

What are some examples of JSON functions in PySpark?

Authored by
stats writer

June 24, 2024

RECOMMENDED CITATION

stats writer (2024). *What are some examples of JSON functions in PySpark?*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=151237>


```
//id |value |
//+---+-----+
//|1 |{"Zipcode":704,"ZipCodeType":"STANDARD","City":"PARC PARQUE","State":"PR"}|
//+---+-----+
```

2. PySpark JSON Functions Examples

2.1. from_json()

PySpark `from_json()` function is used to convert JSON string into Struct type or Map type. The below example converts JSON string to Map key-value pair. I will leave it to you to convert to struct type. Refer, [Convert JSON string to Struct type column](#).

```
#Convert JSON string column to Map type
from pyspark.sql.types import MapType,StringType
from pyspark.sql.functions import from_json
df2=df.withColumn("value",from_json(df.value,MapType(StringType(),StringType())))
)
df2.printSchema()
df2.show(truncate=False)

//root
// |-- id: integer (nullable = false)
// |-- value: map (nullable = true)
//    |-- key: string
//    |-- value: string (valueContainsNull = true)

//+---+-----+
//id |value |
//+---+-----+
//|1 ||
//+---+-----+
```

2.2. to_json()

`to_json()` function is used to convert DataFrame columns MapType or Struct type to JSON string. Here, I am using `df2` that created from above `from_json()` example.

```

from pyspark.sql.functions import to_json,col
df2.withColumn("value",to_json(col("value")))
.show(truncate=False)

//+---+-----+-----+-----+
//|id |value |
//+---+-----+-----+
//|1 |{"Zipcode":"704","ZipCodeType":"STANDARD","City":"PARC PARQUE","State":"PR"}|
//+---+-----+-----+

```

2.3. json_tuple()

Function `json_tuple()` is used to query or extract the elements from a JSON column and create the result as a new column.

```

from pyspark.sql.functions import json_tuple
df.select(col("id"),json_tuple(col("value"),"Zipcode","ZipCodeType","City"))
.toDF("id","Zipcode","ZipCodeType","City")
.show(truncate=False)

```

```

//+---+-----+-----+-----+
//|id |Zipcode|ZipCodeType|City |
//+---+-----+-----+-----+
//|1 |704 |STANDARD |PARC PARQUE|
//+---+-----+-----+-----+

```

2.4. get_json_object()

`get_json_object()` is used to extract the JSON string based on a path from the JSON column.

```

from pyspark.sql.functions import get_json_object
df.select(col("id"),get_json_object(col("value"),"$$.ZipCodeType").alias("ZipCodeType"))
.show(truncate=False)

```

```

//+---+-----+
//|id |ZipCodeType|
//+---+-----+
//|1 |STANDARD |

```

```
//+---+-----+
```

2.5. schema_of_json()

Use `schema_of_json()` to create schema string from JSON string column.

```
from pyspark.sql.functions import schema_of_json,lit
schemaStr=spark.range(1)
.select(schema_of_json(lit("""{"Zipcode":704,"ZipCodeType":"STANDARD","City":"PARC
PARQUE","State":"PR"}"")))
.collect()
print(schemaStr)
```

```
//struct<City:string,State:string,ZipCodeType:string,Zipcode:bigint>
```

3. Complete Example of PySpark JSON Functions

```
from pyspark.sql import SparkSession,Row
spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()

jsonString="""{"Zipcode":704,"ZipCodeType":"STANDARD","City":"PARC
PARQUE","State":"PR"}"""
df=spark.createDataFrame(,)
df.show(truncate=False)

#Convert JSON string column to Map type
from pyspark.sql.types import MapType,StringType
from pyspark.sql.functions import from_json
df2=df.withColumn("value",from_json(df.value,MapType(StringType(),StringType())))
df2.printSchema()
df2.show(truncate=False)

from pyspark.sql.functions import to_json,col
df2.withColumn("value",to_json(col("value")))
.show(truncate=False)

from pyspark.sql.functions import json_tuple
df.select(col("id"),json_tuple(col("value"),"Zipcode","ZipCodeType","City"))
.toDF("id","Zipcode","ZipCodeType","City")
```

```
.show(truncate=False)
```

```
from pyspark.sql.functions import get_json_object
df.select(col("id"),get_json_object(col("value"),"$$.ZipCodeType").alias("ZipCodeType"))
.show(truncate=False)
```

```
from pyspark.sql.functions import schema_of_json,lit
schemaStr=spark.range(1)
.select(schema_of_json(lit("""{"Zipcode":704,"ZipCodeType":"STANDARD","City":"PARC
PARQUE","State":"PR"}"")))
.collect()
print(schemaStr)
```

Related Article

References