ERICSSON

Flow Automation External REST NBI

Interwork Description

1/155 19-CNA 403 3544-1 Uen U

Contents

1 Flow Automation Northbound Interface

Describes the external RESTful web service interface (HTTP) provided by the Flow Automation NBI application.

The Flow Automation Northbound Interface (NBI) is a feature deployed in the ENM domain management for external systems.

The Flow Automation NBI feature provides interfaces for external machine systems to perform custom workflow (use cases) related operations towards nodes managed by ENM. This document describes the interface for Flow Automation NBI.

Functionality Overview

One of the most important features in ENM is to perform Network Management functions to execute workflow based use cases that orchestrate existing ENM system functions (also known as ENM System Task).

The flow automation NBI interface allows the use of machine/external systems to initiate, interact with, and retrieve information/status of custom use cases.

Prerequisites:

— To support secure communication from the external northbound system, a certificate must be installed in the northbound system.

Glossary of Terms

See FA NBI Reference List on page 22 for descriptions of the terms used in this document.

2 Flow Automation NBI Overview

Basic REST Concepts

All services are accessed through the standard HTTP interface. The interface is designed following RESTful principles. These principles include:

— Client-server communication is stateless.

  This information must be stored/managed on the client side, and must be resent with every request.

— REST resources can be addressed using URLs.

— Standard HTTP methods (GET / POST / PUT / DELETE) may be used to manipulate these resources unless otherwise indicated.

— Error reporting is through standard HTTP response codes and messages.

— Expected input parameter encoding is:

  - Standard GET query strings attached to the base URL in GET request.

    Example: ?param1=value1&param2=value2

  - Standard POST form encoding in POST requests with MIME type application/hal+json.

  - Standard RESPONSE form encoding in GET responses with MIME type application/hal+json

— Responses are encoded in JavaScript Object Notation (JSON) format. See RFC 4627 in FA NBI Reference List on page 22 [4] for additional information.

For additional information about the standards, see: RFC 2616 [2], RFC 3986 [3], and RFC 4627 [4] in FA NBI Reference List on page 22.

Base URL

The RESTful services are accessible from https://customer-domain/flowautomation/v.1

Note: The exact root context is subject to change.

Hypermedia (HAL)

HAL is included in this interface: a "_links" keyword is available where it is possible to specify links.

Note: In accordance to the HAL specifications, a "self" link is mandatory.

In the context of an endpoint related to a resource, HAL describes other resources with an association to it.

In the context of root endpoints, HAL describes the endpoints for the supported services (for example: /bulk points to import and export services) without a specific reference to a resource.

Basic Domain Concepts

Flow Automation (FA) is a generic workflow-based framework and application that allows **Flows** to be designed and executed.

A user can import a custom flow and then execute it. Several versions of a flow may exist in parallel at the same time in the system.

Flow is a custom workflow of ENM related tasks to be executed. These may be user tasks that need to be performed by the user, or external tasks that may be used to interact with other ENM applications.

The user may:

— Import a new flow in the system.

— Start a new instance of the imported flow.

— Enable or disable a flow.

— Activate or deactivate version of the flow.

— Get active user tasks of the running instance of the flow.

— Complete user task for the flow instance.

— Get the summary report for the execution of the flow instance.

— Access locale dictionaries for i18n translation.

3 Flow Automation Service Description

3.1 Flows Table

Flows	HTTP Methods			
/v1/flows	GET		POST	
/v1/flows/{flow-id}	GET			DELETE
/v1/flows/{flow-id}/activate		PUT		
/v1/flows/{flow-id}/enable		PUT		
/v1/flows/{flow-id}/execute			POST	
/v1/flows/{flow-id}/resource/flow-input-schema	GET			
/v1/flows/{flow-id}/user-permissions	GET			

3.1.1 /v1/flows

GET	Lists all the flows in the system
POST	Imports a new flow

3.1.2 /v1/flows/{flow-id}

GET	Returns a summary of the flow with the given flow-id
DELETE	Deletes the flow with the given flow-id

3.1.2.1 Query Parameters

Name	Description	Example
force = <boolean>	If this parameter set to true all the flows are deleted irrespective of their running status.	true

3.1.2.2 Flow Response Properties (GET)

The response attribute meanings are clear by name. The response contains the minimum amount of information required to identify an execution object.

