



## Работа с Excel в Java через Apache POI

7 августа 2015 в 18:34, Переводы (<https://tproger.ru/category/translations/>)

6 минут 👁 31 067



### Основы работы с библиотекой Apache POI: чтение и запись на примере Excel файла

Рассказывает автор блога [javarevisited.blogspot.ru](https://javarevisited.blogspot.ru) (<https://javarevisited.blogspot.ru>)

Из этой статьи вы сможете узнать о записи и чтении данных из Excel файлов в Java (будет рассмотрен как `xls`, так и `xlsx` формат). Мы будем использовать библиотеку Apache POI и сосредоточимся на работе с типами `String` и `Date`, работа с последним происходит достаточно хитро. Напомню, что работу с числами мы уже рассмотрели (<https://java67.blogspot.sg/2014/09/how-to-read-write-xlsx-file-in-java-apache-poi-example.html>) в другой статье.

Библиотеку `poi-xx.jar` вы можете использовать для всех старых (`xls`, `doc`, `ppt`) файлов Microsoft Office, для новых (`xlsx`, `docx`, `pptx`) вам понадобится

`poi-ooxml-xx.jar`. Очень важно понимать, что к чему относится, т.к. используемые классы тоже разные — для старых расширений это `HSSFWorkbook`, а для новых — `XSSFWorkbook`.

## Подготовка: загрузка библиотек и зависимостей

Конечно, существует достаточно много открытых библиотек, которые позволяют работать с Excel файлами в Java, например, JXL, но мы будем использовать имеющую самый обширный API и самую популярную — Apache POI. Чтобы её использовать, вам

Спасибо за внимательность.

Опечатка уже отправлена нашим редакторам.

нужно скачать `jar` файлы и добавить их через Eclipse вручную, или вы можете предоставить это Maven.

Во втором случае вам нужно просто добавить следующие две зависимости:

```
1 <dependencies>
2   <dependency>
3     <groupId>org.apache.poi</groupId>
4     <artifactId>poi</artifactId>
5     <version>3.12</version>
6   </dependency>
7   <dependency>
8     <groupId>org.apache.poi</groupId>
9     <artifactId>poi-ooxml</artifactId>
10    <version>3.12</version>
11  </dependency>
12 </dependencies>
```

Самое удобное в Maven — что он загрузит не только указанные `poi.jar` и `poi-ooxml.jar`, но и все `jar` файлы, которые используются внутри, то есть `xmlbeans-2.6.0.jar`, `stax-api-1.0.1.jar`, `poi-ooxml-schemas-3.12.jar` и `commons-codec-1.9.jar`.

Если вы будете добавлять библиотеки вручную — не забудьте о вышеназванных файлах. Скачать всё можно отсюда (<https://www.apache.org/dyn/closer.cgi/poi/release/bin/poi-bin-3.12-20150511.zip>). Помните — если вы загрузите только `poi-XX.jar`, то ваш код скомпилируется без ошибок, но потом упадёт с `java.lang.NoClassDefFoundError: org/apache/xmlbeans/XmlObject`, так как внутри будет вызываться `xmlbeans.jar`.

## Запись

В этом примере мы запишем в `xls` файл следующие данные: в первую ячейку — строку с именем, а во вторую — дату рождения. Вот пошаговая инструкция:

- Создаём объект `HSSFWorkbook`;
- Создаём лист, используя на объекте, созданном в предыдущем шаге, `createSheet()`;
- Создаём на листе строку, используя `createRow()`;
- Создаём в строке ячейку — `createCell()`;
- Задаём значение ячейки через `setCellValue()`;
- Записываем `workbook` в `File` через `FileOutputStream`;
- Закрываем `workbook`, вызывая `close()`.



Спасибо за внимательность.

Для записи строк или чисел этого вполне достаточно, но чтобы записать дату, нам понадобится сделать ещё кое-что:

редактором.

- Создать `DateFormat`;

- Создать `CellStyle` ;
- Записать `DateFormat` в `CellStyle` ;
- Записать `CellStyle` в ячейку;
- Теперь в эту ячейку можно записать объект `Date` через всё тот же `setCellValue` ;
- Чтобы дата поместилась в ячейку, нам нужно добавить столбцу свойство автоматически менять размер: `sheet.autoSizeColumn(1)` .

Всё вместе это будет выглядеть так:

```

1  @SuppressWarnings("deprecation")
2      public static void writeToExcel(String file) throws FileNotFoundException, IOException {
3          Workbook book = new HSSFWorkbook();
4          Sheet sheet = book.createSheet("Birthdays");
5
6          // Нумерация начинается с нуля
7          Row row = sheet.createRow(0);
8
9          // Мы запишем имя и дату в два столбца
10         // имя будет String, а дата рождения --- Date,
11         // формата dd.mm.yyyy
12         Cell name = row.createCell(0);
13         name.setCellValue("John");
14
15         Cell birthdate = row.createCell(1);
16
17         DateFormat format = book.createDateFormat();
18         CellStyle dateStyle = book.createCellStyle();
19         dateStyle.setDataFormat(format.getFormat("dd.mm.yyyy"));
20         birthdate.setCellStyle(dateStyle);
21
22
23         // Нумерация лет начинается с 1900-го
24         birthdate.setCellValue(new Date(110, 10, 10));
25
26         // Меняем размер столбца
27         sheet.autoSizeColumn(1);
28
29         // Записываем всё в файл
30         book.write(new FileOutputStream(file));
31         book.close();
32     }

```

## Чтение

Теперь мы считаем из только что созданного файла то, что мы туда записали.

