CONFD LEGACY REST API TO RESTCONF MIGRATION

tail-f a Cisco company

Table of Contents

CONFD LEGACY REST API TO RESTCONF MIGRATION

1. Overview

With the release in late 2019 of ConfD 7.3, the legacy REST API will be removed from ConfD and customers will need to migrate their use of the legacy REST API to the RESTCONF API. The legacy REST API in ConfD pre-dates the invention of RESTCONF and, for quite some time, ConfD users have been told to discontinue the use of the legacy REST API in favor of the RESTCONF API for new projects. It does not make sense for ConfD to continue to support the proprietary legacy REST API because the standardized RESTCONF API has been defined. This application note covers, at a high level, the legacy REST and RESTCONF APIs with a focus on items that differ between the two APIs. It concludes with a discussion of how to migrate usage from the legacy REST API to RESTCONF.

2. The Legacy REST API

Back in 2011, several ConfD customers whose target market were in areas where REST was commonly used, such as the data center market, asked Tail-f to add REST to the ensemble of northbound interfaces which ConfD provides.

At the time, no standard mapping of YANG data models to REST techniques existed. So, Tail-f invented and implemented a prototype of a REST API. This REST API was released initially in ConfD 3.9 in March 2012. At the time, this API was marked "experimental" and was not to be used in production. The intent of this initial release was to give customers visibility to the approach that Tail-f engineering was taking with mapping YANG data models to a REST API and to allow for comments and feedback to Tail-f. After several iterations over subsequent releases, the legacy REST API solidified and the "experimental" tag was removed. At that time, the legacy REST API became a full-fledged ConfD northbound interface.

Tail-f designed the legacy REST API to operate on the configuration datastores defined in NETCONF and to make use of YANG data models. The legacy REST API also defines a set of create, retrieve, update and delete (CRUD) operations that operate on these datastores.

In the legacy REST API, the root resource is /api. Below the root resource, the legacy REST API has separate resources for the different datastores, /running, /startup and /

candidate, as well as a unified /config resource and a /operational resource for operational data. For user defined RPC operations, there is also the /operation resource.

The legacy REST API defines a set of application specific media types for the various resources, application/vnd.yang.api for the /api resource, application/vnd.yang.datastore for the datastore resources, application/vnd.yang.data for the data resources, and application/vnd.yang.operation for the operation resource.

The legacy REST API supports both XML and JSON, with “+xml” appended to the media types for XML, and “+json” appended to the media types for JSON.

Since the legacy REST API provides separate resources for the different NETCONF datastores, the REST API also provides for operations on the datastores, such as lock, commit for the candidate resource, discard-changes for the candidate datastore, validate, and copy-running-to-startup.

The legacy REST API maps the various HTTP operations to the appropriate NETCONF operation. The HTTP GET operation maps to the NETCONF <get-config> or <get> operation depending on the resource, the HTTP POST, PUT, PATCH and DELETE operations map to the <edit-config> operation with the different HTTP operations implying a specific operation semantic.

For actions that are defined in a YANG data model, the action can be invoked using the POST operation. For example, if a "reboot" action is defined under a YANG container "chassis", in order to invoke this action, you send a POST operation to /api/running/chassis/_operations/reboot.

For RPC operations defined in a YANG data model, the RPC appears under the resource /api/operations.

Examples of using the legacy REST API can be found in a ConfD installation under $CONFD_DIR/examples.confd/rest. More details on the legacy REST API can be found in the ConfD User Guide chapter "The REST Interface".

3. The RESTCONF API

Shortly after the initial release of the legacy REST API, Tail-f began to participate in IETF efforts to develop and standardize a RESTful analogue to NETCONF. Tail-f was a contributor to the first draft of RESTCONF in September 2013. In March 2014, RESTCONF was taken up as a working IETF NETCONF working group item and became RFC 8040 in January 2017.

RESTCONF defines the root resource as /restconf, although it allows this to be configurable. This can be done in ConfD by setting the root resource in the ConfD configuration file.

