

Spark by {Examples} (https://sparkbyexamples.com/)

PySpark Tutorial

PySpark Tutorial For Beginners

(https://sparkbyexamples.com/pyspark-tutorial/)

PySpark – Features

(https://sparkbyexamples.com/pyspark-tutorial/#features)

PySpark – Advantages

(https://sparkbyexamples.com/pyspark-tutorial/#advantages)

PySpark – Modules & Packages

(https://sparkbyexamples.com/pyspark-tutorial/#modules-packages)

PySpark – Cluster Managers

(https://sparkbyexamples.com/pyspark-tutorial/#cluster-manager)

PySpark – Install on Windows

(https://sparkbyexamples.com/pyspark-tutorial/#pyspark-installation)

PySpark – Web/Application UI

(https://sparkbyexamples.com/spark/spark-web-ui-understanding/)

PySpark – SparkSession

(https://sparkbyexamples.com/pyspark/pyspark-what-is-sparksession/)

PySpark – RDD

(https://sparkbyexamples.com/pyspark-rdd)

PySpark – Parallelize

(https://sparkbyexamples.com/pyspark/pyspark-parallelize-create-rdd/)

PySpark – repartition() vs coalesce()

(https://sparkbyexamples.com/pyspark/pyspark-repartition-vs-coalesce/)

PySpark – Broadcast Variables

(https://sparkbyexamples.com/pyspark/pyspark-broadcast-variables/)

PySpark

(https://sparkbyexamples.com/pyspark-tutorial/)

Hive

(https://sparkbyexamples.com/apache-hive-tutorial/)

PySpark StructType & StructField Explained with Examples

(https://sparkbyexamples.com/apache-hbase-tutorial/)

Kafka

(https://sparkbyexamples.com/apache-kafka-tutorials-with-examples/)

FAQ's

(https://sparkbyexamples.com/spark-questions/)

More

(https://sparkbyexamples.com/)

Fresh Flowers & Chocolate
Ferns N Petals

 NNK

(https://sparkbyexamples.com/author/admin/)

 PySpark

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

PySpark StructType & StructField classes are used to programmatically specify the schema to the DataFrame and creating complex columns like nested struct, array and map columns.

StructType (https://github.com/apache/spark/blob/master/sql/catalyst/src/main/scala/org/apache/spark/sql/types/StructType.scala) is a collection of **StructField's** (https://github.com/apache/spark/blob/master/sql/catalyst/src/main/scala/org/apache/spark/sql/types/StructField.scala) that defines column name, column data type, boolean to specify if the field can be nullable or not and metadata.



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

PySpark DataFrame

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

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

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

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

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

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

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

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

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

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

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



In this article, I will explain different ways to define the structure of DataFrame using StructType with PySpark examples. Though PySpark infers a schema from data, some times we may need to define our own column names and data types and this article explains how to define simple, nested, and complex schemas.

- [Using PySpark StructType & StructField with DataFrame](#)
- [Defining Nested StructType or struct](#)
- [Adding & Changing columns of the DataFrame](#)
- [Using SQL ArrayType and MapType](#)
- [Creating StructType or struct from Json file
\(https://sparkbyexamples.com/wp-admin/post.php?post=7769&action=edit#schema-from-json\)](#)
- [Creating StructType object from DDL string](#)
- [Check if a field exists in a StructType](#)

1. StructType – Defines the structure of the Dataframe

PySpark provides from `pyspark.sql.types` import `StructType` class to define the structure of the DataFrame.

StructType is a collection or list of StructField objects.

`printSchema()` method on the DataFrame shows StructType columns as “struct”.

Incredible Savings

Data Science Mercury Learning

PySpark – [withColumnRenamed\(\)](https://sparkbyexamples.com/pyspark/pyspark-rename-dataframe-column/).
(<https://sparkbyexamples.com/pyspark/pyspark-rename-dataframe-column/>).

PySpark – [where\(\)](https://sparkbyexamples.com/pyspark/pyspark-where-filter/) & [filter\(\)](https://sparkbyexamples.com/pyspark/pyspark-where-filter/).
(<https://sparkbyexamples.com/pyspark/pyspark-where-filter/>).

