



BMS

엑셀 파일

유희수



APACHE POI를 활용한 엑셀 파일

* POI?

HSSF, XSSF, SXSSF 방식이 있으며

MultiSheet, CellStyle 등 쉽게 구현 가능하지만

자바에서 ExcelRow생성, 각 Cell 값 입력 등 약간의 번거로움이 있다.

* Workbook?

> HSSF : 엑셀 2007 이전 버전에서 사용 (.xls)

> XSSF : 엑셀 2007 이후 버전에서 사용 (.xlsx)

> SXSSF : XSSF의 Streaming Version으로 메모리를 적게 사용해 대용량 엑셀 다운로드에서 사용



APACHE POI를 활용한 엑셀 파일

1. poi라이브러리 의존성을 pom.xml 파일에 추가

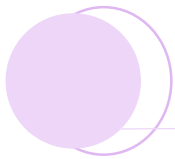
```
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>4.1.1</version>
</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml-schemas</artifactId>
    <version>4.1.1</version>
</dependency>
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>4.1.1</version>
</dependency>
```



APACHE POI를 활용한 엑셀 파일

2. Report1.java

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Date;
import java.text.SimpleDateFormat;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellStyle;
import org.apache.poi.ss.usermodel.FillPatternType;
import org.apache.poi.ss.usermodel.Font;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.util.CellRangeAddress;
import org.apache.poi.hssf.util.HSSFColor.HSSFColorPredefined; //색 이용
import org.apache.poi.ss.usermodel.BorderStyle;
import org.apache.poi.ss.usermodel.HorizontalAlignment;
import org.apache.poi.ss.usermodel.IndexedColors; //이용 못함
import org.apache.poi.ss.usermodel.VerticalAlignment;
import org.apache.poi.xssf.streaming.SXSSFRow;
import org.apache.poi.xssf.streaming.SXSSFSheet;
import org.apache.poi.xssf.streaming.SXSSFWorkbook;
import org.json.simple.JSONObject;
import kr.co.quantumsolution.bms.Main;
```



APACHE POI를 활용한 엑셀 파일

3. 새로운 excel workbook 및 sheet 생성하기

```
private SXSSFSheet s;  
private SXSSFWorkbook wb;
```

```
String filename = "BMS 현황 REPORT.xlsx";
```

```
// 새로운 excel workbook 생성  
wb = new SXSSFWorkbook();  
  
// 새로운 sheet 생성  
s = wb.createSheet("BMS 현황 REPORT");  
Row row = null;  
Cell cell = null;
```



APACHE POI를 활용한 엑셀 파일

4. Font 객체 생성하기

```
Font font = wb.createFont(); // Font객체 생성
font.setFontName("나눔바른고딕 Bold");
font.setBold(true); // 굵게
font.setItalic(true);
font.setFontHeight((short) (20 * 20)); // 폰트 사이즈
font.setColor(HSSFColorPredefined.WHITE.getIndex());
```

원래 색은 이런식으로 지정

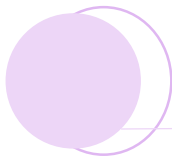
```
// style1.setFillForegroundColor(IndexedColors.GREY_50_PERCENT.index);
```



APACHE POI를 활용한 엑셀 파일

5. Cell 스타일 설정하기

```
CellStyle style1 = wb.createCellStyle(); // 새 style 정의
// 정렬
style1.setAlignment(HorizontalAlignment.CENTER); // 가운데 정렬
style1.setVerticalAlignment(VerticalAlignment.CENTER); // 높이 가운데
// 배경색 지정
style1.setFillForegroundColor(HSSFColorPredefined.GREY_80_PERCENT.getIndex());
style1.setFillPattern(FillPatternType.SOLID_FOREGROUND); //넣어야 배경 적용
// 테두리
style1.setBorderBottom(BorderStyle.THICK); // 바닥 테두리 두껍게
style1.setBorderLeft(BorderStyle.THICK); // 왼쪽 테두리 두껍게
style1.setBorderRight(BorderStyle.THICK); // 오른쪽 테두리 두껍게
style1.setBorderTop(BorderStyle.THICK); // 위쪽 테두리 두껍게
style1.setFont(font); //style1에 font 적용시키기
```



APACHE POI를 활용한 엑셀 파일

6. 셀 병합하기

	0	1	2	3	4	5
	A	B	C	D	E	F
0	<i>BMS 현황 리포트</i>					
1						
2						
3						

