

KIP-995: Allow users to specify initial offsets while creating connectors

- [Status](#)
- [Motivation](#)
- [Public Interfaces](#)
 - ["POST /connectors" REST API endpoint](#)
- [Proposed Changes](#)
 - [Minor details](#)
 - [Validate offset before connector creation](#)
 - [Remanent offsets from connectors with same name](#)
- [Compatibility, Deprecation, and Migration Plan](#)
- [Test Plan](#)

Status

Current state: *Under Discussion*

Discussion thread: [here](#)

JIRA: [here](#)

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

Motivation

During the discussion for [KIP-980](#), which proposed the creation of connectors in STOPPED state, there was a suggestion to also allow setting the initial offset for a connector in the connector creation API. The [proposal was deemed valid](#) (point no.4) but was deferred to a future KIP. This KIP proposes to implement that change.

This feature will be helpful to users who want to create a connector and start from a specific offset.

To do this today, user will have to

1. first create a connector in STOPPED state and check the response for errors
2. alter the offsets for said connector and check the response for errors
3. finally set the connector state to RUNNING.

Public Interfaces

"POST /connectors" REST API endpoint

A new optional field "initial_offsets" will be added to the request body format for the POST /connectors endpoint. The value for this field has to adhere to format specified in [KIP-875](#) else a 400 Bad Request response will be returned.

If the field is omitted in the request body, the current behaviour of creating connectors will be preserved. An example request body would look like:

CreateConnectorRequest

```
{
  "name": "file-src-1",
  "config": {
    "connector.class": "org.apache.kafka.connect.file.FileStreamSourceConnector",
    "file": "test.txt",
    "topic": "test-topic",
    "tasks.max": "1"
  },
  "initial_offsets": [
    {
      "partition": {
        // source partition
      },
      "offset": {
        // Desired initial offset
      }
    }
  ]
}
```

As mentioned in [KIP-875: First-class offsets support in Kafka Connect#Altering/resettingoffsets\(response\)](#), the response message (`initial_offsets_response`) for successfully setting the `initial_offsets` will depend on whether the connector has implemented `alterOffsets()` method.

If the connector has implemented `alterOffsets` and everything has succeeded, the HTTP status will be 200 and the create-connector response body will be:

response (definite success)

```
{
  "name": "ConnectorOne",
  "config": {...},
  "initial_offsets_response": "The offsets for this connector have been set successfully"
}
```

If the connector has not implemented `alterOffsets` but everything else has succeeded, the HTTP status will be 200 and the create-connector response body will be:

response (possible success)

```
{
  "name": "ConnectorOne",
  "config": {...},

  "initial_offsets_response": "The framework-managed offsets for this connector have been set successfully.
However, if this connector manages offsets externally, they will need to be manually set in the system that the
connector uses."
}
```

This method will be supported in both the distributed mode as well as the standalone mode.

KIP-980 had [introduced](#) an option to pass JSON config to `bin/connect-standalone.sh`. This new field will be supported there as well.

Proposed Changes

When Connect runtime receives a request with `initial_offsets` to create a connector via `POST /connectors`, it will perform the following steps

1. Validate connector configs
2. Validate the offset using the same checks performed while altering connector offsets (`PATCH /$connector/offsets`) as specified in KIP-875 (We will skip validations which check if the connector exists or is in STOPPED state)
3. Wipe all existing offsets for the connector (i.e we do not want to merge existing offsets and "initial_offsets")
4. Set the offset of the connector to the value of "initial_offsets"
5. Write connector config (with whatever initial state is specified in the request, or the default if none is specified)

If Connect runtime encounters an error in any of these steps, it will cleanup (if required) and return an error response mentioning the step at which the failure was encountered.

If step no. 4 fails, the existing offsets of the connector would've already been wiped in step no. 3.

If step no. 5 fails, the `initial_offset` set in step 4 will be wiped as part of cleanup.

In either failure case, the offset of the connector, set prior to invoking this API, will be unrecoverable.

Minor details

Validate offset before connector creation

There is a slight chance of a race condition if two connector creation requests with same name, with and without `initial_offsets`, are received. The request without offset might create the connector while the other one validates and writes the offset. We still would like to validate the offset and config first - so that there is lesser chance of a cleanup while creating the connector and setting its offset.

Remanent offsets from connectors with same name

There could be cases when a previous instance of connector with same name has offsets left in the offsets topic. The offsets sent along with the create connector request will be written after wiping out the previous offsets. This is done to ensure a clean slate for the connector, thereby avoiding any surprises later.

Compatibility, Deprecation, and Migration Plan

Since "initial_offsets" is an optional field and when unspecified will default to current behaviour, we do not foresee a compatibility issue while downgrading the Connect cluster. This new feature will reuse existing logic for storing offsets. Hence the connector will be indistinguishable from one which started with no initial offset specified and in due course committed an offset to the store.

Test Plan

The following scenarios will be tested using Unit, Integration or system tests

- Create connector with `initial_offsets` in standalone mode
- Create connector with `initial_state` and `initial_offsets` in standalone mode
- Create connector with `initial_offsets` in distributed mode
- Create connector with `initial_state` and `initial_offsets` in distributed mode
- Failure while creating connector with `initial_offsets` will clean up and return error response
- Standalone mode cli accepts `initial_offsets` in JSON config file

Rejected Alternatives

Restrict `initial_offsets` usage to cases where `initial_state == STOPPED`

We could enforce `initial_offset` usage only to cases when a connector is created in STOPPED state. This would've reduced one point of failure while performing the action. However, we feel that this will impact the usability of this new API and will not offer any major benefit from the steps which a user has to run currently, namely, creating a connector in STOPPED state, altering the offset and then setting the state to RUNNING.