PySpark – [drop\(\)](https://sparkbyexamples.com/pyspark/pyspark-distinct-to-drop-duplicates/) & [dropDuplicates\(\)](https://sparkbyexamples.com/pyspark/pyspark-distinct-to-drop-duplicates/).
(<https://sparkbyexamples.com/pyspark/pyspark-distinct-to-drop-duplicates/>).

PySpark – [orderBy\(\)](https://sparkbyexamples.com/pyspark/pyspark-orderby-and-sort-explained/) and [sort\(\)](https://sparkbyexamples.com/pyspark/pyspark-orderby-and-sort-explained/).
(<https://sparkbyexamples.com/pyspark/pyspark-orderby-and-sort-explained/>).

PySpark – [groupBy\(\)](https://sparkbyexamples.com/pyspark/pyspark-groupby-explained-with-example/).
(<https://sparkbyexamples.com/pyspark/pyspark-groupby-explained-with-example/>).

PySpark – [join\(\)](https://sparkbyexamples.com/pyspark/pyspark-join-explained-with-examples/).
(<https://sparkbyexamples.com/pyspark/pyspark-join-explained-with-examples/>).

PySpark – [union\(\)](https://sparkbyexamples.com/pyspark/pyspark-union-and-unionall/) & [unionAll\(\)](https://sparkbyexamples.com/pyspark/pyspark-union-and-unionall/).
(<https://sparkbyexamples.com/pyspark/pyspark-union-and-unionall/>).

PySpark – [unionByName\(\)](https://sparkbyexamples.com/spark/spark-merge-two-dataframes-with-different-columns/).
(<https://sparkbyexamples.com/spark/spark-merge-two-dataframes-with-different-columns/>).

PySpark – [UDF \(User Defined Function\)](https://sparkbyexamples.com/pyspark/pyspark-udf-user-defined-function/).
(<https://sparkbyexamples.com/pyspark/pyspark-udf-user-defined-function/>).

PySpark – [map\(\)](https://sparkbyexamples.com/pyspark/pyspark-map-transformation/).
(<https://sparkbyexamples.com/pyspark/pyspark-map-transformation/>).

PySpark – [flatMap\(\)](https://sparkbyexamples.com/pyspark/pyspark-flatmap-transformation/).
(<https://sparkbyexamples.com/pyspark/pyspark-flatmap-transformation/>).

pyspark – [foreach\(\)](https://sparkbyexamples.com/pyspark/pyspark-loop-iterate-through-rows-in-dataframe/#use-foreach-loop-through-dataframe).
(<https://sparkbyexamples.com/pyspark/pyspark-loop-iterate-through-rows-in-dataframe/#use-foreach-loop-through-dataframe>).

PySpark – [sample\(\)](#) vs [sampleBy\(\)](#).



2. StructField – Defines the metadata of the DataFrame column

PySpark provides `pyspark.sql.types` import `StructField` class to define the columns which includes column name(`String`), column type (`DataType` (<https://sparkbyexamples.com/spark/spark-sql-dataframe-data-types/>)), nullable column (`Boolean`) and metadata (`MetaData`)

3. Using PySpark StructType & StructField with DataFrame

While [creating a PySpark DataFrame](https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/) (<https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/>) we can specify the structure using `StructType` and `StructField` classes. As specified in the introduction, `StructType` is a collection of `StructField`'s which is used to define the column name, data type, and a flag for nullable or not. Using `StructField` we can also add nested struct schema, `ArrayType` (<https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/>) for arrays, and `MapType` (<https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/>) for key-value pairs which we will discuss in detail in later sections.

Learning

Data Science-Amazing new deal on everything U need to know re: data science

[humblebundle.com](https://www.humblebundle.com)

OPEN

[\(https://sparkbyexamples.com/pyspark/pyspark-sampling-example/\)](https://sparkbyexamples.com/pyspark/pyspark-sampling-example/)

[PySpark – fillna\(\) & fill\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-fillna-fill-replace-null-values/\)](https://sparkbyexamples.com/pyspark/pyspark-fillna-fill-replace-null-values/)

[PySpark – pivot\(\)_\(Row to Column\).
\(https://sparkbyexamples.com/pyspark/pyspark-pivot-and-unpivot-dataframe/\)](https://sparkbyexamples.com/pyspark/pyspark-pivot-and-unpivot-dataframe/)

