LEARNING spark-dataframe

#spark-dataframe

Table of Contents

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: spark-dataframe

It is an unofficial and free spark-dataframe ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official spark-dataframe.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with spark-dataframe

Remarks

This section provides an overview of what spark-dataframe is, and why a developer might want to use it.

It should also mention any large subjects within spark-dataframe, and link out to the related topics. Since the Documentation for spark-dataframe is new, you may need to create initial versions of those related topics.

Examples

Installation or Setup

Detailed instructions on getting spark-dataframe set up or installed.

Loading Data Into A DataFrame

In Spark (scala) we can get our data into a DataFrame in several different ways, each for different use cases.

Create DataFrame From CSV

The easiest way to load data into a DataFrame is to load it from CSV file. An example of this (taken from the official documentation) is:

```
import org.apache.spark.sql.SQLContext

val sqlContext = new SQLContext(sc)
val df = sqlContext.read
    .format("com.databricks.spark.csv")
    .option("header", "true") // Use first line of all files as header
    .option("inferSchema", "true") // Automatically infer data types
    .load("cars.csv")
```

Create DataFrame From RDD Implicitly

Quite often in spark applications we have data in an RDD, but need to convert this into a DataFrame. The easiest way to do this is to use the `.toDF()` RDD function, which will implicitly determine the data types for our DataFrame:

```
val data = List(
   ("John", "Smith", 30),
   ("Jane", "Doe", 25)
)
```

```
val rdd = sc.parallelize(data)

val df = rdd.toDF("firstname", "surname", "age")
```

Create DataFrame From RDD Explicitly

In some scenarios using the `.toDF()` approach is not the best idea, since we need to explicitly define the schema of our DataFrame. This can be achieved using a StructType containing an Array of StructField.

```
import org.apache.spark.sql.types._
import org.apache.spark.sql.Row

val data = List(
   Array("John", "Smith", 30),
   Array("Jane", "Doe", 25)
)

val rdd = sc.parallelize(data)

val schema = StructType(
   Array(
      StructField("firstname", StringType,  true),
      StructField("surname",   StringType,  false),
      StructField("age",       IntegerType, true)
   )
)

val rowRDD = rdd.map(arr => Row(arr : _*))

val df = sqlContext.createDataFrame(rowRDD, schema)
```

Read Getting started with spark-dataframe online: https://riptutorial.com/spark-dataframe/topic/8988/getting-started-with-spark-dataframe

Credits

S. No	Chapters	Contributors
1	Getting started with spark-dataframe	Ben Horsburgh, Community