

KIP-616: Rename implicit Serdes instances in kafka-streams-scala

- [Status](#)
- [Motivation](#)
- [Proposed Change](#)
- [Changed public Interfaces](#)
- [Migration Plan, Compatibility and Deprecation](#)
- [Rejected Alternatives](#)
 - [Rename serdes in the old org.apache.kafka.streams.scala.Serdes](#)

Status

Current state: *Accepted*

Voting thread: [here](#)

Discussion thread: [here](#)

JIRA:

 Unable to render Jira issues macro, execution error.

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

Motivation

Currently `org.apache.kafka.streams.scala.Serdes` contains implicit `Serde` instances. And some of these implicits are named the same as its `Serde` type. For example:

```
implicit def Long: Serde[Long] = ???
```

This naming leads to a name clash after wildcard import:

```
import org.apache.kafka.streams.scala.Serdes._

val x = String.valueOf(5) // Error:(7, 18) value valueOf is not a member of org.apache.kafka.common.serialization.Serde[String]
println(x)
```

Because wildcard import is the way for getting implicits in the current scope, there is no way to use these implicits without possible naming issues.

Personally, I faced this issue on the first day of using `kafka-streams-scala` library. And many of my colleagues faced it too.

This KIP was created as a result of [this](#) pull request.

Proposed Change

Create a new `org.apache.kafka.streams.scala.serialization.Serdes`. It will be like an old one, but with other implicits names. The old version of `Serdes` should be marked as deprecated.

Also, some default Java serdes are missing from the `org.apache.kafka.streams.scala.Serdes`. Since we are creating a new version of `Serdes`, it also makes sense to add the missing default Java serdes to it.

Changed public Interfaces

New serdes in the `org.apache.kafka.streams.scala.serialization.Serdes` object should look like these:

```
object Serdes {
  implicit def stringSerde: Serde[String] = ???
  implicit def longSerde: Serde[Long] = ???
  implicit def javaLongSerde: Serde[java.lang.Long] = ???
  implicit def byteArraySerde: Serde[Array[Byte]] = ???
  implicit def bytesSerde: Serde[org.apache.kafka.common.utils.Bytes] = ???
  implicit def byteBufferSerde: Serde[ByteBuffer] = ???
  implicit def shortSerde: Serde[Short] = ???
  implicit def javaShortSerde: Serde[java.lang.Short] = ???
  implicit def floatSerde: Serde[Float] = ???
  implicit def javaFloatSerde: Serde[java.lang.Float] = ???
  implicit def doubleSerde: Serde[Double] = ???
  implicit def javaDoubleSerde: Serde[java.lang.Double] = ???
  implicit def intSerde: Serde[Int] = ???
  implicit def javaIntegerSerde: Serde[java.lang.Integer] = ???
  implicit def uuidSerde: Serde[UUID] = ???

  implicit def sessionWindowedSerde[T](implicit tSerde: Serde[T]): WindowedSerdes.SessionWindowedSerde[T] = ???

  def fromFn[T >: Null](serializer: T => Array[Byte], deserializer: Array[Byte] => Option[T]): Serde[T] = ???

  def fromFn[T >: Null](serializer: (String, T) => Array[Byte],
    deserializer: (String, Array[Byte]) => Option[T]): Serde[T] = ???
}
```

Not all old names have name clash, but I think it worth to rename all implicits for consistency. Also, new Serdes will contain `uuidSerde`, `shortSerde`, `javaShortSerde` and `byteBufferSerde`, which are missing in the old `org.apache.kafka.streams.scala.Serdes`.

Note, we're not providing an implicit for `timeWindowedSerde` because there's no way to automatically configure it properly (which requires knowing the window size). See [KIP-659: Improve TimeWindowedDeserializer and TimeWindowedSerde to handle window size](#).

Migration Plan, Compatibility and Deprecation

Proposed changes are backward compatible. The old code remains the same.

An old `org.apache.kafka.streams.scala.Serdes` will be marked as deprecated and could be deleted with the major 3.0 release.

Rejected Alternatives

Rename serdes in the old `org.apache.kafka.streams.scala.Serdes`

Renaming serdes will require a complex migration plan. We would need to create a new serdes with the new naming near old serdes. Old serdes will be marked as deprecated. And old, deprecated serdes, could be deleted only with the major 3.0 release. It's too far away from now and too complex for such change.