Name	Description
<id=<String>> (required)	A unique identifier for the flow.
<name=<String>> (required)	The name given to the flow.
<status=<String>> (required)	The current status of the flow.
<source=<String>> (required)	The source of the flow, can be INTERNAL, EXTERNAL or PREDEFINED

Name	Description
<flowVersions=<FlowVersion[ ]>> (required)	An array of FlowVersions (see FlowVersion on page 5)

3.1.2.3 FlowVersion

Name	Description
<version=<String>> (required)	Version of the flow.
<description=<String>> (required)	Description of the flow.
<active=<String>> (required)	Status of the flow.
<createdBy=<String>> (required)	User who imported the flow.
<createdDate=<String>> (required)	Flow Import date.

3.1.3 /v1/flows{flow-id}/activate

PUT	Activate or deactivate the flow with the given flow-id

3.1.3.1 Query Parameters

Name	Description	Example
flow-version = <String>	Version of the flow that need to be activated/deactivated	1.0.0

3.1.4 /v1/flows{flow-id}/enable

PUT	Enable or disable the flow with the given flow-id

3.1.5 /v1/flows{flow-id}/execute

POST	Execute the flow with the given flow-id

3.1.6 /v1/flows/{flow-id}/resource/flow-input-schema

GET	Returns FlowInput-schema of the active version of flow if no version is passed specifically

3.1.6.1 Query Parameters

Name	Description	Example
flow-version = <String>	Version of the flow that need to be downloaded	1.0.0

3.1.7 /v1/flows/{flow-id}/user-permissions

GET	Returns user permissions for a flow

3.1.7.1 User Permission Response Properties

The response contains the minimum information required to identify an user permission object.

The properties of execute object of user permission response object are as follows:

Name	Description	Example
strategy=<String>	Points to the user permission strategy. It can be either warn or reject. warn - user can execute the flow with a warning in the absence of the required user permissions reject - user cannot execute a flow in the absence of the required user permissions	warn
permission=<boolean>	If permission is true, user can always execute the flow, irrespective of the strategy	true

3.2 Executions Table

Executions	HTTP Methods			
/v1/executions	GET			
/v1/executions/usertasks/{task-id}/complete			POST	
/v1/executions/usertasks/{task-id}/schema	GET			
/v1/executions/{flow-execution-name}				DELETE
/v1/executions/{flow-execution-name}/report	GET			
/v1/executions/{flow-execution-name}/report-schema	GET			
/v1/executions/{flow-execution-name}/flowinput-schema-data	GET			
/v1/executions/{flow-execution-name}/usertasks	GET			
/v1/executions/{flow-execution-name}/suspend		PUT		
/v1/executions/suspend		PUT		

Executions	HTTP Methods			
/v1/executions/{flow-execution-name}/stop		PUT		
/v1/executions/{flow-execution-name}/download/{resource}	GET			
/v1/executions/usertasks/{task-id}/dictionary	GET			
/v1/executions/report-dictionary	GET			
/v1/executions/flowinput-schema-data-dictionary	GET			

3.2.1 /v1/executions

GET	Lists all the executions of all the imported flows

3.2.1.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow
flow-execution-name = <String>	A name given to a flow-execution	flow-execution-flow
flow-version = <String>	The current version of flow automation	1.0.0
filter-by-user = <boolean>	Apply filter using logged user	true

3.2.1.2 Execution Response Properties

The response attribute meanings are clear by name. The response contains the minimum amount of information required to identify an execution object.

Name	Description
<createdBy=<String>> (required)	The user name of execution creator.
<duration=<Integer>>	Duration of execution.
<endTime=<Date-only>>	The date the execution was ended.
<executedBy=<String>> (required)	User who started the execution.
<flowId=<String>> (required)	A unique identifier for a flow.
<flowName=<String>> (required)	The name given to a flow.
<flowVersion=<String>> (required)	The current version of flow.
<name=<String>> (required)	The name given to the execution.
<numberActiveUserTasks=<Integer>> (required)	The current number of active user tasks in the instance.
<startTime=<Date-only>> (required)	The date the execution was started.
<state=<String>> (required)	The current state of the execution.
<summaryReport=<string>>	Execution summary report.

