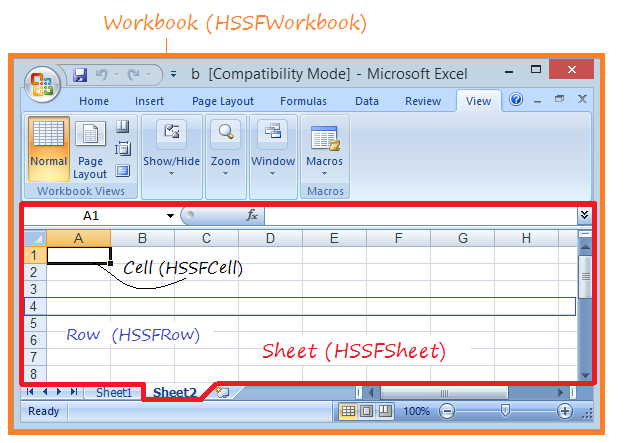
**LAB 3.2: THỰC HÀNH VỚI EXCEL**

**Tổng quan**

Đây là một bài học rất quan trọng trong chuỗi bài học automation.  
Làm việc với excel sẽ rất hữu ích trong việc lưu trữ, xử lý kết quả test, trích xuất báo cáo sau khi chạy xong bộ testcase automation.

Trong bài thực hành này, chúng ta sẽ sử dụng thư viện mã nguồn mở java (**Apache POI**) được cung cấp bởi Apache. Đây là một thư viện mạnh mẽ giúp các bạn làm việc với các tài liệu của Microsoft như Word, Excel, Power point, Visio, ...

**POI** hỗ trợ làm việc dưới định dạng file Excel như dưới đây



Để có thể sử dụng thư viện này trong dự án, chúng ta có 2 cách dưới đây.

* Sử dụng **Maven project**, khai báo thư viện và version của POI trong **pom.xml** file.  
  Khi build project, thư viện sẽ được tải tự động về.
* Download trực tiếp thư viện tại <http://poi.apache.org/download.html> và import vào trong project.

*=> Trong bài thực hành này, chúng ta sẽ làm việc với cách thứ 1 vì tính tiện lợi và đơn giản của Maven. Chi tiết về Maven, chúng ta sẽ được học ở bài học sau.*

**1. Hướng dẫn tạo Maven project.**

**<Tham khảo file: REF\_1\_CREATE\_MAVEN\_PROJECT.docx>**

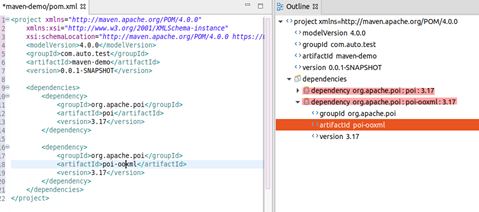
**2. Cài đặt POI trong pom.xml vào project.**

| <**dependency**>  <**groupId**>org.apache.poi</**groupId**>  <**artifactId**>poi</**artifactId**>  <**version**>3.17</**version**> </**dependency**>  <**dependency**>  <**groupId**>org.apache.poi</**groupId**>  <**artifactId**>poi-ooxml</**artifactId**>  <**version**>3.17</**version**> </**dependency**> |
| --- |

Thêm thông tin khai báo thư viện và version vào trong poml.xml file bên dưới thẻ tab <dependencies>. Nếu trong file pom.xml chưa có, thì các bạn có thể tự thêm

Sau đó, sử dụng tổ hợp phím **Ctrl + S** để save file → thư viện sẽ được tự động tải về.

Hoặc các bạn có thể: chuột phải file pom.xml → Run as → Maven clean/install

****

**Hướng dẫn cơ bản về đọc ghi file.**

1. **Đọc dữ liệu từ file**

VD: Ta có 1 file excel Animal.xlsx, trong đó có 1 sheet tên là Breed và giá trị ô A1 = Cat.