[PySpark – partitionBy\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-partitionby-example/\)](https://sparkbyexamples.com/pyspark/pyspark-partitionby-example/)

[PySpark – ArrayType Column \(Array\).
\(https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/\)](https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/)

[PySpark – MapType \(Map/Dict\).
\(https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/\)](https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/)

PySpark SQL Functions

[PySpark – Aggregate Functions
\(https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/\)](https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/)

[PySpark – Window Functions
\(https://sparkbyexamples.com/pyspark/pyspark-window-functions/\)](https://sparkbyexamples.com/pyspark/pyspark-window-functions/)

[PySpark – Date and Timestamp Functions
\(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/\)](https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/)

[PySpark – JSON Functions
\(https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/\)](https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/)

PySpark Datasources

[PySpark – Read & Write CSV File
\(https://sparkbyexamples.com/pyspark/pyspark-read-csv-file-into-dataframe/\)](https://sparkbyexamples.com/pyspark/pyspark-read-csv-file-into-dataframe/)

[PySpark – Read & Write Parquet File
\(https://sparkbyexamples.com/pyspark/pyspark-read-write-parquet-file/\)](https://sparkbyexamples.com/pyspark/pyspark-read-write-parquet-file/)

The below example demonstrates a very simple example of how to create a StructType & StructField on DataFrame and it's usage with sample data to support it.

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField

spark = SparkSession.builder.master("local[*]").appName('SparkByExamples.com').getOrCreate()

data = [
    ("James", "", "Smith", "360", "M", "40288"),
    ("Michael", "Rose", "", "40288", "M", "40288"),
    ("Robert", "", "Williams", "42", "M", "40288"),
    ("Maria", "Anne", "Jones", "39", "F", "40288"),
    ("Jen", "Mary", "Brown", "", "F", "40288")
]

schema = StructType([
    StructField("firstname", StringType, True),
    StructField("middlename", StringType, True),
    StructField("lastname", StringType, True),
    StructField("id", StringType, True),
    StructField("gender", StringType, True),
    StructField("salary", IntegerType, True)
])

df = spark.createDataFrame(data, schema)
df.printSchema()
df.show(truncate=False)
```

By running the above snippet, it displays below outputs.

[pyspark/pyspark-read-and-write-parquet-file/](#)

[PySpark – Read & Write JSON file](#)
(<https://sparkbyexamples.com/pyspark/pyspark-read-json-file-into-dataframe/>)

Incredible Savings

Data Science Mercury Learn

Data Science-Amazing new deal everything U need to know re: data science
humblebundle.com

OPEN

```
root
|-- firstname: string (nullable=true)
|-- middlename: string (nullable=true)
|-- lastname: string (nullable=true)
|-- id: string (nullable = true)
|-- gender: string (nullable = true)
|-- salary: integer (nullable = true)
```

Firstname	middlename	lastname
James		Smith
Michael	Rose	
Robert		Williams
Maria	Anne	Jones
Jen	Mary	Brown



Defining Nested

StructType object struct

While working on DataFrame we often need to work with the nested struct column and this can be defined using StructType.

In the below example column “name” data type is StructType which is nested.

Now Only in
798 Rupees

Shop for Men's
Sneaker Online at
Shoe



PySpark Built-In
Functions

[PySpark – when\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-when-otherwise/>)

[PySpark – expr\(\)](#)
([https://sparkbyexamples.com/pyspark-sql-expr-expression-function/](https://sparkbyexamples.com/pyspark/pyspark-sql-expr-expression-function/))



[PySpark – lit\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-lit-add-literal-constant/>).

[PySpark – split\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-convert-string-to-array-column/>).

[PySpark – concat_ws\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-convert-array-column-to-string-column/>).

[Pyspark – substring\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-substring-from-a-column/>).

[PySpark – translate\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#translate-replace-character-by-character>).

[PySpark – regexp_replace\(\)](#)
(https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#regexp_replace-replace-string-columns).

[PySpark – overlay\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#overlay-function>).

[PySpark – to_timestamp\(\)](#)
(https://sparkbyexamples.com/pyspark/pyspark-to_timestamp-convert-string-to-timestamp-type/).

[PySpark – to_date\(\)](#)
(https://sparkbyexamples.com/pyspark/pyspark-to_date-convert-timestamp-to-date/).

[PySpark – date_format\(\)](#)
(https://sparkbyexamples.com/pyspark/pyspark-date_format-convert-date-to-string-format/).

[PySpark – datediff\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#datediff>).

[PySpark – months_between\(\)](#)
([https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#months_between\(\)](https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#months_between())).

