

KIP-212: Enforce set of legal characters for connector names

- [Status](#)
- [Motivation](#)
- [Proposed Change](#)
- [New or changed public interfaces](#)
- [Migration plan and compatibility](#)
- [Rejected alternatives](#)
- [Appendix A - Research](#)

Status

Current State: *Accepted*

Discussion Thread: [here](#)

Jira:



Unable to render Jira issues macro, execution error.

Released: 1.1.0

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

Motivation

Currently very little checking is done against connector names, when creating a new connector in Kafka Connect. The only check that is performed is that no / is present in the name, other than that it is even possible to create a connector with an empty name that is then impossible to delete.

Additionally there are a large number of special characters that can be used in connector names but create issues afterwards, when they need to be used in the url for the rest call to change or delete the connector.

A pull request is ready for review that will deal with a few of the cases that cause issues, but this is more of a band aid than a proper fix for this issue, as I am sure there are a whole lot more characters out there that were missed and also create issues.

There are a number of related jiras currently open that could probably all be fixed by implementing something like the checks proposed here:



Unable to render Jira issues macro, execution error.



Unable to render Jira issues macro, execution error.



Unable to render Jira issues macro, execution error.

Proposed Change

All changes suggested in this KIP will only be applied when connectors are created, for connectors already deployed at the time this is implemented no changes are made, they can be updated and deleted like before.

For new connectors, the validation of connector names will be changed to trim leading and trailing whitespaces and reject zero length strings after trimming. This would allow for whitespaces in connector names but remove potential confusion caused by accidentally padding the name with whitespaces, which is easily possible due to the create request having the name as a json value, not in the url. This will only affect the creation of new connectors,

Additionally connector names containing control characters or one of their more common escape sequence representations will be rejected.

Additionally a section will be added to the documentation explaining which characters need to be url encoded for rest calls to work properly. This together

with the work done on



should enable Connect to handle a very broad range of

characters for connector names.

New or changed public interfaces

There is no new public interface, however the behavior of the create connector api will change a little bit. Existing connectors are not affected by this change and can be updated and queried just like today.

Whitespace handling

Following implementation of this KIP, leading and trailing whitespaces in connector names will be trimmed before the connector is created. In the response to the create request the updated name will be reflected. See the following table for a few examples:

Original name	Trimmed name	Comment
" test"	"test"	
"test "	"test"	
" test "	"test"	
" "	""	Will be rejected as empty connector name!

This is a change to current behavior, as today when sending a create connector request the connector will be created with the specified name or the request will fail, names are not changed by Connect. However, requests containing whitespaces would fail in the current version (fixed in KAFKA-4827) - so it could be assumed that not many people rely on connectors containing whitespaces in the name at all.

Control characters

The [control characters](#) ASCII 0 - 31 & 127 will be considered illegal characters and connector names containing these will be rejected. In order to avoid injecting these characters via escape sequences the connector name will be *unescaped* before testing for control characters.

See Migration plan and compatibility for scenarios that might be influenced by this change in behavior.

Migration plan and compatibility

Backward compatibility for existing connectors is given, as all changes suggested here only affect connectors when they are created. The possibility to update and delete existing connectors is not impacted.

For new connectors that are created after implementing this change limitations are restricted to a few scenarios described below. Apart from those, nothing that previously worked should break based on this KIP. Connectors with an empty name were broken before and could not be deleted anyway, so there is no reason to half-support these any further. It is however a change in behavior to reject these.

As mentioned there are however two scenarios that are affected by this change:

Changing of connectors with leading or trailing whitespaces

If an external system is used to manage connectors, the change in connector names could potentially cause this system not to find deployed connectors anymore - if their names contain trailing or leading whitespaces!

1. Create connector: "test " -> Connector gets created (request fails though, due to KAFKA-4827)
2. Query status of connector: "test " -> Fails as the connector was created with the name "test"

There are two points worth noting in this scenario though:

1. The system managing the connectors would have to completely ignore the response from connect to the first request, as this would return an error in the current version and it should not have assumed that the connector has been created anyway.
2. Following the fix for KAFKA-4827 being merged the request would be successful, but contain the updated name for the created connector, so there would be an easy check to update the name used.

Based on the above points I would consider this a fringe scenario that should not impact a significant number of people (to be honest, I'd be surprised if anyone is impacted at all). Additionally I can't come up with a valid reason to use trailing or leading whitespaces in connector names.

Creating connectors that only differ in the number of trailing or leading whitespaces

As shown in the table above, multiple input names might be matched to the same output name after this change: " test" and "test " would be considered the same after stripping whitespaces. The second create request would be rejected as a connector with the name "test" already exists.

While this is strictly speaking a limitation my personal opinion is that using connector names that differ only in the number of whitespaces at the beginning or end is a recipe for disaster and should not be encouraged.

Rejected alternatives

Initially the proposal was to reject a number of characters as illegal in connector names based on a whitelist or blacklist. However following discussions on the mailinglist it was agreed that we can be very generous in allowing characters in connector names as long as all rest requests are properly url-encoded.

Original KIP:

Based on this research and since I don't really see the benefit behind supporting a large number of exotic special characters I propose to limit allowed chars for connector names to the unreserved characters:

```
a-z  A-Z  0-9  .  -  _  ~
```

Appendix A - Research

The set of allowed characters should be determined by the fact that the connector name is part of the url in a rest call to update the config or delete the connector, so we should take care to allow only characters that can be "legally" used within urls - however it turns out that this is not an entirely easy distinction. After looking at a few stackoverflow threads([here](#), [here](#) & [here](#)) as well as RFCs [1738](#), [2396](#) and [3896](#) my understanding is the following.

There are some characters that are definitely legal and allowed without any restriction:

```
a-z  A-Z  0-9  .  -  _  ~
```

Then there are reserved characters, these are legal, but can have special meaning depending on which section of the URL they appear in:

```
 ; / ? : @ & = + $ ,
```

And last but not least there are a few "unwise" characters:

```
{ } | \ ^ [ ] `
```

During my work on KAFKA-4930 I found that at least ? from the list of reserved characters also creates issues and leads to connectors that are not accessible anymore after creation, so I'd be hesitant to simply include these in the list of allowed characters without further research into what causes these issues. A good example for one of these creating issues is the ; char. Connect uses jetty internally to serve the rest endpoints. Jetty considers ; to be a special character delimiting two url parameters from each other and stops parsing the url portion at this character. What this means is that you can create a connector with the name "test;test", since during creation you specify the name within the body of the request, but when you try calling the /connectors /test;test/status endpoint jetty will cut at ; - look for a connector named "test" and not find anything.