



(/)

Q Search

 Login (/login)

crealytics (/crealytics) / spark-excel (/crealytics/spark-excel)

A Spark plugin for reading Excel files via Apache POI

GitHub  (<https://github.com/crealytics/spark-excel>)

[spark \(/search?topics=spark\)](/search?topics=spark) [data-frame \(/search?topics=data-frame\)](/search?topics=data-frame) [excel \(/search?topics=excel\)](/search?topics=excel)
[scala \(/search?topics=scala\)](/search?topics=scala)

Version Matrix (/artifacts/crealytics/spark-excel)

scala 2.12 ▼

spark-excel ▼

0.13.7 ▼

(<https://github.com/crealytics/spark-excel/#spark-excel-library>) Spark Excel Library

A library for querying Excel files with Apache Spark, for Spark SQL and DataFrames.

 CI passing (<https://github.com/crealytics/spark-excel/actions>)  Sonatype Nexus

(https://oss.sonatype.org/content/repositories/public/com/crealytics/spark-excel_2.12/)

 Open in Gitpod (<https://gitpod.io/#https://github.com/crealytics/spark-excel>)

(<https://github.com/crealytics/spark-excel/#co-maintainers-wanted>)

Co-maintainers wanted

Due to personal and professional constraints, the development of this library has been rather slow. If you find value in this library, please consider stepping up as a co-maintainer by leaving a comment here (<https://github.com/crealytics/spark-excel/issues/191>). Help is very welcome e.g. in the following areas:

- Additional features
- Code improvements and reviews
- Bug analysis and fixing
- Documentation improvements
- Build / test infrastructure

(<https://github.com/crealytics/spark-excel/#requirements>)

Requirements

This library requires Spark 2.0+

(<https://github.com/crealytics/spark-excel/#linking>)

Linking

You can link against this library in your program at the following coordinates:

(<https://github.com/crealytics/spark-excel/#scala-212>) Scala 2.12

```
groupId: com.crealytics
artifactId: spark-excel_2.12
version: 0.13.1
```

(<https://github.com/crealytics/spark-excel/#scala-211>) Scala 2.11

```
groupId: com.crealytics  
artifactId: spark-excel_2.11  
version: 0.13.1
```

(<https://github.com/crealytics/spark-excel/#using-with-spark-shell>) Using with Spark shell

This package can be added to Spark using the `--packages` command line option. For example, to include it when starting the spark shell:

(<https://github.com/crealytics/spark-excel/#spark-compiled-with-scala-212>) Spark compiled with Scala 2.12

```
$SPARK_HOME/bin/spark-shell --packages com.crealytics:spark-excel_2.12:0.13.1
```

(<https://github.com/crealytics/spark-excel/#spark-compiled-with-scala-211>) Spark compiled with Scala 2.11

```
$SPARK_HOME/bin/spark-shell --packages com.crealytics:spark-excel_2.11:0.13.1
```

(<https://github.com/crealytics/spark-excel/#features>) Features

This package allows querying Excel spreadsheets as Spark DataFrames
(<https://spark.apache.org/docs/latest/sql-programming-guide.html>).

(<https://github.com/crealytics/spark-excel/#scala-api>) Scala API

Spark 2.0+:

(<https://github.com/crealytics/spark-excel/#create-a-dataframe-from-an-excel-file>)

Create a DataFrame from an Excel file

```
import org.apache.spark.sql._

val spark: SparkSession = ???
val df = spark.read
  .format("com.crealytics.spark.excel")
  .option("dataAddress", "'My Sheet'!B3:C35") // Optional, default: "A1"
  .option("header", "true") // Required
  .option("treatEmptyValuesAsNulls", "false") // Optional, default: true
  .option("setErrorCellsToFallbackValues", "true") // Optional, default: false, where errors
  .option("usePlainNumberFormat", "false") // Optional, default: false, If true, format the c
  .option("inferSchema", "false") // Optional, default: false
  .option("addColorColumns", "true") // Optional, default: false
  .option("timestampFormat", "MM-dd-yyyy HH:mm:ss") // Optional, default: yyyy-mm-dd hh:mm:ss
  .option("maxRowsInMemory", 20) // Optional, default None. If set, uses a streaming reader w
  .option("excerptSize", 10) // Optional, default: 10. If set and if schema inferred, number
  .option("workbookPassword", "pass") // Optional, default None. Requires unlimited strength
  .schema(myCustomSchema) // Optional, default: Either inferred schema, or all columns are St
  .load("Worktime.xlsx")
```

