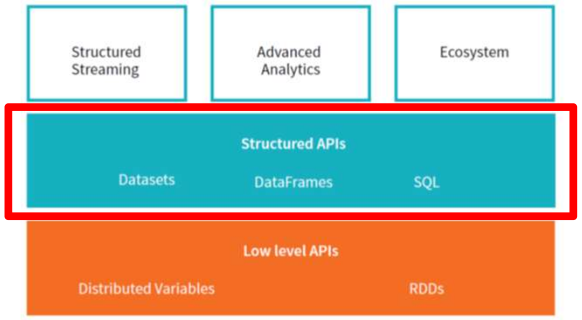


DATASOURCES API

REFERENCES

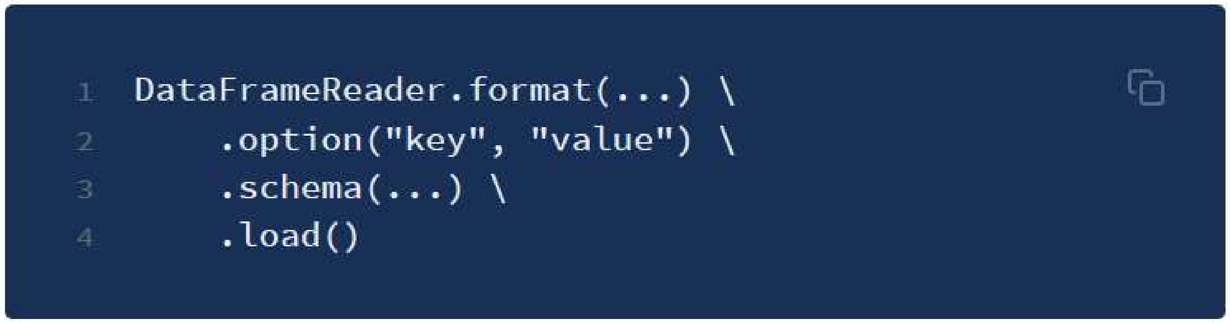
1. Spark: The Definitive Guide

DATASOURCES API

Used to load/save data from/to external sources:

- Csv
- JSON
- Plain-text files
- Parquet
- ORC
- JDBC databases

READ API STRUCTURE

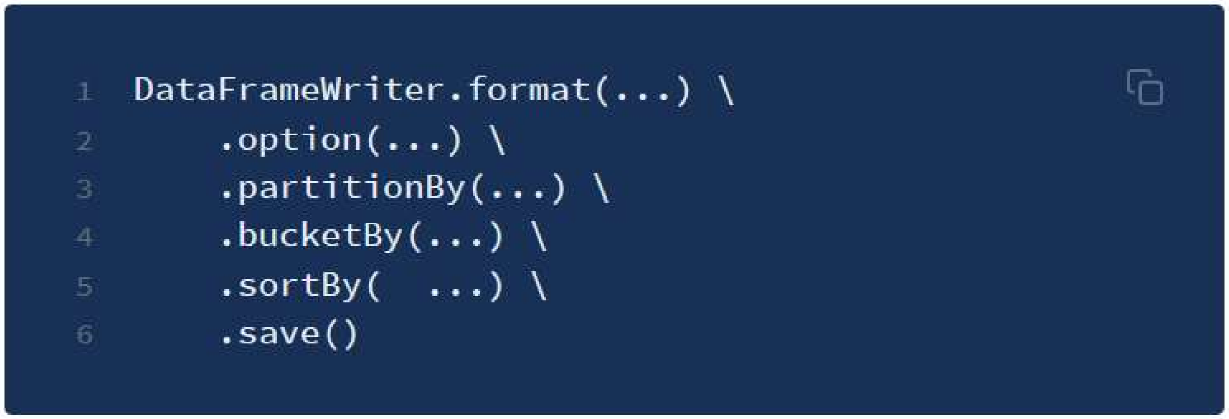


```
DataFrameReader.format(...) \
    .option("key", "value") \
    .schema(...) \
    .load()
```

The option *mode* defines Spark's behavior when a malformed record is fond:

- permissive (default): includes a record with all fields null.
- dropMalformed: discard malformed records
- failFast: fails when a malformed record is found

WRITE API STRUCTURE



```
DataFrameWriter.format(...) \
    .option(...) \
    .partitionBy(...) \
    .bucketBy(...) \
    .sortBy(  ...) \
    .save()
```

The option mode defines Spark’s behavior when it finds data where the DataFrame is supposed to be written

- append
- overwrite
- error or errorifexists (default)
- ignore (ignores the write operation!)