```
structureData = [
    ("James", "", "Smith"), "36636", "M", "1990-01-01", 10000,
    ("Michael", "Rose", ""), "40288", "M", "1990-01-01", 10000,
    ("Robert", "", "Williams"), "42114", "M", "1990-01-01", 10000,
    ("Maria", "Anne", "Jones"), "39192", "F", "1990-01-01", 10000,
    ("Jen", "Mary", "Brown"), "", "F", "1990-01-01", 10000,
]

structureSchema = StructType([
    StructField('name', StringType, True),
    StructField('firstname', StringType, True),
    StructField('middlename', StringType, True),
    StructField('lastname', StringType, True),
    StructField('id', IntegerType, True),
    StructField('gender', StringType, True),
    StructField('salary', IntegerType, True)
])

df2 = spark.createDataFrame(data=structureData, schema=structureSchema)
df2.printSchema()
df2.show(truncate=False)
```

Outputs below schema and the DataFrame

```
root
|-- name: struct (nullable = true)
|   |-- firstname: string (nullable = true)
|   |-- middlename: string (nullable = true)
|   |-- lastname: string (nullable = true)
|-- id: string (nullable = true)
|-- gender: string (nullable = true)
|-- salary: integer (nullable = true)
```

name	id	gender
[James, , Smith]	36636	M
[Michael, Rose,]	40288	M
[Robert, , Williams]	42114	M
[Maria, Anne, Jones]	39192	F
[Jen, Mary, Brown]		F

5. Adding & Changing struct of the DataFrame

[PySpark – explode\(\)](https://sparkbyexamples.com/pyspark/pyspark-explode-nested-array-into-rows/)
(<https://sparkbyexamples.com/pyspark/pyspark-explode-nested-array-into-rows/>).

[PySpark – array_contains\(\)](https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/#array_contains)
(https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/#array_contains).

[PySpark – array\(\)](https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/#array)
(<https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/#array>).

[PySpark – collect_list\(\)](https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#collect-list)
(<https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#collect-list>).

[PySpark – collect_set\(\)](https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#collect-set)
(<https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#collect-set>).

[PySpark – create_map\(\)](https://sparkbyexamples.com/pyspark/pyspark-convert-dataframe-columns-to-maptype-dict/)
(<https://sparkbyexamples.com/pyspark/pyspark-convert-dataframe-columns-to-maptype-dict/>).

[PySpark – map_keys\(\)](https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/#map_keys)
(https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/#map_keys).

[PySpark – map_values\(\)](https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/#map_values)
(https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/#map_values).

[PySpark – struct\(\)](https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/#update-struct-function)
(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/#update-struct-function>).

[PySpark – countDistinct\(\)](https://sparkbyexamples.com/pyspark/pyspark-count-distinct-from-dataframe/)
(<https://sparkbyexamples.com/pyspark/pyspark-count-distinct-from-dataframe/>).

[PySpark – sum\(\).avg\(\)](https://sparkbyexamples.com/pyspark/pyspark-dataframe-groupby-and-sort-by-descending-order/)
(<https://sparkbyexamples.com/pyspark/pyspark-dataframe-groupby-and-sort-by-descending-order/>).

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

[PySpark – rank\(\)](https://sparkbyexamples.com/pyspark/pyspark-window-functions/#rank)
(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/#rank>).

Using [PySpark SQL function](#)

(<https://sparkbyexamples.com/spark/sql-functions-understanding/>)

struct(), we can change the struct of the existing DataFrame and add a new StructType to it. The below example demonstrates how to copy the columns from one structure to another and adding a new column. [PySpark Column Class](#) (<https://sparkbyexamples.com/pyspark/pyspark-column-functions/>) also provides some functions to work with the StructType column.

