

# JavaFX and Apache

# Parsing .xls files



This PA will require parsing files again. We will be dealing with a different type of file (.xls), but the code will still be very similar to the programs before.

One difference: we have to use a new library to read .xls files, since Java cannot handle .xls files natively

# Learn programming by googling

When we don't know how to use a method, or what method to even use, Google it! Here's what you generally look for:

- code examples
- documentation (remember javadoc?)

For this program, the library we want to use to read .xls is Apache POI. We are given the .jar, but we don't really know how to use it. Let's try googling it?

### Other Packages

| Barikana | Description |
|----------|-------------|
|----------|-------------|

- org.apache.poi.ss.util.CellRangeAddressList
- org.apache.poi.ss.usermodel
- org.apache.poi.wml.usermodel
- org.apache.poi.xslf
- org.apache.poi.xslf.extractor
- org.apache.poi.xssf.model
- org.apache.poi.xssf.model.gom
- org.apache.poi.xssf.usermodel
- org.apache.poi.xssf.util
- org.apache.poi.xssf.usermodel
- org.apache.poi.xssf.dev
- org.apache.poi.xssf.eventusermodel
- org.apache.poi.xssf.extractor
- org.apache.poi.xssf.usermodel
- org.apache.poi.xssf.usermodel
- org.apache.poi.xssf.usermodel
- org.apache.poi.xssf.usermodel.charts
- org.apache.poi.xssf.usermodel.extensions
- org.apache.poi.xssf.usermodel.helpers
- org.apache.poi.xssf.util
- org.apache.poi.xwpt.extractor
- org.apache.poi.xwpt.usermodel
- org.apache.poi.xwpt.usermodel

---

- CellRangeAddressList
- CellRangeAddressList
- CellRangeUtil
- CellStyle
- CellReference
- CellReference
- CellReferenceNameType
- CellValue
- CellValue
- CellValueRecordInterface
- CellWalk
- CellWalkContext
- CertificateEmbeddingOption
- CipherException
- CFHeaderRecord
- CFRecordsAggregate
- CFRecord
- ComparisonOperator
- ChainingMode
- CharacterProperties
- CharacterPropertyFetcher
- CharacterRun
- CharacterRun
- CharacterSpmCompressor
- CharacterSpmCompressor
- CharacterTextExp
- CharacterTranslator
- Chart
- ChartAxis
- ChartDataFactory
- ChartData
- ChartDataSource
- ChartDataFactory
- ChartEndObjectRecord
- ChartFormatRecord
- ChartRTTimeRecord
- ChartLegend
- ChartRecord
- ChartSeries
- ChartStockBlockRecord
- ChartTObjectRecord
- ChartSubstreamRecordAggregate
- ChartTitleFormatRecord
- Cnld
- choose
- CPAbstractType
- CPBntTable
- CPFormattedDiskPage
- CPV
- Chunk
- Chunk
- ChunkBookOffsetCommand
- ChunkCommand
- ChunkCommandValue
- ChunkedCiperInputStream
- ChunkedCiperOutputStream
- ChunkedCiperOutputStream
- ChunkUtility
- ChunkFactoryCommandDefinition
- ChunkGroup

```
org.apache.poi.ss.formula.stp
org.apache.poi.ss.formula.constant
org.apache.poi.ss.formula.eval
org.apache.poi.ss.formula.eval.forked
org.apache.poi.ss.formula.function
org.apache.poi.ss.formula.functions
org.apache.poi.ss.formula.ptg
formula package contains binary PTG structures used in Formula
```

formula package contains binary PTG structures used in Formul

# The Apache POI

There's a LOT of information to take in here. For this one, we actually recommend you just google whatever task you want to accomplish.

For example, if you wanted to just read in an .xls file, it might be a better idea to just google “reading in xls file java apache poi” instead of sifting through the documentation.

# Googling advice

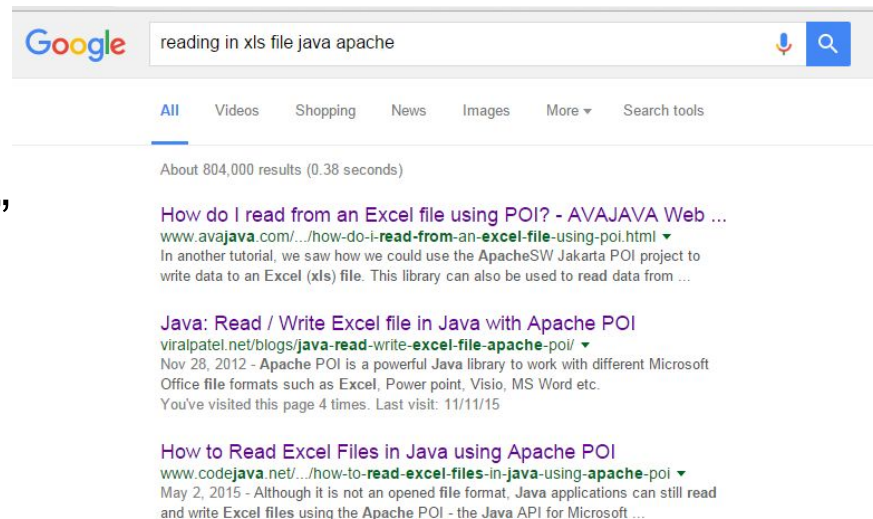
- 1) Search for the specific task you want to accomplish.
- 2) Use specific keywords, leave out “grammar”

Example of a good query:

- “reading in xls file java apache poi”

Example of a bad query:

- “how to get words from prices.xls”



# GUI

Creating and using GUI components will also require a new library. In combination with the Apache library, you will be familiarizing yourselves with JavaFX library.

<https://docs.oracle.com/javase/8/javafx/api/toc.htm>

NOTE: If you have a newer version of Eclipse, you should already have JavaFX. If not, you may have to install it yourself.

# GUI Crash Course

- Components (buttons,etc.) go on the StackPane!
- Set layout for StackPane!
- Scene is initialized using that StackPane!
- Stage sets the scene!
- Stage shows the scene!

For more detail, visit: <http://docs.oracle.com/javase/8/javase-clienttechnologies.htm>

# IDE (optional)

From here on out, use of an IDE (Integrated Development Environment) is now an option. These are programs that are built to make you write code more efficiently.

They usually have features that most dedicated text editors don't. For example, Eclipse has:

- Compiler
- Debugger
- Plugins (JUnit anyone?)
- Javadocs
- ...and much more!

# “I don’t want to use an IDE”

YOU CAN STILL USE EMACS/VIM IF YOU WANT.

IDEs are a very powerful tool, but come with their own learning curves. However, simply coding and running programs on an IDE is still easier than using vim/emacs. We recommend you get exposure to it now, and make your decision.



**QUESTIONS?!?!?**