

Apache Spark API By Example

A Command Reference for Beginners

Matthias Langer, Zhen He
Department of Computer Science and Computer Engineering
La Trobe University
Bundoora, VIC 3086
Australia
m.langer@latrobe.edu.au, z.he@latrobe.edu.au

May 31, 2014

Contents

1	Preface	5
2	Shell Configuration	6
2.1	Adjusting the amount of memory	6
2.2	Adjusting the number of worker threads	6
2.3	Adding a Listener to the Logging System	6
3	The RDD API	7
3.1	aggregate	8
3.2	cartesian	10
3.3	checkpoint	10
3.4	coalesce, repartition	11
3.5	cogroup ^[Pair] , groupWith ^[Pair]	11
3.6	collect, toArray	12
3.7	collectAsMap ^[Pair]	12
3.8	combineByKey ^[Pair]	13
3.9	compute	13
3.10	context, sparkContext	14
3.11	count	14
3.12	countApprox	14
3.13	countByKey ^[Pair]	14
3.14	countByKeyApprox ^[Pair]	15
3.15	countByValue	15
3.16	countByValueApprox	15
3.17	countApproxDistinct	16
3.18	countApproxDistinctByKey ^[Pair]	16
3.19	dependencies	17
3.20	distinct	17
3.21	first	18
3.22	filter	18
3.23	filterWith	19
3.24	flatMap	20
3.25	flatMapValues ^[Pair]	20
3.26	flatMapWith	21
3.27	fold	21
3.28	foldByKey ^[Pair]	22

3.29	foreach	22
3.30	foreachPartition	22
3.31	foreachWith	23
3.32	generator, setGenerator	23
3.33	getCheckpointFile	23
3.34	preferredLocations	24
3.35	getStorageLevel	24
3.36	glom	25
3.37	groupBy	25
3.38	groupByKey ^[Pair]	26
3.39	histogram ^[Double]	27
3.40	id	27
3.41	isCheckpointed	28
3.42	iterator	28
3.43	join ^[Pair]	28
3.44	keyBy	29
3.45	keys ^[Pair]	29
3.46	leftOuterJoin ^[Pair]	30
3.47	lookup ^[Pair]	30
3.48	map	31
3.49	mapPartitions	31
3.50	mapPartitionsWithContext	32
3.51	mapPartitionsWithIndex	33
3.52	mapPartitionsWithSplit	33
3.53	mapValues ^[Pair]	34
3.54	mapWith	34
3.55	mean ^[Double] , meanApprox ^[Double]	35
3.56	name, setName	35
3.57	partitionBy ^[Pair]	35
3.58	partitioner	36
3.59	partitions	36
3.60	persist, cache	36
3.61	pipe	37
3.62	reduce	37
3.63	reduceByKey ^[Pair] , reduceByKeyLocally ^[Pair] , reduceByKeyToDriver ^[Pair]	37
3.64	rightOuterJoin ^[Pair]	38
3.65	sample	38
3.66	saveAsHadoopFile ^[Pair] , saveAsHadoopDataset ^[Pair] , saveAsNewAPIHadoopFile ^[Pair]	39
3.67	saveAsObjectFile	39
3.68	saveAsSequenceFile ^[SeqFile]	40
3.69	saveAsTextFile	40
3.70	stats ^[Double]	41
3.71	sortByKey ^[Ordered]	42
3.72	stdev ^[Double] , sampleStdev ^[Double]	42

3.73	subtract	43
3.74	subtractByKey ^[Pair]	43
3.75	sum ^[Double] , sumApprox ^[Double]	44
3.76	take	44
3.77	takeOrdered	45
3.78	takeSample	45
3.79	toDebugString	46
3.80	toJavaRDD	46
3.81	top	46
3.82	toString	47
3.83	union, ++	47
3.84	unpersist	47
3.85	values ^[Pair]	48
3.86	variance ^[Double] , sampleVariance ^[Double]	48
3.87	zip	48
3.88	zipPartitions	49
4	Further Topics	51
4.1	Reading from HDFS	51

1 Preface

Spark is an advanced open-source cluster computing system that is capable of handling extremely large data sets. It was first published by ? and its popularity has increased ever since. Due to its real-time properties and efficient usage of resources, Spark has become a very popular alternative to well established computational software for big data.

Spark is still actively being maintained and further developed by its original creators from UC Berkeley. Hence, this command reference and the associated, including the code-snippets and sample outputs outputs shown, should be considered as a overview of the status-quo of this amazing piece of software technology. Specifically, the API examples in this document are for **Spark version 0.9**. However, we do not expect the API to change much in future releases.

This document does not cover any installation or distribution related topics. For installation instructions, please refer to the Apache Spark website.

[Click here to download full PDF material](#)