Ministry of SMEs and Startups × KOSME Korea SMEs and Startups Agency × LIKELION

Word, Excel, PDF Handling

Nguyễn Anh Tuấn

K TECH COLLEGE

Nội dung bài giảng

1 Microsoft Word Handle

2 Microsoft Excel Handle

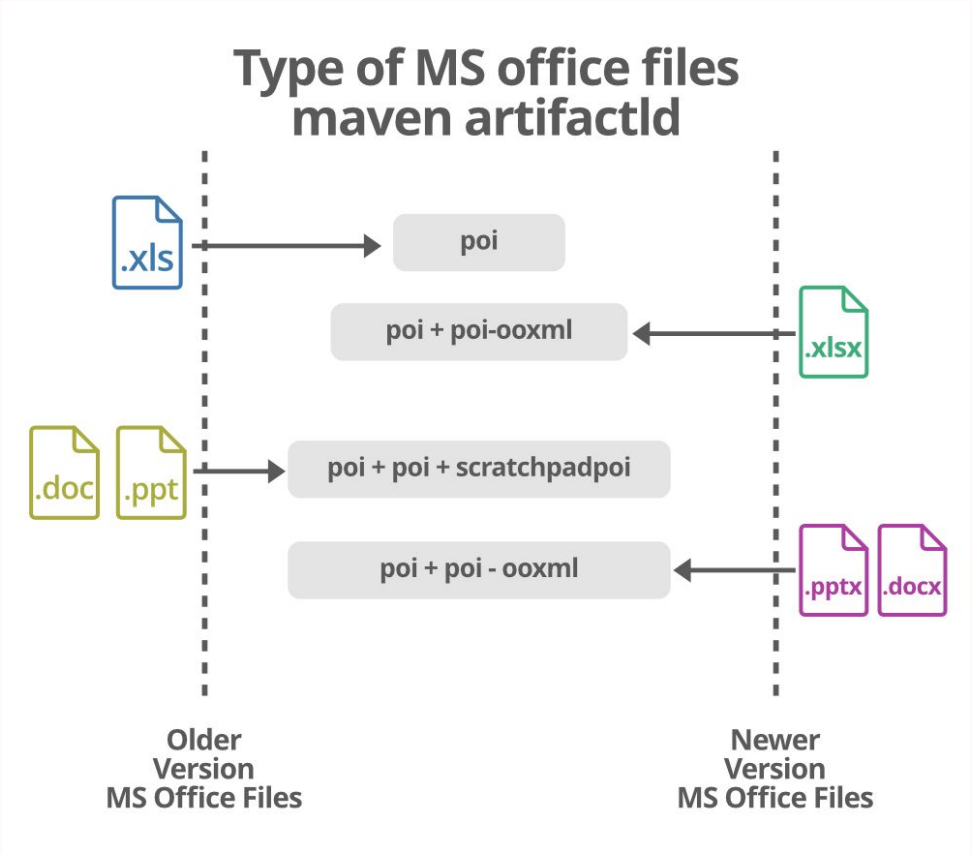
3 Adobe PDF Handle

Microsoft Word Handle

Apache POI

- **Mô tả**:
  - Apache POI giúp Java có thể đọc và ghi các file Word, Excel, Power Point, … trên nhiều nền tảng khác nhau.
  - Apache POI có thể thực hiện các thao tác đọc, ghi trên cả định dạng file .doc, .docx, .xls, .xlsx, .ppt hay .pptx
  - Apache POI là một công cụ mã nguồn mở được cung cấp bởi Apache.
  - https://poi.apache.org/apidocs/5.0/

Import package



Import package

- **Import các package sau vào file pom.xml**:

```
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
        <groupId>org.apache.poi</groupId>
        <artifactId>poi</artifactId>
        <version>5.3.0</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
        <groupId>org.apache.poi</groupId>
        <artifactId>poi-ooxml</artifactId>
        <version>5.3.0</version>
</dependency>
```

