本文使用基于阿里的easyexcel编写的工具类对xls后缀的Excel文件(即03版)读取并写成xlsx后缀的Excel文件(即07版),	博文	来自:	阅读数 1万+ 鲸临于空
easyexcel使用 项目主页地址:https://github.com/alibaba/easyexcel参考地址1:https://blog.csdn.net/jianggujin/article/det	博文	来自:	阅读数 71 三个和尚_的博客
SQL 检查约束整理 运算符:对于所有的运算符,可分为单目运算符和双目运算符。目:操作数一、数学运算符:+-*/%+5-84+58-89	博文	来自:	阅读数 5798 千里之行始于足下
EasyExcel写excel的理解 1.首先要创建一个OutputStream流,保证能够读到文件,OutputStreamout=newFileOutputStream("filep	博文	来自:	阅读数 493 Pruett的博客
easypoi版本自定义字典的使用方法 publicclassGlobalExcelDictHandlerimplementslExcelDictHandler{ privatestaticfinalLoggerLOG=LoggerFac	博文	来自:	阅读数 446 会飞的时钟
Java-使用easyexcel读大文件 使用easyexcel读大excel文件,避免OOM错误。依赖:compilegroup:'com.alibaba',name:'easyexcel',version:'1	博文	来自:	阅读数 2657 wang704987562
easyexcel导出遇到的两个坑 问题一:poi的jar版本过低问题使用easyexcel大数据量导出时,需要依赖org.apache.poi的3.17版的jar包,而poi3	博文	来自:	阅读数 1308 冰魄神光
java百万级别excel导出(easyExcel) 1.为什么需要excel到处?导出功能在各个领域都被广泛的运用,当用户想把数据下载下来的时候,此时excel是一个…	博文	来自:	阅读数 3320 qq_33220089的博客
EasyExcel导入工具(SpringMVC下使用) easyExcel:由阿里巴巴公司开发,由github托管github上有详细使用文档github地址:https://github.com/alibab	博文	来自:	阅读数 491 weixin_34128839
使用EasyExcel对excel进行数据处理(导入导出) 度过了一段忙碌的时间,新需求开发,老系统的技改,现在终于轻松了一点;话不多说,直接进入主题。之前进行ex	博文	来自:	阅读数 517 wmx1992的博客
JAVA阿里Easyexcel使用心得以及发现的一些问题 easyexcel使用心得easyexcel的好处官方描述个人理解excel读取示例easyexcel的好处官方描述Java解析、生成Excel	博文	来自:	阅读数 1930 jerry_guangguan
阿里十大开源项目 本文转载自公众号 Java架构师之路 1JSON处理器fastjsonfastjson是一个性能很好的Java语言实现的JSON解析器和	博文	来自:	阅读数 1万+ 程序员小灰的博客
一行代码完成 JAVA 的 EXCEL 读写——EasyExcel 的方法封装 前段时间在github上发现了阿里的EasyExcel项目,觉得挺不错的,就写了一个简单的方法封装,做到只用一个函数	博文	来自:	阅读数 1868 Howie_Y的博客
Java解析excel工具easyexcel https://blog.csdn.net/jiangjiandecsd/article/details/81115622Java解析、生成Excel比较有名的框架有Apachep	博文	来自:	阅读数 94 NRGAGA的专栏
Java Excel导入导出,基于XML和Easy-excel使用 1.前言•在工作时,遇到过这样的需求,需要灵活的对工单进行导入或导出,以前自己也做过,但使用不灵活繁琐。我	博文	来自:	阅读数 889 u010586151的博客
[EasyPoi] 初识EasyPoi easypoi功能如同名字easy,主打的功能就是容易,让一个没见接触过poi的人员就可以方便的写出Excel导出,Excel模板	博文	来自:	阅读数 1827 旅行才是人生追寻
使用easyexcel操作Excel 我也是凑巧用上了,就在这里分享一下,性能确实是没得说!对easyexcel的介绍是:Java解析、生成Excel比较有名…	博文	来自:	阅读数 4819 babyyaoyao的博客
EasyExcel为单个Cell设置样式 EasyExcel是阿里巴巴对POI封装的一个库,号称解决了POI的OOM问题,并且在使用上也更方便一些Github:然而…	博文	来自:	阅读数 3556 qq_28612673的博客





