Excel File Handling

Program for Reading data from excel

```
public class ReadingExcelData {
	@Test
	public void readExcelFile() throws Exception {
	String file = "./files/Student.xlsx";
	FileInputStream fis=new FileInputStream(file);
	XSSFWorkbook workbook=new XSSFWorkbook(fis);
	XSSFSheet s=workbook.getSheet("Student");
	XSSFRow r=s.getRow(0);
	XSSFCell c=r.getCell(0);
	String data = c.toString();
	workbook.close();
	System.out.println(data);
	}
}
```

Enhanced Version of Reading Excel File

```
public class ReadingExcelData {
        @Test
        public void readExcelFile() throws Exception {
        String file = "./files/Student.xlsx";
        FileInputStream fis=new FileInputStream(file);
        XSSFWorkbook workbook=new XSSFWorkbook(fis);
        String data=workbook.getSheet("Student").getRow(0).getCell(0).toString();
        workbook.close();
        System.out.println(data);
        }
}
```

Reading Entire Excel File(All Rows & Columns)

```
public class ReadingExcelData {
    @Test
    public void readExcelFile() throws Exception {
        String file = "./files/Student.xlsx";
        FileInputStream fis = new FileInputStream(file);
        XSSFWorkbook workbook = new XSSFWorkbook(fis);
        String data = "";
        for (int i = 0; i < 5; i++) {
            for (int j = 0; j < 2; j++) {
                data = workbook.getSheet("Student").getRow(i).getCell(j).toString();
                System.out.print(data + " ");
            }
            System.out.println();
        }
        workbook.close();
    }
}
```

Count number of Rows and Columns in Excel Sheet

```
public class ReadingExcelData {
    @Test
    public void readExcelFile() throws Exception {
        String file = "./files/Student.xlsx";
        FileInputStream fis = new FileInputStream(file);
        XSSFWorkbook workbook = new XSSFWorkbook(fis);
        int colomn=workbook.getSheet("Student").getRow(0).getLastCellNum();
        int rows = workbook.getSheet("Student").getPhysicalNumberOfRows();
        System.out.println(colomn+" "+rows);
        workbook.close();
    }
}
```

Print Odd Data in Excel Sheet

```
public class ReadingExcelData {
    @Test
    public void readExcelFile() throws Exception {
        String file = "./files/Student.xlsx";
        FileInputStream fis = new FileInputStream(file);
        XSSFWorkbook workbook = new XSSFWorkbook(fis);
        String data = "";
        int rows = workbook.getSheet("Student").getLastRowNum();
        for (int i = 0; i <= rows; i++) {
            for (int j = 0; j < workbook.getSheet("Student").getRow(i).getLastCellNum();j++) {
                data = workbook.getSheet("Student").getRow(i).getCell(j).toString();
                System.out.print(data + " ");
            }
            System.out.println();
        }
        workbook.close();
    }
}
```

Writing Data in Excel Sheet

```
public class WriteExcel {
    @Test
    public void write() throws Exception {
        XSSFWorkbook workbook = new XSSFWorkbook();
        XSSFSheet sheet = workbook.createSheet("Emp Info");
        Object empData[][] = {
                { "Empid", "Name", "Job" },
                { 101, "David", "Engineer" },
                { 102, "Miller", "Analyst" },
                { 103, "Steve", "PO" },
                { 104, "Smith", "Director" },
                { 105, "Tyagi", "Manager" },
                { 106, "Natraj", "Lead" },
                { 107, "Morgan", "HR" }
            };
        // Using Normal for loop -> 8 rows and 3 columns
        int rows = empData.length;
        int cols = empData[0].length; //
        System.out.println(rows + " " + cols);
        for (int r = 0; r < rows; r++) {
            XSSFRow row = sheet.createRow(r);
            for (int c = 0; c < cols; c++) {
                XSSFCell cell = row.createCell(c);
                Object value = empData[r][c];
                if (value instanceof String)
                    cell.setCellValue((String) value);
                if (value instanceof Integer)
                    cell.setCellValue((Integer) value);
                if (value instanceof Boolean)
                    cell.setCellValue((Boolean) value);
            }
        }
        Date date = new Date();
        String filepath = "./files/" + date + "_employee.xlsx";
        FileOutputStream fos = new FileOutputStream(filepath);
        workbook.write(fos);
        fos.close();
        System.out.println("Employee File Written Successfully!");
    }
}
```

