

IoT Accelerator Developer Portal

Migrating Labels to Custom Fields with Postman

Tutorial





Contents

1	Purpose and Scope.....	3
2	Install Postman and import the collection	3
3	Configuring the environment	4
4	Dynamic variables	6
5	Run the collection.....	6



1 Purpose and Scope

This document is in the scope of IoT Accelerator Service.

Purpose of the document is to describe how to use Postman and a pre-configured API collection to migrate the content of the legacy "Labels" into the new "Custom Fields", leveraging on Custom Field API.

The Postman Collection will query all the Subscriptions of a specified Company, read the values of the legacy "Labels" and transfer the values into Custom Fields, with a field name that is configurable (see "newLabel" variable described below).

Please Note custom field does not support special chars except:

- _ (Underscore)
- (dash)
- () (round brackets)

All invalid characters in the Label field will be converted to underscore during the copy process in the custom label

2 Install Postman and import the collection

1. Download Postman® Client from [here](#) and install it.
2. Launch Postman®.
3. Click on File in the top-left corner and select Import

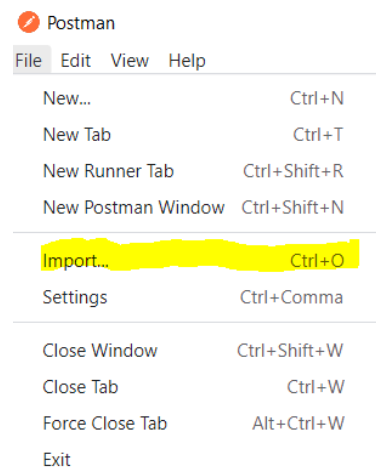


Figure 1

4. In the new window click Upload Files or drag and drop the "Custom Label Creation and Filling_ver_XX.postman_collection.json" file than click Import



5. Verify that the Collection has imported in the current Workspace.

3 Configuring the environment

Once you have imported the collections in your Postman client, they will appear grouped in a folder within your Workspace collection.

To properly configure the environment, follow these steps:

- Click on the imported collections folder in your Postman client (clicking on the three dots at the right of the collection name)
- Click "Edit"

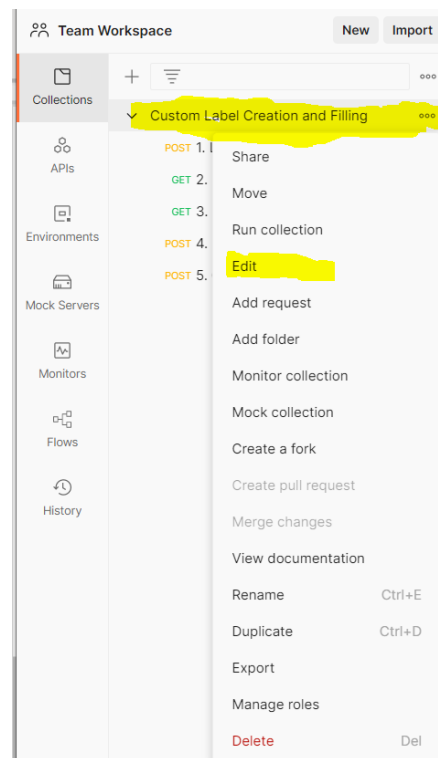


Figure 2

- On the new window which will appear, click on Variables and fill the values as described in the Table1 below.



Custom Label Creation and Filling Share Fork 0 0 ...

Authorization Pre-request Script Tests **Variables**

These variables are specific to this collection and its requests. [Learn more about collection variables.](#)

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	...	Persist All	Reset All
<input checked="" type="checkbox"/>	bearer_token					
<input checked="" type="checkbox"/>	newLabel	<INSERT-NEW-LABEL...>	<INSERT-NEW-LABEL-NAME>			
<input checked="" type="checkbox"/>	username	<INSERT-USERNAME>	<INSERT-USERNAME>			
<input checked="" type="checkbox"/>	password	<INSERT-PASSWORD>	<INSERT-PASSWORD>			
<input checked="" type="checkbox"/>	companyID	<INSERT-COMPANYID>	<INSERT-COMPANYID>			
<input checked="" type="checkbox"/>	base_url	<INSERT-BASE-URL>	<INSERT-BASE-URL>			
<input checked="" type="checkbox"/>	labellmsiArray					
<input checked="" type="checkbox"/>	ScriptExecution	true	true			
<input checked="" type="checkbox"/>	StopExecution					

Figure 3

- Define the following variables, then click on “Save” in the upper right corner of the tab.

newLabel	The name of new Label
username	insert the username of your IoTa account for consuming APIs
password	insert the password of your IoTa account for consuming APIs
companyID	Insert the company ID of your IoTa account
ScriptExecution	Set it to True if you want execute the collection and write the new label, set to false if you want to check the IMSIs involved before create the new label
base_url	The https base URL of Ericsson IoTa

Table 1

NOTE: All the variables not specified in Table 1 should not be filled in.

