PySpark Read CSV file into DataFrame

NNK (https://sparkbyexamples.com/author/admin/) - PySpark (https://sparkbyexamples.com/category/pyspark/)

PySpark provides `csv("path")` on DataFrameReader to read a CSV file into PySpark DataFrame and `dataframeObj.write.csv("path")` to save or write to the CSV file. In this tutorial, you will learn how to read a single file, multiple files, all files from a local directory into DataFrame, applying some transformations, and finally writing DataFrame back to CSV file using PySpark example.

PySpark supports reading a CSV file with a pipe, comma, tab, space, or any other delimiter/separator files.

PySpark – Accumulator (https://sparkbyexamples.com/pyspark/pyspark-accumulator-with-example/)

PySpark DataFrame

PySpark – Create a DataFrame (https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/)

PySpark – Create an empty DataFrame (https://sparkbyexamples.com/pyspark/pyspark-create-an-empty-dataframe/)

PySpark – Convert RDD to DataFrame (https://sparkbyexamples.com/pyspark/convert-pyspark-rdd-to-dataframe/)

PySpark – Convert DataFrame to Pandas (https://sparkbyexamples.com/pyspark/convert-pyspark-dataframe-to-pandas/)

PySpark – show() (https://sparkbyexamples.com/pyspark/pyspark-show-display-dataframe-contents-in-table/)

PySpark – StructType & StructField (https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/)

PySpark – Row Class (https://sparkbyexamples.com/pyspark/pyspark-row-using-rdd-dataframe/)

PySpark – Column Class (https://sparkbyexamples.com/pyspark/pyspark-column-functions/)

PySpark – select() (https://sparkbyexamples.com/pyspark/select-columns-from-pyspark-dataframe/)

PySpark – collect() (https://sparkbyexamples.com/pyspark/pyspark-collect/)

PySpark – withColumn() (https://sparkbyexamples.com/pyspark/pyspark-withcolumn/)

Note: PySpark out of the box supports reading files in CSV, JSON, and many more file formats into PySpark DataFrame.

Table of contents:

1. PySpark Read CSV File into DataFrame

Using `csv("path")` or `format("csv").load("path")` of DataFrameReader, you can read a CSV file into a PySpark DataFrame, These methods take a file path to read from as an argument. When you use `format("csv")` method, you can also specify the Data sources by their fully qualified name, but for built-in sources, you can simply use their short names (`csv`,`json`, `parquet`, `jdbc`, `text` e.t.c).

Refer dataset zipcodes.csv at GitHub (https://github.com/spark-examples/pyspark-examples/blob/master/resources/zipcodes.csv)

```
val spark = SparkSession.builde
          .appName("SparkByExam
          .getOrCreate()
df = spark.read.csv("/tmp/resou
df.printSchema()
```

Using fully qualified data source name, you can alternatively do the following.

```
df = spark.read.format("csv")
                .load("/tmp/r
//       or
df = spark.read.format("org.apa
                .load("/tmp/r
df.printSchema()
```

This example reads the data into DataFrame columns `"_c0"` for the first column and `"_c1"` for the second and so on. and by default data type for all these columns is treated as String.

```
root
 |-- _c0: string (nullable = tr
 |-- _c1: string (nullable = tr
 |-- _c2: string (nullable = tr
```

1.1 Using Header Record For Column Names

If you have a header with column names on your input file, you need to explicitly specify `True` for header option using `option("header",True)` not mentioning this, the API treats header as a data record.

```
df2 = spark.read.option("header
      .csv("/tmp/resources/zipco
```

As mentioned earlier, PySpark reads all columns as a string (StringType) by default. I will explain in later sections on how to read the schema (`inferschema`) from the header record and derive the column type based on the data.