Using Enhanced *for* loop

```
public class WriteExcel {
//      Workbook->Sheet->Rows->Cells
        @Test
        public void write() throws Exception {
                XSSFWorkbook workbook = new XSSFWorkbook();
                XSSFSheet sheet = workbook.createSheet("Emp Info");
                Object empData[][] = {
                                { "Empid", "Name", "Job" },
                                { 101, "David", "Engineer" },
                                { 102, "Miller", "Analyst" },
                                { 103, "Steve", "PO" },
                                { 104, "Smith", "Director" },
                                { 105, "Tyagi", "Manager" },
                                { 106, "Natraj", "Lead" },
                                { 107, "Morgan", "HR" }
                        };
//      Using for...each Loop
        int rowCount = 0;
        for (Object emp[] : empData) {
            XSSFRow row = sheet.createRow(rowCount++);
            int columnCount = 0;
            for (Object value : emp) {
                XSSFCell cell = row.createCell(columnCount++);
                if (value instanceof String)
                    cell.setCellValue((String) value);
                if (value instanceof Integer)
                    cell.setCellValue((Integer) value);
                if (value instanceof Boolean)
                    cell.setCellValue((Boolean) value);
            }
        }
        Date date = new Date();
        String filepath = "./files/" + date + "_employee.xlsx";
        FileOutputStream fos = new FileOutputStream(filepath);
        workbook.write(fos);
        fos.close();
        System.out.println("Employee File Written Successfully!");
    }
}
```

Write data in Excel from HashMap

```
public class WriteExcel {
//	Workbook->Sheet->Rows->Cells
	@Test
	public void write() throws Exception {
		XSSFWorkbook workbook = new XSSFWorkbook();
		XSSFSheet sheet = workbook.createSheet("Student Data");

		Map<String, String> data = new HashMap<String, String>();
		data.put("101", "John");
		data.put("102", "Kim");
		data.put("103", "Steve");
		data.put("104", "David");
		data.put("105", "Mery");
		data.put("106", "Tom");
```

```
        int rownum = 0;
        for (Map.Entry entry : data.entrySet()) {
            XSSFRow row = sheet.createRow(rownum++);
            row.createCell(0).setCellValue((String)entry.getKey());
            row.createCell(1).setCellValue((String)entry.getValue());
        }
        Date date=new Date();
        String filepath="./files/"+date+"_Student.xlsx";
        FileOutputStream fos=new FileOutputStream(filepath);
        workbook.write(fos);
        fos.close();
        System.out.println("Student File Written Successfully!");
    }
}
```

Read Excel Data & Convert to HashMap

```
public class ReadingExcelData {
    @Test
    public void readExcelFile() throws Exception {
        FileInputStream fis = new FileInputStream("./files/Student.xlsx");
        XSSFWorkbook workbook = new XSSFWorkbook(fis);
        XSSFSheet sheet = workbook.getSheet("social");

        int rows = sheet.getLastRowNum();

        Map<String, String> data = new HashMap<String, String>();

//      Reading Data from Excel to HashMap
        for (int r = 0; r <= rows; r++) {
            String key = sheet.getRow(r).getCell(0).getStringCellValue();
            String value = sheet.getRow(r).getCell(1).getStringCellValue();
            data.put(key, value);
        }
//      Read Data from HashMap
        for (Map.Entry<String, String> entry: data.entrySet()) {
            System.out.println(entry.getKey()+"  "+entry.getValue());
        }
    }
}
```