

Openpyxl vs Apache Poi

	A	B	C	D	E	F
1	ID	Name	Math	Physics	Average	Result
2	1	Alice	85	78	81,5	Pass
3	2	Bob	60	55	57,5	Fail
4	3	Charlie	92	88	90	Pass
5	4	Diana	70	66	68	Pass
6	5	Ethan	48	52	50	Fail

	A	B	C	D	E
1	Product	Quantity	Price	Total	
2	Laptop	2	1200	2400	
3	Mouse	10	25	250	
4	Keyboard	5	80	400	
5	Monitor	3	300	900	

Test1 : lire et modifier un fichier Excel

Openpyxl

```
test1.py
1 # lire et modifier un fichier excel avec openpyxl
2 from openpyxl import load_workbook
3 # ouvrir fichier
4 workbook = load_workbook("excel_practice_openpyxl_poi.xlsx")
5 # choisir feuille
6 sheet = workbook["Students"]
7 # lire valeur
8 print(sheet["B2"].value)
9 # modifier valeur
10 sheet["B2"] = "Farah"
11 # sauvegarder
12 workbook.save("excel_practice_openpyxl_poi.xlsx")
13 print("Done")
```

```
PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv/Scripts/Activate.ps1
(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv/Scripts/python.exe c:/Users/owner/Documents/IFT1015/openpyxl/test1.py
Alice
Done
(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl>
```

	A	B	C	D	E	
1	ID	Name	Math	Physics	Average	Result
2	1	Farah	85	78	81,5	Pass
3	2	Bob	60	55	57,5	Fail
4	3	Charlie	92	88	90	Pass
5	4	Diana	70	66	68	Pass
6	5	Ethan	48	52	50	Fail

Apache POI

```
src > main > java > excel > Test1.java > ...
1 package excel;
2
3 import org.apache.poi.ss.usermodel.Sheet; // import la classe sheet
4 import org.apache.poi.ss.usermodel.Workbook; // import le fichier Excel complet
5 import org.apache.poi.ss.usermodel.WorkbookFactory; // pour ouvrir le fichier Excel
6
7 import java.io.FileInputStream; // pour lire le fichier Excel
8 import java.io.FileOutputStream; // pour écrire dans le fichier Excel
9
10 public class Test1 {
11
12     Run | Debug
13     public static void main(String[] args) throws Exception {
14         String file = "excel_practice_openpyxl_poi.xlsx";
15         try (FileInputStream in = new FileInputStream(file);
16             Workbook workbook = WorkbookFactory.create(in)) {
17             Sheet sheet = workbook.getSheet(name: "Students");
18             System.out.println(sheet.getRow(rownum: 1).getCell(cellnum: 1).getStringCellValue());
19             sheet.getRow(rownum: 1).getCell(cellnum: 1).setCellValue(value: "Farah");
20             try (FileOutputStream out = new FileOutputStream(file)) {
21                 workbook.write(out);
22             }
23         }
24         System.out.println("Done");
25     }
26 }
```

