

# Package ‘sparkavro’

November 10, 2018

**Type** Package

**Title** Load Avro file into 'Apache Spark'

**Version** 0.2.0

**Author** Aki Ariga

**Maintainer** Aki Ariga <chezou@gmail.com>

**Description** Load Avro Files into 'Apache Spark' using 'sparklyr'. This allows to read files from 'Apache Avro' <<https://avro.apache.org/>>.

**License** Apache License 2.0 | file LICENSE

**BugReports** <https://github.com/chezou/sparkavro>

**Encoding** UTF-8

**LazyData** true

**Imports** sparklyr, dplyr, DBI

**RoxygenNote** 6.1.1

**Suggests** testthat

**Language** en-us

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-11-10 11:00:03 UTC

## R topics documented:

spark_read_avro . . . . .	2
spark_write_avro . . . . .	3

<b>Index</b>	<b>4</b>
--------------	----------

---

spark_read_avro	<i>Reads a Avro File into Apache Spark</i>
-----------------	--

---

### Description

Reads a Avro file into Apache Spark using sparklyr.

### Usage

```
spark_read_avro(sc, name, path, readOptions = list(), repartition = 0L,
  memory = TRUE, overwrite = TRUE)
```

### Arguments

sc	An active spark_connection.
name	The name to assign to the newly generated table.
path	The path to the file. Needs to be accessible from the cluster. Supports the "hdfs://", "s3n://" and "file://" protocols.
readOptions	A list of strings with additional options.
repartition	The number of partitions used to distribute the generated table. Use 0 (the default) to avoid partitioning.
memory	Boolean; should the data be loaded eagerly into memory? (That is, should the table be cached?)
overwrite	Boolean; overwrite the table with the given name if it already exists?

### Examples

```
## Not run:
## If you haven't got a Spark cluster, you can install Spark locally like this
library(sparklyr)
spark_install(version = "2.0.1")

sc <- spark_connect(master = "local")
df <- spark_read_avro(
  sc,
  "twitter",
  system.file("extdata/twitter.avro", package = "sparkavro"),
  repartition = FALSE,
  memory = FALSE,
  overwrite = FALSE
)

spark_disconnect(sc)

## End(Not run)
```

---

spark_write_avro	<i>Write a Spark DataFrame to a Avro file</i>
------------------	---

---

**Description**

Serialize a Spark DataFrame to the **Parquet** format.

**Usage**

```
spark_write_avro(x, path, mode = NULL, options = list())
```

**Arguments**

x	A Spark DataFrame or dplyr operation
path	The path to the file. Needs to be accessible from the cluster. Supports the "hdfs://", "s3n://" and "file://" protocols.
mode	Specifies the behavior when data or table already exists.
options	A list of strings with additional options. See <a href="http://spark.apache.org/docs/latest/sql-programming-guide.html#configuration">http://spark.apache.org/docs/latest/sql-programming-guide.html#configuration</a> .

# Index

spark\_read\_avro, 2  
spark\_write\_avro, 3