

GenericAvroSerializer
<div>- schemas: Map[Long, String]</div> <div>+ compressCache.HashMap[Schema, Array[Byte]]</div> <div>+ decompressCache.HashMap[ByteBuffer, Schema]</div> <div>+ writerCache.HashMap[Schema, DatumWriter[_]]</div> <div>+ readerCache.HashMap[Schema, DatumReader[_]]</div> <div>+ fingerprintCache.HashMap[Schema, Long]</div> <div>+ schemaCache.HashMap[Long, Schema]</div> <div>+ codec.CompressionCodec</div>
<div>+ def compress(schema: Schema): Array[Byte]</div> <div>+ def decompress(schemaBytes: ByteBuffer): Schema</div> <div>+ def serializeDatum[R <: GenericRecord](datum: R, output: KryoOutput): Unit</div> <div>+ def deserializeDatum(input: KryoInput): GenericRecord</div>

