Spark 2.0介绍：SparkSession创建和使用相关API

《Spark 2.0技术预览：更容易、更快速、更智能》文章中简单地介绍了Spark 2.0带来的新技术等。Spark 2.0是Apache Spark的下一个主要版本。此版本在架构抽象、API以及平台的类库方面带来了很大的变化，为该框架明年的发展方向奠定了方向，所以了解Spark 2.0的一些特性对我们能够使用它有着非常重要的作用。本博客将对Spark 2.0进行一序列的介绍（参见Spark 2.0分类），欢迎关注。

SparkSession-Spark的一个全新的切入点

在Spark的早期版本，sparkContext是进入Spark的切入点。我们都知道RDD是Spark中重要的API，然而它的创建和操作得使用sparkContext提供的API；对于RDD之外的其他东西，我们需要使用其他的Context。比如对于流处理来说，我们得使用StreamingContext；对于SQL得使用sqlContext；而对于hive得使用HiveContext。然而DataSet和Dataframe提供的API逐渐称为新的标准API，我们需要一个切入点来构建它们，所以在 Spark 2.0中我们引入了一个新的切入点(entry point)：SparkSession

SparkSession实质上是SQLContext和HiveContext的组合（未来可能还会加上StreamingContext），所以在SQLContext和HiveContext上可用的API在SparkSession上同样是可以使用的。SparkSession内部封装了sparkContext，所以计算实际上是由sparkContext完成的。

下面我将讨论如何创建和使用SparkSession。

创建SparkSession

SparkSession的设计遵循了工厂设计模式（factory design pattern），下面代码片段介绍如何创建SparkSession

```
val sparkSession = SparkSession.builder.
    master("local")
    .appName("spark session example")
    .getOrCreate()
```

上面代码类似于创建一个SparkContext，master设置为local，然后创建了一个SQLContext封装它。如果你想创建hiveContext，可以使用下面的方法来创建SparkSession，以使得它支持Hive：

```
val sparkSession = SparkSession.builder.
    master("local")
    .appName("spark session example")
    .enableHiveSupport()
```

```
    .getOrCreate()
```

enableHiveSupport 函数的调用使得SparkSession支持hive，类似于HiveContext。

使用SparkSession读取数据

创建完SparkSession之后，我们就可以使用它来读取数据，下面代码片段是使用SparkSession来从csv文件中读取数据：

```
val df = sparkSession.read.option("header","true").
  csv("src/main/resources/sales.csv")
```

上面代码非常像使用SQLContext来读取数据，我们现在可以使用SparkSession来替代之前使用SQLContext编写的代码。下面是完整的代码片段：

```
package com.iteblog

import org.apache.spark.sql.SparkSession

/**
 * Spark Session example
 *
 */
object SparkSessionExample {

  def main(args: Array[String]) {

    val sparkSession = SparkSession.builder.
      master("local")
      .appName("spark session example")
      .getOrCreate()

    val df = sparkSession.read.option("header","true").csv("src/main/resources/sales.csv")

    df.show()

  }

}
```

Spark 2.0现在还支持SQLContext和HiveContext吗？

并没有，Spark的设计是向后兼容的，所有SQLContext和HiveContext相关的API在Spark 2.0还是可以使用的。不过既然有SparkSession了，所有大家还是尽量在Spark 2.0中使用它。