spark-csv (homepage (https://github.com/databricks/spark-csv))

Spark SQL CSV data source

@databricks (/user/databricks) / ★★★★★ (👤9)

This packages implements a CSV data source for Apache Spark. CSV files can be read as DataFrame.

Tags

4 csv **2** sql **2** DataSource **1** SparkSQL

How to [+]

Include this package in your Spark Applications using:

spark-shell, pyspark, or spark-submit

```
> $SPARK_HOME/bin/spark-shell --packages com.databricks:spark-csv_2.11:1.4.0
```

Releases

Version: 1.4.0-s_2.11 (cbc72f (https://github.com/databricks/spark-csv/tree/cbc72fe8fbd414f9e732714abdf9388e8cea7e40) | zip (https://github.com/databricks/spark-csv/archive/cbc72fe8fbd414f9e732714abdf9388e8cea7e40.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.11/1.4.0/spark-csv_2.11-1.4.0.jar)) / Date: 2016-03-05 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/cbc72fe8fbd414f9e732714abdf9388e8cea7e40/LICENSE) / Scala version: 2.11

Version: 1.4.0-s_2.10 (cbc72f (https://github.com/databricks/spark-csv/tree/cbc72fe8fbd414f9e732714abdf9388e8cea7e40) | zip (https://github.com/databricks/spark-csv/archive/cbc72fe8fbd414f9e732714abdf9388e8cea7e40.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.10/1.4.0/spark-csv_2.10-1.4.0.jar)) / Date: 2016-03-05 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/cbc72fe8fbd414f9e732714abdf9388e8cea7e40/LICENSE) / Scala version: 2.10

Version: 1.3.0-s_2.10 (84858c (https://github.com/databricks/spark-csv/tree/84858ce5e78c024aa1682dcac82618e24a616f5e) | zip (https://github.com/databricks/spark-csv/archive/84858ce5e78c024aa1682dcac82618e24a616f5e.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.10/1.3.0/spark-csv_2.10-1.3.0.jar)) / Date: 2015-11-20 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/84858ce5e78c024aa1682dcac82618e24a616f5e/LICENSE) / Scala version: 2.10

Spark Scala/Java API compatibility: 1.0.0 - 6% (/release-compatibility/982) , 1.1.0 - 22% (/release-compatibility/981) , 1.2.0 - 26% (/release-compatibility/985) , 1.3.0 - 79% (/release-compatibility/984) , 1.4.0 - 92% (/release-compatibility/980) , 1.5.0 - 100% (/release-compatibility/983)

Version: 1.3.0-s_2.11 (84858c (https://github.com/databricks/spark-csv/tree/84858ce5e78c024aa1682dcac82618e24a616f5e) | zip (https://github.com/databricks/spark-csv/archive/84858ce5e78c024aa1682dcac82618e24a616f5e.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.11/1.3.0/spark-csv_2.11-1.3.0.jar)) / Date: 2015-11-20 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/84858ce5e78c024aa1682dcac82618e24a616f5e/LICENSE) / Scala version: 2.11

Spark Scala/Java API compatibility: 1.2.0 - 26% (/release-compatibility/979) , 1.3.0 - 79% (/release-compatibility/977) , 1.4.0 - 92% (/release-compatibility/976) , 1.5.0 - 100% (/release-compatibility/978)

Version: 1.2.0-s_2.11 (82344b (https://github.com/databricks/spark-csv/tree/82344b9c2d444f734ac04e50b4f9de5459ae62d4) | zip (https://github.com/databricks/spark-csv/archive/82344b9c2d444f734ac04e50b4f9de5459ae62d4.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.11/1.2.0/spark-csv_2.11-1.2.0.jar)) / Date: 2015-08-07 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/82344b9c2d444f734ac04e50b4f9de5459ae62d4/LICENSE) / Scala version: 2.11

