



Java 核心技术(进阶)

第四章 高级文件处理

第七节 表格文件简介及解析

华东师范大学 陈良育



表格文件

- xls/xlsx 文件 (Microsoft Excel)
- CSV 文件 (Comma-Separated Values 文件)

	A	B	C
1	姓名	学科	成绩
2	张三	语文	80
3	李四	数学	90
4	王五	英语	60
5	王五	数学	100
6			
7			
8			
9			

Sheet1



xlsx(Excel)

- 与word类似，也分成xls和xlsx。
- xlsx以XML为标准，为主要研究对象
- 数据
 - sheet
 - 行
 - 列
 - 单元格

	A	B	C	
1	姓名	学科	成绩	
2	张三	语文	80	
3	李四	数学	90	
4	王五	英语	60	
5	王五	数学	100	
6				
7				
8				
9				

Sheet1



xlsx(Excel)功能和第三方包

- 常见功能
 - 解析
 - 生成
- 第三方的包
 - POI, JXL (免费)
 - COM4J (Windows平台)
 - Aspose等(收费)

POI



- Apache POI

- Apache 出品，必属精品， poi.apache.org
- 可处理docx, xlsx, pptx, visio等office套件
- 纯Java工具包，无需第三方依赖
- 主要类

- XSSFWorkbook 整个文档对象
- XSSFSheet 单个sheet对象
- XSSFRow 一行对象
- XSSFCell 一个单元格对象

	A	B	C	
1	姓名	学科	成绩	
2	张三	语文	80	
3	李四	数学	90	
4	王五	英语	60	
5	王五	数学	100	
6				
7				
8				
9				

Sheet1



CSV文件

- 全称：Comma-Seperated Values文件(逗号分隔)
- 广义CSV文件，可以由空格/Tab键/分号/.../完成字段分隔
- 第三方包：Apache Commons CSV
 - CSVFormat 文档格式
 - CSVParser 解析文档
 - CSVRecord 一行记录
 - CSVPrinter 写入文档

```
姓名,学科,成绩  
张三,语文,80  
李四,数学,90  
王五,英语,60  
王五,数学,100
```

	A	B	C
1	姓名	学科	成绩
2	张三	语文	80
3	李四	数学	90
4	王五	英语	60
5	王五	数学	100
6			

总结



- 针对不同的表格文件格式，选择合适的第三方包
- 大并发情况下，注意读写的速度
- API很多，需要多查询、多练习



代码(1) ReadWriteExcelFile.java

```
package xlsx;

import java.io.FileInputStream;

public class ReadWriteExcelFile
{
    public static void readXLSFile() throws IOException
    {
        InputStream ExcelFileToRead = new FileInputStream("C:/Test.xls");
        HSSFWorkbook wb = new HSSFWorkbook(ExcelFileToRead);

        HSSFSheet sheet = wb.getSheetAt(0);
        HSSFRow row;
        HSSFCell cell;

        Iterator rows = sheet.rowIterator();
```




代码(2) ReadWriteExcelFile.java

```
while (rows.hasNext())
{
    row = (HSSFRow) rows.next();
    Iterator cells = row.cellIterator();

    while (cells.hasNext())
    {
        cell = (HSSFCell) cells.next();

        if (cell.getCellType() == HSSFCell.CELL_TYPE_STRING)
        {
            System.out.print(cell.getStringCellValue() + " ");
        }
        else if (cell.getCellType() == HSSFCell.CELL_TYPE_NUMERIC)
        {
            System.out.print(cell.getNumericCellValue() + " ");
        }
        else
        {
            // U Can Handel Boolean, Formula, Errors
        }
    }
    System.out.println();
}
}
```

代码(3) ReadWriteExcelFile.java



```
public static void writeXLSFile() throws IOException
{
    String excelFileName = "C:/Test.xls";// name of excel file

    String sheetName = "Sheet1";// name of sheet

    HSSFWorkbook wb = new HSSFWorkbook();
    HSSFSheet sheet = wb.createSheet(sheetName);

    // iterating r number of rows
    for (int r = 0; r < 5; r++)
    {
        HSSFRow row = sheet.createRow(r);

        // iterating c number of columns
        for (int c = 0; c < 5; c++)
        {
            HSSFCell cell = row.createCell(c);

            cell.setCellValue("Cell " + r + " " + c);
        }
    }
}
```