* **B1:** Tạo một **FileInputStream** đến đường dẫn lưu trữ file. Mục đích là đọc data .

| FileInputStream fis = new FileInputStream("D:\\path\\Animal .xlsx"); |
| --- |

* **B2:** Tạo một **workbook** thông qua **WorkbookFactory** với tham số đầu vào chính là FileInputStream tại B1. Workbook này chính là đại diện cho file **excel**.

| Workbook wb = WorkbookFactory.create(fis); |
| --- |

* **B3:** Từ **workbook,** chúng sẽ lấy ra **Sheet** như sau:

| Sheet sheet = wb.getSheet("Breed"); |
| --- |

* **B4:** Từ một **Sheet**, chúng ta có thể dễ dàng lấy ra 1 **Row** bất kỳ.

| Row row = sheet.getRow(0); // Lấy ra row đầu tiên |
| --- |

* **B5:** Mỗi một **Row** sẽ bao gồm nhiều **Cell**. Chúng ta có thể lấy ra Cell bất kỳ.

| Cell cell = row.getCell(0); // Lấy ra cell đầu tiên, cell 0 |
| --- |

* **B6:** Bước cuối cùng chính là đọc ra giá trị trong **cell.**

| String data = cell.getStringCellValue() |
| --- |

**2. Ghi dữ liệu vào file**

* **B1:** Khởi tạo workbook cho tập tin xlsx hoặc lấy ra workbook từ 1 file đã có sẵn.

| XSSFWorkbook workbook = new XSSFWorkbook();  or  String fileExcel = "path/books.xlsx";  XSSFWorkbook workbook = new XSSFWorkbook(fileExcel); |
| --- |

* **B2:** Tạo một **Sheet** mới từ **workbook.**

| // tạo sheet có tên là Details  XSSFSheet sheet = workbook.createSheet(**"Details"**); |
| --- |

* **B3:** Tạo 1 **Row** bất kì trên **Sheet.**

| Row row = sheet.createRow(rowIndex); // rowIndex = 0,1,2.. |
| --- |

* **B4:** Tạo ra **Cell** thường hoặc đi kèm công thức.

| Cell cell = row.getCell(columnIndex); Hoặc  cell = row.createCell(columnIndex, CellType.FORMULA); |
| --- |

* **B5:** Set giá trị hoặc công thức cho cell.

| cell.setCellValue(stringValue); or cell.setCellFormula("SUM(B2:B5)"); |
| --- |

* **B6:** Ghi Dữ liệu vào file.

| FileOutputStream out = new FileOutputStream(new File("data.xlsx"));  workbook.write(out);  out.close(); |
| --- |

**I/ Yêu cầu thực hành**

1. Học viên đã nắm được kiến thức về thao tác đọc ghi file **excel**
2. Thực hành đọc hiểu đề bài, phân tích và tìm giải pháp cho đề bài
3. Thực hành đưa giải pháp vào viết code, thực thi và sửa lỗi code nếu có
4. Lưu lại code.

**II/ Ví dụ mẫu**

**1.** Một đối tượng User gồm các thông tin cơ bản sau: Id, firstName, LastName.

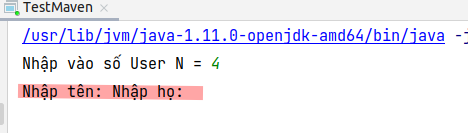
Viết chương trình nhập vào 1 danh sách user từ bàn phím và lưu dữ liệu vào file excel có tên là data.xlsx

*Giải pháp:*

* Nhập vào N là số user  
  *Chú ý:*   
   Trong bài này, nếu các bạn sử dụng nextInt() để đọc vào số người dùng và sau đó sử dụng nextLine() để đọc vào tên và họ thì sẽ xảy ra hiện tượng bị trôi và không thể nhập được tên như dưới đây:

| System.***out***.print(**"Nhập vào số User N = "**); **int** N = scanner.nextInt(); System.***out***.print(**"Nhập tên: "**); String ten = scanner.nextLine(); System.***out***.print(**"Nhập họ: "**); String ho = scanner.nextLine(); |
| --- |

**#Kết quả**



**Lý do:**  
 Nếu sử dụng **nextInt()** trước **nextLine()** thì NextInt() sẽ không đọc ký tự  
 dòng mới khi bạn ấn “ENTER”. NextLine() phía sau sẽ tự động đọc thông  
 tin này và return.  
 **#Phương án**  
 int N = Integer.parseInt(scanner.nextLine());

Sẽ dùng nextLine() để nhập N và chuyển đổi sang kiểu int

* Tạo class User có các thông tin cơ bản (id, firstName, lastName)
* Xây dựng method **writeToExcelFile** để lưu data vào file excel
* Kiểm tra đường dẫn và nội dung file.

