LOGZILLA DOCUMENTATION

Receiving Java Events

Forward events from Java applications to LogZilla using Log4j 2 syslog appenders over TCP with RFC5424 formatting, facility, and program name

Receiving Data in LogZilla · Generated April 29, 2026 · logzilla.ai/docs/receiving-data/receiving-java-events

Receiving Java Events

Java applications can forward events to LogZilla using syslog (recommended) or HTTP/HTTPS. The examples below show minimal, production-safe settings.

Prerequisites

- Confirm syslog listener ports in LogZilla (see Syslog Settings (https://www.logzilla.ai/docs/administration/syslog-settings)).
- Choose the facility and a clear program name for the app (for example, `LOCAL0` and `MyJavaApp`).

Option A: Log4j 2 (syslog over TCP, RFC5424)

Create or edit `log4j2.xml` and define a `Syslog` appender. Replace `LOGZILLA_HOST` with the LogZilla server address and adjust the program name and facility as needed.

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="WARN">
  <Appenders>
    <Syslog name="LZSyslog"
            host="LOGZILLA_HOST"
            port="514"
            protocol="TCP"
            format="RFC5424"
            appName="MyJavaApp"
            facility="LOCAL0"/>
  </Appenders>
  <Loggers>
    <Root level="info">
      <AppenderRef ref="LZSyslog"/>
    </Root>
  </Loggers>
  </Configuration>
```

Restart the Java application after saving changes.

Option B: Log4j 1.x (legacy)

For Log4j 1.x, add a `SyslogAppender` to `log4j.properties`. Note that the legacy appender commonly uses UDP. Upgrade to Log4j 2, or forward locally to a relay when TCP is required.

```
log4j.rootLogger=INFO, SYSLOG
log4j.appender.SYSLOG=org.apache.log4j.net.SyslogAppender
log4j.appender.SYSLOG.syslogHost=LOGZILLA_HOST
log4j.appender.SYSLOG.Facility=LOCAL0
log4j.appender.SYSLOG.layout=org.apache.log4j.EnhancedPatternLayout
log4j.appender.SYSLOG.layout.ConversionPattern=%m - thread=%t class=%C{1} method=%M%n
```

Alternative: HTTP/HTTPS

Applications that prefer HTTP can send structured JSON to the HTTP Receiver at `/incoming`. See HTTP Event Receiver (https://www.logzilla.ai/docs/receiving-data/http-event-receiver) for the JSON envelope, token headers, and minimal tests.

Verification

In LogZilla, confirm reception by searching for the configured program name or message text. For packet-level checks, see Syslog Troubleshooting (https://www.logzilla.ai/docs/administration/syslog-troubleshooting).

Optional: Transform fields in LogZilla

If messages contain class, method, or thread markers, those values can be extracted into tags or used to adjust the `program` field with rewrite rules. See Rewrite Rules (https://www.logzilla.ai/docs/data-transforms/rewrite-rules) for techniques and safe deployment steps.