https://mvnrepository.com/

Microsoft Word Handle

- **Các class cần dùng**:
  - **XWPFDocument**: Đại diện cho toàn bộ tài liệu Word
    - Tạo hoặc mở tài liệu DOCX.
    - Lưu tài liệu DOCX.
    - Truy cập và sửa đổi các phần khác nhau của tài liệu như các đoạn văn, bảng, tiêu đề, …
  - **XWPFParagraph**: Đại diện cho 1 đoạn văn bản.
    - Truy cập văn bản của một đoạn văn.
    - Thêm, sửa đổi, hoặc xóa đoạn văn bản.
    - Định dạng đoạn văn, chẳng hạn như căn chỉnh, thụt lề, và kiểu chữ.
  - **XWPFRun**: Đại diện cho một đoạn (run) của văn bản trong một đoạn văn. *Ví dụ, trong một đoạn văn bản có thể có nhiều đoạn (run) khác nhau với các kiểu chữ, kích thước, hoặc màu sắc khác nhau.*
    - Thêm văn bản vào một đoạn văn.
    - Định dạng văn bản, chẳng hạn như thiết lập kiểu chữ, kích thước, màu sắc, đậm, nghiêng, gạch chân, và các định dạng khác.

Write Docx File

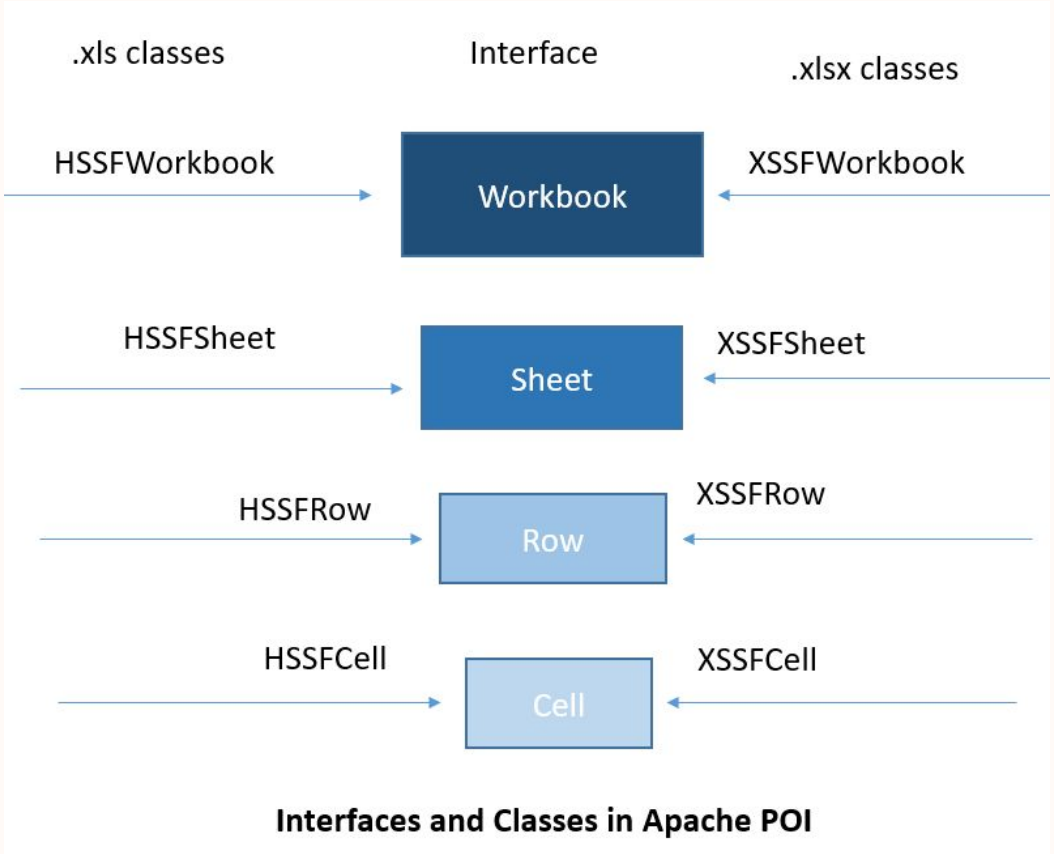
```
public static void main(String[] args) {
        XWPFDocument document = new XWPFDocument();   // Tạo 1 tài liệu mới

        // Tạo 1 đoạn văn bản và thêm nội dung
        XWPFParagraph paragraph = document.createParagraph();
        XWPFRun run = paragraph.createRun();
        run.setText("Xin chào, đây là file đầu tiên tôi viết file word");

        // Ghi ra file output.docx
        FileOutputStream out = new FileOutputStream("output.docx");
        document.write(out);

        document.close();          // Đóng tài liệu
        System.out.println("Đã tạo file docx thành công!");
}
```

