## 前言

根据客户的需要，需要将应用的财务数据导出到Excel表格中导出，使用java库jxl生成Excel表格

## Jxl库使用

1. 导入依赖包

在gradle中添加

implementation 'net.sourceforge.jexcelapi:jxl:2.6.12'

也可以导入jar包

implementation files('libs/jxl-2.6.12.jar')

1. 设置单元格格式

设置单元格字体和字体大小、对齐方式、边框大小、背景颜色

// 设置字体和字体大小

*arial10font* = new WritableFont(WritableFont.*ARIAL*, 10, WritableFont.*BOLD*);  
*arial10format* = new WritableCellFormat(*arial10font*);  
// 对齐格式  
*arial10format*.setAlignment(jxl.format.Alignment.*CENTRE*);  
// 设置边框  
*arial10format*.setBorder(jxl.format.Border.*ALL*, jxl.format.BorderLineStyle.*THIN*);  
// 设置背景色  
*arial10format*.setBackground(Colour.*GRAY\_25*);

1. 初始化表格

初始化设置表格存储路径和文件名称，创建Excel表格并且设置sheet、设置标题栏

WritableWorkbook workbook = null;

try {  
 File file = new File(context.getExternalCacheDir() + "/Export");  
 *makeDir*(file);  
 File saveFile = new File(file, filePath);  
 if (!saveFile.exists()) {  
 saveFile.createNewFile();  
 }  
  
 workbook = Workbook.*createWorkbook*(saveFile);  
 // 设置表格的sheet名字  
 WritableSheet sheet = workbook.createSheet(sheetName, 0);  
 // 创建标题栏  
 sheet.addCell(new Label(0, 0, filePath, *arial14format*));  
 for (int col = 0; col < colName.length; col++) {  
 sheet.addCell(new Label(col, 0, colName[col], *arial10format*));  
 }  
 // 设置行高  
 sheet.setRowView(0, 340);  
 // 表格写入  
 workbook.write();  
} catch (Exception e) {  
 e.printStackTrace();  
} finally {  
 if (workbook != null) {  
 try {  
 // 表格关闭  
 workbook.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

表格sheet名称设置好之后，需要调用write（）进行写入，写入完成后调用close（）进行关闭释放资源

1. 数据写入到表格

WritableWorkbook writebook = null;

InputStream in = null;  
try {  
 WorkbookSettings setEncode = new WorkbookSettings();  
 setEncode.setEncoding(*UTF8\_ENCODING*);  
 File file = new File( context.getExternalCacheDir() + "/Export");  
 *makeDir*(file);  
 File saveFile = new File(file, fileName);  
 if (!saveFile.exists()) {  
 saveFile.createNewFile();  
 }  
  
 in = new FileInputStream(saveFile);  
 Workbook workbook = Workbook.*getWorkbook*(in);  
 writebook = Workbook.*createWorkbook*(saveFile, workbook);  
 WritableSheet sheet = writebook.getSheet(0);  
  
 for (int j = 0; j < dataList.size(); j++) {  
 List<String> list = dataList.get(j);  
 for (int i = 0; i < list.size(); i++) {  
 sheet.addCell(new Label(i, j + 1, list.get(i), *arial12format*));  
 if (list.get(i).length() <= 5) {  
 //设置列宽  
 sheet.setColumnView(i, list.get(i).length() + 8);  
 } else {  
 //设置列宽  
 sheet.setColumnView(i, list.get(i).length() + 5);  
 }  
 }  
 //设置行高  
 sheet.setRowView(j + 1, 350);  
 }  
  
 writebook.write();  
} catch (Exception e) {  
 e.printStackTrace();  
} finally {  
 if (writebook != null) {  
 try {  
 writebook.close();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 }  
 if (in != null) {  
 try {  
 in.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
}

1. 调用导出Excel表格

private void exportExcel() {

// 表格标题  
 String[] title = {"场地费支出", "水费支出", "电费支出", "员工报销支出", "客户收入", "公司授权收入", "月份"};  
 // 表格文件名  
 String fileName = System.*currentTimeMillis*() + "\_" + "财务收支汇总表.xls";  
 // 初始化表格  
 ExcelUtil.*initExcel*(getContext(), fileName, "财务记录", title);  
  
 List<List<String>> dataList = new ArrayList<>();  
 // 第一行数据  
 List<String> rowData = Arrays.*asList*("54000", "560", "1004", "20000", "1040000", "23000", "1");  
 // 第二行数据  
 List<String> rowData1 = Arrays.*asList*("54000", "580", "1084", "20200", "1240000", "23000", "2");  
 // 第三行数据  
 List<String> rowData2 = Arrays.*asList*("54000", "520", "1504", "20078", "1340000", "23000", "3");  
 dataList.add(rowData);  
 dataList.add(rowData1);  
 dataList.add(rowData2);  
  
 // 数据写入表格  
 ExcelUtil.*writeObjListToExcel*(dataList, fileName, getContext());  
}

1. 效果图