1.2 Read Multiple CSV Files

Using the `read.csv()` method you can also read multiple csv files, just pass all file names by separating comma as a path, for example :

```
df = spark.read.csv("path1,path
```

1.3 Read all CSV Files in a Directory

We can read all CSV files from a directory into DataFrame just by passing directory as a path to the `csv()` method.

```
df = spark.read.csv("Folder pat
```

2. Options While Reading CSV File

PySpark CSV dataset provides multiple options to work with CSV files. Below are some of the most important options explained with examples.

You can either use chaining `option(self, key, value)` to use multiple options or use alternate `options(self, **options)` method.

2.1 delimiter

`delimiter` option is used to specify the column delimiter of the CSV file. By default, it is **comma (,)** character, but can be set to any character like **pipe(|)**, **tab (\t)**, **space** using this option.

```
df3 = spark.read.options(delimi
  .csv("C:/apps/sparkbyexamples
```

2.2 inferSchema

The default value set to this option is `False` when setting to `true` it automatically infers column types based on the data. Note that, it requires reading the data one more time to infer the schema.

```
df4 = spark.read.options(inferS
  .csv("src/main/resources/zipc
```

Alternatively you can also write this by chaining `option()` method.

```
df4 = spark.read.option("inferS
                .option("delimi
  .csv("src/main/resources/zipc
```

2.3 header

This option is used to read the first line of the CSV file as column names. By default the value of this option

is `False` , and all column types are assumed to be a string.

```
df3 = spark.read.options(header
  .csv("/tmp/resources/zipcodes
```

2.4 quotes

When you have a column with a delimiter that used to split the columns, use `quotes` option to specify the quote character, by default it is ” and delimiters inside quotes are ignored. but using this option you can set any character.

2.5 nullValues

Using `nullValues` option you can specify the string in a CSV to consider as null. For example, if you want to consider a date column with a value **`"1900-01-01"`** set null on DataFrame.

2.6 dateFormat

`dateFormat` option to used to set the format of the input DateType (https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/#pyspark-sql-date-functions) and TimestampType (https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/#pyspark-sql-timestamp-functions) columns. Supports all `java.text.SimpleDateFormat` formats.

Note: Besides the above options, PySpark CSV API also supports many other options, please refer to this article for details (https://docs.databricks.com/data/data-sources/read-csv.html).

PySpark – struct() (https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/#update-struct-function)

PySpark – countDistinct() (https://sparkbyexamples.com/pyspark/pyspark-count-distinct-from-dataframe/)

PySpark – sum(), avg() (https://sparkbyexamples.com/pyspark/pyspark-dataframe-groupby-and-sort-by-descending-order/)

PySpark – row_number() (https://sparkbyexamples.com/pyspark/pyspark-window-functions/#row_number)

PySpark – rank() (https://sparkbyexamples.com/pyspark/pyspark-window-functions/#rank)

PySpark – dense_rank() (https://sparkbyexamples.com/pyspark/pyspark-window-functions/#dense_rank)

PySpark – percent_rank() (https://sparkbyexamples.com/pyspark/pyspark-window-functions/#percent_rank)

PySpark – typedLit() (https://sparkbyexamples.com/pyspark/pyspark-lit-add-literal-constant/#typedlit)

PySpark – from_json() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#from_json)

PySpark – to_json() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#to_json)

PySpark – json_tuple() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#json_tuple)

PySpark – get_json_object() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#get_json_object)

PySpark – schema_of_json() (https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#schema_of_json)

3. Reading CSV files with a user-specified custom schema

If you know the schema of the file ahead and do not want to use the `inferSchema` option for column names and types, use user-defined custom column names and type (https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/) using `schema` option.

```
schema = StructType() \
      .add("RecordNumber",Integ
      .add("Zipcode",IntegerTyp
      .add("ZipCodeType",String
      .add("City",StringType(),
      .add("State",StringType()
      .add("LocationType",Strin
      .add("Lat",DoubleType(),T
      .add("Long",DoubleType(),
      .add("Xaxis",IntegerType(
      .add("Yaxis",DoubleType()
      .add("Zaxis",DoubleType()
      .add("WorldRegion",String
      .add("Country",StringType
      .add("LocationText",Strin
      .add("Location",StringTyp
      .add("Decommisioned",Bool
      .add("TaxReturnsFiled",St
      .add("EstimatedPopulation
      .add("TotalWages",Integer
      .add("Notes",StringType()

df_with_schema = spark.read.for
      .option("header", True) \
      .schema(schema) \
      .load("/tmp/resources/zip
```