Test 2 : Afficher les formules d'un fichier et leurs dépendances dans une seule feuille

```
File Edit Selection View Go Run ...
openpyxl

EXPLORER
  OPENPYXL
    > venv
    excel_dependencies_betwe...
    excel_practice_openpyxl_p...
    test1.py
    test2.py

test2.py > ...
1 # afficher les formules d'un fichier excel et leurs dépendances dans une seule feuille
2 from openpyxl import load_workbook
3 import re
4 # ouvrir fichier
5 workbook = load_workbook("excel_practice_openpyxl_poi.xlsx")
6 # choisir feuille
7 sheet = workbook["Students"]
8 # regex pour détecter cellules comme A1, B2, C15...
9 pattern = r"[A-Z]+[0-9]+"
10 # parcourir toutes les lignes
11 for row in sheet.iter_rows():
12     # parcourir cellules de chaque ligne
13     for cell in row:
14         value = cell.value
15         # vérifier si c'est une formule
16         if isinstance(value, str) and value.startswith("="):
17             print(f"Cellule : {cell.coordinate}")
18             print(f"Formule : {value}")
19             # chercher dépendances
20             dependencies = re.findall(pattern, value)
21             print("Dépendances :")
22             for dep in dependencies:
23                 print(dep)
24
25
PROBLEMS DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:\Users\owner\Documents\IFT1015\openpyxl\venv\Scripts\Activate.ps1
(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:\Users\owner\Documents\IFT1015\openpyxl\venv\Scripts\python.exe c:\Users\owner\Documents\IFT1015\openpyxl\test2.py

Cellule : E2
Formule : =AVERAGE(C2:D2)
Dépendances :
C2
D2

Cellule : F2
Formule : =IF(E2>60,"Pass","Fail")
Dépendances :
E2

Cellule : E3
Formule : =AVERAGE(C3:D3)
Dépendances :
C3
D3

Cellule : F3
Formule : =IF(E3>60,"Pass","Fail")
Dépendances :
E3

Cellule : E4
Formule : =AVERAGE(C4:D4)
Dépendances :
C4
D4

Cellule : F4
Formule : =IF(E4>60,"Pass","Fail")
Dépendances :
E4

Cellule : E5
Formule : =AVERAGE(C5:D5)
Dépendances :
C5
D5

Cellule : F5
Formule : =IF(E5>60,"Pass","Fail")
Dépendances :
E5
```

```
Preview progress.md  Test1.java  Test2.java
src > main > java > excel > Test2 > main(String[])
1  package excel;
2  import org.apache.poi.ss.usermodel.Cell;
3  import org.apache.poi.ss.usermodel.CellType;
4  import org.apache.poi.ss.usermodel.Row;
5  import org.apache.poi.ss.usermodel.Sheet;
6  import org.apache.poi.ss.usermodel.Workbook;
7  import org.apache.poi.ss.usermodel.WorkbookFactory;
8  import org.apache.poi.ss.util.CellReference;
9  import java.io.FileInputStream;
10 import java.util.regex.Matcher;
11 import java.util.regex.Pattern;
12
13 public class Test2 {
14     private static final Pattern CELL_REF = Pattern.compile("[A-Z]+[0-9]+");
15     public static void main(String[] args) throws Exception {
16         try (FileInputStream in = new FileInputStream("excel_practice_openpyxl_poi.xlsx");
17             Workbook workbook = WorkbookFactory.create(in)) {
18             Sheet sheet = workbook.getSheet(name: "Students");
19             for (Row row : sheet) {
20                 for (Cell cell : row) {
21                     if (cell.getCellType() == CellType.FORMULA) {
22                         String formula = "=" + cell.getCellFormula();
23                         System.out.println();
24                         System.out.println("Cellule : " + new CellReference(cell).formatAsString());
25                         System.out.println("Formule : " + formula);
26                         System.out.println("Dépendances :");
27                         Matcher matcher = CELL_REF.matcher(formula);
28                         while (matcher.find()) {
29                             System.out.println(matcher.group());
30                         }
31                     }
32                 }
33             }
34             for (Row row : sheet) {
35                 for (Cell cell : row) {
36                     System.out.println(new CellReference(cell).formatAsString() + " " + cellContent(cell));
37                 }
38             }
39         }
40     }
41     private static Object cellContent(Cell cell) {
42         if (cell.getCellType() == CellType.FORMULA) {
43             return "=" + cell.getCellFormula();
44         }
45         switch (cell.getCellType()) {
46             case NUMERIC:
47                 return cell.getNumericCellValue();
48             case STRING:
49                 return cell.getStringCellValue();
50             case BOOLEAN:
51                 return cell.getBooleanCellValue();
52             default:
53                 return null;
54         }
55     }
56 }
```

En Python avec openpyxl, le code est beaucoup plus court, lisible et naturel : les cellules, lignes et formules sont manipulées directement sans gestion explicite des types. Par exemple `cell.value` peut retourner automatiquement un texte, un nombre ou une formule, alors qu'en Java avec Apache POI il faut vérifier explicitement le type avec `CellType`, utiliser `getStringCellValue()`, `getNumericCellValue()` ou `getCellFormula()`.

