



Excel(실습)

배 희호 교수
경북대학교
소프트웨어융합과



Excel



- Server 측 Programming을 하다 보면, 특히나 Web Programming을 하다 보면, Excel File을 읽고 쓰는 일이 빈번함
 - 예) 화면에 출력된 내용을 Excel File로 내보내기(export)한다거나, 한꺼번에 다량의 항목을 등록하려 할 때에 Excel File로 등록하기(import) 등의 기능이 이에 해당 됨
- 물론 다양한 방법으로 Excel File 읽고 쓰는 것이 가능하지만, 가장 흔히 사용되고 편리한 것은 POI라는 것을 이용하는 것
- 사실 과거에는 Excel File을 Test File로 읽어 들이면, XML File과 유사한 형태를 띄고 있어 쉽게 읽고 쓰는 것이 가능했지만, 최근에는 보안상의 이유인지 이마저도 제대로 보이지 않고, 특히 *.xlsx File은 똑바로 읽을 방법이 없음



JXL

JXL Library 다운로드

<http://sourceforge.net/projects/jexcelapi/files/>

The screenshot shows the SourceForge project page for 'A Java library for reading/writing Excel'. The page includes a navigation bar with 'Open Source Software', 'Business Software', and 'Resources'. A search bar is present on the right. The main content area features a large header with the project title and a 'Download Latest Version' button. Below this is a table listing files, including 'CSharpJExcel' and 'jexcelapi'. The right sidebar contains a 'Recommended Projects' section and a 'Sign Up' button.

Home / Browse / Business & Enterprise / Office Suites / A Java library for reading/writing Excel / Files

A Java library for reading/writing Excel

Brought to you by: [grimholtz](#), [lamber45](#)

Summary Files Reviews Support Wiki

Download Latest Version
C# port of JExcel 2.6.12 for Visual Studio 2005 or 2008 (5.9 MB) [Get Updates](#)

Home

Name	Modified	Size	Downloads / Week
CSharpJExcel	2010-03-24		85
jexcelapi	2009-10-26		187
Totals: 2 Items			272

<https://googleads.g.doubleclick.net/dbm/clk?sa=L&ai=CfOtKleK7YOrGHsz02ATgsaalCdXmqJ...>




Futuristic Innovator
京福大學校
KYUNGBOK UNIVERSITY



JXL

■ File을 download후 압축풀어 jxl.jar

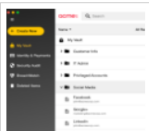
[Open Source Software](#) [Business Software](#) [Resources](#) [Menu](#)





**Awardco**
Customer Love Life
[Learn More](#)



Download Latest Version
C# port of JExcel 2.6.12 for Visual Studio 2005 or 2008 (5.9 MB)
[Get Updates](#)

Home / jexcelapi / 2.6.12

Name	Modified	Size	Downloads / Week
Parent folder			
2_6_12_releasenotes.txt	2009-10-26	606 Bytes	3
jexcelapi_2_6_12.zip	2009-10-26	2.5 MB	48
jexcelapi_2_6_12.tar.gz	2009-10-26	1.9 MB	2
Totals: 3 Items		4.4 MB	53

Other Useful Business Software
**Cybersecurity Starts With Password Security.**
Keeper is the top-rated password manager for protecting you, your family and your business from password-related data breaches and cyberthreats.
Research shows that a whopping 81% of data breaches are due to weak or stolen passwords. Business password managers provide an affordable and simple way for companies to solve the single biggest root cause of most data breaches. By implementing Keeper, your

Recommended Projects
 **iText®**, a JAVA PDF library
PDF Library for Developers
 **EtherCalc**
Open source web spreadsheet
 **jxl**
jxl is a pure java 5 super-project that includes Enclosure...
 **jxls**
jxls is a small and easy-to-use Java library for writing Excel...