```
from pyspark.sql.functions import struct, col, when, cast, IntegerType, StringType

updatedDF = df2.withColumn("OtherInfo",
    struct(
        col("id").alias("identifier"),
        col("gender").alias("gender"),
        col("salary").alias("salary"),
        when(col("salary").cast(IntegerType) > 10000,
            col("salary").alias("Salary_Grade"),
            .otherwise("High").alias("Salary_Grade")
        )
    ).drop("id", "gender", "salary"))

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

Here, it copies “gender“, “salary” and “id” to the new struct “otherInfo” and add’s a new column “Salary_Grade“.

```
root
 |-- name: string (nullable = true)
 |-- firstname: string (nullable = true)
 |-- middlename: string (nullable = true)
 |-- lastname: string (nullable = true)
 |-- OtherInfo: struct (nullable = true)
 |   |-- identifier: string (nullable = true)
 |   |-- gender: string (nullable = true)
 |   |-- salary: integer (nullable = true)
 |   |-- Salary_Grade: string (nullable = true)
```

6. Using SQL ArrayType and MapType

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

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

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

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

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

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

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

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

SQL StructType also supports

[ArrayType](https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/)

[\(https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/\)](https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/) and [MapType](https://sparkbyexamples.com/pyspark/pyspark-maptypes-dict-examples/)

[\(https://sparkbyexamples.com/pyspark/pyspark-maptypes-dict-examples/\)](https://sparkbyexamples.com/pyspark/pyspark-maptypes-dict-examples/) to

define the DataFrame columns for array and map collections respectively. On

the below example, column hobbies defined as ArrayType(StringType) and properties defined as

MapType(StringType,StringType)

meaning both key and value as String.

```
arrayStructureSchema = StructType(
    StructField('name', StructType(
        StructField('firstname', StringType),
        StructField('middlename', StringType),
        StructField('lastname', StringType),
    )),
    StructField('hobbies', ArrayType(StringType)),
    StructField('properties', MapType(StringType, StringType))
)
```

Outputs the below schema. Note that field Hobbies is array type and properties is map type.

```
root
|-- name: struct (nullable = true)
|   |-- firstname: string (nullable = true)
|   |-- middlename: string (nullable = true)
|   |-- lastname: string (nullable = true)
|-- hobbies: array (nullable = true)
|   |-- element: string (contains null)
|-- properties: map (nullable = true)
|   |-- key: string
|   |-- value: string (valueContainsNull = true)
```

7. Creating StructType object struct from JSON file

If you have too many columns and the structure of the DataFrame changes now and then, it's a good practice to load the SQL StructType schema from

JSON file. You can get the schema by using `df2.schema.json()` , store this in a file and will use it to create a the schema from this file.

```
print(df2.schema.json())
```

```
{
  "type" : "struct",
  "fields" : [ {
    "name" : "name",
    "type" : {
      "type" : "struct",
      "fields" : [ {
        "name" : "firstname",
        "type" : "string",
        "nullable" : true,
        "metadata" : { }
      }, {
        "name" : "middlename",
        "type" : "string",
        "nullable" : true,
        "metadata" : { }
      }, {
        "name" : "lastname",
        "type" : "string",
        "nullable" : true,
        "metadata" : { }
      } ]
    },
    "nullable" : true,
    "metadata" : { }
  }, {
    "name" : "dob",
    "type" : "string",
    "nullable" : true,
    "metadata" : { }
  }, {
    "name" : "gender",
    "type" : "string",
    "nullable" : true,
    "metadata" : { }
  }, {
    "name" : "salary",
    "type" : "integer",
    "nullable" : true,
    "metadata" : { }
  } ]
}
```

Alternatively, you could also use `df.schema.simpleString()`, this will return an relatively simpler schema format.

Now let's load the json file and use it to create a `DataFrame`.

```
import json
schemaFromJson = StructType.fromJson(json.loads(jsonStr).get('schema'))
df3 = spark.createDataFrame(
    spark.sparkContext.parallelize(data), schemaFromJson)
df3.printSchema()
```

This prints the same output as the previous section. You can also, have a name, type, and flag for nullable in a comma-separated file and we can use these to create a `StructType` programmatically, I will leave this to you to explore.