代码(4) ReadWriteExcelFile.java

```
FileOutputStream fileOut = new FileOutputStream(excelFileName);

// write this workbook to an OutputStream.
wb.write(fileOut);
fileOut.flush();
fileOut.close();
}

public static void readXLSXFile() throws IOException
{
    InputStream ExcelFileToRead = new FileInputStream("Test.xlsx");
    XSSFWorkbook wb = new XSSFWorkbook(ExcelFileToRead);

    XSSFSheet sheet = wb.getSheetAt(0);
    XSSFRow row;
    XSSFCell cell;

    Iterator rows = sheet.rowIterator();
```

代码(5) ReadWriteExcelFile.java



```
while (rows.hasNext())
{
    row = (XSSFRow) rows.next();
    Iterator cells = row.cellIterator();
    while (cells.hasNext())
    {
        cell = (XSSFCell) cells.next();

        if (cell.getCellType() == XSSFCell.CELL_TYPE_STRING)
        {
            System.out.print(cell.getStringCellValue() + " ");
        }
        else if (cell.getCellType() == XSSFCell.CELL_TYPE_NUMERIC)
        {
            System.out.print(cell.getNumericCellValue() + " ");
        }
        else
        {
            // U Can Handel Boolean, Formula, Errors
        }
    }
    System.out.println();
}
}
```


代码(6) ReadWriteExcelFile.java



```
public static void writeXLSXFile() throws IOException
{
    String excelFileName = "Test.xlsx";// name of excel file
    String sheetName = "Sheet1";// name of sheet
    XSSFWorkbook wb = new XSSFWorkbook();
    XSSFSheet sheet = wb.createSheet(sheetName);

    // iterating r number of rows
    for (int r = 0; r < 5; r++)
    {
        XSSFRow row = sheet.createRow(r);

        // iterating c number of columns
        for (int c = 0; c < 5; c++)
        {
            XSSFCell cell = row.createCell(c);

            cell.setCellValue("Cell " + r + " " + c);
        }
    }
}
```



代码(7) ReadWriteExcelFile.java

```
FileOutputStream fileOut = new FileOutputStream(excelFileName);

// write this workbook to an OutputStream.
wb.write(fileOut);
fileOut.flush();
fileOut.close();
}

public static void main(String[] args) throws IOException
{
    writeXLSFile();
    readXLSFile();

    writeXLSXFile();
    readXLSXFile();
}
}
```

代码(8) CSVTest.java



```
public class CSVTest {  
  
    public static void main(String[] args) throws Exception {  
        readCSVWithIndex();  
        System.out.println("=====华丽丽的分割线1=====");  
        readCSVWithName();  
        System.out.println("=====华丽丽的分割线2=====");  
        writeCSV();  
        System.out.println("write done");  
    }  
  
    public static void readCSVWithIndex() throws Exception {  
        Reader in = new FileReader("c:/temp/score.csv");  
        Iterable<CSVRecord> records = CSVFormat.EXCEL.parse(in);  
        for (CSVRecord record : records) {  
            System.out.println(record.get(0)); //0 代表第一列  
        }  
    }  
}
```


代码(9) CSVTest.java



```
public static void readCSVWithName() throws Exception {
    Reader in = new FileReader("c:/temp/score.csv");
    Iterable<CSVRecord> records = CSVFormat.RFC4180.withHeader("Name", "Subject", "Score").parse(in);
    for (CSVRecord record : records) {
        System.out.println(record.get("Subject"));
    }
}

public static void writeCSV() throws Exception {
    try (CSVPrinter printer = new CSVPrinter(new FileWriter("person.csv"), CSVFormat.EXCEL)) {
        printer.printRecord("id", "userName", "firstName", "lastName", "birthday");
        printer.printRecord(1, "john73", "John", "Doe", LocalDate.of(1973, 9, 15));
        printer.println(); //空自行
        printer.printRecord(2, "mary", "Mary", "Meyer", LocalDate.of(1985, 3, 29));
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
}
```




谢谢!