HCLSoftware

HCL DRYiCE
IEM API Guide

IEM Publish Service API Guide
IEM Bi-directional Integration with ITSM API Guide

Version 1.0.0

HCL has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the HCL website at www.hcltechsw.com

Table of Contents

Table of Figures

List of Tables

Document Revision History

This guide is updated with each release of the product or when necessary.

This table provides the revision history of this API Guide.

Version Date	Description
April, 2024	Dryice IEM v1.0.0 API Guide

1 Preface

This section provides information about the IEM PUBLISH SERVICE API Guide, IEM Bi-directional Integration API guide and includes the following topics.

- Intended Audience
- Conventions

1.1 Intended Audience

This guide is intended for users who have access and authorized to use IEM publish service, IEM Bi-directional Integration APIs.

1.2 Conventions

The following typographic conventions are used in this document.

Table 1 - Conventions

Convention	Element
Boldface	Indicates graphical user interface elements associated with an action, or terms defined in text or the glossary
Blue Underline face	Indicates cross-reference and links
`Courier New (Font)`	Indicates commands within a paragraph, URLs, code in examples, and paths including onscreen text and text input from users
Italic	Indicates document titles, occasional emphasis, or glossary terms
Numbered lists	Indicates steps in a procedure to be followed in a sequence
Bulleted lists	Indicates a list of items that is not necessarily meant to be followed in a sequence

2 IEM API Overview

2.1 IEM Publish Service Overview

IEM Publish is a REST API service which helps to ingest the data into IEM system with different APIs using authentication.

2.2 IEM Bi-directional Integration Overview

IEM Bi-directional Integration is a REST API service which helps to auto update the actionable state to resolved in IEM using an authentication when ticket is getting resolved at ITSM tool.

3 About IEM Publish Service APIs

IEM Publish service offers a set of REST APIs. Using these APIs, a user establishes the connection between IEM and Nifi connectors using IMM console, which will have a continuous live data from Nifi to IEM.

Listed below are the methods that are supported by IEM Publish Service API calls.

- **HTTP POST**: Methods to create or change an object.

3.1 IEM Publish Service Rest API Request

To consume a functionality of IEM, Publish Service provided by a specific API, an enterprise tool needs to place a request. This section details the construct of that request that is supported by IEM Publish Service. Requests are typically categorized in terms of the following requested operation:

- create (POST) or retrieve (GET).

3.1.1 Authentication

HTTP communications between IEM Publish API client and servers are secured with SSL. IEM Publish API supports token based authentication with cookies. To get a token, the user needs to provide a valid username and password using basic authentication to token API.

- To get the credentials, contact the IEM Admin of an organization (Provider/ organization).
- It is important to have the user creds listed below to make a valid API call to IEM Publish API.

- **Email**: A valid username to access the API
- **Password**: A valid password to authenticate the API

3.1.2 Request Header

The following HTTP headers are included in IEM Publish Service API requests.

Table 2 - Request Header Fields

Headers	Description
Cookie	All requests from authenticated clients must include "**access_token**" for cookies in headers

3.2 IEM Publish Service Rest API Responses

Once an enterprise tool requests a IEM Publish service API, an API response is sent to that tool from IEM Publish API post successful authentication. This section details the format of that response.

- All responses include a HTTP status code, e.g., 200 for "success", 403 for "Forbidden" etc. Hence, response content depends on the request.
- The response body to a POST request contains a message if data has been published or not.

3.2.1 HTTP Response Codes

Every response for an IEM Publish Service API call, has a response code embedded within it. The table below lists the meaning associated with all the response codes used in IEM Publish Service API call responses. Each of these response codes identifies the reason behind the action that was taken in an API call.

Table 3 – HTTP Response Codes

Status Code	Status Description
200 OK	The request is valid and was completed. The response includes a document body.
400 BadRequest	The request body is malformed, incomplete, or otherwise invalid.
401 Unauthorized	An authorization header was expected but not found.
403 Forbidden	The response status code indicates that the server understands the request but refuses to authorize it.
404 Not Found	One or more objects specified in the request could not be found in specified IEM Publish Service API directory.
500 Internal Server Error	The request was received but could not be completed because of an internal error in the server.

3.3 How to call IEM Publish Service APIs

This section explains how users can consume IEM Publish Service APIs. The tool that has been used to demonstrate the IEM Publish Service API consumption in this guide is "**Postman**".