```
s.addMergedRegion(new CellRangeAddress(시작줄, 종료줄, 시작행, 종료행));
```

```
for (int i = 0; i <= 3; ++i) {  
    row = s.createRow(i);  
    for (int j = 0; j <= 5; j++) {  
        cell = row.createCell(j);  
        cell.setCellStyle(style1);  
        if (i == 0 && j == 0) {  
            cell.setCellValue("BMS 현황 리포트");  
        }  
    }  
}  
  
s.addMergedRegion(new CellRangeAddress(0, 3, 0, 5));
```




APACHE POI를 활용한 엑셀 파일

7. Cell에 값 입력하기

14				
15	2. MAIN BMS 현황			
16				
17	MAIN BMS 번호	전체 전압 (단위 : V)	전체 전류 (단위 : A)	비고
18	1	0	0	저전압 발생
19	2	0	0	저전압 발생

```
setCellValue(16, 0, "MAIN BMS 번호");
setCellValue(16, 1, "전체 전압 (단위 : V)");
s.addMergedRegion(new CellRangeAddress(16, 16, 1, 2)); //셀 병합
setCellValue(16, 3, "전체 전류 (단위 : A)");
s.addMergedRegion(new CellRangeAddress(16, 16, 3, 4)); //셀 병합
setCellValue(16, 5, "비고");
for (int i = 0; i < 6; i++) {
    cell = getCell(16, i);
    cell.setCellStyle(style3);
} //병합된 셀에서는 for문을 통해 style 적용 가능
```



APACHE POI를 활용한 엑셀 파일

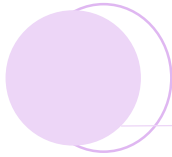
8. 셀 너비 및 높이 설정하기

```
CellStyle red = wb.createCellStyle(); //red 스타일정의
red.setAlignment(HorizontalAlignment.CENTER); // 가운데 정렬
red.setVerticalAlignment(VerticalAlignment.CENTER); // 높이 가운데
red.setWrapText(true); //한 cell에 여러 줄 표시
red.setFont(caution);
```

```
Row r = getRow(6); //7번째 행
r.setHeight((short) 500); //25
```

특정 행 높이 설정하기
단위는 대략 200 = 10point

```
// 셀 너비 자동 조정
for (int i = 0; i < 7; i++) {
    s.trackColumnForAutoSizing(i);
    s.autoSizeColumn(i);
}
```



APACHE POI를 활용한 엑셀 파일

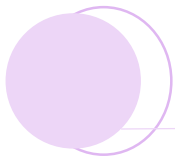
8. 셀 너비 및 높이 설정하기

자동으로 cell 너비 설정

	A	B	C	D	E	F
1	<div>BMS 현황 리포트</div>					
2						
3						
4						
5	1. 일반사항					
6						
7	구분			내용		
8	소회 시간			2020년 02월 06일 16시 01분 24초		
9	배터리 종류			Lithium ion		
10	MAIN BMS 개수			2개		
11	전압 위험 기준			4.2V 이상		
12	저전압 기준			3.4V 이하		
13	온도 위험 기준			0.0℃ 이상		
14						
15	2. MAIN BMS 현황					
16						
17	MAIN BMS 번호	전체 전압 (단위 : V)		전체 전류 (단위 : A)		비고
18	1	0		0		저전압 발생
19	2	0		0		저전압 발생
20						
21	3. 개별 CELL 현황					
22						
23	1) Main BMS #1 _ SUB BMS #1					
24						
25	번호	CELL 번호	전압 (단위 : V)	온도 (단위 : ℃)	내부저항 (단위 : mΩ)	비고
26	1	24	0	0	0	저전압 온도 위험
27	2	23	0	0	0	저전압 온도 위험
28	3	22	0	0	0	저전압 온도 위험

7번째 행 cell 높이 변경

한 cell에 여러 줄 표시



APACHE POI를 활용한 엑셀 파일

9. 엑셀 파일 출력 및 다운로드

```
File file = new File("C:/Users/lims8/Downloads/" + filename);  
FileOutputStream fileOut = new FileOutputStream(file);  
wb.write(fileOut);  
fileOut.close();  
wb.dispose(); //전에 있던 파일이 있다면 삭제  
System.out.println("종료");
```

저장 위치 : user의 다운로드에 저장됨 (이름 : filename)