As with the legacy REST API, RESTCONF uses the various HTTP methods to map to CRUD operations. Rather than allowing operations to be specific datastores, RESTCONF specifies a unified datastore, much like the /config resource in the legacy REST API. This simplifies the REST API supplied by RESTCONF because the datastore operations that the legacy REST API defined are no longer needed.

With the unified datastore supported by RESTCONF, there are some behaviors to be aware of. One is that, if the candidate datastore is enabled, changes made by RESTCONF will be made in the candidate datastore then automatically committed to the running datastore. If other changes have been made to the the candidate datastore via another northbound interface, such as NETCONF or CLI, these changes will be committed along with the changes made by RESTCONF.

Another RESTCONF behavior to be aware of is that if a datastore needed by a RESTCONF operation is locked, then that operation will fail.

RESTCONF also reduced the number of media types to two, application/yang-data+xml and application/yang-data+json.

RESTCONF supports invoking YANG data model defined RPCs and actions. RPCs can be invoked using an HTTP operation and the RPC can be found under the /restconf/operations resource. An action can be invoked using an HTTP POST operation under the /restconf/data resource. For example, if there is a "reboot" action under a YANG container "chassis", the action can be performed by sending a POST operation to /restconf/data/chassis/reboot.

RESTCONF also provides capabilities that the legacy REST API does not, such as the ability to retrieve YANG data modules and a more flexiblibility to modify the configuration in a single REST call using the YANG Patch Media Type as defined in RFC 8072.

More information on RESTCONF can be found in the Tail-f whitepaper "Inside RESTCONF".

Examples of using the RESTCONF interface can be found in a ConfD installation under $CONFD_DIR/examples.confd/restconf. More details on the RESTCONF API can be found in the ConfD User Guide, chapter "The RESTCONF Interface".

Tail-f has also invested in its RESTCONF implementation by adding support for generating Swagger definitions from a YANG data model. More information on Swagger support can be found in the ConfD User Guide, chapter “The RESTCONF API”, section “Generating Swagger for RESTCONF” and in the ConfD application note "RESTCONF, YANG, and Swagger".

4. Migrating from the Legacy REST API to RESTCONF

When migrating from the legacy REST API to RESTCONF, there are several considerations to be made in the move.

One consideration is that the root resource in the legacy REST API is /api, while the default root resource for RESTCONF is /restconf.

The media types used for the two interfaces are different as well. All of the various legacy REST API media types map to either application/yang-data+xml or application/yang-data+json, depending on whether the representation is XML or JSON.

There is only one datastore in RESTCONF, the unified datastore. All references to the specific datastore resources in the legacy REST API, such as /running, /candidate, and /config, need to be modified to use /data in RESTCONF. Usage of /operational in the legacy REST API needs to use the content modifier on /data in RESTCONF, /data?content=nonconfig.

With the unified datastore in RESTCONF, the datastore operations, such as commit, lock, validate, supplied by the legacy REST API are not needed or available.

The representation of keys is also slightly different in RESTCONF. Where you would specify the key value "foo" in the legacy REST API as /api/running/mylist/foo, in RESTCONF, you specify this using /restconf/data/mylist=foo

RPCs and actions are handled similarly in the legacy REST API and RESTCONF, with changes involved in the root resource, /api versus /restconf, for RPCs and actions, and the removal of _operation in the URI from the legacy REST API for actions.

For more details on migration from the legacy REST API to RESTCONF, starting with ConfD 7.1, a new ConfD User Guide chapter "Migrating from REST to the RESTCONF API" has been added and should be consulted for full details.

For More Information

For more information about ConfD, visit: https://www.tail-f.com

"Inside RESTCONF" whitepaper: https://info.tail-f.com/inside-restconf

"RESTCONF, YANG and Swagger" application note:
https://info.tail-f.com/appnote-restconf-yang-swagger

Legacy REST to RESTCONF migration documentation, ConfD User Guide, Chapter 24,
"Migrating from REST to RESTCONF API"

tail-f a Cisco company

www.tail-f.com
info@tail-f.com

Corporate Headquarters

Sveavagen 25
111 34 Stockholm
Sweden
+46 8 21 37 40