*#Code*

| **public class** User {  **private int id**;  **private** String **firstName**;  **private** String **lastName**;   **public** User(**int** id, String firstName, String lastName) {  **this**.**id** = id;  **this**.**firstName** = firstName;  **this**.**lastName** = lastName;  }  **public int** getId() {  **return id**;  }  **public void** setId(**int** id) {  **this**.**id** = id;  }  **public** String getFirstName() {  **return firstName**;  }  **public void** setFirstName(String firstName) {  **this**.**firstName** = firstName;  }  **public** String getLastName() {  **return lastName**;  }  **public void** setLastName(String lastName) {  **this**.**lastName** = lastName;  } } |
| --- |

| **import** org.apache.poi.ss.usermodel.Cell; **import** org.apache.poi.ss.usermodel.Row; **import** org.apache.poi.xssf.usermodel.XSSFSheet; **import** org.apache.poi.xssf.usermodel.XSSFWorkbook;  **import** java.io.File; **import** java.io.FileOutputStream; **import** java.util.\*;  **public class** TestMaven {   **public static void** writeToExcelFile(ArrayList<User> data) {  *// Khởi tạo workbook cho tệp xlsx* XSSFWorkbook workbook = **new** XSSFWorkbook();  *// Khởi tạo một worksheet mới từ workbook* XSSFSheet sheet = workbook.createSheet(**"User Details"**);   *//Create title ID|FIRSTNAME|LASTNAME* Row titleRow = sheet.createRow(0);  Cell cell = titleRow.createCell(0);  cell.setCellValue(**"ID"**);  cell = titleRow.createCell(1);  cell.setCellValue(**"FIRSTNAME"**);  cell = titleRow.createCell(2);  cell.setCellValue(**"LASTNAME"**);   *//create data* **for** (**int** i = 0; i < data.size(); i++) {  User u = data.get(i);  Row localRow = sheet.createRow(i+1) //row 0 for title;  Cell localCell = localRow.createCell(0);  localCell.setCellValue(String.*valueOf*(u.getId()));  localCell = localRow.createCell(1);  localCell.setCellValue(u.getFirstName());  localCell = localRow.createCell(2);  localCell.setCellValue(u.getLastName());  }   **try** {  *// ghi dữ liệu xuống file* FileOutputStream out = **new** FileOutputStream(**new** File(**"D:/demo/data.xlsx"**));  workbook.write(out);  out.close();  System.***out***.println(**"data.xlsx written successfully on disk."**);  } **catch** (Exception e) {  e.printStackTrace();  }  }  **public static void** main(String[] args) {  Scanner scanner = **new** Scanner(System.***in***);  System.***out***.print(**"Nhập vào số User N = "**);  **int** N = Integer.*parseInt*(scanner.nextLine());  *//Tạo arrayList để lưu data* ArrayList<User> data = **new** ArrayList<>();  *//Nhập danh sách user* String firstName = **""**;  String lastName = **""**;  **for** (**int** i = 0; i < N; i++) {  System.***out***.println(**"Nhập User["** + i + **"]: "**);  System.***out***.print(**"First Name: "**);  firstName = scanner.nextLine();  System.***out***.print(**"Last Name: "**);  lastName = scanner.nextLine();  User u = **new** User(i + 1, firstName, lastName);  data.add(u);  }  *//Ghi excel file*  *writeToExcelFile*(data);  } } |
| --- |

**III. Thực hành**

**Bài 1**.

Viết Chương trình nhập vào danh sách Nhân viên từ bàn phím và ghi thông tin vào excel file có tên là “DSNhanvien.xlsx”.

Thông tin của một Nhân viên bao gồm:

* Mã Nhân viên
* Họ tên
* Giới tính
* Năm sinh
* Quê Quán
* Tên phòng ban
* Lương

Gợi ý:

Tương tự ví dụ mẫu