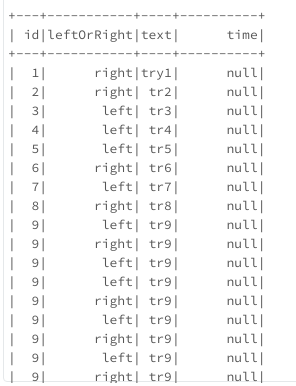
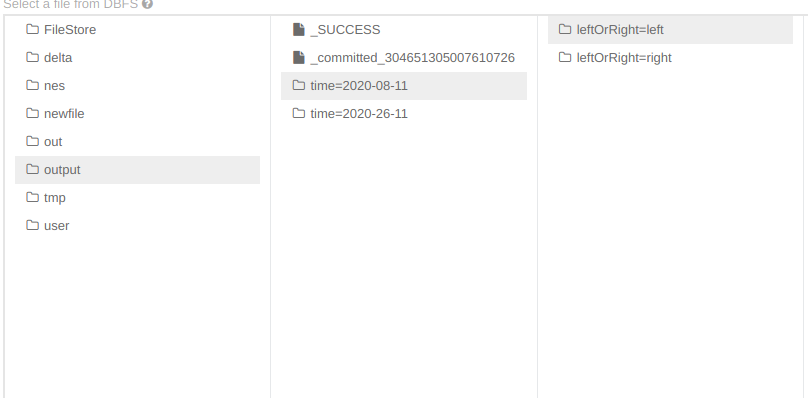
Partition demo

**Input:**



Input is 4 data frames union in one master dataframe.

**Output:**



Output is structured files in our file system.

**Flow**

First we import 4 json files and union them in a single one.

val df4 = spark.read.option("multiline", true).json(args(0))

val df1 = spark.read.option("multiline", true).json(args(0))

val df2 = spark.read.option("multiline", true).json(args(0))

val df3 = spark.read.option("multiline", true).json(args(0))

var df = df4.union(df1).union(df2).union(df3)

var df = df4.union(df1).union(df2).union(df3)

Then we ask if our data frame contains a time column. If the answer is yes we pass through him and where is null we put today date otherwise we live what was there.

try {

df = df.withColumn("time", when(df("time").isNull, now).otherwise(df("time")))

df.show()

}catch {

case e: AnalysisException => {

df = df.withColumn("time", lit(now))

}

}

Finally in the end we save that data frame on some path in partitions time and left or right.

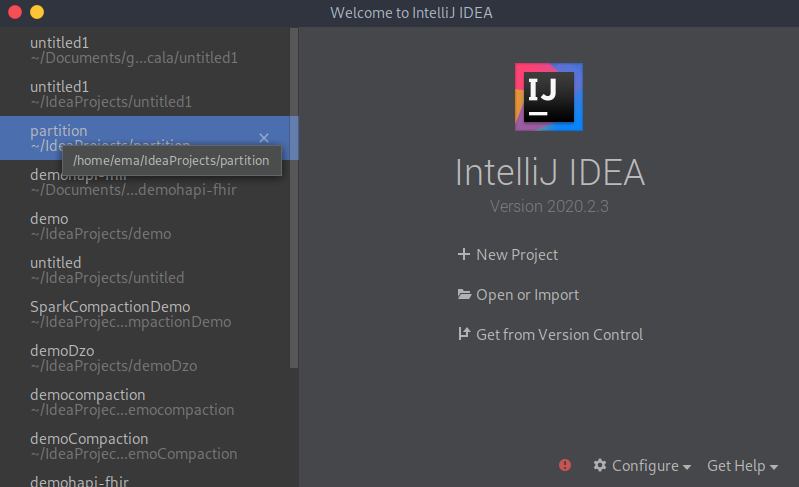
finally {

df.write.partitionBy("time", "leftOrRight").mode("overwrite").json(args(1))

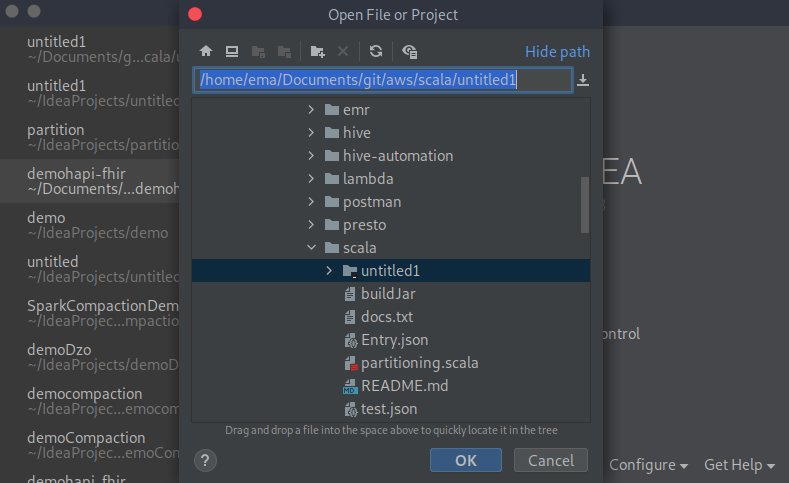
}

**Import project:**

Open intelij idea pop up window and go to Open or Import



Then find project on your disk and select ok



**Create and run jar**

Add build tag:

<build>

<pluginManagement>

<plugins>

<plugin>

<groupId>net.alchim31.maven</groupId>

<artifactId>scala-maven-plugin</artifactId>

<version>4.4.0</version>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>2.0.2</version>

</plugin>

</plugins>

</pluginManagement>

<plugins>

<plugin>

<groupId>net.alchim31.maven</groupId>

<artifactId>scala-maven-plugin</artifactId>

<executions>

<execution>

<id>scala-compile-first</id>

<phase>process-resources</phase>

<goals>

<goal>add-source</goal>

<goal>compile</goal>

</goals>

</execution>

<execution>

<id>scala-test-compile</id>

<phase>process-test-resources</phase>

<goals>

<goal>testCompile</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<executions>

<execution>

<phase>compile</phase>

<goals>

<goal>compile</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<!-- Build an executable JAR -->

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-jar-plugin</artifactId>

<version>2.4</version>

<configuration>

<archive>

<manifest>

<mainClass>sample.Main</mainClass>

</manifest>

</archive>

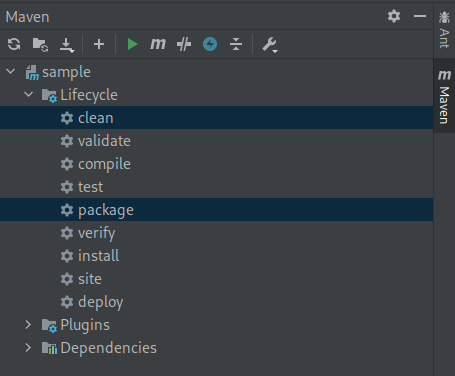
</configuration>

</plugin>

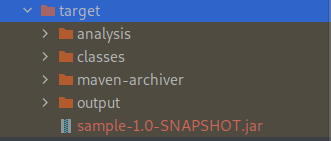
</plugins>

</build>

then go to maven property -> Lifecycle and select clear, package and click run.



That create jar file in target folder



Run:

We execute jar file with spark-submit

**spark-submit sample-1.0-SNAPSHOT.jar s3://scalademo1/Entry.json s3://scalademo1/output**

We pass path to json file and output path as arguments