Python permet aussi un parcours très simple avec `iter_rows()` et une extraction rapide des dépendances grâce à `re.findall()`, tandis qu'en Java il faut utiliser `Matcher`, `Pattern`, des objets `Row`, `Cell`, et davantage de boilerplate.

Apache POI offre un contrôle plus robuste et typé, ce qui est avantageux dans les applications entreprise et les systèmes critiques. Dans ce test précis de détection de formules et dépendances Excel, openpyxl est clairement plus rapide à développer et plus adapté à l'analyse logique des feuilles Excel, alors qu'Apache POI demande plus de code mais donne une architecture plus stricte et professionnelle.

Test 3

EXPLORER

test1.py

test2.py

test3.py

test3.py > ...

```
1 # afficher les valeurs calculées d'un fichier excel
2 from openpyxl import load_workbook
3
4 # data_only=True → valeurs calculées, pas les formules
5 workbook = load_workbook("excel_practice_openpyxl_poi.xlsx", data_only=True)
6
7 # Students - colonne Average (E)
8 print("Students - Average:")
9 for row in range(2, 7):
10     print(workbook["Students"][f"E{row}"].value)
11
12 # Sales - colonne Total (D)
13 print("\nSales - Total:")
14 for row in range(2, 6):
15     print(workbook["Sales"][f"D{row}"].value)
```

PROBLEMS

DEBUG CONSOLE

TERMINAL

PORTS

```
● PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv
(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpy
openpyxl/test.py
Students - Average:
81.5
57.5
90
68
50

Sales - Total:
2400
250
400
900
○ (venv) PS C:\Users\owner\Documents\IFT1015\openpyxl>
```

OUTPUT

Filter (e.g. text, lexclu...)

src > main > java > excel > J Test3.java > ...

```
1 package excel;
2
3 import org.apache.poi.ss.usermodel.Cell;
4 import org.apache.poi.ss.usermodel.CellType;
5 import org.apache.poi.ss.usermodel.DateUtil;
6 import org.apache.poi.ss.usermodel.Row;
7 import org.apache.poi.ss.usermodel.Sheet;
8 import org.apache.poi.ss.usermodel.Workbook;
9 import org.apache.poi.ss.usermodel.WorkbookFactory;
10
11 import java.io.FileInputStream;
12
13 public class Test3 {
14
15     Run | Debug
16     public static void main(String[] args) throws Exception {
17         try (FileInputStream in = new FileInputStream("excel_practice_openpyxl_poi.xlsx");
18             Workbook workbook = WorkbookFactory.create(in)) {
19             Sheet students = workbook.getSheet(name: "Students");
20             System.out.println("Students - Average:");
21             for (int row = 2; row <= 6; row++) {
22                 Row dataRow = students.getRow(row - 1);
23                 Cell cell = dataRow == null ? null : dataRow.getCell(cellnum: 4);
24                 System.out.println(cachedValue(cell));
25             }
26
27             Sheet sales = workbook.getSheet(name: "Sales");
28             System.out.println("\nSales - Total:");
29             for (int row = 2; row <= 5; row++) {
30                 Row dataRow = sales.getRow(row - 1);
31                 Cell cell = dataRow == null ? null : dataRow.getCell(cellnum: 3);
32                 System.out.println(cachedValue(cell));
33             }
34         }
35     }
36
37     private static Object cachedValue(Cell cell) {
38         if (cell == null) {
39             return null;
40         }
41         CellType type = cell.getCellType();
42         if (type == CellType.FORMULA) {
43             type = cell.getCachedFormulaResultType();
44         }
45         switch (type) {
46             case NUMERIC:
47                 return DateUtil.isCellDateFormatted(cell) ? cell.getDateCellValue() : cell.getNumericCellValue();
48             case STRING:
49                 return cell.getStringCellValue();
50             case BOOLEAN:
51                 return cell.getBooleanCellValue();
52             default:
53                 return null;
54         }
55     }
56 }
```