8. Creating StructType object struct from DDL String

Like loading structure from JSON string, we can also create it from DDL (by using `fromDDL()` static function on SQL `StructType` class `StructType.fromDDL()`). You can also generate DDL from a schema using `toDDL()`. `printTreeString()` on struct object prints the schema similar to `printSchema` function returns.

```
ddlSchemaStr = "`fullName` STRING, `middle`: STRING>, `age` INT, `gender` STRING"
ddlSchema = StructType.fromDDL(ddlSchemaStr)
ddlSchema.printTreeString()
```

9. Checking if a Column Exists in a DataFrame

If you want to perform some checks on metadata of the DataFrame, for example, if a column or field exists in a DataFrame or data type of column; we can easily do this using several functions on SQL StructType and StructField.

```
print(df.schema.fieldNames.contains("first name"))
print(df.schema.contains(StructType([StructField("first name", IntegerType)]))
```

This example returns “true” for both scenarios. And for the second one if you have IntegerType instead of StringType it returns false as the datatype for first name column is String, as it checks every property in a field. Similarly, you can also check if two schemas are equal and more.

10. Complete Example of PySpark StructType & StructField

```

import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType, StructField, StringType, IntegerType
from pyspark.sql.functions import concat_ws, concat

spark = SparkSession.builder.master("local[*]").appName('SparkByExample.com').getOrCreate()

data = [
    ("James", "", "Smith", "36636", "M", 300000),
    ("Michael", "Rose", "", "40288", "F", 400000),
    ("Robert", "", "Williams", "42112", "M", 400000),
    ("Maria", "Anne", "Jones", "39112", "F", 400000),
    ("Jen", "Mary", "Brown", "", "F", 500000)
]

schema = StructType([
    StructField("firstname", StringType, True),
    StructField("middlename", StringType, True),
    StructField("lastname", StringType, True),
    StructField("id", StringType, True),
    StructField("gender", StringType, True),
    StructField("salary", IntegerType, True)
])

df = spark.createDataFrame(data, schema)
df.printSchema()
df.show(truncate=False)

structureData = [
    (("James", "", "Smith"), "36636", "M", 300000),
    (("Michael", "Rose", ""), "40288", "F", 400000),
    (("Robert", "", "Williams"), "42112", "M", 400000),
    (("Maria", "Anne", "Jones"), "39112", "F", 400000),
    (("Jen", "Mary", "Brown"), "", "F", 500000)
]

structureSchema = StructType([
    StructField('name', StructType([
        StructField('first', StringType, True),
        StructField('middle', StringType, True),
        StructField('last', StringType, True)
    ])),
    StructField('id', StringType, True),
    StructField('gender', StringType, True),
    StructField('salary', IntegerType, True)
])

df2 = spark.createDataFrame(data, structureSchema)
df2.printSchema()
df2.show(truncate=False)

updatedDF = df2.withColumn("Other",
    struct(
        col("id").alias("id"),
        col("gender").alias("gender")
    )
)

```

```

        col("salary").alias("salary")
        when(col("salary").cast(IntegerType) > 100000)
            .when(col("salary").cast(IntegerType) > 50000)
            .otherwise("High").alias("salary")
    ))).drop("id", "gender", "salary")

updatedDF.printSchema()
updatedDF.show(truncate=False)

""" Array & Map """

arrayStructureSchema = StructType(
    StructField('name', StringType),
    StructField('firstname', StringType),
    StructField('middlename', StringType),
    StructField('lastname', StringType),
    ],
    StructField('hobbies', ArrayType(StringType)),
    StructField('properties', MapType(StringType, StringType))
])

```

The complete example explained here is available also available at [GitHub \(https://github.com/spark-by-examples/pyspark-examples/blob/master/pyspark-structtype.py\)](https://github.com/spark-by-examples/pyspark-examples/blob/master/pyspark-structtype.py) project.

Conclusion:

In this article, you have learned the usage of SQL StructType, StructField, and how to change the structure of the Pyspark DataFrame at runtime, converting case class to the schema and using ArrayType, MapType.

Happy Learning !!

