



链滴

springboot 通过 poi 导出 excel

作者: [carson33459](#)

原文链接: <https://ld246.com/article/1567876700635>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



Maven引入依赖

```
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi</artifactId>
  <version>RELEASE</version>
</dependency>
<dependency>
  <groupId>org.apache.poi</groupId>
  <artifactId>poi-ooxml</artifactId>
  <version>RELEASE</version>
</dependency>
```

导出excel

导出excel比较重要的api有以下几个，其他的并无难点

- 创建一个excel文件工作簿； (HSSFWorkbook workbook = new HSSFWorkbook())
- 创建一张表； HSSFSheet sheet = workbook.createSheet("统计表")
- 创建一行； HSSFRow row = sheet.createRow(0)
- 填充一列数据； row.createCell(0).setCellValue("数据")
- 设置一个单元格样式； cell.setCellStyle(style)

注意!!! 如果文件没有提示下载 则不代表生成EXCEL失败 文件会在项目 根目录 中自动生成!!!

```
public HSSFWorkbook createExcel(SearchParams params) {
  // 第一步创建工作簿
  HSSFWorkbook wb = new HSSFWorkbook();
```

```

// 第二步创建sheet
HSSFSheet sheet = wb.createSheet("列表");
// 设置列宽, setColumnWidth的第二个参数要乘以256, 这个参数的单位是1/256个字符宽度
sheet.setColumnWidth(0, 256 * 20);
sheet.setColumnWidth(1, 256 * 20);
sheet.setColumnWidth(2, 256 * 20);
sheet.setColumnWidth(3, 256 * 10);
sheet.setColumnWidth(4, 256 * 20);
sheet.setColumnWidth(5, 256 * 20);

// 第三步创建行row:添加表头0行
HSSFRow row = sheet.createRow(0);
HSSFCellStyle style = wb.createCellStyle();

// 设置为居中加粗
// HSSFCellStyle style = workbook.createCellStyle();
// HSSFFont font = workbook.createFont();
// font.setBold(true);
// style.setAlignment(HSSFCellStyle.ALIGN_CENTER);
// style.setFont(font);

// 第四步创建单元格
int cellColumn = 0;
// 第一个单元格
HSSFCell cell = row.createCell(cellColumn++);
cell.setCellValue("ID");
cell.setCellStyle(style);

// 第二个单元格
cell = row.createCell(cellColumn++);
cell.setCellValue("用户");
cell.setCellStyle(style);

// 第三个单元格
cell = row.createCell(cellColumn++);
cell.setCellValue("时间");
cell.setCellStyle(style);

// 第四个单元格
cell = row.createCell(cellColumn++);
cell.setCellValue("符号");
cell.setCellStyle(style);

// 第五个单元格
cell = row.createCell(cellColumn++);
cell.setCellValue("数量");
cell.setCellStyle(style);

AtomicInteger index = new AtomicInteger();
// 自增跳过表头
index.incrementAndGet();
fillData(params, index, sheet);

```

```

    return wb;
}

private void fillData(SearchParams params, AtomicInteger index, HSSFSheet sheet) {
    // 自定义分页查找数据
    PageInfo<SearchListForm> pageInfo = this.service.getDataBySearch(params);
    List<SearchListForm> searchList = pageInfo.getList();
    Optional.ofNullable(searchList).ifPresent(list -> list.stream().forEach(v -> {
        HSSFRow row = sheet.createRow(index.getAndIncrement());

        // 设置日期格式
        // HSSFCellStyle style = workbook.createCellStyle();
        // style.setDataFormat(HSSFDataFormat.getBuiltinFormat("m/d/yy h:mm"));

        // HSSFCell cell = row.createCell(3);
        // cell.setCellValue(user.getCreate_time());
        // cell.setCellStyle(style);

        int cellColumn = 0;
        row.createCell(cellColumn++).setCellValue(v.getId());
        row.createCell(cellColumn++).setCellValue(v.getUsername());
        row.createCell(cellColumn++).setCellValue(DateFormatUtils.format(v.getCreatetime(),
MM/dd/yyyy HH:mm:ss"));
        row.createCell(cellColumn++).setCellValue(v.getSymbol());
        row.createCell(cellColumn++).setCellValue(v.getQuantity().setScale(8, RoundingMode
HALF_DOWN).stripTrailingZeros().toString());
    }));

    // 循环获取数据
    if (pageInfo.hasNextPage()) {
        params.setPage(pageInfo.getNextPage());
        fillData(params, index, sheet);
    }
}

@GetMapping(value = "downloadExcel")
public void (
    @RequestBody @ApiParam(required = false, name = "SearchParams", value = "参数")
    SearchParams params,
    HttpServletResponse response
) {
    params.setPage(1);
    params.setRows(1000);
    String fileName = LocalDate.now().format(DateTimeFormatter.ofPattern("yyyyMMdd")) +
" _ExcelData" + ".xls";
    response.setContentType("application/octet-stream");
    response.setHeader("Pragma", "no-cache");
    response.setHeader("Cache-Control", "no-cache");
    response.setDateHeader("Expires", 0);
    response.setHeader("Content-disposition", "attachment;filename=" + fileName);
    OutputStream outputStream = null;
    HSSFWorkbook wb = null;
}

```

```

try {
    outputStream = response.getOutputStream();
    // 刷新缓冲
    response.flushBuffer();
    wb = createExcel(params);
    wb.write(outputStream);
} catch (IOException e) {
    logger.info("IOException {}", e);
} finally {
    IOUtils.closeQuietly(wb);
    IOUtils.closeQuietly(outputStream);
}
}

// 生成excel文件
protected void buildExcelFile(String filename,HSSFWorkbook workbook) throws Exception{
    FileOutputStream fos = new FileOutputStream(filename);
    workbook.write(fos);
    fos.flush();
    fos.close();
}

// 浏览器下载excel
protected void buildExcelDocument(String filename,HSSFWorkbook workbook,HttpServlet
response response) throws Exception{
    response.setContentType("application/vnd.ms-excel");
    response.setHeader("Content-Disposition", "attachment;filename="+URLEncoder.encode
filename, "utf-8"));
    OutputStream outputStream = response.getOutputStream();
    workbook.write(outputStream);
    outputStream.flush();
    outputStream.close();
}

```

通过swagger直接下载可能会出现乱码