- After the variables values have changed remember to save the collection typing CTRL-S.



4 Dynamic variables

The dynamic variables described in this guide are meant to facilitate the handling of the requests. In particular, for the token management, dynamic variables make possible to automatically populate the headers of the requests with the token: simply use one of the available APIs for token generation, and the token will be included in the header of the requests, with no need of manual copy/paste.

5 Run the collection

Once all the previous steps are executed, click on the collection name and then on the "Run" Icon

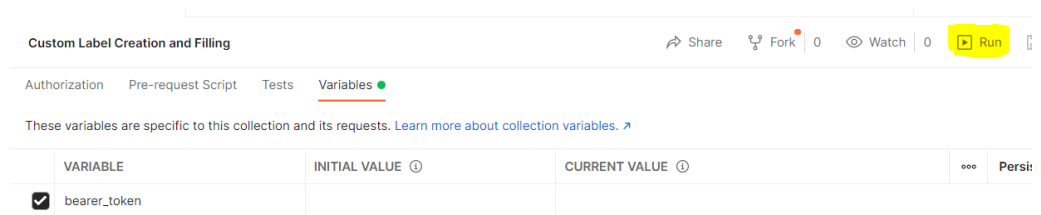


Figure 4

The new window will show the run order.
If you want to check the logs:

- Click on the "console" button on the left-bottom corner.
- Click on the blue button "Run Custom Label" to run the collection

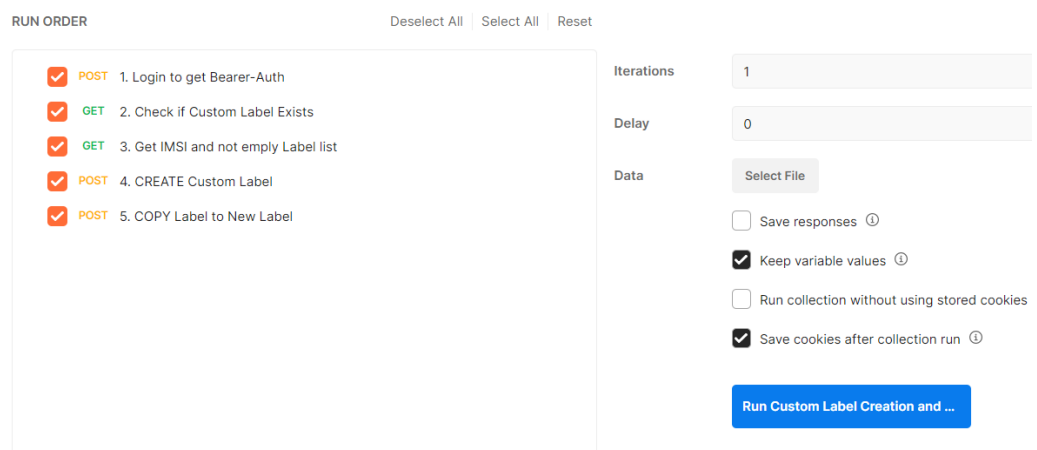


Figure 5



During the collection execution, the iteration of the requests will be shown in the postman window:

Custom Label With ACCESS No Environment, 42 mins ago View Summary Run Again New Export Results

All Tests Passed (1) Failed (0)

Iteration 1

POST	1. Login to get Bearer-Auth	https://m2m.dcp.ericsson.net/iot/api/auth/token	/ 1. Login to get Bearer-Auth	200 OK	395 ms	5.733 KB
	Pass	Status test				
GET	2. Check if Custom Label Exists	https://m2m.dcp.ericsson.net/iot/api/subscriptions/custom-fields	/ 2. Check if Custom Label Exists	200 OK	129 ms	709 B
		This request does not have any tests.				
GET	3. Get IMSI and not empty La...	https://m2m.dcp.ericsson.net/iot/api/subscriptions/details?q-company={{companyID}}&order=IMSi&additio...	/ 3. Get L...	200 OK	194 ms	7.351 KB
		This request does not have any tests.				
POST	4. CREATE Custom Label	https://m2m.dcp.ericsson.net/iot/api/subscriptions/custom-fields	/ 4. CREATE Custom Label			
		This request does not have any tests.				

Figure 6