

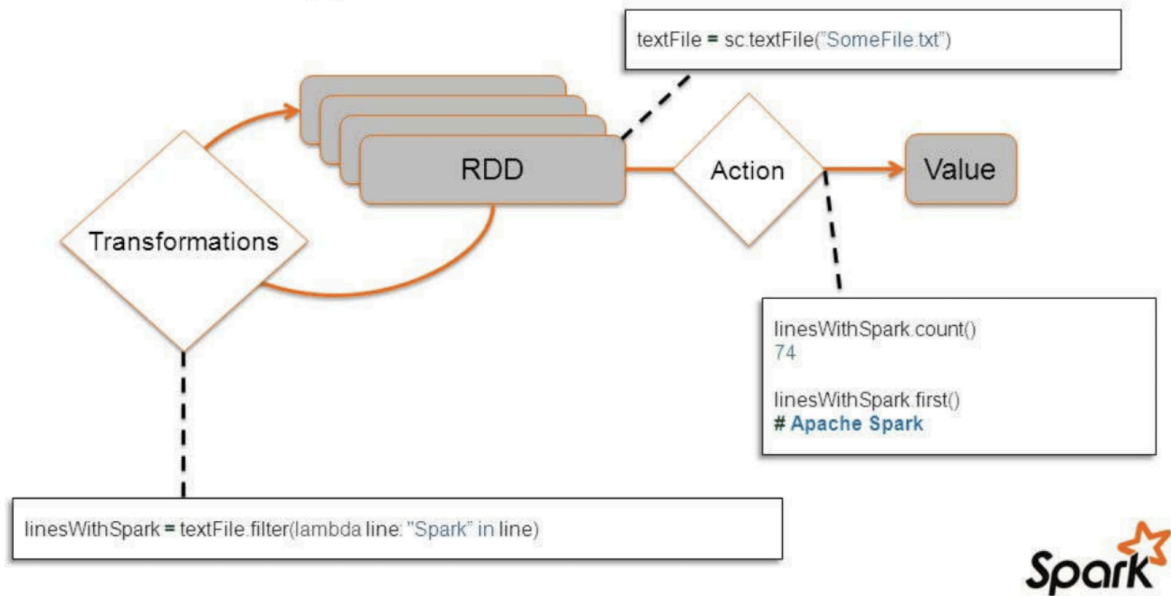
Bu bölümde RDD action metodları hakkında genel bilgiler vereceğiz

Örnekler için

<http://grouplens.org/datasets/movielens/adresinden>

veri indirebiliriz. Action metotları genel olarak Spark verileri üzerinde hesaplama yada dış sistemlere veri aktarma işlemleri yapmamızı sağlar

Working With RDDs



Count

RDD, Dataset içerisindeki kayıt sayısını verir

```
import org.apache.spark.sql.SparkSession

object MovieTransformationCount {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample").
      getOrCreate()

    val
    rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/ApacheSpark/ml-latest-small/movies.csv");
    println("Count : " + rdd.count())
    println("First : " + rdd.first())

    //1.parametre > Sampling with Replacement(false ise aynı kayıtlar gelebilir
    //2.parametre > yüzde olarak kaç tane seçilecek
    val count = rdd.count()

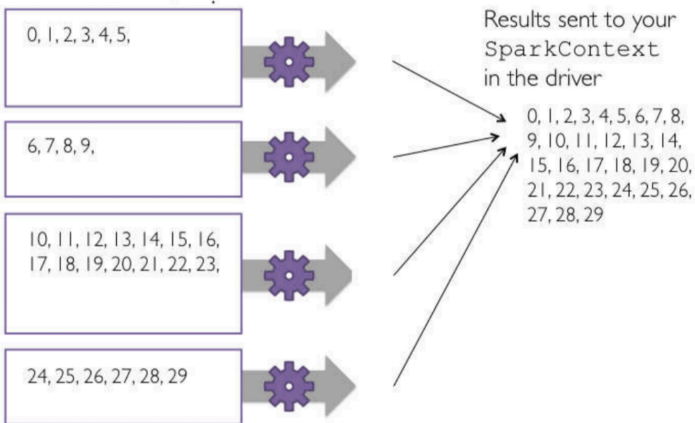
    println("Movie RDD Count : " + count)

  }
}
```

Collect

Worker makineleri üzerinde bulunan veriler **driver** üzerinde List veri yapısında toplanır.