Dans ce test, openpyxl est beaucoup plus simple et rapide : avec data_only=True, Python récupère directement les valeurs calculées des cellules sans gestion complexe des types. En revanche, Apache POI demande plus de code car Java vérifie explicitement le type de chaque cellule (FORMULA, NUMERIC, STRING, BOOLEAN) et gère aussi les dates avec DateUtil.

data_only=True permet à openpyxl de retourner la dernière valeur calculée par Excel au lieu de retourner la formule de la cellule.

Test 5 : ajouter une colonne dans une feuille Excel

EXPLORER

test1.py test2.py test3.py test4.py test5.py excel_practice_openpyxl_poi.xlsx

OPENPYXL

.idea

venv

excel_dependencies_betwe...

excel_practice_openpyxl_p...

test1.py

test2.py

test3.py

test4.py

test5.py

test5.py > ...

```
1 # ajouter une colonne dans une feuille excel avec openpyxl
2 from openpyxl import load_workbook
3 # ouvrir fichier
4 workbook = load_workbook("excel_practice_openpyxl_poi.xlsx")
5 # choisir feuille
6 sheet = workbook["Students"]
7 # ajouter une colonne avant la colonne C
8 sheet.insert_cols(3)
9 # ajouter titre
10 sheet["C1"] = "Age"
11 # ajouter données
12 sheet["C2"] = 20
13 sheet["C3"] = 21
14 sheet["C4"] = 19
15 sheet["C5"] = 22
16 sheet["C6"] = 20
17 # sauvegarder
18 workbook.save("excel_practice_openpyxl_poi.xlsx")
19 print("Colonne ajoutée")
```

PROBLEMS

DEBUG CONSOLE

TERMINAL

PORTS

PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv/Scripts/Activate.ps1

(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv/Scripts/python.exe c:/Users/owner/Documents/IFT1015/openpyxl/test5.py

Colonne ajoutée

(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl>

C1			Age					
	A	B	C	D	E	F	G	
1	ID	Name	Age	Math	Physics	Average	Result	
2	1	Farah	20	85	78	52,5	Pass	
3	2	Bob	21	60	55	40,5	Fail	
4	3	Charlie	19	92	88	55,5	Pass	
5	4	Diana	22	70	66	46	Pass	
6	5	Ethan	20	48	52	34	Fail	
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								

< >

Students Sales +

src > main > java > excel > Test5.java > {} excel

```
1  package excel;
2
3  import org.apache.poi.ss.usermodel.Row;
4  import org.apache.poi.ss.usermodel.Sheet;
5  import org.apache.poi.ss.usermodel.Workbook;
6  import org.apache.poi.ss.usermodel.WorkbookFactory;
7
8  import java.io.FileInputStream;
9  import java.io.FileOutputStream;
10
11 public class Test5 {
12
13     Run | Debug
14     public static void main(String[] args) throws Exception {
15         String file = "excel_practice_openpyxl_poi.xlsx";
16         try (FileInputStream in = new FileInputStream(file);
17             Workbook workbook = WorkbookFactory.create(in)) {
18             Sheet sheet = workbook.getSheet(name: "Students");
19             int lastCol = 0;
20             for (Row row : sheet) {
21                 if (row != null && row.getLastCellNum() > lastCol) {
22                     lastCol = row.getLastCellNum();
23                 }
24             }
25             if (lastCol > 0) {
26                 sheet.shiftColumns(startColumn: 2, lastCol - 1, n: 1);
27             }
28             Row header = sheet.getRow(rownum: 0);
29             if (header == null) {
30                 header = sheet.createRow(rownum: 0);
31             }
32             header.createCell(column: 2).setCellValue(value: "Age");
33             int[] ages = {20, 21, 19, 22, 20};
34             for (int i = 0; i < ages.length; i++) {
35                 Row row = sheet.getRow(i + 1);
36                 if (row == null) {
37                     row = sheet.createRow(i + 1);
38                 }
39                 row.createCell(column: 2).setCellValue(ages[i]);
40             }
41             try (FileOutputStream out = new FileOutputStream(file)) {
42                 workbook.write(out);
43             }
44             System.out.println("Colonne ajoutée");
45         }
46     }
47 }
```

