Pom.xml

<**dependency**>  
 <**groupId**>org.apache.poi</**groupId**>  
 <**artifactId**>poi</**artifactId**>  
 <**version**>3.16</**version**>  
</**dependency**>  
<**dependency**>  
 <**groupId**>org.apache.poi</**groupId**>  
 <**artifactId**>poi-ooxml</**artifactId**>  
 <**version**>3.16</**version**>  
</**dependency**>

一个Exceldata.java

**package** com.cs.exchange.util;  
  
**import** java.io.Serializable;  
**import** java.util.List;  
  
**public class** ExcelData **implements** Serializable {  
  
 **private static final long *serialVersionUID*** = 4444017239100620999L;  
  
 *// 表头* **private** List<String> **titles**;  
  
 *// 数据* **private** List<List<Object>> **rows**;  
  
 *// 页签名称* **private** String **name**;  
  
 **public** List<String> getTitles() {  
 **return titles**;  
 }  
  
 **public void** setTitles(List<String> titles) {  
 **this**.**titles** = titles;  
 }  
  
 **public** List<List<Object>> getRows() {  
 **return rows**;  
 }  
  
 **public void** setRows(List<List<Object>> rows) {  
 **this**.**rows** = rows;  
 }  
  
 **public** String getName() {  
 **return name**;  
 }  
  
 **public void** setName(String name) {  
 **this**.**name** = name;  
 }  
  
}

ExportExcelUtils.java

**package** com.cs.exchange.util;  
  
**import** org.apache.poi.ss.usermodel  
 .BorderStyle;  
**import** org.apache.poi.ss.usermodel.Cell;  
**import** org.apache.poi.ss.usermodel.Font;  
**import** org.apache.poi.ss.usermodel.IndexedColors;  
**import** org.apache.poi.ss.usermodel.Row;  
**import** org.apache.poi.ss.usermodel.Sheet;  
**import** org.apache.poi.xssf.usermodel.XSSFCellStyle;  
**import** org.apache.poi.xssf.usermodel.XSSFColor;  
**import** org.apache.poi.xssf.usermodel.XSSFSheet;  
**import** org.apache.poi.xssf.usermodel.XSSFWorkbook;  
**import** org.apache.poi.xssf.usermodel.extensions.XSSFCellBorder;  
  
**import** javax.servlet.http.HttpServletResponse;  
**import** java.awt.\*;  
**import** java.io.OutputStream;  
**import** java.net.URLEncoder;  
**import** java.util.List;  
  
**public class** ExportExcelUtils {  
 **public static void** exportExcel(HttpServletResponse response, String fileName, ExcelData data) **throws** Exception {  
 *// 告诉浏览器用什么软件可以打开此文件* response.setHeader(**"content-Type"**, **"application/vnd.ms-excel"**);  
 *// 下载文件的默认名称* response.setHeader(**"Content-Disposition"**, **"attachment;filename="**+ URLEncoder.*encode*(fileName, **"utf-8"**));  
 *exportExcel*(data, response.getOutputStream());  
 }  
  
 **public static void** exportExcel(ExcelData data, OutputStream out) **throws** Exception {  
  
 XSSFWorkbook wb = **new** XSSFWorkbook();  
 **try** {  
 String sheetName = data.getName();  
 **if** (**null** == sheetName) {  
 sheetName = **"Sheet1"**;  
 }  
 XSSFSheet sheet = wb.createSheet(sheetName);  
 *writeExcel*(wb, sheet, data);  
  
 wb.write(out);  
 } **finally** {  
 wb.close();  
 }  
 }  
  
 **private static void** writeExcel(XSSFWorkbook wb, Sheet sheet, ExcelData data) {  
  
 **int** rowIndex = 0;  
  
 rowIndex = *writeTitlesToExcel*(wb, sheet, data.getTitles());  
 *writeRowsToExcel*(wb, sheet, data.getRows(), rowIndex);  
 *autoSizeColumns*(sheet, data.getTitles().size() + 1);  
  
 }  
  
 **private static int** writeTitlesToExcel(XSSFWorkbook wb, Sheet sheet, List<String> titles) {  
 **int** rowIndex = 0;  
 **int** colIndex = 0;  
  
