800+ Q&As | Logout | Contact

Java-Success.com

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

search here … Go

Home Why? ▾ 300+ Java FAQs ▾ 300+ Big Data FAQs ▾ Courses ▾

Membership ▾ Your Career ▾

Home › bigdata-success.com › Tutorials - Big Data › TUT - Pig › 2. Apache Pig: Regex (Regular expressions)

2. Apache Pig: Regex (Regular expressions)

Posted on February 5, 2016

This extends the tutorial 1. Apache Pig Getting started.

Input Data

scores.xml in folder:**/Users/arulk/projects** representing marks of 4 students in 3 subjects:

```

<scores>
   <subject>
       <name>Science</name>
       <marks>
          <mark>80</mark>
      <mark>75</mark>
```

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

TOP

```
      <mark>89</mark>
      <mark>90</mark>
      </marks>
   </subject>
   <subject>
       <name>Maths</name>
       <marks>
          <mark>90</mark>
      <mark>87</mark>
      <mark>78</mark>
      <mark>92</mark>
      </marks>
   </subject>
   <subject>
       <name>English</name>
       <marks>
          <mark>78</mark>
      <mark>88</mark>
      <mark>65</mark>
      <mark>99</mark>
      </marks>
   </subject>
</scores>

```

Step 1: Start pig in local file system mode.

```

pig -x local

```

Step 2: Extract the “**Subjects**” from the input XML file.

```

grunt> SUBJECTS_EXTRACT =  LOAD '/Users/arulk/proj

```

Dump the output:

```

grunt> dump SUBJECTS_EXTRACT;

```

```

(<subject>       <name>Science</name>       <marks
(<subject>       <name>Maths</name>       <marks>
```

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

```
(<subject>        <name>English</name>        <marks>
```

Step 3: **Regex** to extract each "Subject" and its corresponding marks.

```
grunt> MARKS_FOR_SUBJECT_CSV = foreach SUBJECTS_EX
```

```
grunt> dump MARKS_FOR_SUBJECT_CSV;
```

Dump it:

```
(Science,80,75,89,90)
(Maths,90,87,78,92)
(English,78,88,65,99)
```

Step 4: Put it all into a single "marks_by_subjects.pig" script.

```
SUBJECTS_EXTRACT =  LOAD '/Users/arulk/projects/sc

MARKS_FOR_SUBJECT_CSV = foreach SUBJECTS_EXTRACT

dump MARKS_FOR_SUBJECT_CSV
```

Run the above pig script:

```
$ pig -x local marks_by_subjects.pig
```

Outputs:

```

(Science,80,75,89,90)
(Maths,90,87,78,92)
(English,78,88,65,99)

```

If you run without “-x local” option it runs in “Map-Reduce” mode against the HDFS (e.g. hdfs://localhost:9000). The name and data nodes need to be running. Otherwise you will get “Caused by: java.net.ConnectException: Connection refused” error.

Mapreduce mode

```

pig -x mapreduce
pig

```

‹ 1. Apache Pig Getting started 3. Apache Pig: XPath for XML ›

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

Top