

# KIP-118: Drop Support for Java 7

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)

## Status

**Current state:** *Adopted in 2.0.0*

**Discussion thread:** <http://search-hadoop.com/m/uyzND1MrUwP522Y52>

**Older Discussion thread:** <http://search-hadoop.com/m/Kafka/uyzND1olhV61GS5Sf2>

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

Java 7 hasn't received public updates since [April 2015](#), Java 8 was released in [March 2014](#) and Java 9 is scheduled to be released in [July 2017](#).

The last public release of Java 7 by Oracle contains a large number of known security vulnerabilities and the effectiveness of Kafka's security features is reduced if the underlying runtime is not itself secure.

The other side of the coin is that Java 8 adds a number of compelling features:

- Lambda expressions and method references (particularly useful for the Kafka Streams DSL)
- Default methods (very useful for maintaining compatibility when adding methods to interfaces)
- `java.util.stream` (helpful for making collection transformations more concise)
- Lots of improvements to `java.util.concurrent` (`CompletableFuture`, `DoubleAdder`, `DoubleAccumulator`, `StampedLock`, `LongAdder`, `LongAccumulator`)
- Other nice things: `SplittableRandom`, `Optional` (and many others I have not mentioned)

Additionally, dropping Java 7 will allow us to support the impending Java 9 without making our testing matrix even more complex (this is particularly useful for system tests that take hours to run). It will also make it easier to support Scala 2.12 properly (we don't currently build with Scala 2.12 by default as it requires Java 8).

Many other open-source projects have taken the leap already. Examples are [Apache Cassandra](#) (and [Java driver](#)), [Apache Spark](#), [Apache Lucene](#), [Apache Storm](#), [Akka](#), [Apache Hadoop 3](#), [Jetty](#), [Guava](#), [Eclipse](#), [IntelliJ](#), [Jenkins](#) and [many others](#). Even Android will support Java 8 in the next version (although it will take a while before most phones will use that version sadly). This reduces (but does not eliminate) the chance that we would be the first project that would cause a user to consider a Java upgrade.

## Public Interfaces

None.

## Proposed Changes

Change `sourceCompatibility` from 1.7 to 1.8 in `build.gradle`.

## Compatibility, Deprecation, and Migration Plan

Users who cannot upgrade to Java 8 can continue to use Kafka 0.10.x until they can upgrade. 0.11.x brokers will support 0.10.x clients and 0.11.x clients will support 0.10.x brokers, so upgrades can be incremental (while keeping the older message format until enough clients have been upgraded). This is useful if some clients and/or brokers are stuck in Java 7.

## Rejected Alternatives

1. Continue supporting Java 7: the benefit of dropping support outweighs the cost.
2. Continue supporting Java 7 in clients library: in some cases, users have legacy clients where upgrading to a new Java version is difficult and they would prefer if the clients library would still support Java 7. Because this tends to affect legacy clients, they can continue using Kafka 0.10.x with newer brokers (if and when the brokers are upgraded).