 Font titleFont = wb.createFont();  
 titleFont.setFontName(**"simsun"**);  
 titleFont.setBold(**true**);  
 *// titleFont.setFontHeightInPoints((short) 14);* titleFont.setColor(IndexedColors.***BLACK***.**index**);  
  
 XSSFCellStyle titleStyle = wb.createCellStyle();  
 titleStyle.setAlignment(XSSFCellStyle.***ALIGN\_CENTER***);  
 titleStyle.setVerticalAlignment(XSSFCellStyle.***VERTICAL\_CENTER***);  
 titleStyle.setFillForegroundColor(**new** XSSFColor(**new** Color(182, 184, 192)));  
 titleStyle.setFillPattern(XSSFCellStyle.***SOLID\_FOREGROUND***);  
 titleStyle.setFont(titleFont);  
 *setBorder*(titleStyle, BorderStyle.***THIN***, **new** XSSFColor(**new** Color(0, 0, 0)));  
  
 Row titleRow = sheet.createRow(rowIndex);  
 *// titleRow.setHeightInPoints(25);* colIndex = 0;  
  
 **for** (String field : titles) {  
 Cell cell = titleRow.createCell(colIndex);  
 cell.setCellValue(field);  
 cell.setCellStyle(titleStyle);  
 colIndex++;  
 }  
  
 rowIndex++;  
 **return** rowIndex;  
 }  
  
 **private static int** writeRowsToExcel(XSSFWorkbook wb, Sheet sheet, List<List<Object>> rows, **int** rowIndex) {  
 **int** colIndex = 0;  
  
 Font dataFont = wb.createFont();  
 dataFont.setFontName(**"simsun"**);  
 *// dataFont.setFontHeightInPoints((short) 14);* dataFont.setColor(IndexedColors.***BLACK***.**index**);  
  
 XSSFCellStyle dataStyle = wb.createCellStyle();  
 dataStyle.setAlignment(XSSFCellStyle.***ALIGN\_CENTER***);  
 dataStyle.setVerticalAlignment(XSSFCellStyle.***VERTICAL\_CENTER***);  
 dataStyle.setFont(dataFont);  
 *setBorder*(dataStyle, BorderStyle.***THIN***, **new** XSSFColor(**new** Color(0, 0, 0)));  
  
 **for** (List<Object> rowData : rows) {  
 Row dataRow = sheet.createRow(rowIndex);  
 *// dataRow.setHeightInPoints(25);* colIndex = 0;  
  
 **for** (Object cellData : rowData) {  
 Cell cell = dataRow.createCell(colIndex);  
 **if** (cellData != **null**) {  
 cell.setCellValue(cellData.toString());  
 } **else** {  
 cell.setCellValue(**""**);  
 }  
  
 cell.setCellStyle(dataStyle);  
 colIndex++;  
 }  
 rowIndex++;  
 }  
 **return** rowIndex;  
 }  
  
 **private static void** autoSizeColumns(Sheet sheet, **int** columnNumber) {  
  
 **for** (**int** i = 0; i < columnNumber; i++) {  
 **int** orgWidth = sheet.getColumnWidth(i);  
 sheet.autoSizeColumn(i, **true**);  
 **int** newWidth = (**int**) (sheet.getColumnWidth(i) + 100);  
 **if** (newWidth > orgWidth) {  
 sheet.setColumnWidth(i, newWidth);  
 } **else** {  
 sheet.setColumnWidth(i, orgWidth);  
 }  
 }  
 }  
  
 **private static void** setBorder(XSSFCellStyle style, BorderStyle border, XSSFColor color) {  
 style.setBorderTop(border);  
 style.setBorderLeft(border);  
 style.setBorderRight(border);  
 style.setBorderBottom(border);  
 style.setBorderColor(XSSFCellBorder.BorderSide.***TOP***, color);  
 style.setBorderColor(XSSFCellBorder.BorderSide.***LEFT***, color);  
 style.setBorderColor(XSSFCellBorder.BorderSide.***RIGHT***, color);  
 style.setBorderColor(XSSFCellBorder.BorderSide.***BOTTOM***, color);  
 }  
}

controller:

