

## Announcing Stack Overflow Documentation

We started with Q&A. Technical documentation is next, and we need your help.


Whether you're a beginner or an experienced developer, you *can* contribute.

[Sign up and start helping →](#)[Learn more about Documentation →](#)

## Creating a Spark DataFrame from an RDD of lists



Tired of recruiter spam?  
Want jobs tailored to your needs?



[Get started](#)

I have an rdd (we can call it myrdd) where each record in the rdd is of the form:

```
[('column 1',value), ('column 2',value), ('column 3',value), ... , ('column 100',value)]
```

I would like to convert this into a DataFrame in pyspark - what is the easiest way to do this?

[apache-spark](#) [dataframe](#) [pyspark](#)

asked Apr 7 '15 at 20:53



[mgoldwasser](#)

1,807 1 14 33

It's not exactly clear from your question where you're having trouble. Is it the fact that you have so many columns? Or just that records of your RDD are lists of tuples? – [Snoozer](#) Apr 7 '15 at 23:25

### 3 Answers

How about use the `toDF` method? You only need add the field names.

```
df = rdd.toDF(['column', 'value'])
```

answered Apr 9 '15 at 19:23



[dapangmao](#)

780 5 9

this answer works, and the solution I posted below (based on your answer) would convert an rdd as described above to a DataFrame – [mgoldwasser](#) Apr 10 '15 at 20:50

```
36 if (dev.isBored() || job.sucks()) {
37   searchJobs({flexibleHours: true, companyCulture: 100});
38 }
39 A career site that's by developers, for developers.
```



Get started

The answer by [@dapangmao](#) got me to this solution:

```
my_df = my_rdd.map(lambda l: Row(**dict(l))).toDF()
```

edited Jul 19 at 12:50

answered Apr 10 '15 at 20:48



[mgoldwasser](#)

1,807 1 14 33

Take a look at the [DataFrame documentation](#) to make this example work for you, but this should work. I'm assuming your RDD is called `my_rdd`

```
from pyspark.sql import SQLContext, Row
sqlContext = SQLContext(sc)

# You have a ton of columns and each one should be an argument to Row
# Use a dictionary comprehension to make this easier
def record_to_row(record):
    schema = {'column{i:d}'.format(i = col_idx):record[col_idx] for col_idx in
range(1,100+1)}
    return Row(**schema)

row_rdd = my_rdd.map(lambda x: record_to_row(x))

# Now infer the schema and you have a DataFrame
schema_my_rdd = sqlContext.inferSchema(row_rdd)

# Now you have a DataFrame you can register as a table
schema_my_rdd.registerTempTable("my_table")
```

I haven't worked much with DataFrames in Spark but this should do the trick

edited Apr 7 '15 at 22:25

answered Apr 7 '15 at 21:51



Snoozer

2,413 1 13 27

---

you might need to add a line after the sqlContext is created to load the implicits library: "import sqlContext .implicits.\_". See [spark.apache.org/docs/1.3.0/sql-programming-guide.html](http://spark.apache.org/docs/1.3.0/sql-programming-guide.html) – Glenn Strycker May 4 '15 at 20:10

---

Isn't that a scala-only thing? My answer is written in Python – Snoozer May 4 '15 at 22:51

---