Spark Scala/Java API compatibility: 1.2.0 - 43% (/release-compatibility/430) , 1.3.0 - 77% (/release-compatibility/429) , 1.4.0 - 100% (/release-compatibility/428)

Version: 1.2.0-s_2.10 (82344b (https://github.com/databricks/spark-csv/tree/82344b9c2d444f734ac04e50b4f9de5459ae62d4) | zip (https://github.com/databricks/spark-csv/archive/82344b9c2d444f734ac04e50b4f9de5459ae62d4.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.10/1.2.0/spark-csv_2.10-1.2.0.jar)) / Date: 2015-08-07 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/82344b9c2d444f734ac04e50b4f9de5459ae62d4/LICENSE) / Scala version: 2.10

Spark Scala/Java API compatibility: 1.0.0 - 11% (/release-compatibility/426) , 1.1.0 - 38% (/release-compatibility/424) , 1.2.0 - 43% (/release-compatibility/427) , 1.3.0 - 77% (/release-compatibility/425) , 1.4.0 - 100% (/release-compatibility/423)

Version: 1.0.3 (464a3e (https://github.com/databricks/spark-csv/tree/464a3e09529d97cdbb7f74b3d1bb5767c3c27ca2) | zip (https://github.com/databricks/spark-csv/archive/464a3e09529d97cdbb7f74b3d1bb5767c3c27ca2.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.11/1.0.3/spark-csv_2.11-1.0.3.jar)) / Date: 2015-04-04 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/464a3e09529d97cdbb7f74b3d1bb5767c3c27ca2/LICENSE) / Scala version: 2.11

Spark Scala/Java API compatibility: 1.2.0 - 43% (/release-compatibility/108) , 1.3.0 - 100% (/release-compatibility/107)

Version: 1.0.2 (8a2503 (https://github.com/databricks/spark-csv/tree/8a2503c00300c83ca2b6f44957a9f2d4e7a5f95c) | zip (https://github.com/databricks/spark-csv/archive/8a2503c00300c83ca2b6f44957a9f2d4e7a5f95c.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.10/1.0.2/spark-csv_2.10-1.0.2.jar)) / Date: 2015-04-04 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/8a2503c00300c83ca2b6f44957a9f2d4e7a5f95c/LICENSE) / Scala version: 2.10

Spark Scala/Java API compatibility: 1.0.0 - 11% (/release-compatibility/105) , 1.1.0 - 37% (/release-compatibility/103) , 1.2.0 - 43% (/release-compatibility/106) , 1.3.0 - 100% (/release-compatibility/104)

Version: 1.0.0 (074388 (https://github.com/databricks/spark-csv/tree/074388a0d12590224e007899a448f137977ae9e1) | zip (https://github.com/databricks/spark-csv/archive/074388a0d12590224e007899a448f137977ae9e1.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.10/1.0.0/spark-csv_2.10-1.0.0.jar)) / Date: 2015-03-17 / License: Apache-2.0 (https://github.com/databricks/spark-csv/blob/074388a0d12590224e007899a448f137977ae9e1/LICENSE) / Scala version: 2.10

Spark Scala/Java API compatibility: 1.0.0 - 11% (/release-compatibility/39) , 1.1.0 - 37% (/release-compatibility/37) , 1.2.0 - 43% (/release-compatibility/40) , 1.3.0 - 100% (/release-compatibility/38)

Version: 0.1.1 (a0b932 (https://github.com/databricks/spark-csv/tree/a0b932735b24b0b75d076a8fd364d582526de79a) | zip (https://github.com/databricks/spark-csv/archive/a0b932735b24b0b75d076a8fd364d582526de79a.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.10/0.1.1/spark-csv_2.10-0.1.1.jar)) / Date: 2015-01-12 / License: Apache-2.0