3.3.1 Calling Token API to get a token

1. Define **methodtype** = **POST.**

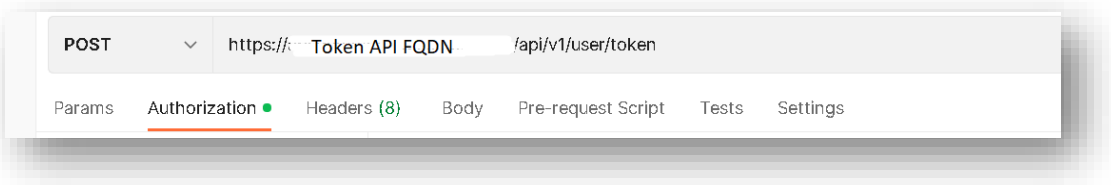

Figure 1 – How to consume token API

2. Define the basic authentication with **username** and **password**.

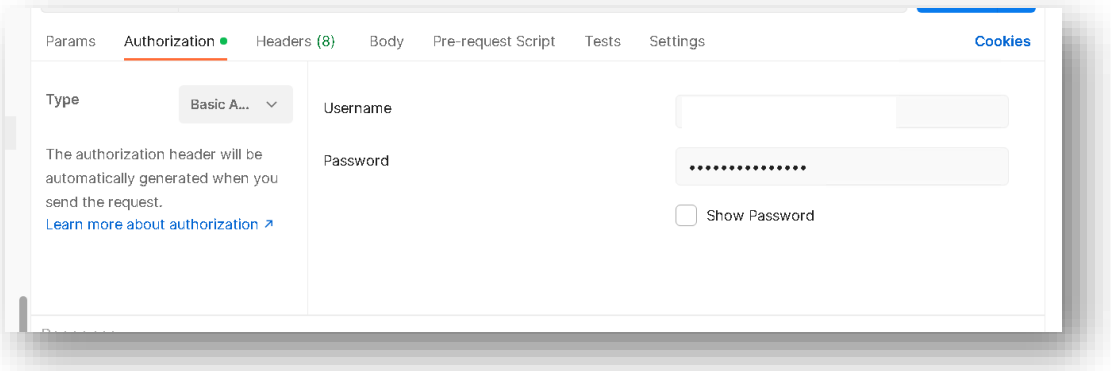

Figure 2 – How consume token API (cont.)

3. Click on Send button to get a token in response body which will be used to access all the APIs of IEM Publish Service in a session.
4. If user creds are valid, API will return access token in response.

Figure 3 – How to consume Token API (cont.)

5. If user creds are invalid, API will return warning message

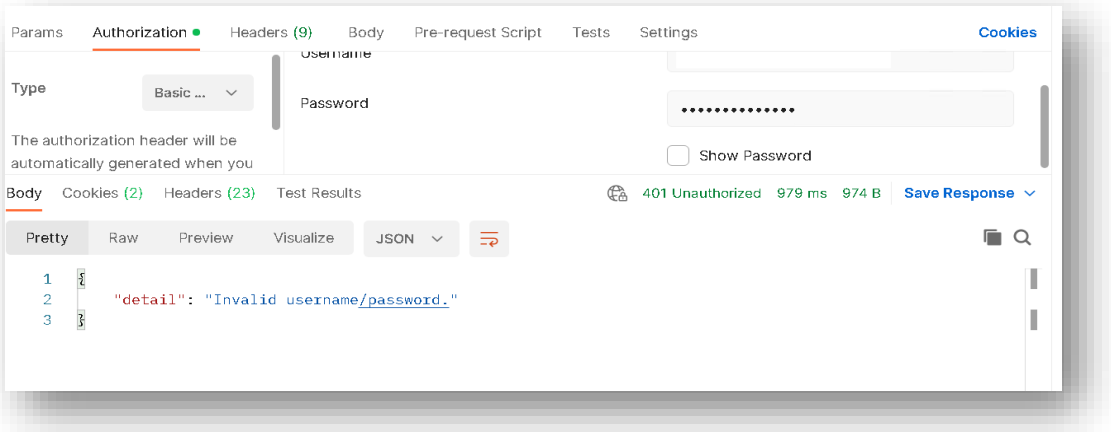

Figure 4 – How to consume Token API (cont.)

3.3.2 Calling Publish API endpoint with access token

Set cookies with access token for Publish service API domain

Method 1:

1. Click on cookies.

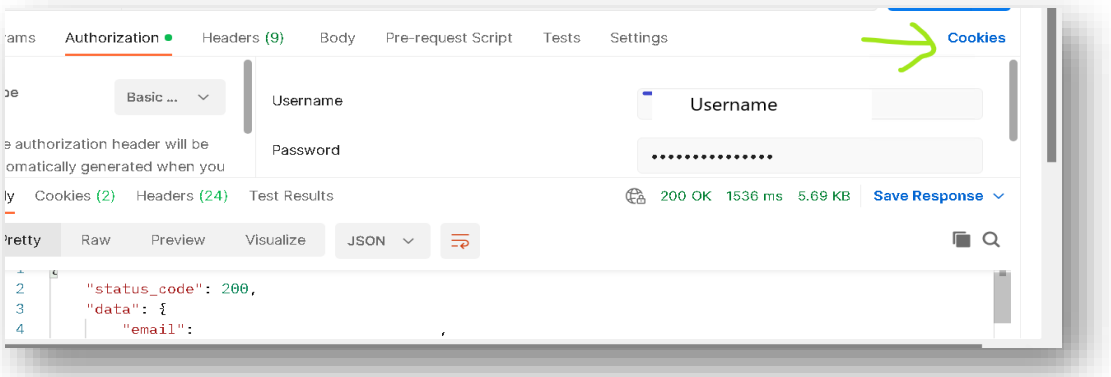

Figure 5 – How to consume Publish APIs

2. Add a domain of Publish Service API.

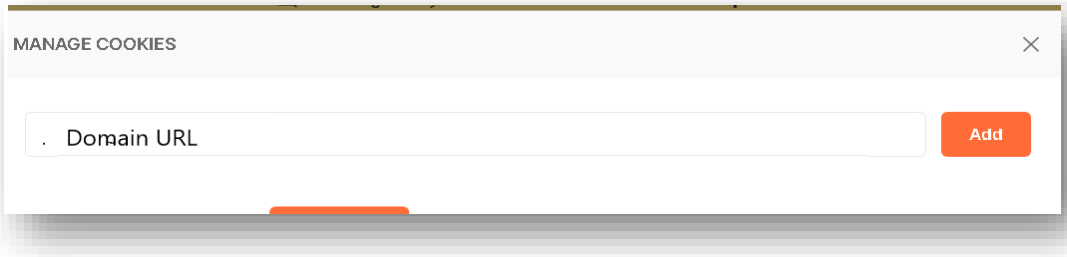

Figure 6 - How to consume Publish APIs (cont.)

3. Click on add cookies.

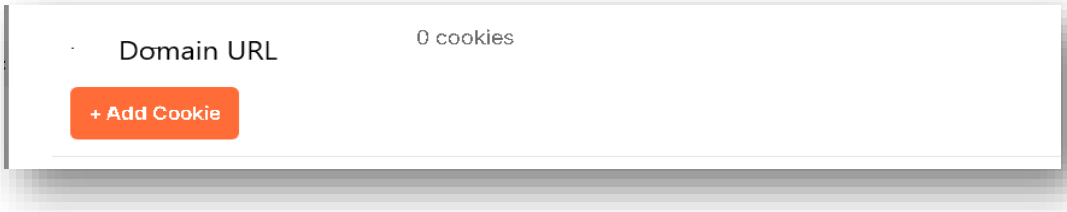

Figure 7 – How to consume Publish APIs (cont.)

4. Set access token key with token value.

Figure 8 – How to consume Publish APIs (cont.)

5. Close the cookies window.

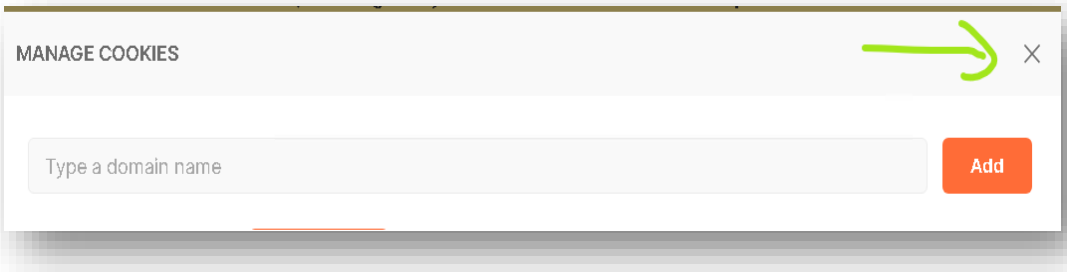

Figure 9 – How to consume Publish APIs (cont.)

Method 2:

1. Set cookie with access token in API headers

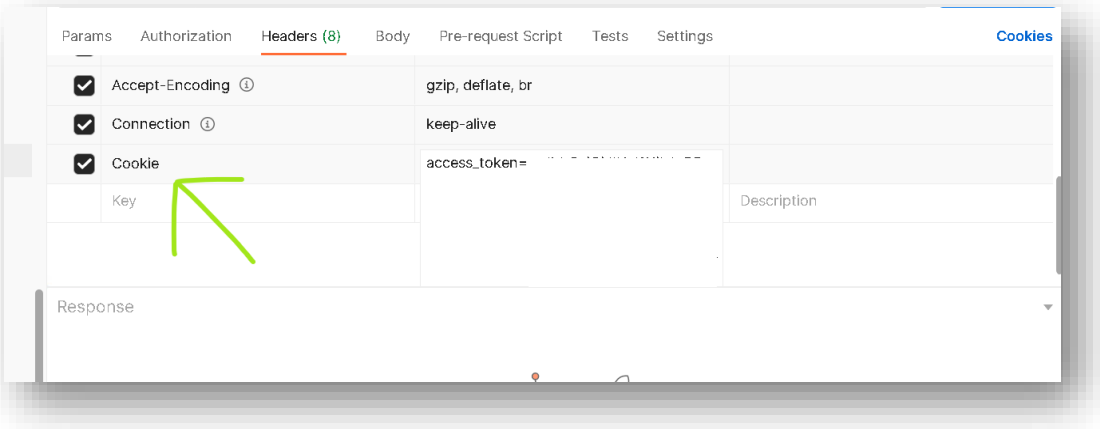

Figure 10 – How to consume Publish APIs (cont.)

2. Enter the Publish API endpoint with "**POST**" as method type.

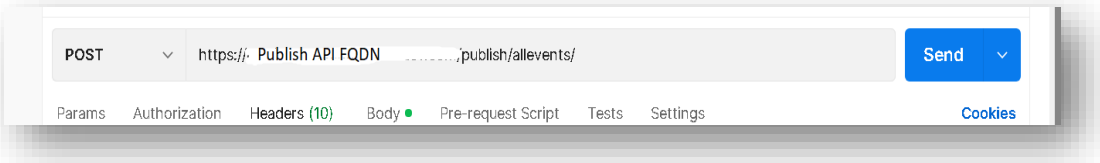

Figure 11 – How to consume Publish APIs (cont.)