For convenience, there is an implicit that wraps the `DataFrameReader` returned by `spark.read` and provides a `.excel` method which accepts all possible options and provides default values:

```
import org.apache.spark.sql._
import com.crealytics.spark.excel._

val spark: SparkSession = ???
val df = spark.read.excel(
  header = true, // Required
  dataAddress = "'My Sheet'!B3:C35", // Optional, default: "A1"
  treatEmptyValuesAsNulls = false, // Optional, default: true
  setErrorCellsToFallbackValues = false, // Optional, default: false, where errors will be co
  usePlainNumberFormat = false, // Optional, default: false. If true, format the cells witho
  inferSchema = false, // Optional, default: false
  addColorColumns = true, // Optional, default: false
  timestampFormat = "MM-dd-yyyy HH:mm:ss", // Optional, default: yyyy-mm-dd hh:mm:ss[.ffffff
  maxRowsInMemory = 20, // Optional, default None. If set, uses a streaming reader which can
  excerptSize = 10, // Optional, default: 10. If set and if schema inferred, number of rows
  workbookPassword = "pass" // Optional, default None. Requires unlimited strength JCE for o
).schema(myCustomSchema) // Optional, default: Either inferred schema, or all columns are Strin
.load("Worktime.xlsx")
```

If the sheet name is unavailable, it is possible to pass in an index:

```
val df = spark.read.excel(  
  header = true,  
  dataAddress = "0!B3:C35"  
).load("Worktime.xlsx")
```

or to read in the names dynamically:

```
val sheetNames = WorkbookReader( Map("path" -> "Worktime.xlsx")  
                                , spark.sparkContext.hadoopConfiguration  
                                ).sheetNames  
  
val df = spark.read.excel(  
  header = true,  
  dataAddress = sheetNames(0)  
)
```

(<https://github.com/crealytics/spark-excel/#create-a-dataframe-from-an-excel-file-using-custom-schema>)

Create a DataFrame from an Excel file using custom schema

```
import org.apache.spark.sql._  
import org.apache.spark.sql.types._  
  
val peopleSchema = StructType(Array(  
  StructField("Name", StringType, nullable = false),  
  StructField("Age", DoubleType, nullable = false),  
  StructField("Occupation", StringType, nullable = false),  
  StructField("Date of birth", StringType, nullable = false)))  
  
val spark: SparkSession = ???  
val df = spark.read  
  .format("com.crealytics.spark.excel")  
  .option("sheetName", "Info")  
  .option("header", "true")  
  .schema(peopleSchema)  
  .load("People.xlsx")
```

(<https://github.com/crealytics/spark-excel/#write-a-dataframe-to-an-excel-file>)

Write a DataFrame to an Excel file

```
import org.apache.spark.sql._

val df: DataFrame = ???
df.write
  .format("com.crealytics.spark.excel")
  .option("dataAddress", "'My Sheet'!B3:C35")
  .option("header", "true")
  .option("dateFormat", "yy-mm-d") // Optional, default: yy-m-d h:mm
  .option("timestampFormat", "mm-dd-yyyy hh:mm:ss") // Optional, default: yyyy-mm-dd hh:mm:ss.0
  .mode("append") // Optional, default: overwrite.
  .save("Worktime2.xlsx")
```

(<https://github.com/crealytics/spark-excel/#data-addresses>) Data Addresses

As you can see in the examples above, the location of data to read or write can be specified with the `dataAddress` option. Currently the following address styles are supported:

- `B3` : Start cell of the data. Reading will return all rows below and all columns to the right. Writing will start here and use as many columns and rows as required.
- `B3:F35` : Cell range of data. Reading will return only rows and columns in the specified range. Writing will start in the first cell (`B3` in this example) and use only the specified columns and rows. If there are more rows or columns in the DataFrame to write, they will be truncated. Make sure this is what you want.
- `'My Sheet'!B3:F35` : Same as above, but with a specific sheet.
- `MyTable[#A11]` : Table of data. Reading will return all rows and columns in this table. Writing will only write within the current range of the table. No growing of the table will be performed. PRs to change this are welcome.

