SparkSession：新的切入点

在Spark 1.x版本，我们收到了很多询问SparkContext, SQLContext和HiveContext之间关系的问题。当人们想使用DataFrame API的时候把HiveContext当做切入点的确有点奇怪。在Spark 2.0，引入了SparkSession，作为一个新的切入点并且包含了SQLContext和HiveContext的功能。为了向后兼容，SQLContext和HiveContext被保存下来。SparkSession拥有许多特性，下面将展示SparkSession的一些重要的功能。

本文是使用Scala编写的，但是Python和Java中同样可用。

Creating a SparkSession

SparkSession可以通过建造者模式创建。如果SparkContext存在，那么SparkSession将会重用它；但是如果SparkContext不存在，则创建它。在build上设置的参数将会自动地传递到Spark和Hadoop中。

```
>   // A SparkSession can be created using a builder pattern
import org.apache.spark.sql.SparkSession
val sparkSession = SparkSession.builder
  .master("local")
  .appName("my-spark-app")
  .config("spark.some.config.option", "config-value")
  .getOrCreate()
import org.apache.spark.sql.SparkSession
sparkSession: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@46d6b87c
```

在Databricks notebooks 和Spark REPL中，SparkSession实例会被自动创建，名字是spark，如下：

```
>   spark
res9: org.apache.spark.sql.SparkSession = org.apache.spark.sql.SparkSession@46d6b87c
```

Unified entry point for reading data

SparkSession是数据读取的切入点，和之前SQLContext读取数据的方式类似：

```
> val jsonData = spark.read.json("/home/webinar/person.json")
jsonData: org.apache.spark.sql.DataFrame = [email: string, iq: bigint ... 1 more field]

> display(jsonData)
```

email	iq	name
matei@databricks.com	180	Matei Zaharia
rxin@databricks.com	80	Reynold Xin

Running SQL queries

SparkSession可以在数据上运行SQL查询，并且将结果作为DataFrame返回(比如 Dataset[Row])

```
> display(spark.sql("select * from person"))
```

email	iq	name
matei@databricks.com	180	Matei Zaharia
rxin@databricks.com	80	Reynold Xin

Working with config options

SparkSession可以在运行时设置一些参数，这些参数可以触发性能优化，或者I/O行为：

```
> spark.conf.set("spark.some.config", "abcd")
res12: org.apache.spark.sql.RuntimeConfig = org.apache.spark.sql.RuntimeConfig@55d93752

> spark.conf.get("spark.some.config")
res13: String = abcd
```

我们还可以在sql中通过变量替换来传递配置

```
> %sql select "${spark.some.config}"
abcd
```

abcd

Working with metadata directly

SparkSession中还包含了catalog方法，可以使用它来和metastore打交道，结果返回类型是DataSet，所有你可以直接在上面使用DataSet的方法。

```
>  // To get a list of tables in the current database
val tables = spark.catalog.listTables()
tables: org.apache.spark.sql.Dataset[org.apache.spark.sql.catalog.Table] = [name: string, database: string ... 3 more fields]

>  display(tables)
name  database  description  tableType  isTemporary

person  default  null  MANAGED  false

smart  default  null  MANAGED  false
>  // Use the Dataset API to filter on names
display(tables.filter(_.name contains "son"))
name  database  description  tableType  isTemporary

person  default  null  MANAGED  false
>  // Get the list of columns for a table
display(spark.catalog.listColumns("smart"))
name  database  dataType  nullable  isPartition  isBucket

email  null  string  true  false  false

iq  null  bigint  true  false  false

name  null  string  true  false  false
```

Access to the underlying SparkContext

SparkSession.sparkContext方法将会返回内置的SparkContext，可以使用它创建RDD或者是管理集群资源：

```
>  spark.sparkContext
res17: org.apache.spark.SparkContext = org.apache.spark.SparkContext@2debe9ac
```

本文翻译自：https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/6122906529858466/431554386690884/4814681571895601/latest.html