

# Java-Success.com

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

[Home](#) [Why? ▾](#) [300+ Java FAQs ▾](#) [300+ Big Data FAQs ▾](#) [Courses ▾](#)[👤 Membership ▾](#) [Your Career ▾](#)[Home](#) › [bigdata-success.com](#) › [Tutorials - Big Data](#) › [TUT - Spark Scala on Zeppelin](#) ›

02: Spark on Zeppelin – read a file from local file system

## 02: Spark on Zeppelin – read a file from local file system

 Posted on [July 25, 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.

```
1 $ docker pull apache/zeppelin:0.7.3
2
```

### 300+ Java Interview FAQs

300+ Java FAQs



16+ Java Key Areas Q&amp;As



150+ Java Architect FAQs



80+ Java Code Quality Q&amp;As



150+ Java Coding Q&amp;As



### 300+ Big Data Interview FAQs

300+ Big Data FAQs



Tutorials - Big Data

TUT -  Starting Big Data

TUT - Starting Spark &amp; Scala

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

**Step 2:** The input file to read “employees.txt” in the \$(pwd)/seed.

```
1 1, John, USA, 100000.00
2 2, Peter, Australia, 200000.00
3 3, Sam, USA, 76000.00
4 4, Daniel, France, 86000.00
5 5, Simon, Australia, 96000.00
6 6, Roseanne, France, 156000.00
7
```

**Step 3:** Run the container with the above image.

```
1 $ docker run --rm -it -p 8080:8080 -v "$(pwd)/seed
2
```

**Note:** \$(pwd)/seed – is the folder where the **employees.txt** input file will be placed on the host system, and will be synchronized with the container path “/zeppelin/seed”.

You can inspect the container files/logs with the following commands in a separate terminal window:

Get the container id with:

```
1 $ docker ps
2
```

**sh** to the container with:

```
1 $ docker exec -it <container id> /bin/bash
2
```

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



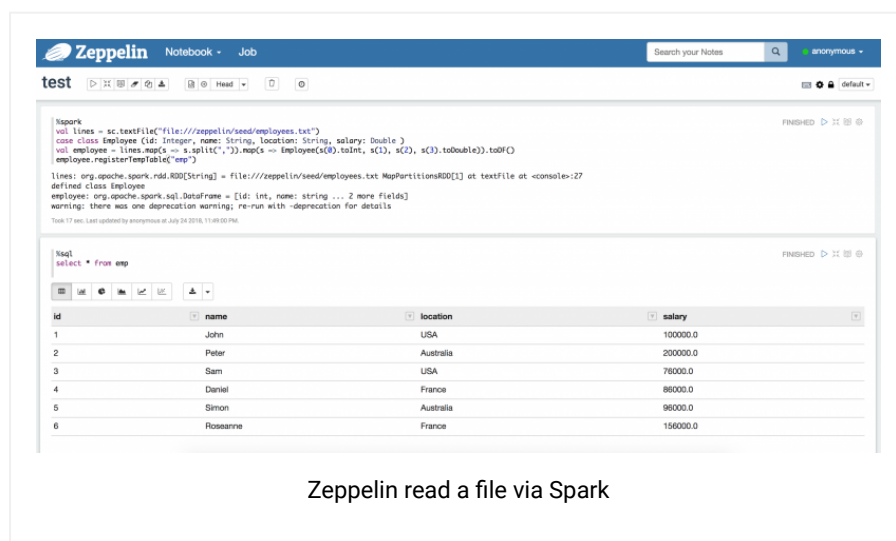
## Step 4: Open Zeppelin notebook via a web browser

"http:localhost:8080". Create a note book with "spark" as a default interpreter.

```
1
2 %spark
3
4 val lines = sc.textFile("file:///zeppelin/seed/employees.txt")
5 case class Employee (id: Integer, name: String, location: String, salary: Double)
6 val employee = lines.map(s => s.split(",")).map(s => Employee(s(0).toInt, s(1), s(2), s(3).toDouble)).toDF()
7 employee.registerTempTable("emp")
8
```

```
1 %sql
2
3 select * from emp
4
```

You can view the "SQL" output in multiple formats like tabular, graph chart, pie chart, etc.