 **PROVISIO LLC**






JXL



■ JXL Download

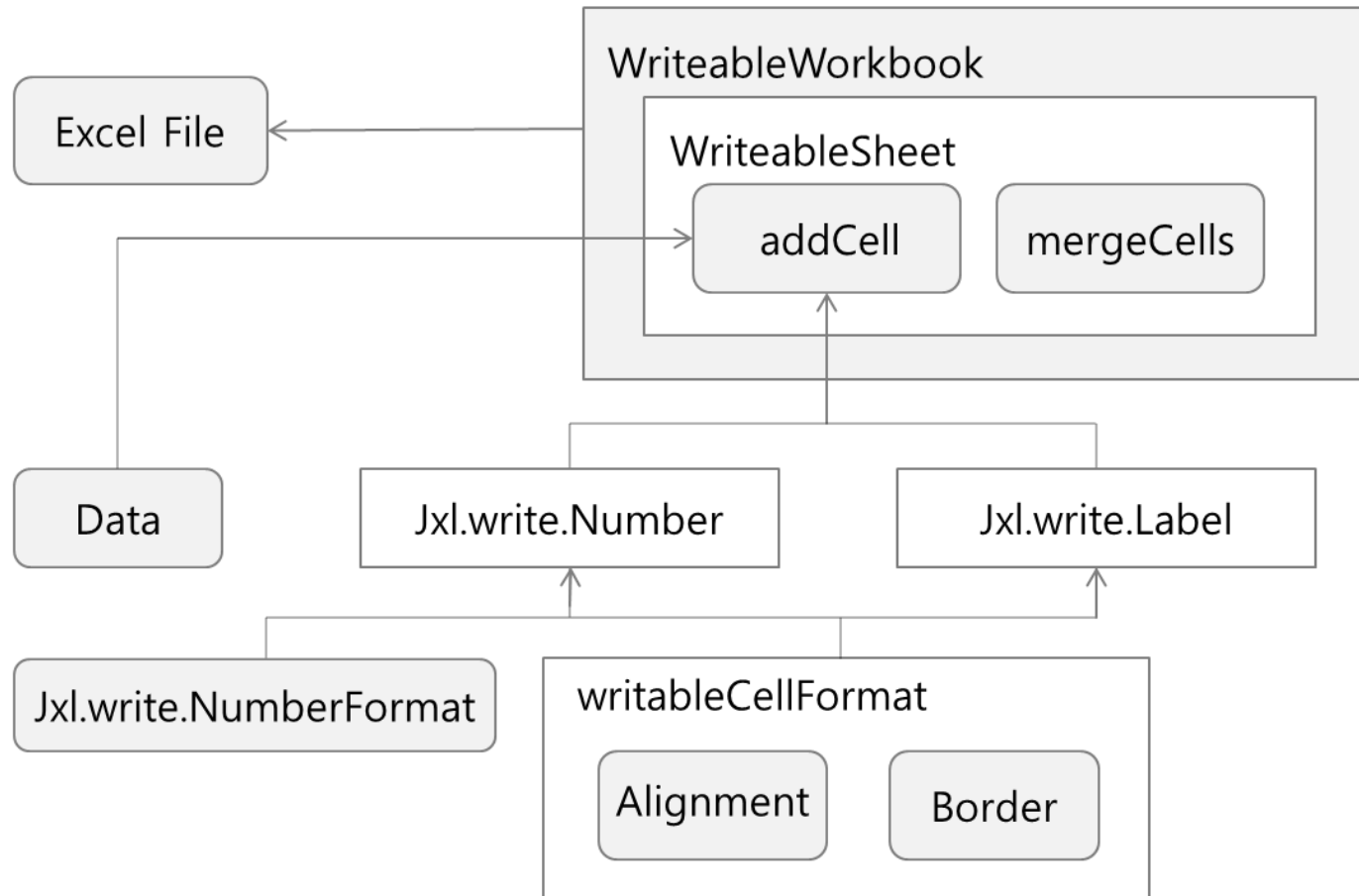
- Data를 Excel File 형식으로 Download하기 위해 JXL에서 제공하는 WriteableWorkbook 클래스가 있음
 - 이는 Excel에서 File에 해당
- WriteableSheet 클래스는 Excel에서 Sheet를 의미
- 원리는 Data를 WriteableSheet 클래스의 addCell() 메소드를 통해 다양한 Cell Option을 적용해서 Excel File로 출력
- 이때 Cell의 Option 적용은 Number와 String형을 구분해서 공통적으로 Cell의 Alignment와 Border등을 설정하며, Number형일 경우에는 해당 Number의 Format을 설정할 수 있음