3. Enter valid Json for API body/payload.

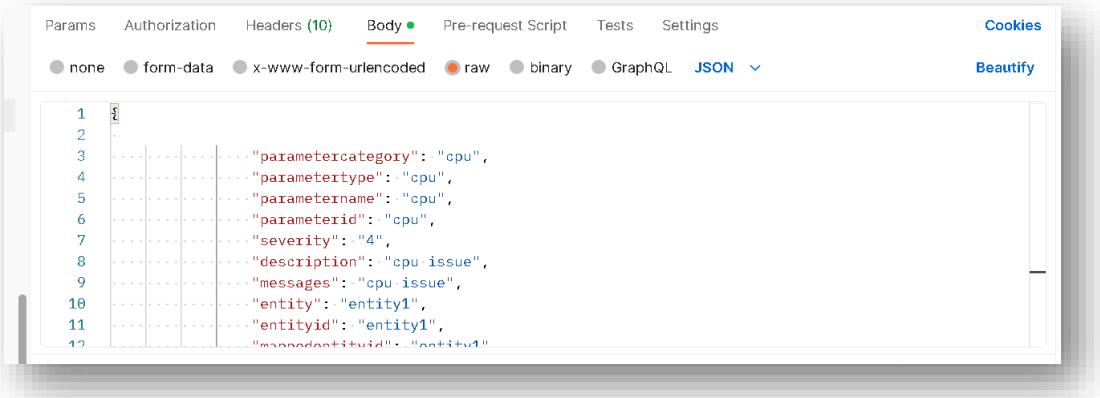

Figure 12 – How to consume Publish APIs (cont.)

4. Click on **Send** to get API response.
5. API response will contain a success or failure message based on json provided

Success case:

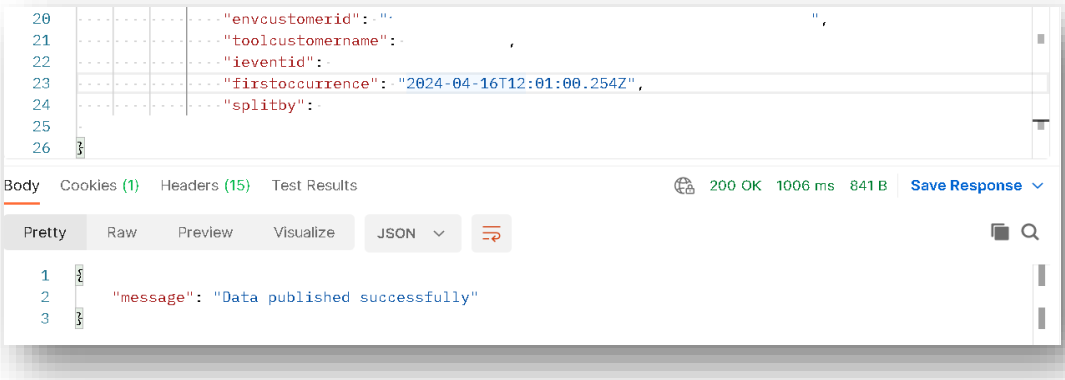

Figure 13 – How to consume Publish APIs (cont.)

Failure cases:

1. When access token is not provided.

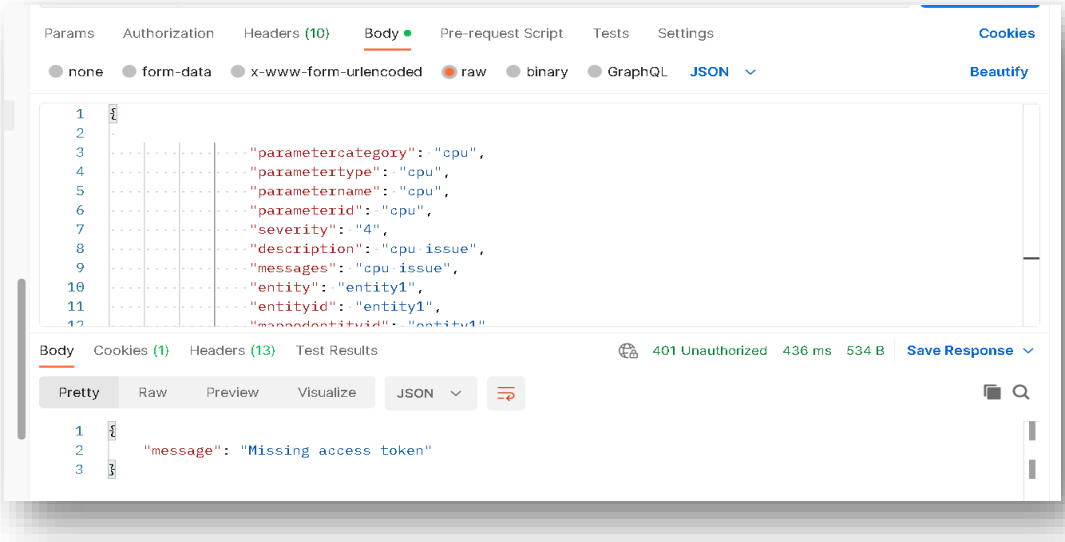

Figure 14 – How to consume Publish APIs (cont.)