Read Docx File

```
public static void main(String[] args) {

        // Mở tài liệu output.docx đã tạo từ bước trước
        File file = new File("output.docx");
        InputStream fis = new FileInputStream(file);
        XWPFDocument document = new XWPFDocument(OPCPackage.open(fis))) {

        // Đọc các đoạn văn bản từ tài liệu
        for (XWPFParagraph paragraph : document.getParagraphs()) {
                System.out.println(paragraph.getText());
        }

        document.close();          // Đóng tài liệu
}
```

Microsoft Excel Handle

Microsoft Excel Handle



Interfaces and Classes in Apache POI

Microsoft Excel Handle

- **Các class cần dùng**:
  - **XSSFWorkbook**: Đại diện cho toàn bộ tài liệu Excel
    - Tạo hoặc mở tài liệu XLSX.
    - Lưu tài liệu XLSX.
    - Truy cập và sửa đổi các sheets trong tài liệu XLSX.
  - **Sheet**: Đại diện cho sheet trong tài liệu Excel.
    - Truy cập các hàng (rows) và ô (cells) trong sheet.
    - Thêm, sửa đổi, hoặc xóa sheet.
    - Đặt tên và bảo vệ sheet.
  - **Row**: Đại diện cho 1 hàng trong 1 sheet.
    - Truy cập các ô (cells) trong hàng.
    - Thêm, sửa đổi, hoặc xóa hàng.
    - Định dạng hàng.
  - **Cell**: Đại diện cho một ô trong 1 row.
    - Truy cập và sửa đổi giá trị của ô.
    - Định dạng ô, chẳng hạn như kiểu dữ liệu, phông chữ, màu sắc, và căn chỉnh.
    - Thêm công thức vào ô.

Write Xlsx File

```
public static void main(String[] args) {
        XSSFWorkbook workbook = new XSSFWorkbook();      // Tạo một workbook mới

        Sheet sheet = workbook.createSheet("Sheet1");          // Tạo một sheet mới

        Row row = sheet.createRow(0);                  // Tạo một hàng mới

        Cell cell1 = row.createCell(0);            // Tạo các ô và đặt giá trị
        cell1.setCellValue("Hello");
        Cell cell2 = row.createCell(1);
        cell2.setCellValue("World");

        FileOutputStream out = new FileOutputStream("output.xlsx");        // Ghi ra file
        workbook.write(out);
        workbook.close();
        System.out.println("Đã tạo file Xlsx thành công!");
}
```

Read Xlsx File

```
public static void main(String[] args) {
        // Đường dẫn đến file Excel
        FileInputStream inputStream = new FileInputStream(new File("output.xlsx"));

        Workbook workbook = WorkbookFactory.create(inputStream);      // Tạo workbook từ file Excel

        Sheet sheet = workbook.getSheetAt(0);      // Lấy sheet đầu tiên từ workbook

        // Duyệt qua từng dòng của sheet và in ra giá trị của từng ô
        for (Row row : sheet) {     for (Cell cell : row) {
                                            switch (cell.getCellType()) {
                                                    case STRING:
                                                            System.out.print(cell.getStringCellValue() + "\t");
                                                            Break; } } }
                workbook.close();
                inputStream.close();
        }
```

Adobe PDF Handle

Import package

- **Import các package sau vào file pom.xml**:

```
<!-- https://mvnrepository.com/artifact/org.apache.pdfbox/pdfbox -->
<dependency>
        <groupId>org.apache.pdfbox</groupId>
        <artifactId>pdfbox</artifactId>
        <version>3.0.2</version>
</dependency>
```

https://mvnrepository.com/

Adobe PDF Handle

- **Các class cần dùng**:
  - **PDDocument**: Đại diện cho toàn bộ tài liệu PDF.
    - Tạo hoặc mở tài liệu PDF.
    - Lưu tài liệu PDF.
    - Truy cập và sửa đổi các trang (pages) trong tài liệu PDF
  - **PDPage**: Đại diện cho một trang trong tài liệu PDF.
    - Để thêm, sửa đổi, hoặc xóa các trang trong tài liệu PDF.
    - Để truy cập và vẽ nội dung trên các trang, chẳng hạn như văn bản, hình ảnh, và hình học.
  - **PDPageContentStream**: Cung cấp các phương thức để vẽ văn bản, hình ảnh, và hình học trên một trang PDF.
    - Bắt đầu và kết thúc việc vẽ nội dung trên một trang PDF.
    - Vẽ văn bản, hình ảnh, đường thẳng, hình chữ nhật, và các hình dạng khác.
    - Thiết lập thuộc tính văn bản và đồ họa như phông chữ, kích thước chữ, màu sắc, và độ dày nét vẽ.