JXL



■ JXL Download



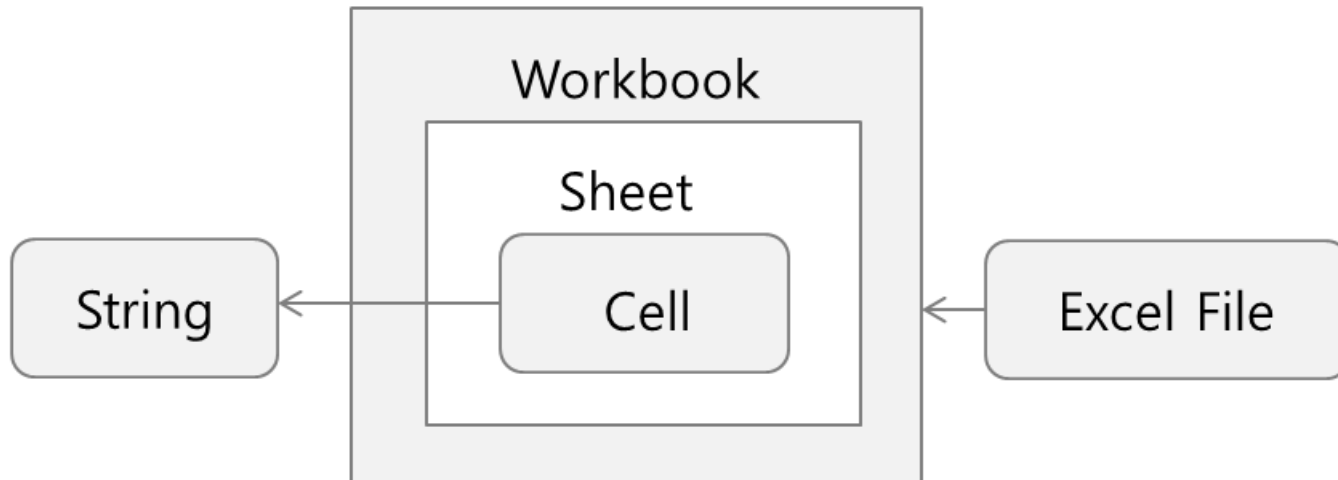


JXL



■ JXL Upload

- 다운로드에서 사용하는 WriteableWorkbook 클래스 대신 Workbook 클래스를 사용하고 WriteableSheet 클래스 대신 Sheet 클래스를 사용
- 사용자가 선택한 Excel File을 Sheet 단위로 나누고 다시 Cell 단위로 각각 조회를 해서 String 형식으로 값을 읽을 수 있음





JXL



■ JXL을 선택한 이유

- Excel Data를 처리할 때 대규모 Data를 처리하는 특성상 주로 CSV 형식으로 변환하고 필요에 따라 JXL을 사용해서 Excel 처리함
- CSV 형식을 사용하면 Data 크기에 제약이 적으며, 속도도 빠르고 구현하기도 간단하지만, 단순히 표현하는 기능 외에는 설정할 수 없어서 아쉬운 점이 많음
- JXL을 사용하게 되면 CSV 보다 느리지만 속도를 어느 정도 유지하면서 기본적인 정렬이나 Data의 Format 형식을 지정할 수 있기 때문에 무난하게 사용할 수 있음
- Excel 2003 이후 버전부터는 지원이 되지 않는 단점이 있음. 이를 감안 하더라도 JXL은 충분한 매력이 있는 기술



JXL



■ JXL 사용방법

- JXL을 사용하기 위해 필요한 Library는 jxl.jar File 임
- jar File을 추가하고, Upload(Download)일 때 Workbook(writeableWorkbook)과 Sheet(WritableSheet)를 생성
- Download일 경우에는 Cell 서식을 설정하고, Excel에 표현할 Data를 Cell에 담아서 표현을 하면 됨