```
String filename = "BMS 현황 REPORT.xlsx";
```



APACHE POI를 활용한 엑셀 파일

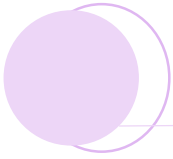
Report1.java

```
public static Row getRow(int i) {  
    Row r = s.getRow(i);  
    if (r == null)  
        r = s.createRow(i);  
    return r;  
}
```

```
public static Cell getCell(int row, int cell) {  
    Row r = getRow(row);  
    Cell c = r.getCell(cell);  
    if (c == null)  
        c = r.createCell(cell);  
    return c;  
}
```

row를 생성한다.
만약 row를 생성하지 않는다면 적용했던
cell들이 나오지 않는다.

cell을 생성한다.
row는 위의 getRow를 통해 받고
row에 일치하는 cell 찾아 생성한다.



APACHE POI를 활용한 엑셀 파일

Report1.java

```
public static void setCellValue(int row, int cell, String cellvalue) {  
    Cell c = getCell(row, cell);  
    c.setCellValue(cellvalue);  
}  
  
public static void setCellValue(int row, int cell, Double cellvalue) {  
    Cell c = getCell(row, cell);  
    c.setCellValue(cellvalue);  
}  
  
public static void setCellValue(int row, int cell, float f) {  
    Cell c = getCell(row, cell);  
    c.setCellValue(f);  
}
```

setCellValue를 통해 cell의 값에 원하는 값을 입력할 수 있다.
형식이 다양해서 int, int ,String / Double/ float 으로 넣었다.



엑셀 데이터 입력 - 조회 시간

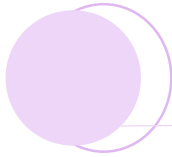
Report1.java

```
Long time = System.currentTimeMillis();
SimpleDateFormat dayTime = new SimpleDateFormat("yyyy년 MM월 dd일 HH시 mm분 ss초");
String now = dayTime.format(new Date(time)); //time에 적용시켜 now로 저장한다.
System.out.println(now);
```

```
for (int i = 3; i < 6; i++) {
    cell = getCell(7, i);
    cell.setCellStyle(style4);
    if (i == 3) {
        cell.setCellValue(now);
    }
}

s.addMergedRegion(new CellRangeAddress(7, 7, 3, 5));
```

병합된 셀에 데이터 값(현재 시간) 넣기



엑셀 데이터 입력 - 조회 시간

조회 시간 출력하기

	A	B	C	D	E	F
1	<div>BMS 현황 리포트</div>					
2						
3						
4						
5	1. 일반사항					
6						
7	구분			내용		
8	조회 시간			2020년 02월 06일 16시 01분 24초		
9	배터리 종류			Lithium ion		
10	MAIN BMS 개수			2개		
11	전압 위험 기준			4.2V 이상		
12	저전압 기준			3.4V 이하		
13	온도 위험 기준			0.0℃ 이상		
14						



엑셀 데이터 입력 - 일반사항 내용

WebSockerServer.java

```
else if (("Report1").equals(command)) {  
    JSONObject Excel = DBManager.selectSiteInfo();  
    Excel.put("cmd", "Report1");  
    Report1.ReportExcel1(Excel);  
}
```

DBManager의 selectSiteInfo 를 실행시키고
ReportExcel1 실행



엑셀 데이터 입력 - 일반사항 내용

DBManager.java

```
public static JSONObject selectSiteInfo() {  
  
    Connection conn = getConnection();  
  
    String sql = "" + " SELECT name, voltage, capacity, date, battery_type ,reqinterval, "  
+ "    cellminvoltage, cellmaxvoltage, cellmintemperature, cellmaxtemperature, "  
+ "    cellminimpedance, cellmaximpedance, mincurrent, maxcurrent, "  
+ "    (SELECT COUNT(*) FROM mbms_conf) AS mbms_count " + " FROM bms_conf" + " LIMIT 0, 1 ";  
  
    JSONObject siteInfo = null;
```

selectSiteInfo 에서 필요한 값인

배터리 종류, MAIN BMS 개수, 전압 위험 기준, 저전압 기준, 온도 위험 기준이 있다.



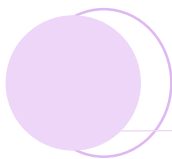
엑셀 데이터 입력 - 일반사항 내용

DBManager.java