Version: 0.1 (09ebf6 (https://github.com/databricks/spark-csv/tree/09ebf6e7ec8777ecbb4debb21fa440c7295af4a9) | zip (https://github.com/databricks/spark-csv/archive/09ebf6e7ec8777ecbb4debb21fa440c7295af4a9.zip) | jar (http://repo1.maven.org/maven2/com/databricks/spark-csv_2.10/0.1/spark-csv_2.10-0.1.jar)) / Date: 2014-12-31 / License: BSD 3-Clause

24 Comments **Spark Packages** **Login**

Recommend 6 **Share** **Sort by Best**

Join the discussion…

Krish Gop • 12 days ago

I tried installing this package along with the spark-redshift on EMR. Do you know, what I need to do to make it work on EMR?

• Reply • Share ›

Jterrell • a month ago

When using Jupyter, whenever I run a write.save it breaks the results into multiple parquet files and stores them in a folder with my path name. I can't see what's in the files because they aren't UTF-8 encoded for Jupyter to show me. Shouldn't I get a csv file somewhere? What am I missing?

• Reply • Share ›

Inam • 2 months ago

I managed to start jupyter notebook with loaded spark-csv by setting this PYSPARK_DRIVER_PYTHON=ipython and this PYSPARK_DRIVER_PYTHON_OPTS=notebook as environment variables and then running This command spark-1.6.1-bin-hadoop2.6\bin\pyspark --packages com.databricks:spark-csv_2.11:1.4.0 --master local[*] it would start my jupyter notebook but when I type in sc in cell I get this ‘ ’.
and when I start my jupyter notebook using this command ipython notebook --profile=pyspark and writes sc it works perfectly and gives main this <pyspark.context.sparkcontext at=""

0x584a490="">
I want to load spark-csv in jupyter notebook.

• Reply • Share ›

Narendra Prasath • 2 months ago

I'm using zeppelin binary package. How to add spark-csv jar in my zeppelin? while using %dep is also not found in my interpreter. Please help me how to add external jars in zeppelin?

• Reply • Share ›

Vicky • 4 months ago

Can any one tell me if I can import spark-csv package from SparkR using R studio under windows 7 environment? My local machine has R, spark-1.6.1-bin-hadoop2.6 and java installed, not maven, scala etc... I don't know if I miss anything in order to call spark-csv. Shall I install this package (.jar file) and put in some folder?? Here is my script:

```
library(rJava)
Sys.setenv(SPARK_HOME = 'C:/Users/***/spark-1.6.1-bin-hadoop2.6')

.libPaths(c(file.path(Sys.getenv('SPARK_HOME'), 'R', 'lib'), .libPaths()))
library(SparkR)

Sys.setenv('SPARKR_SUBMIT_ARGS'='"--packages" "com.databricks:spark-csv_2.11:1.4.0"
"sparkr-shell"')

sc <- sparkR.init(master = "local[*]", sparkEnvir = list(spark.driver.memory="2g"))
sqlContext <- sparkRSQL.init(sc)

flights <- read.df(sqlContext, "nycflights13.csv", "com.databricks.spark.csv", header="true")
```

see more

• Reply • Share ›

Jacques Peeters ➚ Vicky • 3 months ago

Hey i did struggle a little bit with CSV and sparkR, here is what i've done:

- make sure you correctly install the package "$SPARK_HOME/bin/spark-shell --packages com.databricks:spark-csv_2.11:1.4.0" in your shell
- use read.df like this

```
write.csv(iris, file = "iris.csv", row.names = F) #In order to have a correct CSV file
# Now we load it as DataFrame sparkR
irisDF = read.df(sqlContext
, "iris.csv"
, header='true'
, source = "com.databricks.spark.csv"
, encoding = "UTF-8"
)
head(irisDF)
```

Worked like a charm for me.

Reply • Share ›

vangao • 4 months ago

Can anybody tell me the detailed steps to install this package on the linux system?