3.2.2 /v1/executions/usertasks/{task-id}/complete

POST	Completes the usertask with the given task-id

Input Parameters

Name	Description	Example
usertask-input-files= <FileType>	Files can be attached if required by the user task	test.txt
usertask-input = <json>	User task input	{"uploadFile": {"uploadedFile": {"fileName":"hosts.txt"}}}

Note: For completing `set user task with file input` the only supported file format is zip.

The zip file must have following structure:

- The zip file must contain a setup folder.
- The setup folder must contain a file called `flow-input.json`.
- If there are user tasks in the flow that require file input, these files must be in the setup folder.

3.2.3 /v1/executions/usertasks/{task-id}/schema

GET	Retrieves the schema of the usertask with given task-id

3.2.3.1 Schema Response Properties

The response attribute meanings are clear by name. The response contains the minimum amount of information required to identify an execution object.

Name	Description
<id=<String>> (required)	A unique identifier for the schema.
<name=<String>> (required)	Schema name.
<status=<String>> (required)	User task status.
<taskDefinitionId=<String>> (required)	Definition id of the task.
<schema=<String>> (required)	json schema.

3.2.4 /v1/executions/{flow-execution-name}

DELETE	Deletes the execution with the given flow-execution-name. It applies to setup phase and execution that is in EXECUTED, STOPPED, or SUSPENDED state.

3.2.4.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow

3.2.5 /v1/executions/{flow-execution-name}/report

GET	Retrieves a report of the execution with the given flow-execution-name

3.2.5.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow

3.2.6 /v1/executions/{flow-execution-name}/report-schema

GET	Retrieves the report schema of the execution with the given flow-execution-name

3.2.6.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow.	com.ericsson.oss.services.shm.asu.flow

3.2.7 /v1/executions/{flow-execution-name}/flowinput-schema-data

GET	Retrieves the data entered during the setup phase of a running flow instance with given flow-execution-name and flow id

3.2.7.1 Query Parameter

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow

3.2.8 /v1/executions/{flow-execution-name}/usertasks

GET	Retrieves outstanding user tasks of a running flow instance with given flow-execution-name and flow id

3.2.8.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow

3.2.8.2 User Task Response Properties

The response attribute meanings are clear by name. The response contains the minimum amount of information required to identify an execution object.

Name	Description
<id=<String>> (required)	A unique identifier for the user task.
<name=<String>> (required)	User task name.
<status=<String>> (required)	The current status of the user task.
<taskDefinitionId=<String>> (required)	Task definition id of user task.

3.2.9 /v1/executions/{flow-execution-name}/suspend

GET	Suspends a running instance of the flow with the given flow-execution-name

3.2.9.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow

3.2.10 /v1/executions/{flow-execution-name}/suspend

GET	Suspends a running instance of a flow with given flow-execution-name

3.2.10.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow
flow-version = <String>	The current version of flow automation	1.0.0

3.2.11 /v1/executions/{flow-execution-name}/stop

PUT	Stop the flow with the given flow-id.

3.2.11.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow
flow-execution-name = <String>	A name given to a flow-execution.	flow-execution-flow

3.2.12 /v1/executions/{flow-execution-name}/download/{resource}

GET	Downloads the requested resource specified using the "resource" path variable for the given "flow-execution-name". The resource in the request path can have the following values: 1. **flowinput** - To download the setup configuration used in the setup phase of the flow execution in the zip file format. 2. **report** - To download the execution report of the flow execution in zip file format. 3. **all** - To download the setup configuration and execution report of the flow execution in zip file format.

3.2.12.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow	com.ericsson.oss.services.shm.asu.flow

3.2.12.2 /v1/executions/usertasks/{task-id}/dictionary

GET	Gets the i18n dictionary to translate a specific usertask, based on given task-id.

3.2.13 /v1/executions/usertasks/{task-id}/dictionary

GET	Gets the i18n dictionary to translate a specific usertask, based on given task-id.

3.2.14 /v1/executions/report-dictionary

GET	Gets the i18n dictionary to translate the execution report.

3.2.14.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow.	com.ericsson.oss.services.shm.asu.flow
flow-version = <String>	The current version of a flow.	1.0.0

3.2.15 /v1/executions/flowinput-schema-data-dictionary

GET	Gets the i18n dictionary to translate the flow-input-schema data.