```
siteInfo = new JSONObject();

siteInfo.put("name", rs.getString("name"));
// siteInfo.put("type", rs.getString("type"));
siteInfo.put("voltage", rs.getString("voltage"));
siteInfo.put("capacity", rs.getString("capacity"));
siteInfo.put("date", rs.getString("date"));

siteInfo.put("battery_type", rs.getString("battery_type"));
siteInfo.put("reqinterval", rs.getInt("reqinterval"));
siteInfo.put("cellminvoltage", rs.getDouble("cellminvoltage"));
siteInfo.put("cellmaxvoltage", rs.getDouble("cellmaxvoltage"));
siteInfo.put("cellmintemperature", rs.getDouble("cellmintemperature"));
siteInfo.put("cellmaxtemperature", rs.getDouble("cellmaxtemperature"));
siteInfo.put("cellminimpedance", rs.getDouble("cellminimpedance"));
siteInfo.put("cellmaximpedance", rs.getDouble("cellmaximpedance"));
siteInfo.put("mincurrent", rs.getDouble("mincurrent"));
siteInfo.put("maxcurrent", rs.getDouble("maxcurrent"));
siteInfo.put("mbms_count", rs.getInt("mbms_count"));
```

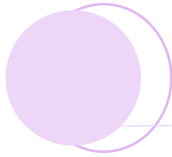


엑셀 데이터 입력 - 일반사항 내용

Report1.java

```
String battery_type = (String) JSONObject.get("battery_type"); // 배터리 종류
double cellminvoltage = (double) JSONObject.get("cellminvoltage"); // 셀전압 최소
double cellmaxvoltage = (double) JSONObject.get("cellmaxvoltage"); // 셀전압 최대
double cellmaxtemperature = (double) JSONObject.get("cellmaxtemperature"); // 셀 온도 최대
double cellmintemperature = (double) JSONObject.get("cellmintemperature"); // 셀 온도 최소
double mincurrent = (double) JSONObject.get("mincurrent"); // 전류 최소
double maxcurrent = (double) JSONObject.get("maxcurrent"); // 전류 최대
int mbms_count = (int) JSONObject.get("mbms_count"); // MAIN BMS 개수
```

JSONObject를 통해 받아온 값들을 변수로 저장한다.



엑셀 데이터 입력 - 일반사항 내용

	A	B	C	D	E	F
1	<div>BMS 현황 리포트</div>					
2						
3						
4						
5	1. 일반사항					
6						
7	구분			내용		
8	조회 시간			2020년 02월 06일 16시 01분 24초		
9	배터리 종류			Lithium ion		
10	MAIN BMS 개수			2개		
11	전압 위험 기준			4.2V 이상		
12	저전압 기준			3.4V 이하		
13	온도 위험 기준			0.0℃ 이상		
14						

DB에서 데이터 가지고 오기 때문에 db에서 데이터 값을 수정하면
자동으로 '내용'에도 바뀌어서 나옴



엑셀 데이터 입력 - 전압/온도/내부저항 값

Report1.java

```
float[][] mbmsInfo = Main.SharedData.nowMbmsInfo;  
float[][] cellVolInfo = Main.SharedData.nowCellVolInfo;  
float[][] cellTempInfo = Main.SharedData.nowCellTempInfo;  
float[][] cellImpInfo = Main.SharedData.nowCellImpInfo;
```

Main SharedDate에서 각 값(전압/전류/온도/내부저항)을 받아온다.

Main.java

```
public static class SharedData {  
    public static JSONObject siteInfo = null;  
    public static JSONObject siteInfo2 = null;  
    public static JSONObject mbmsInfo = null;  
    public static float[][] nowMbmsInfo = null;  
    public static float[][] nowCellVolInfo = null;  
    public static float[][] nowCellTempInfo = null;  
    public static float[][] nowCellImpInfo = null;
```



엑셀 데이터 입력- 전압/온도/내부저항 값

3. 개별 CELL 현황

1) Main BMS #1 _ SUB BMS #1

번호	CELL 번호	전압 (단위 : V)	온도 (단위 : °C)	내부저항 (단위 : mΩ)
1	24	0	0	0
2	23	0	0	0
3	22	0	0	0
4	21	0	0	0
5	20	0	0	0
6	19	0	0	0
7	18	0	0	0
8	17	0	0	0
9	16	0	0	0

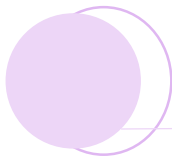
번호 / cell 번호는 받아오는 것이 아닌 지정한 값을 넣었다.

Report1.java

db값을 cell에 넣기

// 3. 개별 CELL 현황 -1) 내용