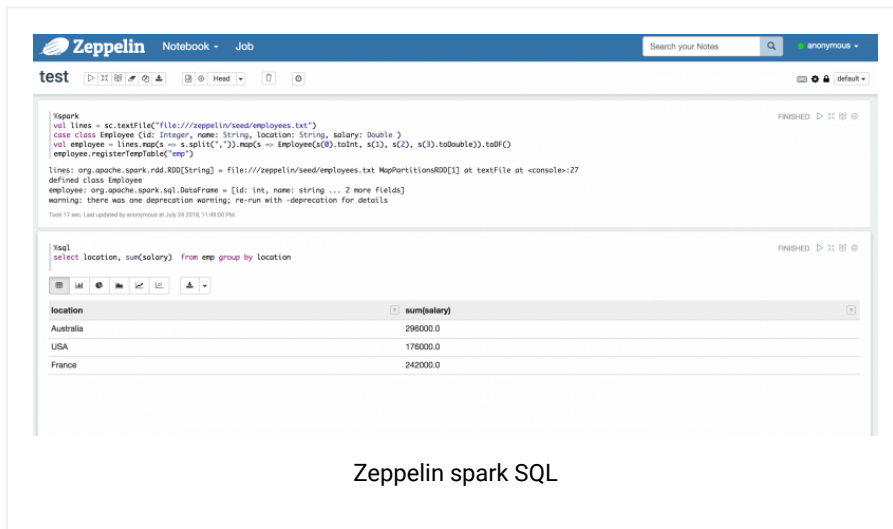
The screenshot shows the Zeppelin Notebook interface. The top bar includes the Zeppelin logo, 'Notebook - Job', a search bar, and a user profile icon labeled 'anonymous'. Below the bar, the notebook title 'test' is displayed. The main area contains two code blocks. The first block is a Spark job that reads a file from 'file:///zeppelin/seed/employees.txt' and registers it as a temporary table named 'emp'. The second block is a SQL query: 'select \* from emp'. Below the SQL query, the output is displayed in a tabular format with columns 'id', 'name', 'location', and 'salary'. The output shows 6 rows of data.

id	name	location	salary
1	John	USA	100000.0
2	Peter	Australia	200000.0
3	Sam	USA	75000.0
4	Daniel	France	85000.0
5	Simon	Australia	95000.0
6	Roseanne	France	155000.0

Zeppelin read a file via Spark

You can group by location, and output the total salary per location with the following query:

```
1 %sql
2
3 select location, sum(salary) from emp group by location
4
```



The screenshot shows the Zeppelin Notebook interface. The top bar includes the Zeppelin logo, 'Notebook - Job', a search bar, and a user dropdown set to 'anonymous'. The main area is titled 'test' and contains two code blocks. The first block is a Spark job that reads a file from the local file system, processes it, and registers it as a table named 'emp'. The second block is a SQL query: `select location, sum(salary) from emp group by location`. Below the SQL query, the results are displayed in a table format.

location	sum(salary)
Australia	296000.0
USA	176000.0
France	242000.0

Zeppelin spark SQL

Alternatively, you can achieve the similar results via Spark dataframe operations as shown below.

```

1 %spark
2
3 employee.groupBy("location").agg(sum("salary")).show()
4

```

```

1
2 +-----+-----+
3 | location|sum(salary)|
4 +-----+-----+
5 | Australia| 296000.0|
6 |      USA| 176000.0|
7 |   France| 242000.0|
8 +-----+-----+
9

```

◀ 01A: Spark on Zeppelin – Docker pull from Docker hub

03: Spark on Zeppelin – DataFrame Operations in Scala ▶

## 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](https://java-success.com)