2. When invalid token value is provided.

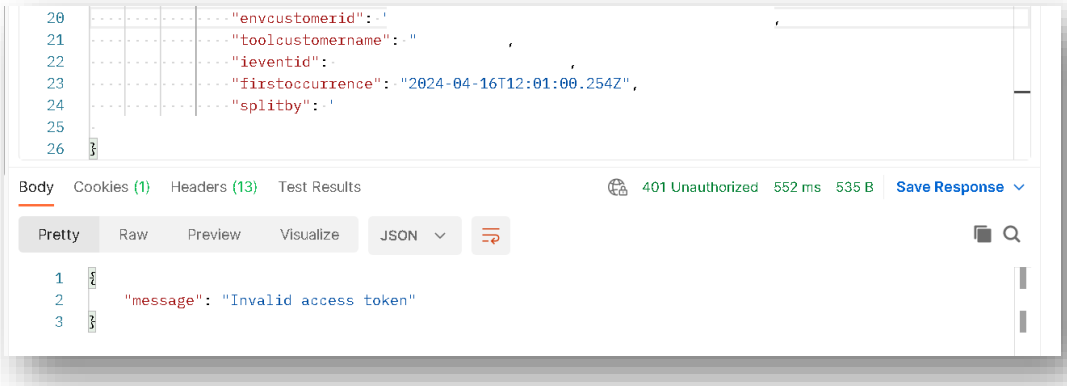

Figure 15 – How to consume Publish APIs (cont.)

3. When data has been sent for inactive customer or customerid is invalid.

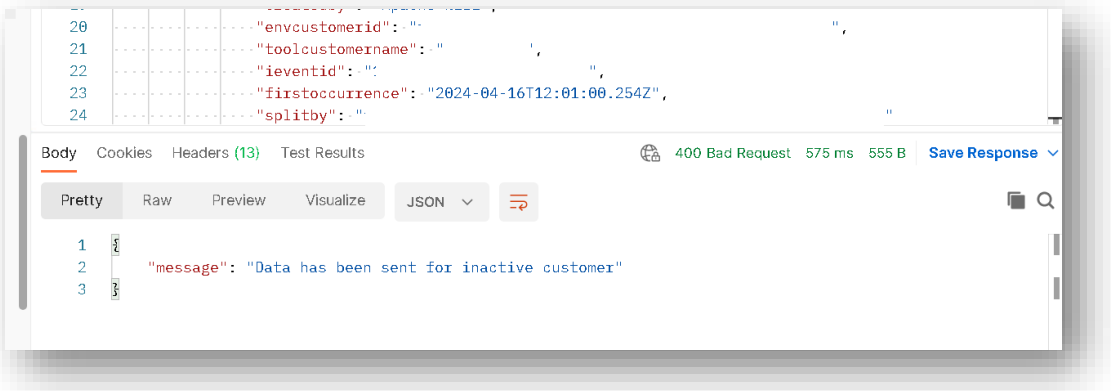

Figure 16 – How to consume Publish APIs (cont.)

4. When customer id is valid, but user is not mapped to that customer

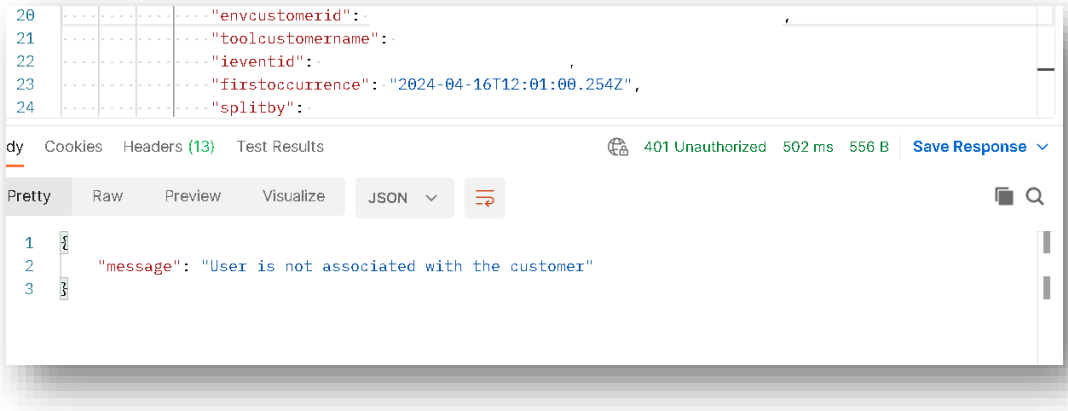

Figure 17 – How to consume Publish APIs (cont.)

3.3.3 IEM Publish Service API Details

This section provides the details of all the existing APIs for IEM Publish Service. The APIs in the current version of IEM Publish Service utilize the GET and POST methods. Listed in the table below are the APIs categorized based on these methods.

