# Polling Consumer

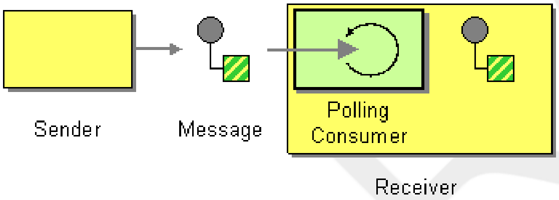
## Overview

The *polling consumer* pattern is a pattern for implementing the consumer endpoint in a Mediation Router component and is thus only relevant to programmers who need to develop a custom component in Mediation Router. Existing components already have a consumer implementation pattern hard-wired into them.

Consumers that conform to this pattern expose polling methods, `receive()`, `receive(long timeout)`, and `receiveNoWait()` that return a new exchange object, if one is available from the monitored resource. A polling consumer implementation must provide its own thread pool to perform the polling.

For more details about this implementation pattern, see Consumer Patterns in the *Programmer's Guide* and *Consumer Interface* in the *Programmer's Guide*.

*Figure 27. Polling Consumer Pattern*



## Scheduled poll component

Many of the Mediation Router source endpoints employ a scheduled poll pattern to receive messages at the start of a route. That is, the endpoint appears to implement an event-driven consumer interface, but internally a scheduled poll is used to monitor a resource that provides the incoming messages for the endpoint.

See Implementing the Consumer Interface in the *Programmer's Guide* for details of how to implement this pattern.

## Quartz component

You can use the quartz component to provide scheduled delivery of messages using the *Quartz* enterprise scheduler. See Quartz in the *Component Reference* and Quartz Component [http://activemq.apache.org/camel/quartz.html] for details.