@RequestMapping(value = **"/toexport"**)//查询列表并保存session  
@ResponseBody  
**public** String toExport(HttpServletRequest request,@RequestBody ViewTransationIO params) {  
 **if** (params == **null**) {  
 params=**new** ViewTransationIO();  
 }  
 List<ViewTransationInfoPull> hyperchainTransationInfos = **transationService**.queryAllList(params);  
  
 request.getSession().setAttribute(**"list"**,hyperchainTransationInfos);  
  
 **return "1"**;  
}

*/\*\*  
 \* 导出兑换明细列表 获取session中的列表并导出  
 \** ***@param*** *\** ***@return*** *\** ***@throws*** *Exception  
 \*/*@RequestMapping(value = **"/export"**)  
**public** HttpServletResponse upload(HttpServletRequest request, HttpServletResponse response) **throws** Exception{  
 List<ViewTransationInfoPull> list = (List<ViewTransationInfoPull>) request.getSession().getAttribute(**"list"**);  
 List<ViewTransationInfoPullVo> hyperchainTransationInfoList=**new** ArrayList<ViewTransationInfoPullVo>();  
 **if**(list!=**null**){  
  
 **for** (ViewTransationInfoPull i:list){//转换格式的可以忽略  
 hyperchainTransationInfoList.add(**new** ViewTransationInfoPullVo(i));  
 }  
 }  
 ExcelData data = **new** ExcelData();  
 data.setName(**"兑换明细"**);  
 List<String> titles = **new** ArrayList();  
 titles.add(**"交易时间"**);  
 titles.add(**"交易号"**);  
 titles.add(**"兑换来源商户"**);  
 **if** (**cz**==0){titles.add(**"兑换来源会员号"**);}  
 titles.add(**"兑换目标商户"**);  
 **if** (**cz**==0){ titles.add(**"兑换目标会员号"**);}  
 titles.add(**"类型"**);  
 titles.add(**"兑出"**);  
 titles.add(**"兑入"**);  
 titles.add(**"提取时间"**);  
 titles.add(**"交易标记"**);  
 titles.add(**"交易状态"**);  
 data.setTitles(titles);  
  
 List<List<Object>> rows = **new** ArrayList();  
 List<Object> row = **new** ArrayList();  
 **if**(**null**!=hyperchainTransationInfoList&&hyperchainTransationInfoList.size()>0){  
 **for** (**int** i = 0; i < hyperchainTransationInfoList.size(); i++) {  
 row.add(hyperchainTransationInfoList.get(i).getTransDate());  
 row.add(hyperchainTransationInfoList.get(i).getTransId());  
 row.add(hyperchainTransationInfoList.get(i).getSourceOrgName());  
 **if** (**cz**==0){row.add(hyperchainTransationInfoList.get(i).getSourceMemberId()); }  
 row.add(hyperchainTransationInfoList.get(i).getAcceptOrgName());  
 **if** (**cz**==0){ row.add(hyperchainTransationInfoList.get(i).getAcceptMemberId());}  
 row.add(hyperchainTransationInfoList.get(i).getTransType());  
 row.add(hyperchainTransationInfoList.get(i).getTransOut());  
 row.add(hyperchainTransationInfoList.get(i).getTransIn());  
 row.add(hyperchainTransationInfoList.get(i).getCreateTime());  
 row.add(hyperchainTransationInfoList.get(i).getTransFlag());  
 row.add(hyperchainTransationInfoList.get(i).getTransStatus());  
 rows.add(row);  
 row = **new** ArrayList<>();  
 }  
 }  
  
 data.setRows(rows);  
  
  
 *//生成本地  
 /\*File f = new File("c:/test.xlsx");  
 FileOutputStream out = new FileOutputStream(f);  
 ExportExcelUtils.exportExcel(data, out);  
 out.close();\*/* ExportExcelUtils.*exportExcel*(response,**"兑换明细.xlsx"**,data);  
 request.getSession().removeAttribute(**"list"**);  
  
 **return** response;  
}

Js:

*//点击导出excel***$**(**"#export"**).click(**function** () {  
  
 ***api***.getPurchaseOrderExport( v).done(**function** (data) {  
 ***console***.log(data);  
 **if** (data=**"1"**)  
 ***window***.location.href=**"/purchaseorder/v1/export"**;  
 }).fail(**function** (res) {  
  
 });  
});