

# JMS Transport

## Using the Java Message Service (JMS)

CXF provides a transport plug-in that enables endpoints to use Java Message Service (JMS) queues and topics. CXF's JMS transport plug-in uses the Java Naming and Directory Interface (JNDI) to locate and obtain references to the JMS provider that brokers for the JMS destinations. Once CXF has established a connection to a JMS provider, CXF supports the passing of messages packaged as either a JMS `ObjectMessage` or a JMS `TextMessage`. The JMS transport also supports the [SOAP over JMS specification](#). See [SOAP over JMS 1.0 support](#) for more information.

### Configuration styles

- [SOAP / JMS spec compliant configuration](#) - recommended style for CXF 3
- [JMSSConfiguration style](#) - works in CXF 2 and 3 but is deprecated for CXF 3
- [CXF 2.x JMS configuration](#) - removed in CXF 3

### Advanced examples (CXF >= 3)

- [Performance and pooling](#)
- [JMS transactions](#)

### Reconnect functionality

The JMS transport automatically reconnects when any JMS exceptions happen. The first exception will be logged with "Exception on JMS connection. Trying to reconnect" and the exception details.

Then CXF will try to reconnect every 5 seconds and log "Exception on reconnect. Trying again, attempt num x".

In conduits every exception when sending a message will be thrown to the caller. The conduit then closes the connection and will start a new one when sending the next message.



#### Reconnects and pooling

When using pooling make sure the pool throws away the Connection in case of errors. In ActiveMQ `PooledConnection` this can be achieved using `setExpiryTimeout(milliseconds)`.

In CXF 2.x the JMS server must be present before JMS endpoints are started. In CXF 3 the endpoint will start and periodically reconnect like in the case of an exception while running.