Write PDF File

```
public static void main(String[] args) {
        PDDocument document = new PDDocument();   // Tạo một document PDF mới
        PDPage page = new PDPage();  document.addPage(page);       // Thêm một trang mới vào document
        // Tạo một content stream để vẽ trên trang mới
        PDPageContentStream contentStream = new PDPageContentStream(document, page);
        contentStream.setFont(PDType1Font.HELVETICA_BOLD, 12);  // Đặt font và kích thước

        contentStream.beginText();      // Bắt đầu vẽ
        contentStream.newLineAtOffset(25, 700); // Đặt vị trí bắt đầu vẽ cho dòng 1
        contentStream.showText("Xin chào lớp KTC class 3 BE!");
        contentStream.endText();

        // Lưu document PDF vào file
        File file = new File("output.pdf");       document.save(file);
        // Đóng content stream và document
        contentStream.close();    document.close();
}
```

Read PDF File

```
public static void main(String[] args) {
        // Tạo đối tượng PDDocument từ file PDF
        PDDocument document = PDDocument.load(new File(output.pdf));

        // Tạo đối tượng PDFTextStripper để trích xuất văn bản từ PDF
        PDFTextStripper pdfStripper = new PDFTextStripper();

        String text = pdfStripper.getText(document);        // Lấy nội dung văn bản từ PDF

        System.out.println("PDF Content:");          // In ra nội dung văn bản từ PDF
        System.out.println(text);

        document.close();          // Đóng tài liệu PDF
}
```

Cách xác định điểm vẽ nội dung trong file PDF

- **getMediaBox(), getCropBox(), getArtBox(), getBleedBox()**:
  - Các phương thức của **PDPage** dùng để lấy thông tin chi tiết về kích thước và ranh giới của trang PDF hiện tại.
- Kết quả [0.0, 0.0, 0.0, 0.0] là một mảng chứa các giá trị số thực, thể hiện kích thước của 1 trang PDF.
  - Giả sử kết quả trên là [X, Y, X', Y']:
    - **X**: Tọa độ **X** của góc dưới bên trái (bottom-left corner).
    - **Y**: Tọa độ **Y** của góc dưới bên trái (bottom-left corner).
    - **X'**: Tọa độ **X'** của góc trên bên phải (top-right corner).
    - **Y'**: Tọa độ **Y'** của góc trên bên phải (top-right corner).
  - Từ đó suy ra cách tìm toạ độ để truyền vào **newLineAtOffset()**
    - **newLineAtOffset()** sử dụng tham số đầu tiên là khoảng cách theo chiều ngang tăng dần từ trái sang phải. Và tham số thứ hai là khoảng cách theo chiều dọc tăng dần từ dưới lên trên.
  - Giả sử kích thước trang PDF là: [0.0, 0.0, 612.0, 792.0]
    - Toạ độ của **newLineAtOffset()** sẽ trong khoảng **newLineAtOffset(25, 700)**

Thực hành 60’

Tạo 1 chương trình trong đó:

- Lấy danh sách từ file txt (lấy file danh sách học viên trong bài thực hành về File Handling).
- Thống kê học viên có mặt trong ngày vào file excel (bao gồm cả 2 thuộc tính stt và tên, mỗi thuộc tính 1 cột).
- Thống kê học viên vắng mặt trong ngày vào file word (bao gồm cả 2 thuộc tính stt và tên, mỗi thuộc tính cách nhau 1 tab).
- Cho phép xem lại danh sách học viên trong file word và file excel.

Lưu ý:

- Tên file có định dạng: **[fileName]_[ddMMyyyy]**
- Mã hoá base64 tên của học viên trong cả 2 file để bảo mật thông tin. Chỉ giải mã tên học viên khi đọc file bằng chương trình Java.

Nâng cao (30’), chọn 1 trong 2 nội dung sau:

- Thống kê học viên có mặt vào 1 file excel mẫu cho trước. Yêu cầu ko thay đổi style của file excel mẫu.
- Thống kê học viên có mặt vào 1 file excel mới, yêu cầu có header và tên header, mỗi cột đều có title, title có style bôi đậm, căn giữa mà đổ màu để phân biệt với phần thân.

ROAD TO KOREA

Nếu có bất kỳ thắc mắc nào, hãy đặt câu hỏi qua

mail@mail.com hoặc Zalo 0xxx xxx xxx