4. Applying DataFrame transformations

Once you have created DataFrame from the CSV file, you can apply all transformation and actions DataFrame support. Please refer to the link for more details.

5. Write PySpark DataFrame to CSV file

Use the `write()` method of the PySpark DataFrameWriter object to write PySpark DataFrame to a CSV file.

```
df.write.option("header",True)
 .csv("/tmp/spark_output/zipcod
```

5.1 Options

While writing a CSV file you can use several options. for example, `header` to output the DataFrame column names as header record and `delimiter` to specify the delimiter on the CSV output file.

```
df2.write.options(header='True'
 .csv("/tmp/spark_output/zipcod
```

Other options available `quote`,`escape`,`nullValue`,`dateFormat`,`quoteMode` .

5.2 Saving modes

PySpark DataFrameWriter also has a method mode() to specify saving mode.

`overwrite` – mode is used to overwrite the existing file.

`append` – To add the data to the existing file.

`ignore` – Ignores write operation when the file already exists.

`error` – This is a default option when the file already exists, it returns an error.

```
df2.write.mode('overwrite').csv
//you can also use this
df2.write.format("csv").mode('o
```

6. PySpark Read CSV Complete Example

```
import pyspark
from pyspark.sql import SparkSe
from pyspark.sql.types import S
from pyspark.sql.types import A
from pyspark.sql.functions impo

spark = SparkSession.builder.ap

df = spark.read.csv("/tmp/resou

df.printSchema()

df2 = spark.read.option("header
     .csv("/tmp/resources/zipco
df2.printSchema()

df3 = spark.read.options(header
  .csv("/tmp/resources/zipcodes
df3.printSchema()

schema = StructType() \
      .add("RecordNumber",Integ
      .add("Zipcode",IntegerTyp
      .add("ZipCodeType",String
      .add("City",StringType(),
      .add("State",StringType()
      .add("LocationType",Strin
      .add("Lat",DoubleType(),T
      .add("Long",DoubleType(),
      .add("Xaxis",IntegerType(
      .add("Yaxis",DoubleType()
      .add("Zaxis",DoubleType()
      .add("WorldRegion",String
      .add("Country",StringType
      .add("LocationText",Strin
      .add("Location",StringTyp
      .add("Decommisioned",Bool
      .add("TaxReturnsFiled",St
      .add("EstimatedPopulation
      .add("TotalWages",Integer
      .add("Notes",StringType()

df_with_schema = spark.read.for
      .option("header", True) \
      .schema(schema) \
      .load(/tmp/resources/zipc
df_with_schema.printSchema()

df2.write.option("header",True)
 .csv("/tmp/spark_output/zipcod
```

7. Conclusion:

In this tutorial, you have learned how to read a CSV file, multiple CSV files and all files from a local folder into PySpark DataFrame, using multiple options to change the default behavior and write CSV files back to DataFrame using different save options.

Happy Learning !!

