201432360
YANGJUNGYO

Accessing Excel Files with JExcelAPI in Java

An Introduction to Reading and Writing Excel Files

Table of Contents

- Introduction to JExcelAPI
- Installation Guide
- Key Classes and Methods
- Workbook Class Overview
- WritableWorkbook Class Overview
- WritableSheet Class Overview
- Writing to an Excel File
- Reading from an Excel File
- Special Features
- Conclusion

Introduction to JExcelAPI

- **Description:**

• JExcelAPI (Java Excel API) is a Java library for reading, writing, and modifying Excel spreadsheets.

• It supports the older binary Excel format (.xls).

• Lightweight and easy to use.

- **URL:** JExcelAPI

Installation Guide

Steps to Install:

1.Download jxl.jar from JExcelAPI

2.Add jxl.jar to your project's classpath.

3.Example for adding to classpath in Eclipse:

- Right-click on project > Build Path > Configure Build Path > Libraries > Add External JARs > Select jxl.jar.

Key Classes and Methods

- **Primary Classes:**
- Workbook
- WritableWorkbook
- WritableSheet
- Label
- Number

- **Important Methods:**
- createWorkbook(File file)
- getWorkbook(File file)
- createSheet(String sheetName, int index)
- addCell(WritableCell cell)

Workbook Class Overview

- **Workbook Class:**
- createWorkbook(File file)

```
File file = new File("example.xls");
Workbook workbook = Workbook.getWorkbook(file);
```

- getWorkbook(File file)

```
File file = new File("example.xls");
WritableWorkbook workbook = Workbook.createWorkbook(file);
```

WritableWorkbook Class Overview

- **WritableWorkbook Class:**
- createSheet(String sheetName, int index)

```
WritableSheet sheet = workbook.createSheet("Sheet1", 0);
```

WritableSheet Class Overview

- WritableSheet Class
- addCell(WritableCell cell)

```
Label label = new Label(0, 0, "Hello World");
sheet.addCell(label);

Number number = new Number(1, 0, 123.45);
sheet.addCell(number);
```

Writing to an Excel File

```
import java.io.File;
import java.io.IOException;
import java.util.Locale;
import jxl.Workbook;
import jxl.WorkbookSettings;
import jxl.write.Label;
import jxl.write.Number;
import jxl.write.WritableSheet;
import jxl.write.WritableWorkbook;
import jxl.write.WriteException;

public class WriteExcel {
    private String inputFile;

    public void setOutputFile(String inputFile) {
        this.inputFile = inputFile;
    }

    public void write() throws IOException, WriteException {
        File file = new File(inputFile);
        WorkbookSettings wbSettings = new WorkbookSettings();
        wbSettings.setLocale(new Locale("en", "EN"));
        WritableWorkbook workbook = Workbook.createWorkbook(file, wbSetting
s);
        WritableSheet sheet = workbook.createSheet("Sheet1", 0);

        Label label = new Label(0, 0, "Hello World");
        sheet.addCell(label);

        Number number = new Number(1, 0, 123.45);
        sheet.addCell(number);

        workbook.write();
        workbook.close();
    }

    public static void main(String[] args) throws IOException, WriteExcepti
on {
        WriteExcel test = new WriteExcel();
        test.setOutputFile("example.xls");
        test.write();
        System.out.println("Excel file created successfully!");
    }
}
```

Reading from an Excel File

```
import java.io.File;
import java.io.IOException;
import jxl.Cell;
import jxl.Sheet;
import jxl.Workbook;
import jxl.read.biff.BiffException;

public class ReadExcel {
    public static void main(String[] args) throws IOException, BiffExceptio
n {
        Workbook workbook = Workbook.getWorkbook(new File("example.xls"));
        Sheet sheet = workbook.getSheet(0);
        Cell cell = sheet.getCell(0, 0);
        System.out.println(cell.getContents());
        workbook.close();
    }
}
```

Special Features

- **Formatting:**
- Customize fonts, colors, borders.
- **Formulas:**
- Add and evaluate formulas in cells.
- **Data Validation:**
- Restrict input values.
- **Example of adding a formula:**

```
import jxl.write.Formula;
// ...
Formula formula = new Formula(2, 0, "A1 + B1");
sheet.addCell(formula);
```

Conclusion

- **Summary:**
- JExcelAPI provides a simple and effective way to handle Excel files in Java.
- Ideal for basic Excel file operations.