```
for (int i = 25; i < 37; i++) {  
    setCellValue(i, 0, i - 24);  
    setCellValue(i, 1, 49 - i);  
    setCellValue(i, 2, cellVolInfo[0][48 - i]);  
    setCellValue(i, 3, cellImpInfo[0][48 - i]);  
    setCellValue(i, 4, cellTempInfo[0][48 - i]);  
    for (int k = 0; k < 7; k++) {  
        cell = getCell(i, k);  
        cell.setCellValue(style4);  
    }  
}
```



엑셀 데이터 입력 - 비교

Report1.java

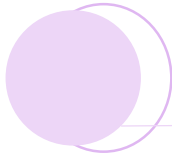
```
for (int i = 41; i < 53; i++) {  
    Cell cell_v = getCell(i, 2);  
    double totalv = cell_v.getNumericCellValue();  
    Cell cell_a = getCell(i, 4);  
    double totala = cell_a.getNumericCellValue();  
    Cell cell_t = getCell(i, 3);  
    double totalt = cell_t.getNumericCellValue();  
    setCellValue(i, 5, "");
```

특정 값보다 크거나 작으면
저전압 발생/전압위험 발생/ 온도 위험 발생 표시

온도 위험 발생은 전압 위험이 발생한다면 -> 한 칸 띄고 작성
전압 위험이 없다면 그 cell에 그대로 온도 위험 발생 표시

cell style은 빨간 색 폰트를 이용한 red로 하고
setWrapText를 이용해서 한 cell에 여러 줄 보이도록 함

```
if (totalv <= cellminvoltage) {  
    setCellValue(i, 5, "저전압 발생");  
}  
if (totalv >= cellmaxvoltage) {  
    setCellValue(i, 5, "전압위험 발생");  
}  
if (totalt > cellmintemperature) {  
    cell = getCell(i, 5);  
    String volt = cell.getStringCellValue();  
    if (volt != null) {  
        setCellValue(i, 5, volt + "\n 온도 위험 발생 ");  
    } else {  
        setCellValue(i, 5, "온도 위험 발생");  
    }  
}  
cell = getCell(i, 5);  
cell.setCellStyle(red);  
red.setWrapText(true);  
}
```

엑셀 데이터 입력 - 비교

비고에서 특정 값 이상이거나 이하일 경우 빨간색으로 표시

3. 개별 CELL 현황

1) Main BMS #1 _ SUB BMS #1

번호	CELL 번호	전압 (단위 : V)	온도 (단위 : °C)	내부저항 (단위 : mΩ)	비고
1	24	0	0	0	저전압 온도 위험
2	23	0	0	0	저전압 온도 위험
3	22	0	0	0	저전압 온도 위험
4	21	0	0	0	저전압 온도 위험
5	20	0	0	0	저전압 온도 위험
6	19	0	0	0	저전압 온도 위험
7	18	0	0	0	저전압 온도 위험
8	17	0	0	0	저전압 온도 위험
9	16	0	0	0	저전압 온도 위험
10	15	0	0	0	저전압 온도 위험



엑셀 데이터 입력 - 사용자가 선택한 기간 (BMS 통계 리포트)

report.js

```
//report2
function Report2(StartDate,EndDate,mbmsNo,sbmscount){
var search = new Object();
search.start = StartDate; // 시작 날짜
search.end = EndDate; // 끝나는 날짜
search.mbmsno = String(mbmsNo); //main bms no
search.sbmscount = String(sbmscount)
search.cmd = 'Report2';
var jsonData = JSON.stringify(search);
QSWebSocket.send(jsonData)
}
```

새로운 오브젝트를 생성한다.

시작날짜/ 끝나는 날짜를 가져온다. (이후 mbmsno와 sbms count도 가져옴)

Report2에 대한 cmd 를 추가한다.

이들을 WebSocket으로 보낸다. (jsonData형식)



엑셀 데이터 입력 - 사용자가 선택한 기간

report.html