(<https://github.com/crealytics/spark-excel/#building-from-source>) Building From Source

This library is built with SBT (<http://www.scala-sbt.org/0.13/docs/Command-Line-Reference.html>). To build a JAR file simply run `sbt assembly` from the project root. The build configuration includes support for Scala 2.12 and 2.11.

Scaladoc (https://www.javadoc.io/doc/com.crealytics/spark-excel_2.12/0.13.7)

Sbt Amm Maven Gradle Mill

```
libraryDependencies += "com.crealytics" %% "spark-excel" % "0.13.7"
```

Copy

Try online with Scastie



([https://scastie.scala-lang.org/try?g=com.crealytics&a=spark-](https://scastie.scala-lang.org/try?g=com.crealytics&a=spark-excel&v=0.13.7&t=JVM&sv=2.12.6)

[excel&v=0.13.7&t=JVM&sv=2.12.6](https://scastie.scala-lang.org/try?g=com.crealytics&a=spark-excel&v=0.13.7&t=JVM&sv=2.12.6))

Statistics

👁 38 watchers

(<https://github.com/crealytics/spark-excel/watchers>)

👤 18 Contributors

(<https://github.com/crealytics/spark-excel/graphs/contributors>)

★ 209 Stars

(<https://github.com/crealytics/spark-excel/stargazers>)

🍴 98 Forks

(<https://github.com/crealytics/spark-excel/network>)

🔄 324 Commits

(<https://github.com/crealytics/spark-excel/commits>)

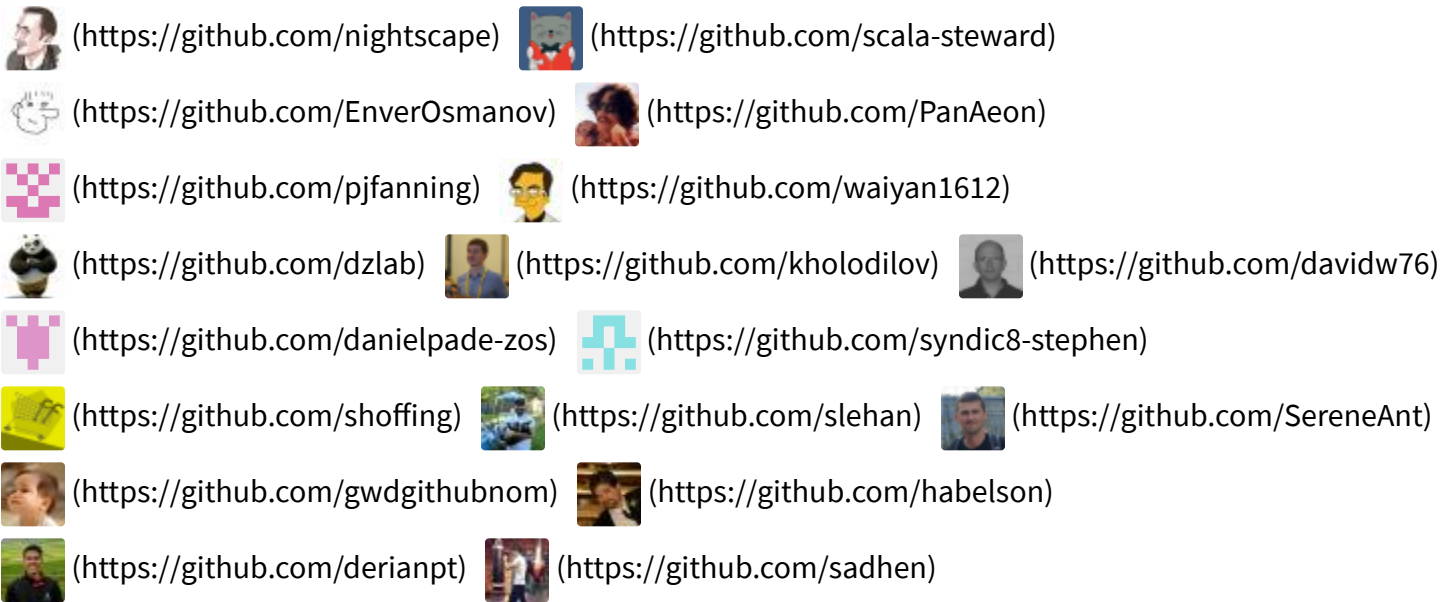
🔔 63 Open issues

(<https://github.com/crealytics/spark-excel/issues>)

