



链滴

# 【类库教程】Java 实现导出 excel 表 POI

作者: [moonce](#)

原文链接: <https://ld246.com/article/1539618077181>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# Java 实现导出excel表 POI

1.首先下载poi-3.6-20091214.jar, 下载地址如下:

[http://download.csdn.net/download/qq\\_31635851/10026048](http://download.csdn.net/download/qq_31635851/10026048)

官网地址: <http://poi.apache.org/>

## 2.Student.java

```
import java.util.Date;

public class Student
{
    private int id;
    private String name;
    private int age;
    private Date birth;

    public Student()
    {
    }

    public Student(int id, String name, int age, Date birth)
    {
        this.id = id;
        this.name = name;
        this.age = age;
        this.birth = birth;
    }

    public int getId()
    {
        return id;
    }

    public void setId(int id)
    {
        this.id = id;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    public int getAge()
    {

```

```

        return age;
    }

    public void setAge(int age)
    {
        this.age = age;
    }

    public Date getBirth()
    {
        return birth;
    }

    public void setBirth(Date birth)
    {
        this.birth = birth;
    }
}

```

### 3.CreateSimpleExcelToDisk.java

```

import java.io.FileOutputStream;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.List;

import org.apache.poi.hssf.usermodel.HSSFCell;
import org.apache.poi.hssf.usermodel.HSSFCellStyle;
import org.apache.poi.hssf.usermodel.HSSFRow;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.HSSFWorkbook;

public class CreateSimpleExcelToDisk
{
    /**
     * @功能: 手工构建一个简单格式的Excel
     */
    private static List getStudent() throws Exception
    {
        List list = new ArrayList();
        SimpleDateFormat df = new SimpleDateFormat("yyyy-mm-dd");

        Student user1 = new Student(1, "张三", 16, df.parse("1997-03-12"));
        Student user2 = new Student(2, "李四", 17, df.parse("1996-08-12"));
        Student user3 = new Student(3, "王五", 26, df.parse("1985-11-12"));
        list.add(user1);
        list.add(user2);
        list.add(user3);

        return list;
    }

    public static void main(String[] args) throws Exception

```

```

{
    // 第一步, 创建一个webbook, 对应一个Excel文件
    HSSFWorkbook wb = new HSSFWorkbook();
    // 第二步, 在webbook中添加一个sheet,对应Excel文件中的sheet
    HSSFSheet sheet = wb.createSheet("学生表一");
    // 第三步, 在sheet中添加表头第0行,注意老版本poi对Excel的行数列数有限制short
    HSSFRow row = sheet.createRow((int) 0);
    // 第四步, 创建单元格, 并设置值表头 设置表头居中
    HSSFCellStyle style = wb.createCellStyle();
    style.setAlignment(HSSFCellStyle.ALIGN_CENTER); // 创建一个居中格式

    HSSFCell cell = row.createCell((short) 0);
    cell.setCellValue("学号");
    cell.setCellStyle(style);
    cell = row.createCell((short) 1);
    cell.setCellValue("姓名");
    cell.setCellStyle(style);
    cell = row.createCell((short) 2);
    cell.setCellValue("年龄");
    cell.setCellStyle(style);
    cell = row.createCell((short) 3);
    cell.setCellValue("生日");
    cell.setCellStyle(style);

    // 第五步, 写入实体数据 实际应用中这些数据从数据库得到,
    List list = CreateSimpleExcelToDisk.getStudent();

    for (int i = 0; i < list.size(); i++)
    {
        row = sheet.createRow((int) i + 1);
        Student stu = (Student) list.get(i);
        // 第四步, 创建单元格, 并设置值
        row.createCell((short) 0).setCellValue((double) stu.getId());
        row.createCell((short) 1).setCellValue(stu.getName());
        row.createCell((short) 2).setCellValue((double) stu.getAge());
        cell = row.createCell((short) 3);
        cell.setCellValue(new SimpleDateFormat("yyyy-mm-dd").format(stu
            .getBirth()));
    }
    // 第六步, 将文件存到指定位置
    try
    {
        FileOutputStream fout = new FileOutputStream("E:/students.xls");
        wb.write(fout);
        fout.close();
    }
    catch (Exception e)
    {
        e.printStackTrace();
    }
}
}

```