Reply • Share ›

Max Power • 4 months ago

Hello,

I will use this package in the spark-shell (without maven or eclipse). Can anybody tell me, how i install thispackage on the unix system?
When i start my shell with the command "./bin/spark-shell --jars lib/spark-csv_2.11-1.4.0.jar" it's not working. Thank you

Reply • Share ›

Alexander Sutyagin ➚ Max Power • 4 months ago

Make sure that version of Scala on your system is really 2.11 and not 2.10. In latter case you'd use spark-csv_2.10-1.4.0.jar

Reply • Share ›

Benjamin Kim • 4 months ago

Is there a way to have this package accept a string instead of a file? I have a situation where CSV strings are streamed to us in chunks at a time, each with headers and delimiters inline.

Reply • Share ›

Vineet Menon • 5 months ago

The methods 'save' etc are implemented only for DataFrame. Any idea how to save an RDD without converting the RDD to DF?

Reply • Share ›

a_Mommy1 • 6 months ago

can anyone share the settings wrt CSV import if there are Line Feed in the data itself. So normally a new line is Carriage Return + LineFeed. But because some of the data has Line Feed as a data itself, the import is not happening correctly.

Reply • Share ›

dataPulverizer • 6 months ago

How do we install this package permanently into spark without having to always use --packages ... when running spark-shell? is there a way to include this package into the conf file?

Reply • Share ›

Keith • 8 months ago

What's changed in 1.3?

Reply • Share ›

Hossein Falaki ➚ Keith • 8 months ago

Here you can find the change list: https://github.com/databricks/...

Reply • Share ›

Tuhin Sharma • 9 months ago

Is there a way to pass "--packages com.databricks:spark-csv_2.10:1.2.0" while creating the spark context inside the python code? I believe it can be done while defining SparkConf() object

but how? any help is highly appreciated.

• Reply • Share ›

Christos ➚ Tuhin Sharma • 9 months ago

export PYSPARK_SUBMIT_ARGS="--packages com.databricks:spark-csv_2.10:1.3.0 $PYSPARK_SUBMIT_ARGS"

• Reply • Share ›

Kamil ➚ Christos • 3 months ago

Any comments on what to actually write in the code to achieve this export? Trying everything with no success...

• Reply • Share ›

Max Krakovyan • 9 months ago

Note that the package jar itself depends on univocity-parsers and commons-csv jars. That's in case U r trying to use the pre-loaded version instead of going with --packages param.

• Reply • Share ›

Sundeep P V • 10 months ago

This is cool stuff, but I'm running into a strange issue here in converting a simple JSON to CSV. Using Spark shell, the error I get is "java.lang.UnsatisfiedLinkError: org.apache.hadoop.util.NativeCrc32.nativeComputeChunkedSumsByteArray(II[BI[BIILjava/lang/S I have set up the class path for Hadoop\..\native as we as Hadoop\..\winutils. And this error throws me out of the Spark-shell with this error: "Error: Could not find or load main class org.apache.spark.deploy.SparkSubmit" . Am I missing anything else here, any help is greatly apreciated

• Reply • Share ›

Mo Omer • a year ago

Looks great, but I'm unable to import it for some reason "error: object databricks is not a member of package com". Any ideas? Commands/log starting spark-shell etc. available at https://gist.github.com/momer/...

• Reply • Share ›

Julien Amelot • a year ago

Nice Package for Java/Scala. It would be great to have the Python API covered as well.

  • Reply • Share ›

Yin Huai Julien Amelot • a year ago

@Julien Amelot

You can use it to load csv to DataFrame or save your DataFrame to csv in Python (through our generic load/save methods https://spark.apache.org/docs/....

To load your csv files as a DataFrame, you can use df = sqlContext.load(path=<path of="" your="" csv="" files="">, source="com.databricks.spark.csv").

To save a DataFrame, you can use df.save(path=<path for="" your="" saved="" data="">, source="com.databricks.spark.csv").

For options like "header", you can also add them when you call the load/save (use named parameters).