```
$(function() {  
    $("#btnExport2").click(function (e) {  
        Report2(StartDate, EndDate, mbmsNo, sbms_count);  
    });  
});
```

사용자가 선택한 기간과, mbmsno 등을 파라미터로 보낸다.



엑셀 데이터 입력 - 사용자가 선택한 기간

WebSockerServer.java

```
JSONParser jsonParser = new JSONParser();  
try {  
    JSONObject jsonObject = (JSONObject) jsonParser.parse(message);  
    String command = (String) jsonObject.get("cmd");  
  
    else if (("Report2").equals(command)) {  
        Report2.ReportExcel2(jsonObject);  
    }
```

cmd가 Report2일 경우 ReportExcel2실행한다.

Report2.java

```
// 기간 가져오기  
String startdate = (String) jsonObject.get("start");  
String enddate = (String) jsonObject.get("end");
```

사용자가 선택한 시작날짜/ 종료날짜를 string으로 저장하고 받아온다.



시작날짜 ~ 종료날짜 가져오기

[illegible]



엑셀 데이터 입력 - MAIN BMS DATA (BMS 통계 리포트)

report.js

```
//report2
function Report2(StartDate,EndDate,mbmsNo,sbmscount){
    var search = new Object();
    search.start = StartDate; // 시작 날짜
    search.end = EndDate; // 끝나는 날짜
    search.mbmsno = String(mbmsNo); //main bms no
    search.sbmscount = String(sbmscount)
    search.cmd = 'Report2';
    var jsonData = JSON.stringify(search);
    QWebSocket.send(jsonData)
```

시작날짜/종료날짜 가져오는 것과 동일하다.



엑셀 데이터 입력 - MAIN BMS DATA (BMS 통계 리포트)

WebSockerServer.java

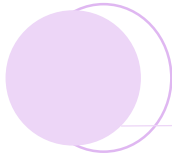
```
JSONParser jsonParser = new JSONParser();  
try {  
    JSONObject jsonObject = (JSONObject) jsonParser.parse(message);  
    String command = (String) jsonObject.get("cmd");  
  
    else if (("Report2").equals(command)) {  
        Report2.ReportExcel2(jsonObject);  
    }  
}
```

시작날짜/종료날짜 가져오는 것과 동일하다.

Report2.java

```
int mbmsno = Integer.parseInt((String) jsonObject.get("mbmsno"));
```

숫자형의 문자열인 mbmsno를 Integer 형으로 반환시킨다.



엑셀 데이터 입력 - MAIN BMS DATA (BMS 통계 리포트)

MAIN BMS DATA 가져오기(MAIN BMS NO 값)

2. MAIN BMS DATA

MAIN BMS NO	전압 (단위 : V)				전류 (단위 : A)				비고
	MAX	DATE	MIN	DATE	MAX	DATE	MIN	DATE	
1	600	2020-02-07 10:39:59	600	2020-02-07 10:39:59	15	2020-02-07 10:39:59	15	2020-02-07 10:39:59	전압위험 발생



엑셀 데이터 입력 - MAIN BMS DATA (BMS 통계 리포트)

```
JSONObject info = new JSONObject();
try {
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery(sql);
    String t = "";
    if (rs.next()) {
        if (rs.getString("totalvoltage") == null) {
            info.put("max_v", null);
            info.put("max_v_time", null);
        } else {
            info.put("max_v", rs.getString("totalvoltage"));
            t = "";
            t = rs.getString("time");
            t = t.substring(0, t.length() - 2);
            info.put("max_v_time", t);
        }
    }
    rs.close();
    rs = stmt.executeQuery(sql2);
    if (rs.next()) {
        if (rs.getString("totalvoltage") == null) {
            info.put("min_v", null);
            info.put("min_v_time", null);
        } else {
            info.put("min_v", rs.getString("totalvoltage"));
            t = "";
            t = rs.getString("time");
            t = t.substring(0, t.length() - 2);
            info.put("min_v_time", t);
        }
    }
    rs.close();
}
```

DBManager.java