3.2.15.1 Query Parameters

Name	Description	Example
flow-id = <String>	A unique identifier for a flow.	com.ericsson.oss.services.shm.asu.flow
flow-version = <String>	The current version of a flow.	1.0.0

3.3 Response Codes

A minimum set of response codes are provided below. Additional details can be found in the response body.

Response Code	Description
HTTP 200 OK	The request has succeeded. The response returns an information on the message-body.
HTTP 201 Created	The request has been fulfilled and resulted in a new resource being created.
HTTP 204 No Content	The request has succeeded. The response does not need to return a message-body.
HTTP 400 Bad Request	The request could not be understood by the server due to malformed syntax. Information message about the specific error is also returned. Examples: Missing mandatory properties, conflicting names.
HTTP 401 Unauthorized	The user does not have the necessary authorization to access the resource. Contact the system administrator.
HTTP 403 Forbidden	The system does not have the necessary license to access the resource. Contact the system administrator.
HTTP 404 Not Found	The server has not found anything matching the Request-URI.
HTTP 5XX	This represents any response code from 500-599. The request has failed. The response code indicates problem on the service side.

4 Flow Automation: Working with the NBI

This section explains, with examples, how to use the Flow Automation NBI REST feature. The examples provided demonstrate how the client of Flow Automation NBI must use the interface.

Authorization

A Resource is defined for the Flow Automation NBI:

— flowautomation

Six capabilities are available for use with custom roles:

1. install
2. enable
3. remove
4. activate
5. execute
6. read

Two predefined roles with the mentioned capabilities are available:

Flow Automation application supports two predefined application specific roles:

— Flowautomation_Administrator

  Allows access to Flow Automation application as administrator, so it would be possible to install new flows and remove existing ones. It also should be possible to enable or disable existing flows and activate and deactivate versions from the existing flows and finally it should be possible to see all the flows.

— Flowautomation_Operator

  Allows access to Flow Automation application as operator, so it would be possible to see all the flows and execute them.

Resources, actions, and associated commands allowed for each role:

Role	Resource	Operations	Action/Command
Flowautomation_Administrator	flowautomation	install	Allows the installation of new flows into the flowautomation application.

Role	Resource	Operations	Action/Command
		remove	Allows the removal of existing flows from the flowautomation application.
		enable	Allows enabling/disabling flows from the flowautomation application.
		activate	Allows activating/ deactivating flows from the flowautomation application.
		execute	Allows the execution of flows from the flowautomation application. **Note:** The user will still need the underlying capabilities existing within the flow to successfully finish the execution of the flow.
		read	Allows seeing flows details and listing them from the flowautomation application.
Flowautomation_Operator	flowautomation	execute	Allows the execution of flows from the flowautomation application. **Note:** The user will still need the underlying capabilities existing within the flow to successfully finish the execution of the flow.
		read	Allows seeing flows details and listing them from the flowautomation application.

4.1 Authentication

Prerequisites:

— A valid Username and Password, with a Flow Automation Rest NBI role or capability to access the ENM System.

— A fully deployed ENM System.

Authentication via NBI

Establish a session with the ENM system. See *How to Export a Valid Certificate for ENM*, *Establish a User Session over REST*, and *Curl examples for Identity and Access Management* in the ENM Identity and Access Management Programmer's Guide [5] for details.

The session is required to call the Flow Automation NBI operations.

4.2 Usage Scenarios

Describes some examples of the Flow automation NBI interface. In these examples, the curl command is used to send the REST requests.

Before the requests can be submitted, you must be authorized and a cookie file received. This cookie file must be provided for the subsequent requests.

For security purposes a file containing the user credentials should be created using an editor. The contents of the file should be in the following format:

```
&IDToken1=<username>&IDToken2=<password>
```

In the following <customer-domain> must be replaced by the hostname of the ENM instance, <user-credentials-filename> must be replaced by the user credentials filename created, and include the filepath with the filename if the file is not in the current directory.

Use the following to request authorization and get the cookie:

```
curl --insecure --request POST --data-ascii @<user-
credentialsfilename> --cookie-jar cookie.txt 'https://<customer-
domain>/login'
```

4.2.1 Flow Queries

4.2.1.1 Get List of Flows

List the flows in the system.