Excel 예제 1

```
public static void main(String[] args) {
    File file = new File("../data//today.xls");
    try {
        if (!file.exists()) {
            file.createNewFile();
        }
        WritableWorkbook workbook = null;
        workbook = Workbook.createWorkbook(file);
        WritableSheet sheet = workbook.createSheet("Sheet1", 0);
        Label label;
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < 3; j++) {
                label = null;
                label = new Label(j, i, "용테크(" + String.valueOf(i) + "," +
                                     String.valueOf(j) + ")");
                sheet.addCell(label);
            }
        }
    }
}
```



Excel 예제 1

```
workbook.write();  
workbook.close();  
System.out.println("생성 완료");  
} catch (IOException e) {  
    System.out.println(e.getMessage());  
} catch (RowsExceededException e) {  
    System.out.println(e.getMessage());  
} catch (WriteException e) {  
    System.out.println(e.getMessage());  
}  
}
```



Excel 예제 1



- Cell에 내용을 기록할 때에는 new Label()을 사용하면 됨
 - new Label(열 순서, 행 순서, "작성 문자열");
 - 참고로 열/행 순서는 0부터 시작



Excel 예제 2

```
public static void main(String[] args) {  
    File file = new File("../WWDATAWW/member.xls");  
    String[] name = {"홍길동", "김영희"};  
    int[] age = {21, 20};  
  
    List list = new ArrayList();  
    for (int i = 0; i < 2; i++) {  
        HashMap map = new HashMap();  
        map.put("name", name[i]);  
        map.put("age", age[i]);  
        list.add(map);  
    }  
  
    try {  
        WritableWorkbook workbook = Workbook.createWorkbook(file);  
        workbook.createSheet("sheet1", 0);  
        WritableSheet sheet = workbook.getSheet(0);  
        Label label = new Label(0, 0, "name");  
        sheet.addCell(label);  
    }  
}
```



Excel 예제 2



```
label = new Label(1, 0, "age");
sheet.addCell(label);

for (int i = 0; i < list.size(); i++) {
    HashMap map = (HashMap) list.get(i);

    label = new Label(0, (i + 1), (String) map.get("name"));
    sheet.addCell(label);

    label = new Label(1, (i + 1), map.get("age").toString());
    sheet.addCell(label);
}
workbook.write();
workbook.close();
System.out.println("생성 완성");
} catch (Exception e) {
    System.out.println(e.getMessage());
}
}
```



Excel 예제 3



```
public static void main(String[] args) {  
    File file = new File("../data//today.xls");  
    try {  
        Workbook workbook = Workbook.getWorkbook(file);  
        Sheet sheet = workbook.getSheet(0);  
  
        for (int i = 0; i < sheet.getRows(); i++) {  
            for (int j = 0; j < sheet.getColumns(); j++) {  
                Cell cell = sheet.getCell(j, i);  
                System.out.print(" " + cell.getContents());  
            }  
            System.out.println();  
        }  
    } catch (IOException | BiffException e) {  
        System.out.println("실패 " + e.getMessage());  
    }  
}
```



Apache POI



- Apache POI는 Apache Software 재단에서 만든 Library
- Microsoft Office File Format을 순수 JAVA 언어로서 읽고 쓰는 기능을 제공
- Apache POI는 Excel File과 관련된 내용을 처리하기 위해 다른 많은 Open Sourced Library 중에서 신뢰할 수 있는 Library
- 또한 Apache POI Library를 사용하여 MS Word 및 MS PowerPoint File을 읽고 쓸 수 있음