Table 4 – IEM Publish Service GET API Details

GET APIS – To check if API service is running or not		
S. No.	**API**	**API Description**
1	https: {{domain}}/publish/healthcheck/	The API returns a message "API is running" with 200 status code

Table 5 – IEM Publish Service POST APIs

POST APIS – To publish data into IEM		
S. No.	**API**	**API Description**
1	Generate Token	This API is used to generate the security token with the valid user credentials.
2	Publish Events data	This API is to send the events data into GCP pubsublite topics, It will be processed further and will be visible over IEM console

3	Publish Metrics data	This API is to send the metrics data into GCP pubsublite topics, It will be processed further and will be visible over IEM console
4	Publish Entity Topology data	This API is to send the entity topology data into GCP pubsublite topics, It will be processed further and will be visible over IEM console
5	Publish Service Topology data	This API is to send the service topology data into GCP pubsublite topics, It will be processed further and will be visible over IEM console
6	Publish Healthcheck data	This API is to send the Nifi adaptors health check data into GCP pubsublite topics, it will be processed further and will be visible over IEM console

3.3.4 Generate Token

Table 6 – GENERATE TOKEN

Element	Description
API	Generate Token
DESCRIPTION	API returns the access token based on username and password to authenticate IEM Publish APIs.
METHOD	POST
URL	https://<Hosted_API>/api/v1/user/token
BODY	NA
HEADER	Authorization headers using basic authentication with username and password

RESPONSE (SUCCESS CASE: 200 OK STATUS CODE)	{
	"status_code": 200,
	"data": {
	"email": "<user_email>",
	"name": "<USERNAME>",
	"is_superadmin": false,
	"access_token": "<token_value>",
	"refresh_token": "<refresh_token_value>",
	"associated_with_customers": [
	{ }
	]
	},
	"msg": ""
	}

RESPONSE PARAMETER	ACCESS_TOKEN

RESPONSE (FAILED CASE: 401 UNAUTHORIZED STATUS CODE)	{
	"detail": "Invalid username/password."
	}

3.3.5 API: Publish Events Data

Table 7 – Publish Events Data

Element	Description

API	Publish Events data
Description	This API is to send the events data into GCP pubsublite topics, It will be processed further and will be visible over IEM console
Method	POST
URL	https://<Hosted_API>/publish/allevents/
Header	Cookies with access_token value

Body	{
	"parametercategory": "cpu",
	"parametertype": "cpu",
	"parametername": "cpu",
	"parameterid": "cpu",
	"severity": "4",
	"description": "cpu issue",
	"messages": "cpu issue",
	"entity": "entity1",
	"entityid": "entity1",
	"mappedentityid": "entity1",
	"agentlocation": "<agentlocation_value>",
	"createdon": "2024-04-16T12:01:00.254Z",
	"toolmanager": "<toolmanager>",
	"manageragent": "<manager_agent>",
	"datasourcename": "<datasource>",
	"toolcustomerid": "<customer_name>",
	"createdby": "",
	"envcustomerid": "<customer_id>",
	"toolcustomername": "<customer_name>",
	"ieventid": "11-j2-5-c-asd-nd34b46644",
	"firstoccurrence": "2024-04-16T12:01:00.254Z",
	"splitby": "<customer_id>_entity1_cpu"
	}

Response (Success Case : 200 OK status code)	{
	"message": "Data published successfully"
	}

Response (Failed Case 1 : 401 Unauthorized status code)	{
	"message": "Missing access token"
	}

Response	{ "message": "Invalid access token"

(Failed Case 2 : 401 Unauthorized status code)	}

Response (Failed Case 3 : 401 Unauthorized status code)	{
	"message": "Access token has expired"
	}

Response (Failed Case 4 : 401 Unauthorized status code)	{
	"message": "Data has been sent for inactive customer"
	}

Response (Failed Case 5: 401 Unauthorized status code)	{
	"message": "User is not associated with the customer"
	}

3.3.6 API: Publish Metrics Data

Table 8 – Publish Metrics Data

Element	Description
API	Publish Metrics data
Description	This API is to send the metrics data into GCP pubsublite topics, It will be processed further and will be visible over IEM console
Method	POST
URL	https://<Hosted_API>/publish/metrics/
Header	Cookies with access_token value
Body	{ "parametername" : "Hardware Resources in percentage", "severity" : "1", "datasum" : 0, "dataavg" : 0, "datamax" : 0, "datamin" : 0, "datacount" : "1", "entity" : "e145", "mappedentityid" : "gamma_entity2", "createdon" : "2024-01-10 10:57:00", "toolmanager" : "<toolmanager>", "manageragent" : "<toolmanager>", "datasourcename" : "<datasource>", "createdby" : "",

	"envcustomerid" : "<customer_id>", "toolcustomername" : "<customer_name>", "starttime" : "2024-01-10 10:57:00", "endtime" : "2024-01-10 10:57:00", "iperfid" : "78b9cdb4-5446-4b25-9e40-b3b36a40c638" }

Response (Success Case: 200 OK status code)	{
	"message": "Data published successfully"
	}

Response (Failed Case 1 : 401 Unauthorized status code)	{
	"message": "Missing access token"
	}

Response (Failed Case 2 : 401 Unauthorized status code)	{
	"message": "Invalid access token"
	}

Response (Failed Case 3 : 401 Unauthorized status code)	{
	"message": "Access token has expired"
	}

