KIP-197 Connect REST API should include the connector type when describing a connector

Status

Current state: Accepted

Discussion thread: here

JIRA: KAFKA-5657

Released: 1.0.0

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

Motivation

Currently we don't expose information about whether a connector is a source or sink in its description. This is useful when, e.g., categorizing connectors in a UI. Given suggested naming conventions, you **might** be able to determine this via the connector's class name, but that isn't reliable. This proposal makes the type of connector explicit in the REST responses.

Public Interfaces

We will modify the following REST API endpoints for Connect to include the type of the connector:

```
GET /connectors/(string:name)
GET /connectors/(string:name)/status
```

Proposed Changes

The endpoint `GET /connectors/(string:name)` currently returns the following structure:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "name": "hdfs-sink-connector",
    "config": {
        "connector.class": "io.confluent.connect.hdfs.HdfsSinkConnector",
        "tasks.max": "10",
        "topics": "test-topic",
        "hdfs.url": "hdfs://fakehost:9000",
        "hadoop.conf.dir": "/opt/hadoop/conf",
        "hadoop.home": "/opt/hadoop",
        "flush.size": "100",
        "rotate.interval.ms": "1000"
    },
    "tasks": [
        { "connector": "hdfs-sink-connector", "task": 1 },
        { "connector": "hdfs-sink-connector", "task": 2 },
        { "connector": "hdfs-sink-connector", "task": 3 }
    ]
}
```

We will add a `type` field to the document to indicate whether the given Connector is a Source or Sink. e.g.:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "name": "hdfs-sink-connector",
    "config": {
        "connector.class": "io.confluent.connect.hdfs.HdfsSinkConnector",
        "tasks.max": "10",
        "topics": "test-topic",
        "hdfs.url": "hdfs://fakehost:9000",
        "hadoop.conf.dir": "/opt/hadoop/conf",
        "hadoop.home": "/opt/hadoop",
        "flush.size": "100",
        "rotate.interval.ms": "1000"
    },
    "type": "sink",
    "tasks": [
        { "connector": "hdfs-sink-connector", "task": 1 },
        { "connector": "hdfs-sink-connector", "task": 2 },
        { "connector": "hdfs-sink-connector", "task": 3 }
    ]
}
```

Similarly, the endpoint GET /connectors/(string:name)/status currently returns the following structure:

```
HTTP/1.1 200 OK
{
    "name": "hdfs-sink-connector",
    "connector": {
        "state": "RUNNING",
        "worker_id": "fakehost:8083"
    },
    "tasks":
    [
        {
            "id": 0,
            "state": "RUNNING",
            "worker_id": "fakehost:8083"
        },
        {
            "id": 1,
            "state": "FAILED",
            "worker_id": "fakehost:8083",
            "trace": "org.apache.kafka.common.errors.RecordTooLargeException\n"
        }
    ]
}
```

We will add a `type` field to the response to indicate whether the given Connector is a Source or Sink. e.g.:

```
HTTP/1.1 200 OK
{
    "name": "hdfs-sink-connector",
    "connector": {
        "type": "sink",
        "state": "RUNNING",
        "worker_id": "fakehost:8083"
    },
    "tasks":
    [
        {
            "id": 0,
            "state": "RUNNING",
            "worker_id": "fakehost:8083"
        },
        {
            "id": 1,
            "state": "FAILED",
            "worker_id": "fakehost:8083",
            "trace": "org.apache.kafka.common.errors.RecordTooLargeException\n"
        }
    ]
}
```

Compatibility, Deprecation, and Migration Plan

This propsal only adds a field to existing JSON response messages from existing REST API endpoints, and does not otherwise change the structure of the responses. Many applications will be able to handle the response documents and ignore fields they don't know about. Therefore, this should be backwards compatible.

KIP-151 brought in the connector type enum which generates lowercase. This is kept in this KIP.

Though consumers of the API should not make assumptions about the capitalization of connector type.

Rejected Alternatives