JDBC OPTIONS

Option	Description
url	The JDBC URL to which to connect.
dbtable	The JDBC table to read. Note that anything that is valid in a FROM clause of a SQL query can be used. For example, instead of a full table you could also use a subquery in parentheses.
driver	The class name of the JDBC driver to use to connect to this URL.
partitionColumn, lowerBound, upperBound	If any one of these options is specified, then all others must be set as well. In addition, numPartitions must be specified. These properties describe how to partition the table when reading in parallel from multiple workers. partitionColumn must be a numeric column from the table in question. Notice that lowerBound and upperBound are used only to decide the partition stride, not for filtering the rows in the table. Thus, all rows in the table will be partitioned and returned. This option applies only to reading.
numPartitions	The maximum number of partitions that can be used for parallelism in table reading and writing. This also determines the maximum number of concurrent JDBC connections.

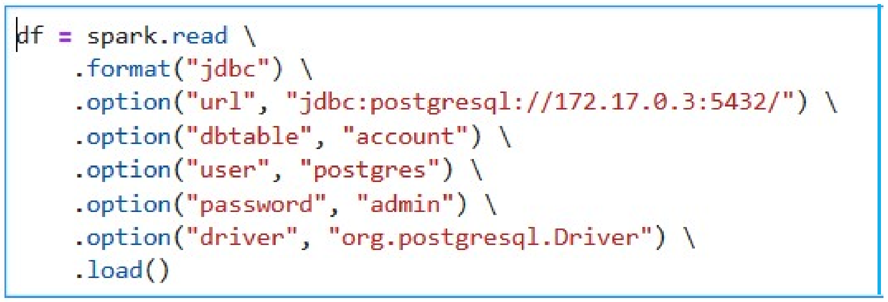
Source: Spark: The Definitive Guide

JDBC OPTIONS

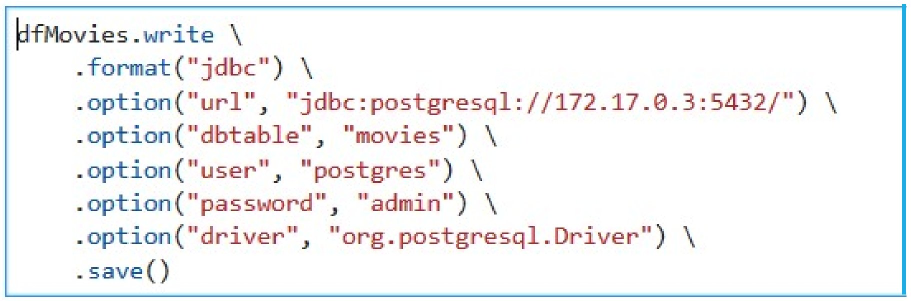
Option	Description
fetchsize	The JDBC fetch size, which determines how many rows to fetch per round trip. This can help performance on JDBC drivers, which default to low fetch size (e.g., Oracle with 10 rows). This option applies only to reading.
batchsize	The JDBC batch size, which determines how many rows to insert per round trip. This can help performance on JDBC drivers. This option applies only to writing. The default is 1000.
isolationLevel	The transaction isolation level, which applies to current connection. It can be one of NONE, READ_COMMITTED, READ_UNCOMMITTED, REPEATABLE_READ, or SERIALIZABLE, corresponding to standard transaction isolation levels defined by JDBC's Connection object. The default is READ_UNCOMMITTED. This option applies only to writing.
truncate	This is a JDBC writer-related option. When SaveMode.Overwrite is enabled, Spark truncates an existing table instead of dropping and re-creating it.

Source: Spark: The Definitive Guide

EXAMPLE: ACCESSING A DATABASE



```
df = spark.read \
    .format("jdbc") \
    .option("url", "jdbc:postgresql://172.17.0.3:5432/") \
    .option("dbtable", "account") \
    .option("user", "postgres") \
    .option("password", "admin") \
    .option("driver", "org.postgresql.Driver") \
    .load()
```



```
dfMovies.write \
    .format("jdbc") \
    .option("url", "jdbc:postgresql://172.17.0.3:5432/") \
    .option("dbtable", "movies") \
    .option("user", "postgres") \
    .option("password", "admin") \
    .option("driver", "org.postgresql.Driver") \
    .save()
```

QUESTIONS???