Apache POI

■ Library 검색 방법

■ Version에 영향을 받음



The screenshot shows the Apache POI website's download page. The browser address bar shows the URL: <https://poi.apache.org/download.html#archive>. The page header includes the Apache Software Foundation logo and the Apache POI logo. A navigation menu on the left lists: Overview, Home, Download (selected), Changelog, Javadocs, Text Extraction, Encryption support, Secure processing, Case Studies, Related projects, Legal, and Apache Wide. The main content area is titled "Apache POI - Download Release Artifacts". It contains a section "Available Downloads" with the text: "This page provides instructions on how to download and verify the Apache POI release artifacts. There are different versions available depending on how stable your code should be." followed by links: "The latest stable release is Apache POI 5.2.5" and "Archives of all prior releases". Below this, it states: "Apache POI releases are available under the Apache License, Version 2.0. See the NOTICE file contained in each release artifact for applicable copyright attribution notices. To ensure that you have downloaded the true release you should verify the integrity of the files using the signatures and checksums available from this page." A blue banner announces "25 November 2023 - POI 5.2.5 available". The text continues: "The Apache POI team is pleased to announce the release of 5.2.5. Featured are a handful of new areas of functionality and numerous bug fixes. A summary of changes is available in the Release Notes. A full list of changes is available in the change log. People interested should also follow the dev list to track progress. The POI source release is listed below. Pre-built versions of all POI components are available in the central Maven repository under Group ID "org.apache.poi" and Version "5.2.5". A section "Source Distribution" lists two files: "poi-src-5.2.5-20231118.tgz" (112 MB, signature (.asc), checksum: SHA-512) and "poi-src-5.2.5-20231118.zip" (116 MB, signature (.asc), checksum: SHA-512). A section "Binary Artifacts" states: "POI 5.2.3 was the last version where we produced a set of poi-bin*.zip and poi-bin*.tgz files. We will continue to publish jars to Maven Central. If you are not using a build tool like Apache Maven or Gradle, you can still find these jars by traversing the directories at <https://repo1.maven.org/maven2/org/apache/poi/>." The footer of the page shows the Windows taskbar with the search bar and various application icons.