```
rs = stmt.executeQuery(sql3);
if (rs.next()) {
    if (rs.getString("current") == null) {
        info.put("max_c", null);
        info.put("max_c_time", null);
    } else {
        info.put("max_c", rs.getString("current"));
        t = "";
        t = rs.getString("time");
        t = t.substring(0, t.length() - 2);
        info.put("max_c_time", t);
    }
}
rs.close();

rs = stmt.executeQuery(sql4);
if (rs.next()) {
    if (rs.getString("current") == null) {
        info.put("min_c", null);
        info.put("min_c_time", null);
    } else {
        info.put("min_c", rs.getString("current"));
        t = "";
        t = rs.getString("time");
        t = t.substring(0, t.length() - 2);
        info.put("min_c_time", t);
    }
}
rs.close();
}
```

DBManager의 seacrch1에서 max, min, time(date) 값을 받는다.



엑셀 데이터 입력 - MAIN BMS DATA (BMS 통계 리포트)

Report2.java

```
// max. date, min. date => data 가져오기
JSONObject bmsInfo2 = DBManager.search1(jsonObject);
Double max_v = Double.parseDouble((String) bmsInfo2.get("max_v"));
String max_v_time = (String) bmsInfo2.get("max_v_time");
Double min_v = Double.parseDouble((String) bmsInfo2.get("min_v"));
String min_v_time = (String) bmsInfo2.get("min_v_time");
Double max_c = Double.parseDouble((String) bmsInfo2.get("max_c"));
String max_c_time = (String) bmsInfo2.get("max_c_time");
Double min_c = Double.parseDouble((String) bmsInfo2.get("min_c"));
String min_c_time = (String) bmsInfo2.get("min_c_time");
```

DBManager의 search1 을 이용해 데이터들을 가져온다.



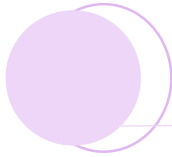
엑셀 데이터 입력 - MAIN BMS DATA (BMS 통계 리포트)

Report2.java

```
// 2. MAIN BMS - 내용

    setCellValue(19, 0, mbmsno);
    s.addMergedRegion(new CellRangeAddress(19, 19, 0, 1));
    setCellValue(19, 2, max_v);
    setCellValue(19, 3, max_v_time);
    setCellValue(19, 4, min_v);
    setCellValue(19, 5, min_v_time);
    setCellValue(19, 6, max_c);
    setCellValue(19, 7, max_c_time);
    setCellValue(19, 8, min_c);
    setCellValue(19, 9, min_c_time);
    for (int i = 0; i < 11; i++) {
        cell = getCell(19, i);
        cell.setCellStyle(style4);
        style4.setWrapText(true);
    }
```

cell 값에 받아온 데이터 값들을 넣어준다.



엑셀 데이터 입력 - MAIN BMS DATA (BMS 통계 리포트)

MAIN BMS DATA 가져오기(MAX, DATE, MIN 값)

2. MAIN BMS DATA

MAIN BMS NO	전압 (단위 : V)				전류 (단위 : A)				비고
	MAX	DATE	MIN	DATE	MAX	DATE	MIN	DATE	
1	600	2020-02-07 10:39:59	600	2020-02-07 10:39:59	15	2020-02-07 10:39:59	15	2020-02-07 10:39:59	전압위험 발생



엑셀 데이터 입력 - 개별 CELL DATA (BMS 통계 리포트)

Report2.java

```
JSONObject.put("cellstartno", "24");  
JSONObject.put("cellno", "24");  
JSONObject cellinfo = DBManager.search2(JSONObject);  
JSONArray celldata = (JSONArray) cellinfo.get("data");
```

celldata 에서 선택한 해당 mbmsno의 개별 cell data에 대한 모든 정보 (max_t, min_t, avg_t 등을 가져온다.)

```
cellinfo["sbmsno":null,"data": [{"cellno":1,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":1,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":2,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":2,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":3,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":3,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":4,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":4,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":5,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":5,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":6,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":6,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":7,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":7,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":8,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":8,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":9,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":9,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":10,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":10,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":11,"imp_usr":"1.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":11,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":12,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":12,"cell_avg_t":"21.3","cell_max_r":"45.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"7.0"}, {"cellno":13,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":13,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":14,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":14,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":15,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":15,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":16,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":16,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":17,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":17,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":18,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":18,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":19,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":19,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":20,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":20,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":21,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":21,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":22,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":22,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":23,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"8.00","sbms_cellno":23,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}, {"cellno":24,"imp_usr":"2.00","imp_prgr":"0.00","cell_min_v":"10.00","sbms_cellno":24,"cell_avg_t":"21.3","cell_max_r":"33.0","cell_avg_v":"11.30","cell_max_t":"23.0","cell_min_r":"30.0","cell_max_v":"13.00","cell_min_t":"20.0"}], "mbmsno": "1", "mbms_name": "Main BMS #1", "order": "desc"}]
```



엑셀 데이터 입력 - 개별 CELL DATA (BMS 통계 리포트)

Report2.java