`collect()` : Gathers the entries from all partitions into the driver



spark-collect

```
import org.apache.spark.sql.SparkSession

object MovieTransformationCollect {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample").
      getOrCreate()

    val
    rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/ApacheSpark/ml-latest-small/movies.csv");
    println("Count : " + rdd.count())
    println("First : " + rdd.first())

    val list = rdd.collect()

    println("Movie RDD Count : " + list.length)

  }
}
```

First

RDD, Dataset içerisindeki ilk kaydı verir

```
import org.apache.spark.sql.SparkSession

object MovieTransformationFirst {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample").
      getOrCreate()

    val
    rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/ApacheSpark/ml-latest-small/movies.csv");
    println("Count : " + rdd.count())
    println("First : " + rdd.first())

    val row = rdd.first()

    println("Movie RDD First row : " + row)

  }
}
```

Take

RDD ve Dataset içerisinde parametre olarak verilen sayı kadar kayıt verir(ilk n kayıt)

```
import org.apache.spark.sql.SparkSession

object MovieTransformationTake {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample").
      getOrCreate()

    val
    rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/ApacheSpark/ml-latest-small/movies.csv");
    println("Count : " + rdd.count())
    println("First : " + rdd.first())

    val list = rdd.take(2)

    println("Two records : " + list.length)

  }
}
```

takeSample

RDD ve Dataset içerisinde parametre olarak verilen sayı kadar örnek kayıt verir

```
import org.apache.spark.sql.SparkSession

object MovieTransformationTakeSample {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample").
      getOrCreate()

    val
    rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/ApacheSpark/ml-late
st-small/movies.csv");
    println("Count : " + rdd.count())
    println("First : " + rdd.first())

    //random value - true (same data can consist)
    val list = rdd.takeSample(true,2)

    for (row <-list)
      println(row)

  }
}
```

saveAsTextFile

Local bilgisayar sistemine yada HDFS'e verikaydetmemizi sağlar

```
import org.apache.spark.sql.SparkSession

object MovieTransformationSaveAsText {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample").
      getOrCreate()

    val
    rdd=spark.sparkContext.textFile("/Users/serkan/Desktop/Training/ApacheSpark/ml-latest-small/movies.csv");
    println("Count : " + rdd.count())
    println("First : " + rdd.first())

    val filteredRdd = rdd.filter(row => row.startsWith("20"))

    //save as text

    filteredRdd.saveAsTextFile("/Users/serkan/Desktop/Training/ApacheSpark/ml-latest-small/output")

    //save to hdfs
    //rdd.saveAsTextFile("hdfs://quickstart.cloudera:8020/user/data")

  }
}
```

takeOrdered ve foreach

takeOrdered RDD ve Dataset içerisindeki verileri belirli bir sıralamaya göre getirir. Verdiğimiz parametre ise kaç tane kayıt getirileceğini belirtir.

Foreach ise bir metodun(fonksiyonun) tüm RDD ve Dataset elemanları için çalıştırılmasını sağlar

```
import org.apache.spark.sql.SparkSession
import scala.math.Ordering

case class Person(name:String, age:Int)

object MovieTransformationTakeOrdered {
  def main(args: Array[String]): Unit = {

    val spark = SparkSession.builder.master("local").
      appName("SparkByExample").
      getOrCreate()

    val people = Array(Person("bob", 30), Person("ann", 32), Person("carl", 19))
    val rdd = spark.sparkContext.parallelize(people, 2)
    rdd.takeOrdered(1)(Ordering[Int].reverse.on(x => x.age))

    rdd.foreach(row => {
      println("Name : " + row.name)
    })

  }
}
```