

# KIP-495: Dynamically Adjust Log Levels in Connect

- [Status](#)
- [Motivation](#)
- [Proposed Changes](#)
  - [Get a list of all named loggers](#)
  - [Get the log level of a specific logger](#)
  - [Set the log level of a specific logger](#)
- [Configuration Changes](#)
- [Example Usage](#)
  - [Example 1: Disable the admin endpoints](#)
  - [Example 2: Use a separate listener for /admin](#)
  - [Example 3: Use a separate listener for /admin \(https\)](#)
- [Public Interfaces](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Rejected Alternatives](#)
- [Related Work](#)

## Status

**Current state:** *Accepted*

**Discussion thread:** <https://www.mail-archive.com/dev@kafka.apache.org/msg99656.html>

**Vote thread:**

JIRA:



Unable to render Jira issues macro, execution error.

**Released:**

**Pull request:** <https://github.com/apache/kafka/pull/7403>

Please keep the discussion on the mailing list rather than commenting on the wiki (wiki discussions get unwieldy fast).

## Motivation

Kafka Connect does not provide an out-of-the-box facility to change log levels. When debugging connectors or the Connect framework, users have to update the `log4j.properties` file and restart the worker to see new logs. This is cumbersome in most cases, and restarting the worker sometimes hides bugs by resetting the internal state.

## Proposed Changes

This change will introduce an `/admin/loggers` endpoint to the Connect worker that can be used to get/modify the log levels of any named loggers in the Connect worker. These modifications will not be persisted across worker restarts, and will only affect the worker whose endpoint received this REST request (other workers in the cluster will be unaffected). The `/admin/loggers` endpoint will support the following operations:

### Get a list of all named loggers

Returns a list of loggers along with their current log levels. Entries are sorted by the logger's name. A named logger is one that is created by either the worker's classes using the `LoggerFactory.getLogger` method or is an ancestor denoted by the name of the package it corresponds to. Their levels are set by the `log4j` configuration file used by the worker JVM, or by using this API.

```
$ curl -s http://localhost:8083/admin/loggers | jq
[
  ...
  "org.apache.kafka.connect.runtime.WorkerSinkTask": {
    "level": "INFO"
  },
  "org.apache.kafka.connect.runtime.WorkerSourceTask": {
    "level": "DEBUG"
  },
  ...
]
```

### Get the log level of a specific logger

If the name of a logger is specified, only return the level to which that logger is writing messages at. This level may be set by a `log4j` configuration file, or by this API. If neither of those sources specified a level, `log4j` uses the level of an ancestor that was set by one of these means (the root logger's level is used if none of the other ancestor had a level set to it).

```
$ curl -s http://localhost:8083/admin/loggers/org.apache.kafka.connect.runtime.WorkerSinkTask | jq
{
  "level": "INFO"
}
```

Even if it is not used by any runtime classes to log any messages, requesting the level of an ancestor package will also return a log level using the above rules.

```
$ curl -s http://localhost:8083/admin/loggers/org.apache | jq
{
  "level": "INFO",
}
```

### Set the log level of a specific logger

Use `PUT` to set the level of a logger. The API returns the list of loggers whose levels were modified.

```
$ curl -s -X PUT '{"level": "TRACE"}' http://localhost:8083/admin/loggers/org.apache.kafka.connect.runtime.WorkerSinkTask
{
  modified: [
    "org.apache.kafka.connect.runtime.WorkerSinkTask"
  ]
}
```

Setting the log level of an ancestor (for example, `org.apache.kafka.connect` as opposed to a classname) will update the levels of all child classes. Any levels previously set by this API will also be overridden.

```
$ curl -s -X PUT '{"level": "TRACE"}' http://localhost:8083/admin/loggers/org.apache.kafka.connect
{
  modified: [
    ...
    "org.apache.kafka.connect.runtime.WorkerSinkTask",
    "org.apache.kafka.connect.runtime.WorkerSourceTask",
    ...
  ]
}
```

The above call returns `org.apache.kafka.connect.runtime.WorkerSinkTask` and `org.apache.kafka.connect.runtime.WorkerSourceTask` because they fall under the specified ancestor. If no name parameter is present, this API will set the log level of the root logger.

## Configuration Changes

In addition, a new configuration property `admin.listeners` will be provided to specify which listener to use for the `/admin` endpoint. An additional configuration property will be used to control how the admin endpoint is made available.

Configuration Name	Description	Default Behavior	Domain
<code>admin.listeners</code>	Control where the admin endpoint will be made available.	the endpoint will be added to the rest of Connect's existing endpoints	List of comma-separated URIs the REST API will listen on. An empty or blank string will disable this feature. Default behavior is to use the normal listener (default port 8083).
<code>admin.listeners.https</code>	Prefix for SSL settings when using HTTPS	Look at <a href="#">Connect REST docs</a> for a list of supported configs that with this prefix.	Look at <a href="#">Connect REST docs</a> for a list of supported configs that can go with this prefix.

## Example Usage

### Example 1: Disable the admin endpoints

Set a blank string to the `admin.listeners` property to disable the entire `/admin` endpoint.

```
admin.listeners=
```

### Example 2: Use a separate listener for /admin

Bring up the admin endpoint on a separate listener:

```
admin.listeners=http://localhost:9093
```

### Example 3: Use a separate listener for /admin (https)

Set up SSL properties in a separate https listener:

```
admin.listeners=https://localhost:9093
admin.listeners.https.ssl.truststore.location=/path/to/truststore.jks
admin.listeners.https.ssl.truststore.password=tpass
admin.listeners.https.ssl.keystore.location=/path/to/keystore.jks
admin.listeners.https.ssl.keystore.password=kpass
```

## Public Interfaces

- A `/admin` endpoint that can be configured to be attached to a different listener than the existing Connect endpoints.
- The `/admin/loggers` endpoint that be used to modify log levels.
- A new configuration property in `WorkerConfig` to enable/disable the `/admin` endpoint along with optionally configuring any HTTPS listeners if needed.

Log level changes are not persisted across worker restarts. These changes also only apply to the worker that receives the PUT request described above.

## Compatibility, Deprecation, and Migration Plan

This is a new feature. Existing features will not be changed.

It must be noted that the changes proposed only work when `log4j 1.x` is used to log messages and is available on classpath. This is the current default in the AK distribution as well. Other loggers (`logback`, `log4j2` etc) will not be supported and this feature (and others mentioned below in related work) will have to be reworked to support these frameworks.

## Rejected Alternatives

- Changing log levels in a single node of an application will affect other nodes in the cluster (for example, changing log level of a class in one Connect worker will update levels in all workers in a Connect cluster) and this new level will be persisted across node restarts. This is beyond the scope of this proposal.
- Using JMX to change log levels is too cumbersome, difficult to secure and the other components are going to move away from it (look at [KIP-412](#) where brokers are now using Admin Client to change log levels).

## Related Work

- [KIP-412: Extend Admin API to support dynamic application log levels](#)
- [KIP-449: Add connector contexts to Connect worker logs](#)