Test 6: supprimer une ligne

test1.py test2.py test3.py test4.py test5.py test6.py excel_practice_openpyxl_poi.xlsx

```
test6.py > ...
1 from openpyxl import load_workbook
2 # ouvrir fichier
3 workbook = load_workbook("excel_practice_openpyxl_poi.xlsx")
4 # choisir feuille
5 sheet = workbook["Students"]
6 # supprimer ligne 3
7 sheet.delete_rows(3)
8 # supprimer dernière ligne inutile
9 sheet.delete_rows(sheet.max_row)
10 # sauvegarder
11 workbook.save("excel_practice_openpyxl_poi.xlsx")
12 print("Ligne supprimée")
```

PROBLEMS DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv/Scripts/Activate.ps1
(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv/Scripts/python.exe c:/Users/owner/Documents/IFT1015\openpyxl/test6.py
Ligne supprimée
(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl>

F4 =MOYENNE(C5:D5)

	A	B	C	D	E	F	G
1	ID	Name	Age	Math	Physics	Average	Result
2	1	Farah	20	85	78	52,5	Pass
3	3	Charlie	19	92	88	46	Pass
4	4	Diana	22	70	56	#DIV/0!	Fail
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							

Students Sales +

File Edit Selection View Go Run openpyxl

test1.py test2.py test3.py restore_after_test6.py test4.py test5.py test6.py

```
test6.py > ...
1 from openpyxl import load_workbook
2
3 # À lancer UNE SEULE FOIS sur un fichier avec 3 élèves (lignes 2-4).
4 # Relancer supprime encore la ligne 3 (ex. Diana) et ne garde que Farah.
5
6 # ouvrir fichier
7 workbook = load_workbook("excel_practice_openpyxl_poi.xlsx")
8 # choisir feuille
9 sheet = workbook["Students"]
10 # supprimer ligne 3
11 sheet.delete_rows(3)
12 # supprimer dernière ligne seulement si elle est vide (évite de supprimer des données)
13 while sheet.max_row > 1 and sheet[f"A{sheet.max_row}"].value is None:
14     sheet.delete_rows(sheet.max_row)
15
16 # openpyxl ne met pas à jour les formules après delete_rows
17 for row in range(2, sheet.max_row + 1):
18     if sheet[f"A{row}"].value is None:
19         break
20     sheet[f"F{row}"] = f"AVERAGE(C{row}:D{row})"
21     sheet[f"G{row}"] = f"=IF(E{row}>=60,\"Pass\",\"Fail\")"
22
23 # sauvegarder
24 workbook.save("excel_practice_openpyxl_poi.xlsx")
25 print("Ligne supprimée")
```

H13							
	A	B	C	D	E	F	G
1	ID	Name	Age	Math	Physics	Average	Result
2	1	Farah	20	85	78	52,5	Pass
3	3	Charlie	19	92	88	55,5	Pass
4	4	Diana	22	70	66	46	Pass
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							

