

# KIP-438: Expose task, connector IDs in Connect API

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
- [Proposed Changes](#)
- [Rejected Alternatives](#)

## Status

**Current state:** *Under Discussion*

**Discussion thread:** [here](#) [Change the link from the KIP proposal email archive to your own email thread]

**JIRA:** [here](#) [Change the link from KAFKA-1 to your own ticket]

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

## Motivation

The Connect runtime assigns unique task IDs to Source/SinkTasks, which are useful for metrics and logging. Currently, task-level metrics and some log messages produced by the runtime use these IDs; however, there is no way for a Source/SinkTask instance to know its task ID. Source/SinkTasks may wish to produce their own log messages and metrics, which is unnecessarily difficult without access to this property.

For example, an operator may wish to debug a specific failed task. The operator knows the task ID, e.g. from the REST API or from runtime log messages. However, since the task's own log messages and metrics do not include the task ID, she is unable to correlate them with the specific failed task.

Connector IDs/names are useful for the same reasons. Currently, Source/SinkTasks do not have access to their associated connector names, though it is straightforward to pass them along in the task config. I propose we expose both task ID and connector name to Source/SinkTasks for consistency.

## Public Interfaces

I propose to add connector() and id() methods to both the SourceTaskContext and SinkTaskContext interfaces:

```
public interface SourceTaskContext {
    ---->%---
    String connector();
    int id();
}

public interface SinkTaskContext {
    ---->%---
    String connector();
    int id();
}
```

## Proposed Changes

Just the four new methods.

## Compatibility, Deprecation, and Migration Plan

No issues.

## Rejected Alternatives

- These methods should not be added to Task or Source/SinkTask, as they are not methods that need to be implemented by Tasks.