References:

- Databricks read CSV (https://docs.databricks.com/data/data-sources/read-csv.html)
- PySpark CSV library (https://github.com/databricks/spark-csv)

TAGS: CSV (HTTPS://SPARKBYEXAMPLES.COM/TAG/CSV/), HEADER (HTTPS://SPARKBYEXAMPLES.COM/TAG/HEADER/), PYSPARK WRITE CSV (HTTPS://SPARKBYEXAMPLES.COM/TAG/PYSPARK-WRITE-CSV/), SCHEMA (HTTPS://SPARKBYEXAMPLES.COM/TAG/SCHEMA/)

[NNK (Https://Sparkbyexamples.Com/Author/Admin/)](https://sparkbyexamples.com/author/admin/)

SparkByExamples.com is a Big Data and Spark examples community page, all examples are simple and easy to understand and well tested in our development environment Read more .. (https://sparkbyexamples.com/about-sparkbyexamples/)

THIS POST HAS 8 COMMENTS

Anonymous

18 APR 2021 REPLY

where is zipcode CSV file ?

NNK

18 APR 2021 REPLY

You can find it at zipcodes.csv @ GitHub (https://github.com/spark-examples/pyspark-examples/blob/master/resources/zipcodes.csv)

Anonymous

11 FEB 2021 REPLY

Really very helpful pyspark example..Thanks for the details!!

Anonymous

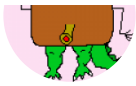
5 JAN 2021 REPLY

Very much helpful!! Thank you for the article!!

NNK

5 JAN 2021 REPLY

Thank you!!



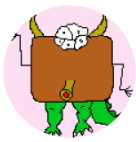
Anonymous

2 NOV 2020 REPLY

Thank you so much for this article ☺

NNK 2 NOV 2020 REPLY

Glad you like it.



Anonymous

19 SEP 2020 REPLY

Thanks for the example. could you please explain how to define/initialise the “spark” in the above example (e.g. spark.read.csv)?

Leave a Reply

Categories

Apache Hadoop (https://sparkbyexamples.com/category/hadoop/)

Apache Spark (https://sparkbyexamples.com/category/spark/)

Apache Spark Streaming (https://sparkbyexamples.com/category/spark/apache-spark-streaming/)

Apache Kafka (https://sparkbyexamples.com/category/kafka/)

Apache HBase (https://sparkbyexamples.com/category/hbase/)

Apache Cassandra (https://sparkbyexamples.com/category/cassandra/)

Snowflake Database (https://sparkbyexamples.com/category/snowflake/)

H2O Sparkling Water (https://sparkbyexamples.com/category/h2o-sparkling-water/)

PySpark (https://sparkbyexamples.com/category/pyspark/)

Recent Posts

Spark regexp_replace() – Replace String Value (https://sparkbyexamples.com/spark/spark-regexp_replace-replace-string-value/)

How to Run a PySpark Script from Python? (https://sparkbyexamples.com/pyspark/run-pyspark-script-from-python-subprocess/)

Spark SQL like() Using Wildcard Example (https://sparkbyexamples.com/spark/spark-sql-like-using-wildcard-example/)

Spark isin() & IS NOT IN Operator Example (https://sparkbyexamples.com/spark/spark-isin-is-not-in-operator-example/)

Spark – Get Size/Length of Array & Map Column (https://sparkbyexamples.com/spark/spark-get-size-length-of-array-map-column/)

Spark Using Length/Size Of a DataFrame Column (https://sparkbyexamples.com/spark/spark-using-length-size-of-a-dataframe-column/)

Spark rlike() Working with Regex Matching Examples (https://sparkbyexamples.com/spark/spark-rlike-regex-matching-examples/)

Spark Check String Column Has Numeric Values (https://sparkbyexamples.com/spark/spark-check-string-column-has-numeric-values/)

Spark Check Column Data Type is Integer or String (https://sparkbyexamples.com/spark/spark-check-column-data-type-is-integer-or-string/)

About SparkByExamples.Com

SparkByExamples.com is a Big Data and Spark examples community page, all examples are simple and easy to understand, and well tested in our development environment Read more .. (https://sparkbyexamples.com/about-sparkbyexamples/)

Follow Us

(https://twitter.com/sparkbyexamples) (https://www.facebook.com/sparkbyexamples) (https://www.linkedin.com/in/n-b860a8193/) (https://github.com/spark-examples/)