💎 59 Releases

🏗 0 Dependents

Contributors



License

Apache-2.0 (<https://spdx.org/licenses/Apache-2.0.html>)

18 Dependencies

Java Dependencies

com.github.pjfanning/excel-streaming-reader

(<http://search.maven.org/#artifactdetails|com.github.pjfanning|excel-streaming-reader|2.3.6|jar>) v2.3.6

compile

com.github.pjfanning/poi-shared-strings

(<http://search.maven.org/#artifactdetails|com.github.pjfanning|poi-shared-strings|1.0.4|jar>) v1.0.4

compile

com.norbitltd/spoiwo_2.12 (http://search.maven.org/#artifactdetails|com.norbitltd|spoiwo_2.12|1.8.0|jar)

v1.8.0 **compile**

org.apache.commons/commons-compress

(<http://search.maven.org/#artifactdetails|org.apache.commons|commons-compress|1.20|jar>) v1.20

compile

org.apache.poi/poi (<http://search.maven.org/#artifactdetails|org.apache.poi|poi|4.1.2|jar>) v4.1.2

compile

org.apache.poi/poi-ooxml (<http://search.maven.org/#artifactdetails|org.apache.poi|poi-ooxml|4.1.2|jar>)

v4.1.2 **compile**

org.scala-lang/scala-library (<http://search.maven.org/#artifactdetails|org.scala-lang|scala-library|2.12.10|jar>)

v2.12.10 **compile**

org.apache.spark/spark-core_2.12 (http://search.maven.org/#artifactdetails|org.apache.spark|spark-core_2.12|2.4.7|jar) v2.4.7 provided

org.apache.spark/spark-hive_2.12 (http://search.maven.org/#artifactdetails|org.apache.spark|spark-hive_2.12|2.4.7|jar) v2.4.7 provided

org.apache.spark/spark-sql_2.12 (http://search.maven.org/#artifactdetails|org.apache.spark|spark-sql_2.12|2.4.7|jar) v2.4.7 provided

org.slf4j/slf4j-api (<http://search.maven.org/#artifactdetails|org.slf4j|slf4j-api|1.7.30|jar>) v1.7.30 provided

com.github.alexarchambault/scalacheck-shapeless_1.14_2.12
(http://search.maven.org/#artifactdetails|com.github.alexarchambault|scalacheck-shapeless_1.14_2.12|1.2.5|jar) v1.2.5 test

com.github.nightscape/spark-testing-base_2.12
(http://search.maven.org/#artifactdetails|com.github.nightscape|spark-testing-base_2.12|c2bc44caf4|jar)
vc2bc44caf4 test

org.scalacheck/scalacheck_2.12
(http://search.maven.org/#artifactdetails|org.scalacheck|scalacheck_2.12|1.15.2|jar) v1.15.2 test

org.scalamock/scalamock-scalatest-support_2.12
(http://search.maven.org/#artifactdetails|org.scalamock|scalamock-scalatest-support_2.12|3.6.0|jar) v3.6.0
test

org.scalatest/scalatest_2.12 (http://search.maven.org/#artifactdetails|org.scalatest|scalatest_2.12|3.2.4|jar)
v3.2.4 test

org.scalatestplus/scalatestplus-scalacheck_2.12
(http://search.maven.org/#artifactdetails|org.scalatestplus|scalatestplus-scalacheck_2.12|3.1.0.0-RC2|jar)
v3.1.0.0-RC2 test

org.typelevel/cats-core_2.12 (http://search.maven.org/#artifactdetails|org.typelevel|cats-core_2.12|2.0.0|jar)
v2.0.0 test

Artifact Dependents (</search?q=depends-on:crealytics/spark-excel>)



0 Dependent

Scaladex

FAQs (<https://github.com/scalacenter/scaladex/wiki/FAQ>)

Report an Issue (<https://github.com/scalacenter/scaladex/issues/new>)

report an issue (<https://github.com/scalacenter/scaladex/issues/new/>)

Follow us:  (https://twitter.com/scala_index)  (<https://github.com/scalacenter/scaladex>)



Scala Center (<https://scala.epfl.ch/>)

Bintray (<https://www.jfrog.com/bintray/>)



Sonatype (<https://www.sonatype.com/>)