

+ timeTrackingStream:OutputStream

+ def write(key: Any, value: Any): Unit

+ def closeIfNonNull[T <: Closeable]

+ def updateBytesWritten(): Unit

- + def createMetricsReporter(context: TaskContext):

+ def write(rdd: RDD[],dep: ShuffleDependency[, _, _]

,mapId: Long,context: TaskContext,partition: Partition):

ShuffleWriteProcessor

+ objOut:SerializationStream

+ def recordWritten(): Unit

+ wrappedStream

+ def open(): Unit

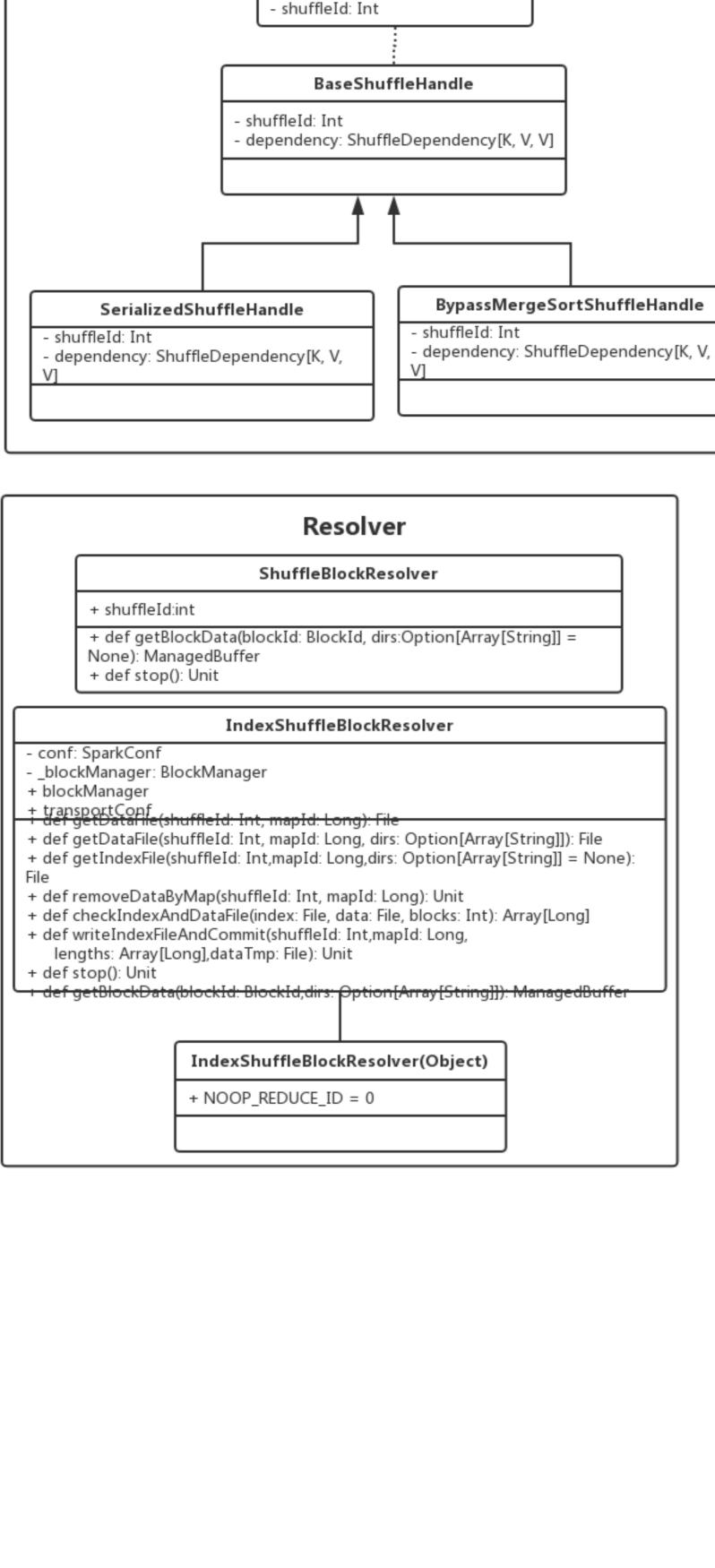
+ def close(): Unit

(closeable: T): T

ShuffleWriteMetricsReporter

+ numRecordsWritten

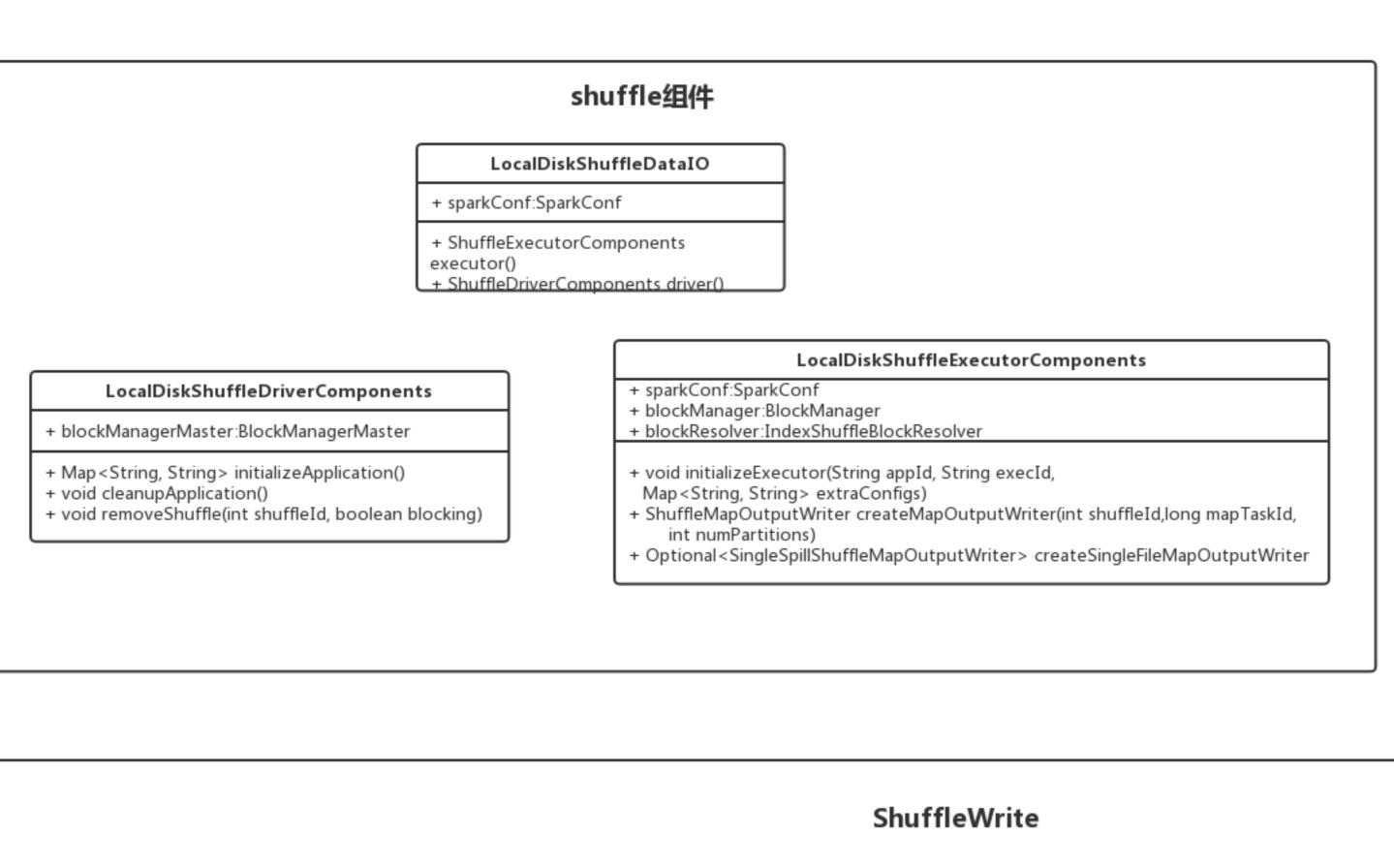
+ curNumBytesWritten



Handle

ShuffleHandle

MetricsReporter Exception ShuffleWriteMetricsReporter FetchFailedException + def incBytesWritten(v: Long): Unit - bmAddress: BlockManagerId + def incRecordsWritten(v: Long): Unit shuffleId: Int + def incWriteTime(v: Long): Unit - mapId: Long + def decBytesWritten(v: Long): Unit mapIndex: Int + def decRecordsWritten(v: Long): Unit reduceId: Int - message: String - cause: Throwable ShuffleReadMetricsReporter + def to TaskFailedReason: Task Failed Reason MetadataFetchFailedExceptio + def incRemoteBlocksFetched(v: Long): Unit + def incLocalBlocksFetched(v: Long): Unit - shuffleId: Int + defincRemoteBytesRead(v: Long): Unit reduceId: Int + defincRemoteBytesReadToDisk(v: Long): Unit - message:Any + def incLocalBytesRead(v: Long): Unit + defincFetchWaitTime(v: Long): Unit + def incRecordsRead(v: Long): Unit



+ void copyRange(LongArray src, int srcPos, LongArray dst, int dstPos, int length)

