



# Other Communication APIs Supported by Spring

In this lesson, we'll look at more communication APIs that the Spring framework can support.



**{ R E S T } A P I**

Spring Boot also supports the **JAX RS API**. For **JAX RS**, Spring Boot uses the library **Jersey**. JAX RS is standardized as part of the **Java Community Process (JCP)**.

For messaging, Spring Boot supports the **Java Messaging Service (JMS)**.

This is a standardized API that can be used to address different messaging solutions from Java.

Spring Boot has starters for the JMS implementations HornetQ (<http://hornetq.jboss.org/>), ActiveMQ (<http://activemq.apache.org/>) and ActiveMQ Artemis (<https://activemq.apache.org/artemis/>). In addition, there is a Spring Boot starter for AMQP (<https://www.amqp.org/>). This protocol is also a standard, but on the network protocol level.

The AMQP starter uses RabbitMQ (<https://www.rabbitmq.com/>) as an implementation of the protocol. For AMQP as well as for JMS, Spring offers an API that makes it easier to send messages. In addition, simple Java objects (Plain Old Java Objects, POJOs) with no dependencies on any of the APIs can process AMQP and JMS messages with Spring and also return responses to messages.

Spring Cloud offers Spring Cloud Streams (<https://cloud.spring.io/spring-cloud-stream/>) for implementing applications for the processing of data streams. This library supports messaging systems such as Kafka, RabbitMQ (see above) and Redis (<https://redis.io/>).

Spring Cloud Stream builds on these technologies and extends them with concepts such as streams and therefore goes beyond just simplifying the use of the technology's APIs.

The integration of technologies in Spring Boot with Spring Boot starters has the advantage that Spring Boot provides the configuration of the environment.

The example Spring Boot application in this chapter uses an infrastructure such as a Tomcat server to handle HTTP requests. This does not require a separate configuration and no additional dependencies. Spring Boot starters also offer such simplifications for messaging and other REST technologies.

Spring Boot applications can also use technologies without a Spring Boot starter. A Spring Boot application can use any technology that supports Java.

In the end, a Spring Boot project is a Java project and can be extended with Java libraries.

However, it is possible that the configuration can be more complex than a Spring Boot starter.

# QUIZ

## Z

1 What is Kafka?

- ☐ A) A 20th century German-speaking novelist
- ☐ B) A messaging system that processes data streams
- ☐ C) A JMS implementation starter

**Submit Answer**



Question 1 of 2  
0 attempted



**Reset Quiz**



In the next lesson, we'll look at how Spring Boot fulfills the operation requirement for a microservice.

**← Back**

Spring Boot for Microservices: Commu...

**Next →**

Spring Boot for Microservices: Operati...



Mark as Completed



Report an Issue