KIP-661: Expose task configurations in Connect REST API

This page is meant as a template for writing a KIP. To create a KIP choose Tools->Copy on this page and modify with your content and replace the heading with the next KIP number and a description of your issue. Replace anything in italics with your own description.

Status

Current state: Adopted

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

JIRA: *KAFKA-10833 [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 exposes the configuration of connectors. This is very useful for administrators as it allows to determine the workload of connectors.

However some connectors process the configuration before passing it to tasks. This can happen to split work between tasks or to enrich the configuration. This is the case for example with MirrorSourceConnect (MirrorMaker2) or with JdbcSourceConnector.

When the configuration is processed, administrators have no direct way to retrieve the computed values. In many cases it's useful to be able to retrieve the exact configuration is task was started with. This can be used for debugging but also for understanding the impact of failures (for example a task crashing).

I propose exposing the configuration of tasks in the Connect REST API.

Public Interfaces

A new Connect REST API endpoint: /{connector}/tasks-config available via GET that returns the configuration of all tasks for the connector.

For example, with MirrorSourceConnector if the connector configuration is:

connector config

```
{
  "connector.class": "org.apache.kafka.connect.mirror.MirrorSourceConnector",
  "tasks.max": "30",
  "topics": ".*",
  "name": "MirrorSourceConnector",
  "target.cluster.bootstrap.servers": "kafka-0:9092,kafka-1:9092,kafka-2:9092"
}
```

The new endpoint would return:

tasks configurations

```
{
  "MirrorSourceConnector-0": {
    "connector.class": "org.apache.kafka.connect.mirror.MirrorSourceConnector",
    "tasks.max": "30",
    "source.cluster.alias": "source",
    "topics": ".*",
    "target.cluster.alias": "target",
    "task.class": "org.apache.kafka.connect.mirror.MirrorSourceTask",
    "name": "MirrorSourceConnector",
    "task.assigned.partitions": "topic0-0,topic0-2,topic0-4"
  },
  "MirrorSourceConnector-1": {
    "connector.class": "org.apache.kafka.connect.mirror.MirrorSourceConnector",
    "tasks.max": "30",
    "source.cluster.alias": "source",
    "topics": ".*",
    "target.cluster.alias": "target",
    "task.class": "org.apache.kafka.connect.mirror.MirrorSourceTask",
    "name": "MirrorSourceConnector",
    "task.assigned.partitions": "topic0-1,topic0-3,topic1-5"
  }
}
```

Proposed Changes

A new endpoint will be defined in ConnectorsResource.java

```
@GET
@Path("/{connector}/tasks-config")
public Map<ConnectorTaskId, Map<String, String>> getTasksConfig(
    final @PathParam("connector") String connector,
    final @Context HttpHeaders headers,
    final @QueryParam("forward") Boolean forward) throws Throwable {
}
```

A new method will be added to Herder to provide the functionnality.

```
/**
 * Get the configuration for all tasks.
 * @param connName name of the connector
 * @param callback callback to invoke with the configuration
 */
void tasksConfig(String connName, Callback<Map<ConnectorTaskId, Map<String, String>>> callback);
```

The 2 Herder implementations, StandaloneHerder and DistributedHerder will implement this new method.

Compatibility, Deprecation, and Migration Plan

This is a new endpoint, existing endpoints are not changed

Rejected Alternatives

- Parse the connect config topic. The configuration of tasks is stored in the config topic. Administrators would need to filter and parse its content to recover the data.