```
for (int i = 13; i < 25; i++) {  
    int index = celldata.size() - i;  
    JSONObject cells = (JSONObject) celldata.get(index);  
  
    Double cell_max_v = Double.parseDouble((String) cells.get("cell_max_v"));  
    Double cell_min_v = Double.parseDouble((String) cells.get("cell_min_v"));  
    Double cell_avg_v = Double.parseDouble((String) cells.get("cell_avg_v"));  
    Double cell_max_t = Double.parseDouble((String) cells.get("cell_max_t"));  
    Double cell_min_t = Double.parseDouble((String) cells.get("cell_min_t"));  
    Double cell_avg_t = Double.parseDouble((String) cells.get("cell_avg_t"));  
    Double cell_max_r = Double.parseDouble((String) cells.get("cell_max_r"));  
    Double cell_min_r = Double.parseDouble((String) cells.get("cell_min_r"));  
    Double imp_usr = Double.parseDouble((String) cells.get("imp_usr"));
```

index에 따라 값이 다르게 나오기 때문에 for문을 통해 값을 나오게 한다.
각 값을 변수로 저장한다.

셀에 값을 넣어준다.

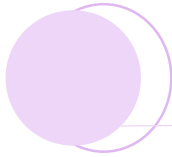
```
setCellValue(31 + i, 1, cell_max_v);  
setCellValue(31 + i, 2, cell_min_v);  
setCellValue(31 + i, 3, cell_avg_v);  
setCellValue(31 + i, 4, cell_max_t);  
setCellValue(31 + i, 5, cell_min_t);  
setCellValue(31 + i, 6, cell_avg_t);  
setCellValue(31 + i, 7, cell_max_r);  
setCellValue(31 + i, 8, cell_min_r);  
setCellValue(31 + i, 9, imp_usr);
```

```
for (int k = 0; k < 10; k++) {  
    cell = getCell(31 + i, k);  
    cell.setCellStyle(style4);  
}
```



1) Main BMS #1_SUB BMS #2

[illegible]



엑셀 데이터 입력 - 개별 CELL DATA (BMS 통계 리포트)

현재 비교 값 넣고 있는 중.

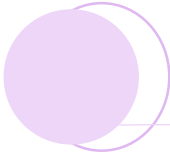


엑셀 데이터 입력 - 개별 CELL DATA (BMS 통계 리포트)

```
for (int j = 44; j < 56; j++) {  
    setCellValue(j, 10, "");  
    if (cell_min_v <= cellminvoltage) {  
        setCellValue(j, 10, "저전압 발생");  
    }  
    if (cell_max_v >= cellmaxvoltage) {  
        setCellValue(j, 10, "전압위험 발생");  
    }  
    if (cell_max_t >= cellmaxtemperature) {  
        cell = getCell(j, 10);  
        String volt = cell.getStringCellValue();  
        if (volt != null) {  
            setCellValue(j, 10, volt + "\n 온도 위험 발생 ");  
        } else {  
            setCellValue(j, 10, "온도 위험 발생 ");  
        }  
    }  
    cell = getCell(j, 10);  
    cell.setCellStyle(red);  
    red.setWrapText(true);  
}
```

BMS 현황 리포트와 동일한 방식으로
진행중인데 오류 발생.

- 온도 위험 발생이 표시가 안됨
- 모두 다 값이 안 나옴



참고 및 출처

JXLS, POI JAVA에서 Excel 사용하는 구현 방법 및 종류 비교

출처: <https://ddoriya.tistory.com/entry/JXLS-POI-JAVA에서-Excel-사용하는-구현-방법-및-종류-비교> [또리야 개발하자]

<http://blog.naver.com/PostView.nhn?blogId=ksseo63&logNo=221431479021&parentCategoryNo=&categoryNo=37&viewDate=&isShowPopularPosts=true&from=search>

POI를 사용하여 엑셀 출력하기

<https://offbyone.tistory.com/250>