Java-Success.com

Prepare to fast-track, choose & go places with 800+ Java & Big Data Q&As with lots of code & diagrams.

13: Spark on Zeppelin – Dataframe date & timestamp

Posted on September 28, 2018

Pre-requisite: Docker is installed on your machine for Mac OS X (E.g. $ brew cask install docker) or Windows 10. Docker interview Q&As. This extends setting up Apache Zeppelin Notebook.

Step 1: Pull this from the docker hub, and build the image with the following command.

```
$ docker pull apache/zeppelin:0.7.3
```

300+ Java Interview FAQs

300+ Java FAQs 🔥

16+ Java Key Areas Q&As

150+ Java Architect FAQs

80+ Java Code Quality Q&As

150+ Java Coding Q&As

300+ Big Data Interview FAQs

300+ Big Data FAQs 🔥

Tutorials - Big Data

TUT - ⏯ Starting Big Data

TUT - Starting Spark & Scala

You can verify the image with the “docker images” command.

Step 2: Run the container with the above image.

```
$ docker run --rm -it -p 8080:8080 apache/zeppelin

```

Step 3: Open Zeppelin notebook via a web browser “http:localhost:8080”. Create a note book with “spark” as a default interpreter.

Comparing Timstamps

A new column “eligibleForBonus” is added based on Timestamp comparison. “**when/otherwise**” is used to assign a value. “**lit**” function is used to convert “cutOffTs” to a Column of literal value.

Step 4:

```
%spark


import org.apache.spark.sql.types.TimestampType
import java.sql.Timestamp
import org.apache.spark.sql.functions.lit

case class Employee (id: Integer, name: String, jo

val employees = Seq(
    Employee(1, "John", Timestamp.valueOf("2016-08
    Employee(2, "Peter", Timestamp.valueOf("2016-0
    Employee(6, "Elliot", Timestamp.valueOf("2016-
)

val employeeDF  = spark.createDataFrame(
  spark.sparkContext.parallelize(employees)
)

val cutOffTs: java.sql.Timestamp =  Timestamp.valu

```

TUT - Starting with Python
TUT - Kafka
TUT - Pig
TUT - Apache Storm
TUT - Spark Scala on Zeppelin
TUT - Cloudera
TUT - Cloudera on Docker
TUT - File Formats
TUT - Spark on Docker
TUT - Flume
TUT - Hadoop (HDFS)
TUT - HBase (NoSQL)
TUT - Hive (SQL)
TUT - Hadoop & Spark
TUT - MapReduce
TUT - Spark and Scala
TUT - Spark & Java
TUT - PySpark on Databricks
TUT - Zookeeper

800+ Java Interview Q&As

300+ Core Java Q&As
300+ Enterprise Java Q&As
150+ Java Frameworks Q&As
120+ Companion Tech Q&As
Tutorials - Enterprise Java

```
val employeeWithEligibilityDf =  employeeDF.withC

employeeWithEligibilityDf.show()

```

Output:

```
+---+------+-------------------+---------------+
| id|  name|             joined|eligibleForBonus|
+---+------+-------------------+---------------+
|  1|  John|2016-08-25 15:30:...|             no|
|  2| Peter|2016-08-25 10:30:...|            yes|
|  6|Elliot|2016-08-25 09:30:...|            yes|
+---+------+-------------------+---------------+

```

Comparing Dates

```
%spark


import org.apache.spark.sql.types.TimestampType
import java.sql.Timestamp
import java.sql.Date
import org.apache.spark.sql.functions.{lit, to_da

case class Employee (id: Integer, name: String, j

val employees = Seq(
    Employee(1, "John", Timestamp.valueOf("2016-0
    Employee(2, "Peter", Timestamp.valueOf("2016-
    Employee(6, "Elliot", Timestamp.valueOf("2016
)

val employeeDF  = spark.createDataFrame(
  spark.sparkContext.parallelize(employees)
)

val cutOffTs: java.sql.Timestamp =  Timestamp.valu

val employeeWithEligibilityDf =  employeeDF.withC

employeeWithEligibilityDf.show()

```

Output:

```
+---+------+-------------------+---------------+
| id|  name|             joined|eligibleForBonus|
+---+------+-------------------+---------------+
|  1|  John|2016-08-25 15:30:...|            yes|
|  2| Peter|2016-08-25 10:30:...|            yes|
|  6|Elliot|2016-08-25 09:30:...|            yes|
+---+------+-------------------+---------------+
```

Nested when/otherwise

```
%spark


import org.apache.spark.sql.types.TimestampType
import java.sql.Timestamp
import java.sql.Date
import org.apache.spark.sql.functions.{lit, to_dat

case class Employee (id: Integer, name: String, de

val employees = Seq(
    Employee(1, "John", "Marketing",  Timestamp.va
    Employee(2, "Peter", "Accounting", Timestamp.v
    Employee(6, "Elliot", "Marketing", Timestamp.v
)

val employeeDF  = spark.createDataFrame(
  spark.sparkContext.parallelize(employees)
)

val cutOffTs: java.sql.Timestamp =  Timestamp.valu

val employeeWithEligibilityDf =  employeeDF.withCo
                                          when(c
                                            .oth
                                            .oth


employeeWithEligibilityDf.show()
```

Output:

```
+---+------+----------+-------------------+------
| id|  name|department|             joined|eligib
+---+------+----------+-------------------+------
|  1|  John| Marketing|2016-08-25 15:30:...|
|  2| Peter|Accounting|2016-08-25 10:30:...|
|  6|Elliot| Marketing|2016-08-25 09:30:...|
```

```
7 +---+------+----------+--------------------+------+
8
```

‹ 10+ Apache NiFi interview Q&As 8 Spark streaming interview Q&As ›

Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty ltd. The EmpoweringTech pty ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. Privacy Policy

© 2022 java-success.com