

Forum:[Hack Forums](#)

Thread: [How to write data from MySql table to Excel using Java](#)

Ashish Bisht - November 14, 2016, 1:54 am

Hi, I have built an application using java programming and I want to write MySQL table data to excel sheet by using Apache's POI libraries and MySQL connector in java. I have explored many mobile development blogs for my previous query on Android Programming & get a satisfactory solution. I hope some of the tech experts in this community can help to figure out my issue on java programming.

FeatureBug - November 14, 2016, 2:28 am

All you need to do is select the rows you need and produce a comma separated string for each one. Then you can write each line to a file and save it as a CSV that can be opened with Excel.

GammaG - November 14, 2016, 3:27 am

Yeah its probably easiest/fastest way to just create a CSV file.

Here a little class that may help you.

```
Code import java.io.BufferedWriter;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
```

```
public class CsvUtil {

    public static String emptyColumn() {
        return "\"\", \"\"";
    }

    public static String quotedColumn(String content) {
        return "\"" + content + "\"";
    }

    public static String quotedColumnAlternative(String content) {
        return content + ",";
    }

    public static String quotedLastColumn(String content) {
        return "\"" + content + "\"\\n";
    }

    public static String emptyColumnSemicolon() {
        return "\"\", \"\"";
    }
}
```

```

}

public static String quotedColumnSemicolon(String content) {
    return "\"" + content + "\";";
}

public static String quotedLastColumnSemicolon(String content) {
    return "\"" + content + "\"\n";
}

public static String removeInvalidCharacters(String content) {
    if (content != null)
        return content.replaceAll("\n", " ").replaceAll("\"", "");
    return "";
}

public static void saveFile(String name, String content) {
    GregorianCalendar calNow = new GregorianCalendar();
    SimpleDateFormat sdf2 = new SimpleDateFormat("yyyyMMddHHmmss");
    String fileTimestamp = "-" + sdf2.format(calNow.getTime());

    try {
        File file = new File(name + fileTimestamp + ".csv");
        // file.getParentFile().mkdirs();
        file.createNewFile();

        BufferedWriter out = null;
        FileOutputStream fos = new FileOutputStream(file);
        out = new BufferedWriter(new OutputStreamWriter(fos, "UTF8"));
        out.write("\u0001");

        out.write(content);
        out.close();

        System.out.println(file.getName() + " has been created.");
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

public static Date getDate(Calendar cal) {
    if (cal != null) {
        return cal.getTime();
    }
    return null;
}
}

```

The usage is like this:

Code // Write the first text line

```
StringBuilder sbOut = new StringBuilder();
sbOut.append(CsvUtil.quotedColumnSemicolon("Mitarbeiter"));
sbOut.append(CsvUtil.quotedColumnSemicolon("Beschreibung"));
sbOut.append(CsvUtil.quotedColumnSemicolon("Datum"));
sbOut.append(CsvUtil.quotedColumnSemicolon("Dauer"));
sbOut.append(CsvUtil.quotedColumnSemicolon("Startzeit"));
sbOut.append(CsvUtil.quotedColumnSemicolon("Ende"));
sbOut.append(CsvUtil.quotedColumnSemicolon("Interne Notiz"));
sbOut.append(CsvUtil.quotedColumnSemicolon("Pruefung"));
sbOut.append("\n");
```

```
//the other lines
```

```
//save
```

```
CsvUtil.saveFile("Leistungsnachweis", sbOut.toString());
```