Extract column values of Dataframe as List in Apache Spark

Asked 5 years, 11 months ago Active 11 days ago Viewed 240k times



100



38

scala

square brackets.

apache-spark apache-spark-sql

Any suggestions would be appreciated. Thank you!

Share Improve this question Follow

edited Oct 19 '19 at 6:05

I want to convert a string column of a data frame to a list. What I can find from the Dataframe API is RDD, so I tried converting it back to RDD first, and then apply to Array function to the RDD. In this case, the length and

SQL work just fine. However, the result I got from RDD has square brackets around every element like this [A00001]. I was wondering if there's an appropriate way to convert a column to a list or a way to remove the

mrsrinivas

119

asked Aug 14 '15 at 0:39

ways to solve it with Spark 2.x - mrsrinivas May 20 '17 at 10:31

10 Answers

Oldest Active Votes

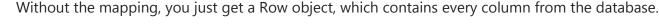


This should return the collection containing single list:

124

dataFrame.select("YOUR COLUMN NAME").rdd.map($r \Rightarrow r(0)$).collect()







Keep in mind that this will probably get you a list of Any type. If you want to specify the result type, you can use .asInstanceOf[YOUR_TYPE] in $r \Rightarrow r(0)$.asInstanceOf[YOUR_TYPE] mapping

P.S. due to automatic conversion you can skip the .rdd part.

Share Improve this answer Follow

edited Aug 25 '16 at 6:28

answered Aug 14 '15 at 7:49



For some strange reason it works the other way round (Spark 2.1.0) collect().map($r \Rightarrow r(0)$) - does this order have any disadvantages? - Boern Mar 14 '17 at 13:17

Can be slower - your solution first collects all the data on the driver, and after that it does the mapping on the driver (without executors aid), using only the processing power of single driver. - TheMP Mar 14 '17 at 15:44

81

With Spark 2.x and Scala 2.11

I'd think of 3 possible ways to convert values of a specific column to List.



Common code snippets for all the approaches

1

```
import org.apache.spark.sql.SparkSession

val spark = SparkSession.builder.getOrCreate
import spark.implicits._ // for .toDF() method

val df = Seq(
    ("first", 2.0),
    ("test", 1.5),
    ("choose", 8.0)
    ).toDF("id", "val")
```

Approach 1

```
df.select("id").collect().map(_(0)).toList
// res9: List[Any] = List(one, two, three)
```

What happens now? We are collecting data to Driver with <code>collect()</code> and picking element zero from each record.

This could not be an excellent way of doing it, Let's improve it with next approach.

Approach 2

```
df.select("id").rdd.map(r => r(0)).collect.toList
//res10: List[Any] = List(one, two, three)
```

How is it better? We have distributed map transformation load among the workers rather than single Driver.

I know $rdd.map(r \Rightarrow r(0))$ does not seems elegant you. So, let's address it in next approach.

Approach 3

```
df.select("id").map(r => r.getString(0)).collect.toList
//res11: List[String] = List(one, two, three)
```

Here we are not converting DataFrame to RDD. Look at $_{map}$ it won't accept $_{r} \Rightarrow _{r(0)}$ (or $_{(0)}$) as the previous approach due to encoder issues in DataFrame. So end up using $_{r} \Rightarrow _{r.getString(0)}$ and it would be addressed in the next versions of Spark.

Conclusion

All the options give the same output but 2 and 3 are effective, finally 3rd one is effective and elegant(I'd think).

Databricks notebook

Share Improve this answer Follow

edited Jan 26 at 2:22

answered May 20 '17 at 6:44



28.6k 11 109 119



26

I know the answer given and asked for is assumed for Scala, so I am just providing a little snippet of Python code in case a PySpark user is curious. The syntax is similar to the given answer, but to properly pop the list out I actually have to reference the column name a second time in the mapping function and I do not need the select statement.



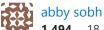
i.e. A DataFrame, containing a column named "Raw"

To get each row value in "Raw" combined as a list where each entry is a row value from "Raw" I simply use:

```
MyDataFrame.rdd.map(lambda x: x.Raw).collect()
```

Share Improve this answer Follow

answered Sep 30 '16 at 23:41



This gives a list of Row objects. What if you want a list of the values? - ThatDataGuy Nov 21 '16 at 16:45

This gives a list of values. – abby sobh Dec 1 '16 at 19:25

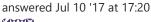
Thanks for sharing this! This works for me great just wondering if there is a way to speed this up, it runs pretty slow – Mojgan Mazouchi Jul 23 '20 at 23:39



In Scala and Spark 2+, try this (assuming your column name is "s"): df.select('s).as[String].collect



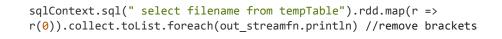
Share Improve this answer Follow





1,005 10 13







it works perfectly

Share Improve this answer Follow

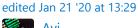
edited Jun 19 '19 at 14:03

answered Dec 16 '17 at 5:58 Shaina Raza



here collect is functions which in turn convert it to list. Be ware of using the list on the huge data set. It will decrease performance. It is good to check the data.

Share Improve this answer Follow





3,271 8 18 29

answered Jan 21 '20 at 12:47



amarnath pimple 139 1 6



2





logger.info(String.format("list is %s",whatever_list)); //verification

Since no one has given any solution in java(Real Programming Language) Can thank me later

Share Improve this answer Follow





Bob **1,301** 10 27 answered Apr 16 '20 at 20:13





Below is for Python-



df.select("col_name").rdd.flatMap(lambda x: x).collect()



Share Improve this answer Follow

answered Nov 26 '20 at 21:24



Nitin Mahajan

Others answers (such as stackoverflow.com/a/59841515/6807769) are similar - Vincent Doba Nov 26 '20 at 22:58

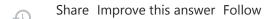
0 @VincentDoba - There will always be similar answers for a query. It's not the same and none of the answers use flatMap in python in this thread. It's very easy to downvote rather than helping people. Anyways. – Nitin Mahajan Jan 28 at 2:59



An updated solution that gets you a list:

 \cap

dataFrame.select("YOUR_COLUMN_NAME").map(r ⇒ r.getString(0)).collect.toList







This is java answer.



df.select("id").collectAsList();



Share Improve this answer Follow

answered Feb 14 '20 at 8:43