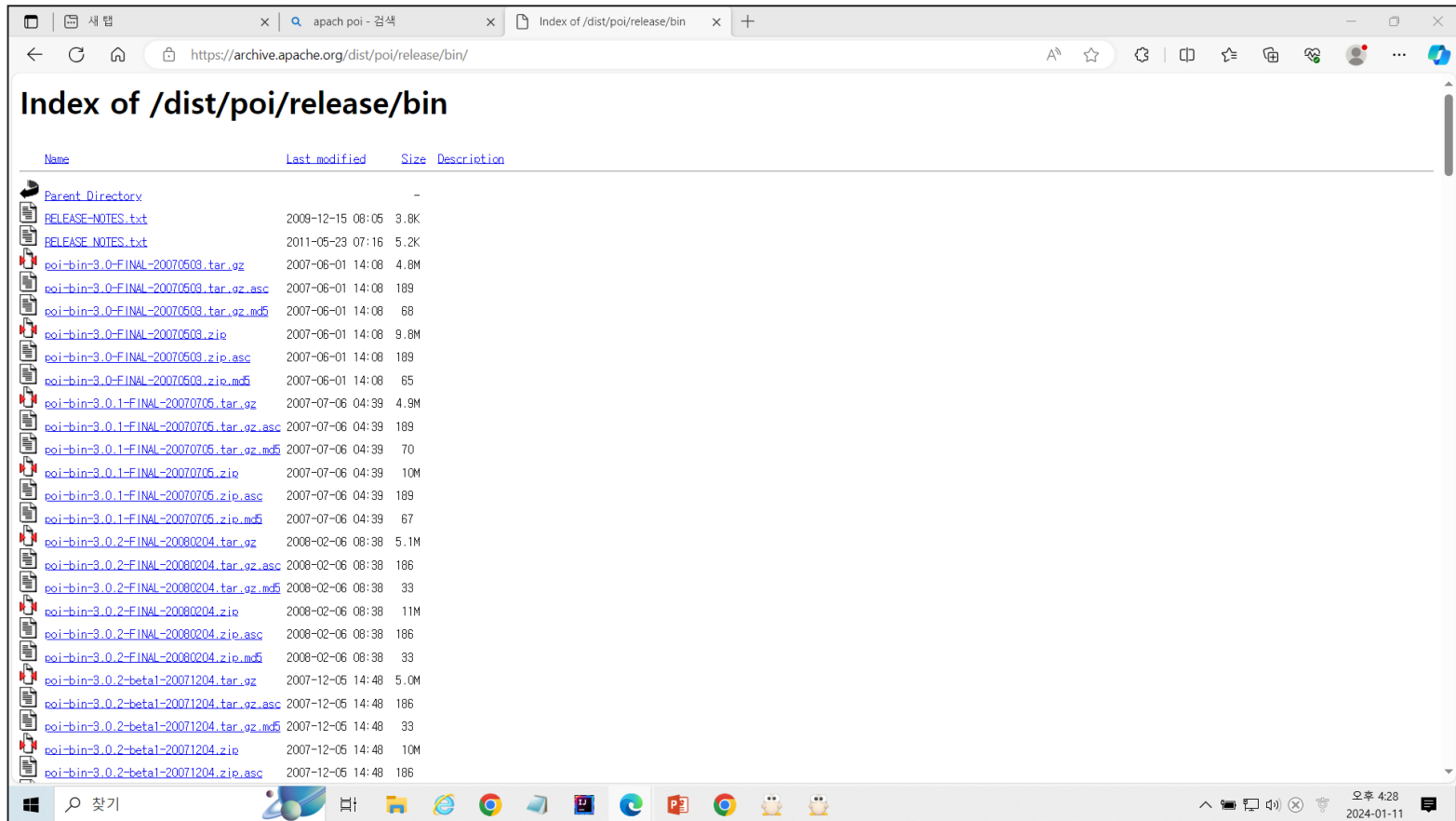




Apache POI

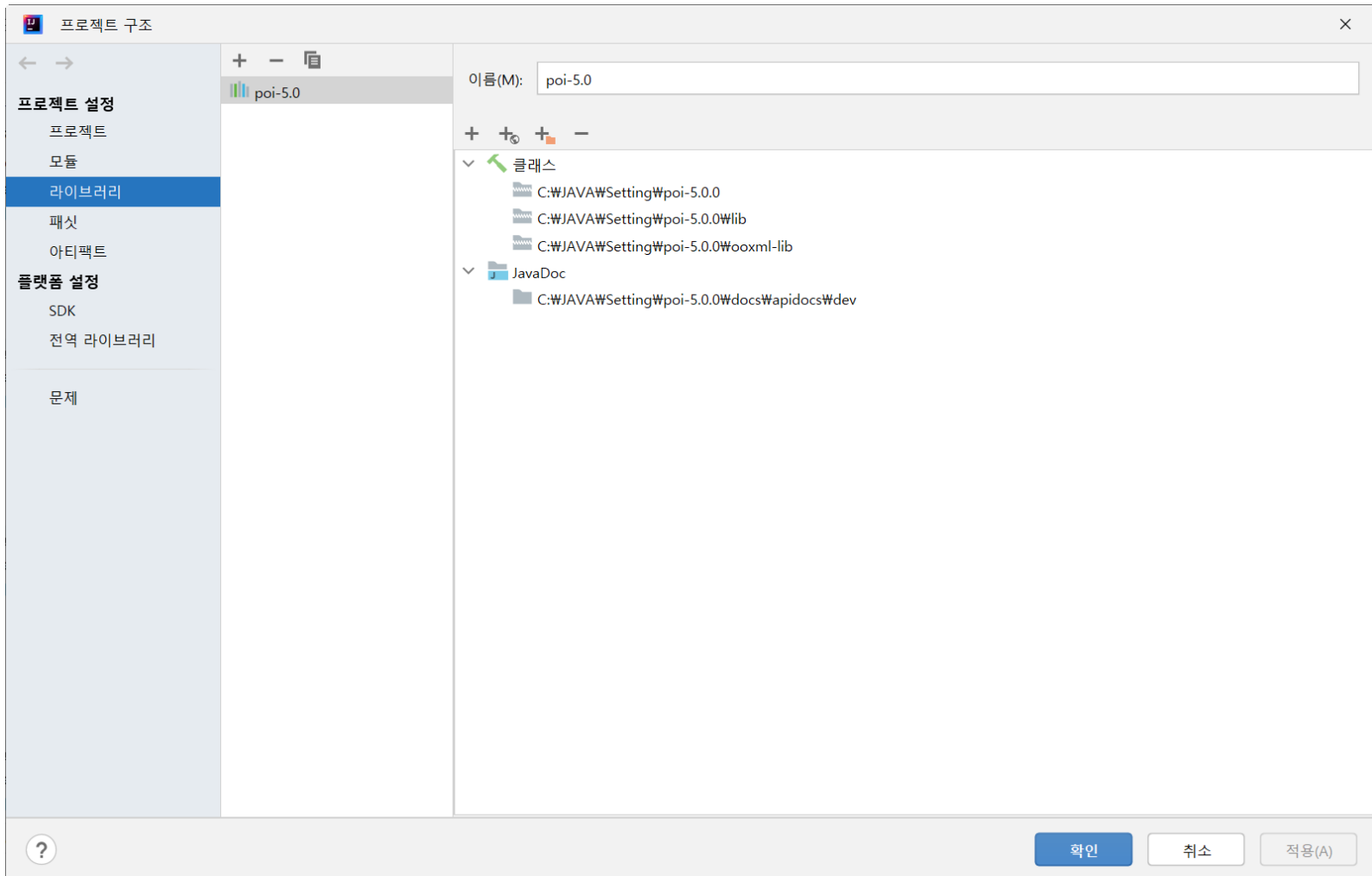
■ Library 검색 방법

■ 마지막을 가면 'Binary Artifacts'를 클릭





Apache POI





Apache POI



■ 핵심 용어

- apache POI의 주요 클래스들은 주로 HSSF, XSSF로 시작

- HSSF

- Excel 97(-2007) File Format 사용할 때 사용

- 예) HSSFWorkbook, HSSFSheet

- XSSF

- Excel 2007 OOXML(*.xlsx) File Format을 사용할 때 사용

- 예) XSSFWorkbook, XSSFSheet

- Workbook, Sheet, Row, Cell

- Workbook은 하나의 Excel File을 의미

- Sheet는 Excel File(= Workbook)의 Sheet를 의미

- Row, Cell은 Sheet안에 있는 행과 열을 의미



Excel 예제 1

```
public class Main {  
    public static String filePath = "..\\data\\";  
    public static String fileName = "test.xls";  
  
    public static void main(String[] args) {  
        Workbook workbook = new XSSFWorkbook();  
        XSSFSheet sheet = (XSSFSheet) workbook.createSheet("employee");  
  
        // Sheet를 채우기 위한 데이터들을 Map에 저장  
        Map<String, Object[]> data = new TreeMap<>();  
        data.put("1", new Object[]{"ID", "NAME", "PHONE_NUMBER"});  
        data.put("2", new Object[]{"1", "cookie", "010-1111-1111"});  
        data.put("3", new Object[]{"2", "sickBBang", "010-2222-2222"});  