Response (Failed Case 4 : 401 Unauthorized status code)	{
	"message": "Data has been sent for inactive customer"
	}

Response (Failed Case 5: 401 Unauthorized status code)	{
	"message": "User is not associated with the customer"
	}

3.3.7 API: Publish Entity Topology Data

Table 9 – Publish Entity Topology Data

Element	Description
API	Publish Entity Topology data
Description	This API is to send the entity topology data into GCP pubsublite topics, It will be processed further and will be visible over IEM console
Method	POST
URL	https://<Hosted_API>/publish/entity/

Header	Cookies with access_token value

Body	<pre>{
	  "custcol" : {
	  "Category" : "Business Application",
	  "Location" : "India"
	  },
	  "entityid" : "testentityj1",
	  "topology" : {
	  "parententity" : "testentityj1"
	  },
	  "envcustomer_id" : "<customer_id>",
	  "createdon" : "2024-01-23T11:02:59.315Z"
	}</pre>

Response (Success Case : 200 OK status code)	<pre>{
	  "message": "Data published successfully"
	}</pre>

Response (Failed Case 1 : 401 Unauthorized status code)	<pre>{
	  "message": "Missing access token"
	}</pre>

Response (Failed Case 2 : 401 Unauthorized status code)	<pre>{
	  "message": "Invalid access token"
	}</pre>

Response (Failed Case 3 : 401 Unauthorized status code)	<pre>{
	  "message": "Access token has expired"
	}</pre>

Response (Failed Case 4 : 401 Unauthorized status code)	<pre>{
	  "message": "Data has been sent for inactive customer"
	}</pre>

Response (Failed Case 5: 401 Unauthorized status code)	<pre>{
	  "message": "User is not associated with the customer"
	}</pre>

3.3.8 API: Publish Service Topology Data

Table 10 – Publish Service Topology Data

Element	Description
API	Publish Service Topology data

Description	This API is to send the service topology data into GCP pubsublite topics, It will be processed further and will be visible over IEM console
Method	POST
URL	https://<Hosted_API>/publish/service/
Header	Cookies with access_token value

Body	{
	"servicename":"testservicej1",
	"service_entity_mapping":[
	{"entityid":"testentityj1"},
	{"entityid":"testentityj2"}],
	"service_topology":[
	{"parentservicename":"testservicej2"},
	{"parentservicename":"testservicej3"}
	],
	"envcustomer_id":"<customer_id>"
	}

Response (Success Case : 200 OK status code)	{
	"message": "Data published successfully"
	}

Response (Failed Case 1 : 401 Unauthorized status code)	{
	"message": "Missing access token"
	}

Response (Failed Case 2 : 401 Unauthorized status code)	{
	"message": "Invalid access token"
	}

Response (Failed Case 3 : 401 Unauthorized status code)	{
	"message": "Access token has expired"
	}

Response (Failed Case 4 : 401 Unauthorized status code)	{
	"message": "Data has been sent for inactive customer"
	}

Response	{
	"message": "User is not associated with the customer"
	}

(Failed Case 5: 401 Unauthorized status code)	

3.3.9 API: Publish Healthcheck Data

Table 11 – Publish Healthcheck Data

Element	Description
API	Publish Healthcheck data
Description	This API is to send the nifi adaptors healthcheck data into GCP pubsublite topics, It will be processed further and will be visible over IEM console
Method	POST
URL	https://<Hosted_API>/publish/healthcheck/
Header	Cookies with access_token value

Body	{
	"host":"test host2",
	"envcustomer_id": "<customer_id>",
	"kpi_name":"heartbeat",
	"kpi_type":"heartbeat",
	"status":"Green",
	"value":"",
	"description":"Heartbeat received on 2024-02-19 10:20:00.477",
	"remark":"Heartbeat received on 2024-02-19 10:20:00.477",
	"timestamp":"2024-02-19 10:20:00.477",
	"updatetimestamp":"2024-02-19 10:20:00.477",
	"updateid":"Manual testing",
	"additional_info": "{}"
	}

Response (Success Case : 200 OK status code)	{
	"message": "Data published successfully"
	}

Response (Failed Case 1 : 401 Unauthorized status code)	{
	"message": "Missing access token"
	}

Response (Failed Case 2 : 401 Unauthorized status code)	{
	"message": "Invalid access token"
	}

Response	{ "message": "Access token has expired"

(Failed Case 3 : 401 Unauthorized status code)	`}`

Response (Failed Case 4 : 401 Unauthorized status code)	`{`
	`"message": "Data has been sent for inactive customer"`
	`}`

Response (Failed Case 5: 401 Unauthorized status code)	`{`
	`"message": "User is not associated with the customer"`
	`}`

4 About IEM Bi-directional Integration with ITSM APIs

IEM Bi-directional Integration with ITSM offers a set of REST APIs. Using these APIs, a user establishes the connection between IEM and ITSM tools (ex: SNOW, Service Exchange), which will help to auto update the actionable state in IEM when resolving the ticket at ITSM tool.

- Listed below are the methods that are supported by IEM Bi-directional Integration with ITSM API calls.
  - **HTTP POST**: Methods to create or change an object.

