Apache spark tutorial java pdf

I'm not robot
reCAPTCHA

Continue

The Spark Java API exposes all of the spark features available in the Scala version to Java. To learn the basics of Spark, it's a good idea to read the Scala programming guide first. Even if you don't know Scala, it should be easy to follow. This guide will show you how to use the spark features described in Java. The Spark Java API is defined in the org.apache.spark.api.java package and includes JavaSparkContext to initialize the Spark and JavaRDD classes that support the same methods as Scala, but perform java functions to return Java data and collection types. The main difference is related to passing a function to an RDD operation (for example, a map) and handling other types of RDDs as described next. The main differences in Java APIs are that there are some key differences between java APIs: Java does not support anonymous or first-class features, so you should extend classes such as org.apache.spark.api.java.function, function2, and implement functionality. To maintain type safety, the Java API defines special functions and RDD classes for key value pairs and doubles. For example, JavaPairRDD stores a pair of key value. RDD methods such as collection() and countByKey() return java collection types such as java.util.List and java.util.Map. In Scala, a pair of key values written briefly as (key, value) is represented by Scala. Tuple2 class, must be created using the new Tuple2<K, v=>(key, value). RDD class sparks define additional actions for key value pairs and double RDDs such as Reduced ByKey, Joins, and Stdev. The Scala API uses Scala's implicit conversion mechanism to automatically add these methods. Additional methods in the Java API are defined in the JavaPairRDD and JavaDoubleRDD classes. RDD methods, such as maps, are overloaded by special pairing saorities and DoubleFunction classes, so you can return the appropriate type of RDD. Because common methods such as filters and samples are implemented by each professional RDD class, filtering pairRDDreturns a new PairRDD, etc., (this method triggers the same result type principles used by the Scala collection framework). The following table of function classes lists the function classes used by the Java API. Each class has a call () which is a single abstract method that must be implemented. Class feature type <T, r=>function T => R Double Functions&lt>T>T> Double Pair Features T<T,k,==>Tuple2>K, v=>lt;T;T, r=>FlatMapFunction T=>T>T>T>DoubleMapFunction =>T>DoubleMapFunction Iterative Fair Map<t, k,= v=>function T=<lt;>> function2>T1, t2,= r=>T1, T2=> R (function of both arguments) storage level RDD storage level constant (e.MEMORY_AND_DISK.a. To define your own storage level, you can use StorageLevel.create (...) you can use. Other features Java API supports other spark features, including accumulators, broadcast variables, and caching. For example, implement word counts.</T1, > > K,>>-T&T;Lt;Double;T;T;T;-R>T;-T;--Lt;-K,>>T&T&T&T>T>T&gt