<

>

StudentsSales+

E2								
	A	B	C	D	E	F	G	H
1	ID	Name	Age	Math	Physics	Average	Result	
2	1	Farah	20	85	78	52,5	Pass	
3	4	Diana	22	70	66	46	Pass	
4								

```

src > main > java > excel > J Test6.java > {} excel
1 package excel;
2
3 import org.apache.poi.ss.usermodel.Cell;
4 import org.apache.poi.ss.usermodel.CellType;
5 import org.apache.poi.ss.usermodel.Row;
6 import org.apache.poi.ss.usermodel.Sheet;
7 import org.apache.poi.ss.usermodel.Workbook;
8 import org.apache.poi.ss.usermodel.WorkbookFactory;
9
10 import java.io.FileInputStream;
11 import java.io.FileOutputStream;
12
13 public class Test6 {
14
15     Run | Debug
16     public static void main(String[] args) throws Exception {
17         String file = "excel_practice_openpyxl_poi.xlsx";
18         try (FileInputStream in = new FileInputStream(file);
19             Workbook workbook = WorkbookFactory.create(in)) {
20             Sheet sheet = workbook.getSheet(name: "Students");
21             int rowIndex = 2;
22             Row toRemove = sheet.getRow(rowIndex);
23             if (toRemove != null) {
24                 sheet.removeRow(toRemove);
25             }
26             int lastRow = sheet.getLastRowNum();
27             if (rowIndex < lastRow) {
28                 sheet.shiftRows(rowIndex + 1, lastRow, -1);
29             }
30
31             while (sheet.getLastRowNum() > 0) {
32                 Row last = sheet.getRow(sheet.getLastRowNum());
33                 Cell idCell = last == null ? null : last.getCell(cellnum: 0);
34                 if (idCell == null || idCell.getCellType() == CellType.BLANK) {
35                     int max = sheet.getLastRowNum();
36                     Row empty = sheet.getRow(max);
37                     if (empty != null) {
38                         sheet.removeRow(empty);
39                     }
40                     if (max == 0) {
41                         break;
42                     }
43                 } else {
44                     break;
45                 }
46             }
47
48             for (int row = 2; row <= sheet.getLastRowNum() + 1; row++) {
49                 Row dataRow = sheet.getRow(row - 1);
50                 if (dataRow == null || dataRow.getCell(cellnum: 0) == null
51                     || dataRow.getCell(cellnum: 0).getCellType() == CellType.BLANK) {
52                     break;
53                 }
54                 int r = row;
55                 dataRow.createCell(column: 5).setCellFormula("AVERAGE(C" + r + ":D" + r + ")");
56                 dataRow.createCell(column: 6).setCellFormula("IF(E" + r + ">=60,\"Pass\",\"Fail\")");
57             }
58
59             try (FileOutputStream out = new FileOutputStream(file)) {
60                 workbook.write(out);
61             }
62             System.out.println("Ligne supprimée");
63         }
64     }
65 }

```

Test 4 : dépendances entre les feuilles

Product				
A	B	C	D	E
1	Product	Price	Quantity	Total
2	Laptop	1200	2	
3	Mouse	25	10	
4	Keyboard	80	5	
5	Monitor	300	3	
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				
20				
21				

=SOMME(Products!D2:D5)		
A	B	C
1	Description	Value
2	Total Sales	3950
3	Average Price	401,25
4	Max Sale	2400
5	Min Sale	250
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		

=Summary!B4-Summary!B5		
A	B	C
1	Metric	Result
2	Sales + Tax	4542,5
3	Double Average	802,5
4	Difference	2150
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		

```
EXPLORER
...
test1.py
test2.py
test3.py
test4.py
excel_practice_openpyxl_poi.xlsx

OPENPYXL
> .idea
> venv
~$excel_dependencies_bet...
excel_dependencies_betwe...
excel_practice_openpyxl_p...
test1.py
test2.py
test3.py
test4.py

test4.py > ...
1 # afficher les formules d'un fichier excel et leurs dépendances dans plusieurs feuilles
2 from openpyxl import load_workbook
3 import re
4 workbook = load_workbook("excel_dependencies_between_sheets.xlsx")
5 pattern = r"[A-Za-z0-9_]+![A-Z]+[0-9]+"
6 # parcourir toutes les feuilles
7 for sheet in workbook.worksheets:
8     print(f"\n==== FEUILLE : {sheet.title} =====")
9     # parcourir toutes les lignes
10    for row in sheet.iter_rows():
11        # parcourir cellules
12        for cell in row:
13            value = cell.value
14            # vérifier si formule
15            if isinstance(value, str) and value.startswith("="):
16                print(f"\nCellule : {sheet.title}!{cell.coordinate}")
17                print(f"Formule : {value}")
18                # trouver dépendances
19                dependencies = re.findall(pattern, value)
20                # afficher dépendances
21                if dependencies:
22                    print("Dépendances :")
23                    for dep in dependencies:
24                        print(dep)
25
```

- PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv/Scripts/Activate.ps1
- (venv) PS C:\Users\owner\Documents\IFT1015\openpyxl> & c:/Users/owner/Documents/IFT1015/openpyxl/venv/Scripts/python.exe c:/Users/owner/Documents/IFT1015/openpyxl/test4.py

```

===== FEUILLE : Products =====

Cellule : Products!D2
Formule : =B2*C2

Cellule : Products!D3
Formule : =B3*C3

Cellule : Products!D4
Formule : =B4*C4

Cellule : Products!D5
Formule : =B5*C5

===== FEUILLE : Summary =====

Cellule : Summary!B2
Formule : =SUM(Products!D2:D5)
Dépendances :
Products!D2

Cellule : Summary!B3
Formule : =AVERAGE(Products!B2:B5)
Dépendances :
Products!B2

Cellule : Summary!B4
Formule : =MAX(Products!D2:D5)
Dépendances :
Products!D2

Cellule : Summary!B5
Formule : =MIN(Products!D2:D5)
Dépendances :
Products!D2

===== FEUILLE : Dashboard =====

Cellule : Dashboard!B2
Formule : =Summary!B2*1.15
Dépendances :
Summary!B2

Cellule : Dashboard!B3
Formule : =Summary!B3*2
Dépendances :
Summary!B3

Cellule : Dashboard!B4
Formule : =Summary!B4-Summary!B5
Dépendances :
Summary!B4
Summary!B5
(venv) PS C:\Users\owner\Documents\IFT1015\openpyxl>

```

```

src > main > java > excel > J Test4.java > {} excel
1  package excel;
2
3  import org.apache.poi.ss.usermodel.Cell;
4  import org.apache.poi.ss.usermodel.CellType;
5  import org.apache.poi.ss.usermodel.Row;
6  import org.apache.poi.ss.usermodel.Sheet;
7  import org.apache.poi.ss.usermodel.Workbook;
8  import org.apache.poi.ss.usermodel.WorkbookFactory;
9  import org.apache.poi.ss.util.CellReference;
10
11 import java.io.FileInputStream;
12 import java.util.regex.Matcher;
13 import java.util.regex.Pattern;
14
15 public class Test4 {
16
17     private static final Pattern CROSS_SHEET_REF =
18         Pattern.compile("[A-Za-z0-9_]+![A-Z]+[0-9]+");
19
20     Run | Debug
21     public static void main(String[] args) throws Exception {
22         try (FileInputStream in = new FileInputStream("excel_dependencies_between_sheets.xlsx");
23             Workbook workbook = WorkbookFactory.create(in)) {
24             for (int i = 0; i < workbook.getNumberOfSheets(); i++) {
25                 Sheet sheet = workbook.getSheetAt(i);
26                 System.out.println("\n===== FEUILLE : " + sheet.getSheetName() + " =====");
27                 for (Row row : sheet) {
28                     for (Cell cell : row) {
29                         if (cell.getCellType() == CellType.FORMULA) {
30                             String formula = "=" + cell.getCellFormula();
31                             String coord = sheet.getSheetName() + "!"
32                                 + new CellReference(cell.getRowIndex(), cell.getColumnIndex(), pAbsRow: false, pAbsCol: false)
33                                 .formatAsString();
34                             System.out.println();
35                             System.out.println("Cellule : " + coord);
36                             System.out.println("Formule : " + formula);
37                             Matcher matcher = CROSS_SHEET_REF.matcher(formula);
38                             boolean found = false;
39                             while (matcher.find()) {
40                                 if (!found) {
41                                     System.out.println("Dépendances :");
42                                     found = true;
43                                 }
44                                 System.out.println(matcher.group());
45                             }
46                         }
47                     }
48                 }
49             }
50         }
51     }
52 }

```