- Для начала создадим `HSSFWorkbook` , передав в конструктор `FileInputStream` ;
- Получаем лист, передавая в `getSheet()` его номер или название;
- Получаем строку, используя `getRow()` ;
- Получаем ячейку, используя `getCell()` ;
- Узнаём тип ячейки, используя на ней `getCellType()` ;



Спасибо за внимательность.

Опечатка уже отправлена нашим редакторам.

- В зависимости от типа ячейки, читаем её значения, используя `getStringCellValue()`, `getNumericCellValue()` или `getDateCellValue()`;
- Закрываем `workbook` используя `close()`.

Напомню, что дату Excel хранит как числа, т.е. тип ячейки всё равно будет `CELL_TYPE_NUMERIC`.

В виде кода это будет выглядеть следующим образом:

```
1 public static void readFromExcel(String file) throws IOException{
2     HSSFWorkbook myExcelBook = new HSSFWorkbook(new FileInputStream(file));
3     HSSFSheet myExcelSheet = myExcelBook.getSheet("Birthdays");
4     HSSFRow row = myExcelSheet.getRow(0);
5
6     if(row.getCell(0).getCellType() == HSSFCell.CELL_TYPE_STRING){
7         String name = row.getCell(0).getStringCellValue();
8         System.out.println("name : " + name);
9     }
10
11     if(row.getCell(1).getCellType() == HSSFCell.CELL_TYPE_NUMERIC){
12         Date birthdate = row.getCell(1).getDateCellValue();
13         System.out.println("birthdate :" + birthdate);
14     }
15
16     myExcelBook.close();
17
18 }
```

## В заключение

Как уже упоминалось выше, чтение из `xlsx` файлов ничем принципиально не отличается — нужно только вместо `HSSFWorkbook`, `HSSFSheet`, `HSSFRow` (и прочих) из `poi-XX.jar` использовать `XSSFWorkbook`, `XSSFSheet`, `XSSFRow` из `poi-ooxml-XX.jar`. Это всё, что вам нужно знать для чтения и записи в файлы Excel. Разумеется, с помощью библиотеки Apache POI вы можете сделать гораздо больше, но эта статья должна помочь вам быстрее в ней освоиться.

Перевод статьи «How to Read Write Excel file in Java — POI Example» (<https://javarevisited.blogspot.ru/2015/06/how-to-read-write-excel-file-java-poi-example.html>)

Наши тесты для вас:

- Что вы знаете о работе мозга? (<https://goo.gl/SBKoVY>)
- Найдите уязвимость в веб-проекте. (<https://goo.gl/zroiY>)

API (<https://tproger.ru/tag/api/>), Java (<https://tproger.ru/tag/java/>)

Спасибо за внимательность.

Опечатка уже отправлена нашим редакторам.



Также рекомендуем:

3 комментария

Ваш комментарий...

Отпр



**Дмитрий Казаков**

при работе возникла одна проблема: в моем Excel-файле данные сохранены как веб-страница, а не как эксель. Каким образом прочитать данный файл?

8 сен 2016 [Комментировать](#)



**Юрік Абдулаєв**

Как узнать что ячейка объединенная с другой ?

4 мар 2016 [Комментировать](#)



**Ваня Ворчуновский**

только у меня close() метод не находит ?

9 дек 2015



**Анна Малашихина**

у меня тоже не находит close для объекта класса HSSFWorkbook (

8 июл 2016 [Ответить](#)



**Тимур Агямов**

у меня всё нашлось

18 дек 2017 [Ответить](#)

Написать комментарий...



Спасибо за внимательность.

Опечатка уже отправлена нашим редакторам.

“

Никого работа программы не удивляет так часто, как ее создателя.

”

## События



6–8 апреля, Москва: 48-часовой хакатон Unblock Blockchain Hackaton

(<https://tproger.ru/events/unblock-blockchain-hackathon/>)



6–7 апреля, Москва: международная Java-конференция JPoint

(<https://tproger.ru/events/jpoint/>)



6–7 апреля, Ульяновск: IT-конференция «Стачка-2018»

(<https://tproger.ru/events/stachka-2018/>)



7 апреля, Санкт-Петербург: конференция IT Trends

(<https://tproger.ru/events/it-trends-spb/>)



11 апреля, Санкт-Петербург: международный IT-форум BIT-2018

(<https://tproger.ru/events/bit-2018-spb/>)



13–15 апреля, Москва: AR/VR-хакатон AVRA Hack

(<https://tproger.ru/events/avra-hack-moscow/>)



13 апреля, Екатеринбург: конференция разработчиков DUMP-2018

Спасибо за внимательность.

Опечатка уже отправлена нашим редакторам.  
(<https://tproger.ru/events/dump-2018/>)

13 апреля, Москва: «Яндекс изнутри: качество поиска», (<https://tproger.ru/events/yandex->