        data.put("4", new Object[]{"3", "workingAnt", "010-3333-3333"});  
        data.put("5", new Object[]{"4", "wow", "010-4444-4444"});  
    }  
}
```



Excel 예제 1

```
// data에서 keySet를 가져온다. 이 Set 값들을 조회하면서
// 데이터들을 sheet에 입력한다.

Set<String> keyset = data.keySet();
int rownum = 0;
// 알아야할 점, TreeMap을 통해 생성된 keySet는 for를 조회시,
// 키값이 오름차순으로 조회된다.

for (String key : keyset) {
    Row row = sheet.createRow(rownum++);
    Object[] objArr = data.get(key);
    int cellnum = 0;
    for (Object obj : objArr) {
        Cell cell = row.createCell(cellnum++);
        if (obj instanceof String) {
            cell.setCellValue((String)obj);
        } else if (obj instanceof Integer) {
            cell.setCellValue((Integer)obj);
        }
    }
}
```



Excel 예제 1



```
try {  
    FileOutputStream out = new FileOutputStream(  
                                                new File(filePath, fileName));  
  
    workbook.write(out);  
    out.close();  
    System.out.println("파일 생성 완료");  
} catch (IOException e) {  
    System.out.println(e.getMessage());  
}  
}
```



Excel 예제 1



- Excel File 생성하기
 - workbook을 생성
 - workbook내에 sheet를 생성
 - sheet내에 row를 생성
 - 하나의 row에 여러 개의 cell을 생성
(하나의 행에 여러 열을 생성)
 - 위의 과정을 계속해서 반복