Share this:



(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/?share=facebook&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/?share=reddit&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/?share=pinterest&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/?share=tumblr&nb=1>)





(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/?share=pocket&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/?share=linkedin&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/?share=twitter&nb=1>)

TAGS: [ARRAYTYPE](#)

([HTTPS://SPARKBYEXAMPLES.COM/TAG/ARRAYTYPE/](https://sparkbyexamples.com/tag/arraytype/)), [DATATYPE](#)

([HTTPS://SPARKBYEXAMPLES.COM/TAG/DATATYPE/](https://sparkbyexamples.com/tag/datatype/)), [MAPTYPE](#)

([HTTPS://SPARKBYEXAMPLES.COM/TAG/MAPTYPE/](https://sparkbyexamples.com/tag/maptype/)), [PYSPARK SCHEMA](#)

([HTTPS://SPARKBYEXAMPLES.COM/TAG/PYSPARK-SCHEMA/](https://sparkbyexamples.com/tag/pyspark-schema/)), [SCHEMA](#)

([HTTPS://SPARKBYEXAMPLES.COM/TAG/SCHEMA/](https://sparkbyexamples.com/tag/schema/)), [STRUCTFIELD](#)

([HTTPS://SPARKBYEXAMPLES.COM/TAG/STRUCTFIELD/](https://sparkbyexamples.com/tag/structfield/)), [STRUCTTYPE](#)

([HTTPS://SPARKBYEXAMPLES.COM/TAG/STRUCTTYPE/](https://sparkbyexamples.com/tag/structtype/)).



[NNK](#)

([Https://Sparkbyexamples.Com/Author/Admin/](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/>).

Leave a Reply

Enter your comment here...




<div>Apache Hadoop</div> <div>(https://sparkbyexamples.com/category/hadoop/)</div>
<div>Apache Spark</div> <div>(https://sparkbyexamples.com/category/spark/)</div>
<div>Apache Spark Streaming</div> <div>(https://sparkbyexamples.com/category/spark/apache-spark-streaming/)</div>
<div>Apache Kafka</div> <div>(https://sparkbyexamples.com/category/kafka/)</div>
<div>Apache HBase</div> <div>(https://sparkbyexamples.com/category/hbase/)</div>
<div>Apache Cassandra</div> <div>(https://sparkbyexamples.com/category/cassandra/)</div>
<div>Snowflake Database</div> <div>(https://sparkbyexamples.com/category/snowflake/)</div>
<div>H2O Sparkling Water</div> <div>(https://sparkbyexamples.com/category/h2o-sparkling-water/)</div>
<div>PySpark</div> <div>(https://sparkbyexamples.com/category/pyspark/)</div>


<div>Spark regexp_replace() – Replace String Value</div> <div>(https://sparkbyexamples.com/spark/spark-regexp_replace-replace-string-value/)</div>
<div>How to Run a PySpark Script from Python?</div> <div>(https://sparkbyexamples.com/pyspark/run-pyspark-script-from-python-subprocess/)</div>
<div>Spark SQL like() Using Wildcard Example</div> <div>(https://sparkbyexamples.com/spark/spark-sql-like-using-wildcard-example/)</div>
<div>Spark isin() & IS NOT IN Operator Example</div> <div>(https://sparkbyexamples.com/spark/spark-isin-is-not-in-operator-example/)</div>
<div>Spark – Get Size/Length of Array & Map Column</div> <div>(https://sparkbyexamples.com/spark/spark-get-size-length-of-array-map-column/)</div>
<div>Spark Using Length/Size Of a DataFrame Column</div> <div>(https://sparkbyexamples.com/spark/spark-using-length-size-of-a-dataframe-column/)</div>
<div>Spark rlike() Working with Regex Matching Examples</div> <div>(https://sparkbyexamples.com/spark/spark-rlike-regex-matching-examples/)</div>
<div>Spark Check String Column Has Numeric Values</div> <div>(https://sparkbyexamples.com/spark/spark-check-string-column-has-numeric-values/)</div>
<div>Spark Check Column Data Type is Integer or String</div> <div>(https://sparkbyexamples.com/spark/spark-check-column-data-type-is-integer-or-string/)</div>

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://www.facebook.com/sparkbyexamples/>)



(<https://www.linkedin.com/company/sparkbyexamples/>)



(<https://twitter.com/sparkbyexamples>)



(<https://github.com/sparkbyexamples>)