Prerequisites:

— Flowautomation_Operator role

or

— Flowautomation_Administrator role

Request:

```
curl -G -s --insecure --request GET --cookie cookie.txt 'https://
<customer-domain>/flowautomation/v1/flows'
```

Response:

```
[
    {
        "id": "com.ericsson.oss.fa.flows.helloworld",
```

```
        "name": "Hello World",
        "status": "enabled",
        "flowVersions": [
            {
                "version": "1.0.1",
                "description": "Hello World",
                "active": true,
                "createdBy": "administrator",
                "createdDate": "2019-01-09 14:48:57.0"
            }
        ]
    },
    {
        "id": "com.ericsson.oss.fa.flows.package-restructure",
        "name": "Flow with user task schema",
        "status": "enabled",
        "flowVersions": [
            {
                "version": "1.0.3",
                "description": "Long delay in groovy script",
                "active": true,
                "createdBy": "administrator",
                "createdDate": "2019-01-09 14:59:31.0"
            }
        ]
    }
]
```


4.2.1.2 Import a Flow

Imports a new flow into the system.

Request:

```
curl --insecure --request POST 'https://<customer-domain>/
flowautomation/v1/flows' --cookie cookie.txt -H 'Accept:
application/json' -H 'content-type: multipart/form-data;
boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW' -F 'flow-
package=@C:dakka\resources\FBB\Joel\fbb-demo-1.0.0.zip'
```

Response:

Returns the new flow object that is created:

```
{
    "id": "com.ericsson.oss.fa.flows.fbb",
    "name": "FBB Flow",
    "status": "enabled",
    "flowVersions": [
        {
            "version": "1.0.0",
            "description": "FBB Flow example",
            "active": true,
            "createdBy": "administrator",
            "createdDate": "2019-01-14 14:30:34.0"
        }
    ]
}
```


4.2.1.3 Enable or Disable a Flow

A flow can be enabled or disabled using this API.

Request:

Enable a Flow:

```
curl --insecure --request PUT --cookie cookie.txt 'https://
<customer-domain>/flowautomation/v1/flows/{flowId}/enable' -H
'Content-Type: application/json' -d '{ "value": "true" }'
```

Disable a Flow:

```
curl --insecure --request PUT --cookie cookie.txt 'https://
<customer-domain>/flowautomation/v1/flows/{flowId}/enable' -H
'Content-Type: application/json' -d '{ "value": "false" }'
```

Response:

```
204 No Content
```

4.2.2 Executions Queries

4.2.2.1 Get List of Executions

To get list of all the executions in the system:

Request:

```
curl --insecure --request GET 'https://<customer-domain>/
flowautomation/v1/executions' --cookie cookie.txt -H 'Accept:
application/json'
```

Response:


```
[
    {
        "name": "James",
        "flowId": "com.ericsson.oss.fa.flows.helloworld",
        "flowName": "Hello World",
        "flowVersion": "1.0.1",
        "createdBy": "administrator",
        "executedBy": "administrator",
        "startTime": 1547045494404,
        "endTime": 1547045530502,
        "duration": 36098,
        "summaryReport": null,
        "state": "COMPLETED",
        "numberActiveUserTasks": 0
    },
    {
        "name": "test-1",
        "flowId": "com.ericsson.oss.fa.flows.rest-client",
        "flowName": "Rest Client Flow without Setup phase",
        "flowVersion": "1.0.1",
        "createdBy": "administrator",
        "executedBy": "administrator",
        "startTime": 1547121821225,
        "endTime": 1547121821481,
        "duration": 256,
        "summaryReport": null,
        "state": "COMPLETED",
        "numberActiveUserTasks": 0
    }
]
```


To list all the executions of a flow

Request:

```
curl --insecure --request GET 'https://<customer-domain>/
flowautomation/v1/executions?flow-id=<Flow ID>' --cookie
cookie.txt -H 'Accept: application/json'
```