Excel 예제 2



```
public static void main(String[] args) {  
    String path = "..\\data\\data";  
    XSSFWorkbook workbook = new XSSFWorkbook();  
    XSSFSheet sheet = workbook.createSheet("Java Books");  
  
    Object[][] bookData = {  
        {"Head First Java", "Kathy Serria", 79},  
        {"Effective Java", "Joshua Bloch", 36},  
        {"Clean Code", "Robert martin", 42},  
        {"Thinking in Java", "Bruce Eckel", 35},  
    };  
  
    int rowCount = 0;  
    for (Object[] aBook : bookData) {  
        Row row = sheet.createRow(++rowCount);  
        int columnCount = 0;
```



Excel 예제 2



```
for (Object field : aBook) {  
    Cell cell = row.createCell(++columnCount);  
    if (field instanceof String) {  
        cell.setCellValue((String) field);  
    } else if (field instanceof Integer) {  
        cell.setCellValue((Integer) field);  
    }  
}  
  
try (FileOutputStream outputStream = new FileOutputStream(  
    path+ "JavaBooks.xlsx")) {  
    workbook.write(outputStream);  
    outputStream.close();  
    workbook.close();  
    System.out.println("엑셀 파일 생성 완료");  
} catch (IOException e) {  
    System.out.println(e.getMessage());  
}  
}
```




Excel 예제 3



```
try {  
    FileOutputStream outputStream = new FileOutputStream(  
                                                path + "test11.xlsx");  
    workbook.write(outputStream);  
    outputStream.close();  
    workbook.close();  
    System.out.println("엑셀 파일 생성 성공");  
} catch (IOException e) {  
    System.out.println(e.getMessage());  
}  
}
```



Excel 예제 4

```
public static void main(String[] args) {  
    String file = "..\\data\\example.xlsx";  
    try {  
        FileInputStream inputStream = new FileInputStream(file);  
        XSSFWorkbook workbook = new XSSFWorkbook(inputStream);  
        int rowindex;  
        int columnindex;  
        //시트 수 (첫번째에만 존재하므로 0을 준다)  
        //만약 각 시트를 읽기위해서는 FOR문을 한번더 돌려준다  
        XSSFSheet sheet = workbook.getSheetAt(0);  
        int rows = sheet.getPhysicalNumberOfRows(); //행의 수  
        for(rowindex = 0; rowindex < rows; rowindex++){  
            XSSFRow row = sheet.getRow(rowindex); //행을 읽는다  
            if(row != null){  
                int cells = row.getPhysicalNumberOfCells(); //셀의 수
```



Excel 예제 4

```
for(columnindex = 1; columnindex <= cells; columnindex++){  
    XSSFCell cell = row.getCell(columnindex); //셀값을 읽는다  
    String value = "";  
    if(cell == null){ //셀이 빈값일경우를 위한 널체크  
        continue;  
    }else{  
        switch (cell.getCellType()){ //타입별로 내용 읽기  
            case FORMULA:  
                value = cell.getCellFormula();  
                break;  
            case NUMERIC:  
                value = cell.getNumericCellValue()+"";  
                break;  
            case STRING:  
                value = cell.getStringCellValue()+"";  
                break;
```



Excel 예제 4



```
        case BLANK:
            value = cell.getBooleanCellValue()+"";
            break;
        case ERROR:
            value = cell.getErrorCellValue()+"";
            break;
    }
}
System.out.println(rowIndex+"번 행 : "+columnindex+
                                "번 열 값은: "+value);
}
}
InputStream.close();
workbook.close();
}
}catch(Exception e) {
    System.out.println(e.getMessage());
}
}
```



Excel 예제 4



■ Excel File 읽기

- 이미 있는 Excel File을 사용해서 workbook 인스턴스를 생성
- 해당 workbook에서 원하는 sheet를 가져옴
- sheet내에서 읽고자 하는 행 번호를 지정
- row에 있는 모든 cell을 순회하면서 읽음
- 3과 4의 과정을 sheet 내의 모든 행을 읽을 때까지 반복