Response:


```
[
    {
        "name": "test34234234",
        "flowId": "com.ericsson.oss.fa.flows.helloworld",
        "flowName": "Hello World",
        "flowVersion": "1.0.1",
        "createdBy": "administrator",
        "executedBy": "administrator",
        "startTime": 1547136645614,
        "endTime": 1547136680370,
        "duration": 34756,
        "summaryReport": null,
        "state": "COMPLETED",
        "numberActiveUserTasks": 0
    },
    {
        "name": "James",
        "flowId": "com.ericsson.oss.fa.flows.helloworld",
        "flowName": "Hello World",
        "flowVersion": "1.0.1",
        "createdBy": "administrator",
        "executedBy": "administrator",
        "startTime": 1547045494404,
        "endTime": 1547045530502,
        "duration": 36098,
        "summaryReport": null,
        "state": "COMPLETED",
        "numberActiveUserTasks": 0
    }
]
```


4.2.2.2 Executing a Flow

Execute a Flow:

Request:

```
curl --insecure --request POST 'https://<customer-domain>/
flowautomation/v1/flows/<FLOW-ID>/execute' --cookie cookie.txt -H
'Content-Type: application/json' -d '{ "name":"<Execution-
Name>"}'
```

Response:

```
{
    "name": "demo-execution-1"
}
```

4.2.2.3 Executing a Flow with Flow Input File

A flow can be executed with the following API, using a JSON file.

This API is used for those flows that require an input in the setup phase.

Request 1:

```
curl --insecure --request POST --cookie cookie.txt 'https://<customer-domain>/fl →
owautomation/v1/flows/<FLOW-ID>/execute' -H 'Accept: application/json' -H 'cache →
-control: no-cache' -H 'Content-Type: multipart/form-data; boundary=---- WebKitF →
ormBoundary7MA4YWxkTrZu0gW' -F 'name=demo-execution-1' -F 'file-name=<flowInput. →
json>' -F 'flow-input=@/ericsson/tor/data/<flowInput.json>'
```

Note: `file-name` attribute value must match the selected file from the `flow-input` attribute (example: `flowInput.json`). The file must not be empty.

Response 1:

```
{
    "name": "demo-execution-1"
}
```

A flow can be executed using following API, using a Zip file:

Request 2:

```
curl --insecure --request POST 'https:///flowautomation/v1/flows//execute' --coo →
kie cookie.txt -H 'Accept:application/json' -H 'cache-control: no-cache' -H 'con →
tent-type: multipart/form-data; boundary=----WebKitFormBoundary7MA4YWxkTrZu0gW' →
-F 'name=demo-execution-2' -F 'file-name=flow-input.json' -F 'flow-input=@111-se →
tup.zip'
```

Note: The `file-name` value, `flow-input.json`, must exist within the content of the zip file. The file must not be empty.

Response 2:

```
{
    "name": "demo-execution-2"

}
```

4.2.2.4 Executing a Flow with Permission Requirements

The request is similar to Executing a Flow on page 19 request above but with an extra Header variable.

Request:

```
curl --insecure --request POST 'https://<customer-domain>/flowautomation/v1/flow
s/<Flow ID>/execute' --cookie cookie.txt -H 'Accept: application/json' -H 'Conte
nt-Type: application/json' -d '{ "name": "<EXECUTION-NAME>"}' -H 'checkUserFlowE
xecutionPermission: true'
```

Note: `checkUserFlowExecutionPermission` is true if permission check is to be performed and false if not. Default is true.

Response:

```
{
    "status": 403,
    "reasonPhrase": "The flow has declared set of capabilities needed by the use
r to successfully execute it",
    "errorDetail": "",
    "errors": [
        { "code": "flow-execution-2181", "errorMessage": "User does not have req
uired capabilities to execute this flow" }
    ]
}
```

Note: If the user has all the permissions required to execute the flow, flow will start successfully. If the user does not then services will throw an error with the above response. If the user still wishes to execute the flow, the REST endpoint would change the header parameter `checkUserFlowExecutionPermission` as false .

FA NBI Reference List

	Reference	Description	Document Number
[1]	ENM Glossary of Terms	Glossary of ENM terminology	1/0033-AOM 901 151
[2]	RFC 2616	Hypertext Transfer Protocol – HTTP/1.1	N/A
[3]	RFC 3986	(plus errata) - the current generic URI syntax specification	N/A
[4]	RFC4627	The application/ json Media Type for JavaScript Object Notation (JSON)	N/A
[5]		ENM Identity and Access Management Programmer's Guide	19817-CNA 403 3016