4.1 IEM Bi-directional Integration with ITSM Rest API Request

To consume a functionality of IEM Bi-directional Integration with ITSM provided by a specific API, an enterprise tool needs to place a request. This section details the construct of that request that is supported by IEM Bi-directional Integration with ITSM. Requests are typically categorized in terms of the requested operation: create (POST).

4.1.1 Authentication

HTTP communications between IEM Integration API client and servers are secured with SSL. IEM Bi-directional Integtation API supports 1) token based authentication with cookies 2) Basic authentication using username and password. To get a token, the user needs to provide a valid username and password using basic authentication to token API.

- To get the credentials, contact the IEM Admin of an organization (Provider/ organization).
- It is important to have the user creds listed below to make a valid API call to IEM Bi-directional Integration API.
  - **Email**: A valid username to access the API.
  - **Password**: A valid password to authenticate the API.

4.1.2 Request Header

The following HTTP headers are included in IEM Bi-directional Integration API requests.

Table 12 – Request Header Fields

Headers	Description
Cookie	All requests from authenticated clients must include "access_token" for cookies in headers
Authorization	Basic authentication header using username and password

4.2 IEM Bi-directional Integration with ITSM Rest API Responses

Once an enterprise tool requests an IEM Bi-directional Integration API, an API response is sent to that tool from IEM Integration API post successful authentication. This section details the format of that response.

- All responses include a HTTP status code, e.g., 200 for "success", 403 for "Forbidden" etc. Hence, response content depends on the request.
- The response body to a POST request contains a message if request is successful or not.

4.2.1 HTTP Response Codes

Every response for an IEM Bi-directional Integration API call, has a response code embedded within it. The table below lists the meaning associated with all the response codes used in IEM Bi-directional Integration API call responses. Each of these response codes identifies the reason behind the action that was taken in an API call.

Table 13 – HTTP Response Codes

Status Code	Status Description
200 OK	The request is valid and was completed. The response includes a document body.
400 BadRequest	The request body is malformed, incomplete, or otherwise invalid.
401 Unauthorized	An authorization header was expected but not found.
403 Forbidden	The response status code indicates that the server understands the request but refuses to authorize it.
404 Not Found	One or more objects specified in the request could not be found in specified IEM Bi-directional Integration with ITSM API directory.
500 Internal Server Error	The request was received but could not be completed because of an internal error in the server.

4.2.2 How to call IEM Bi-directional Integration with ITSM APIs

This section explains how users can consume IEM Bi-directional Integration APIs. In this guide "**Postman**" tool has been used to explain IEM Integration API calls and considering "**SNOW**" as ITSM tool.

4.2.3 API Authentication

Bi-directional Integration API supports two types of authentications as follows:

1. Basic authentication using username and password.
2. Provide username and password for authorization using basic authentication to Integtation API.

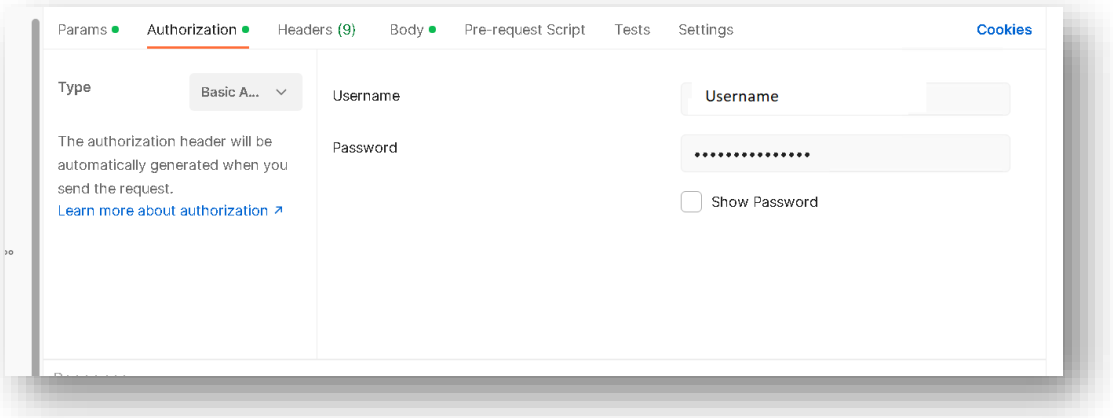

Figure 18 – How to consume token API

3. Token based authentication.

4.2.4 Calling Token API to get a token

1. Define **methodtype** = **POST**

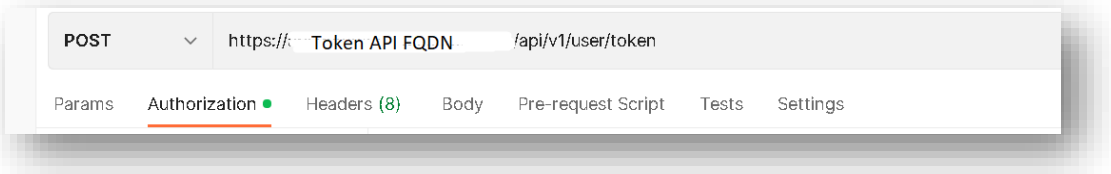

Figure 19 – How to consume token API (cont.)

2. Define the basic authentication with **username** and **password** to token API.

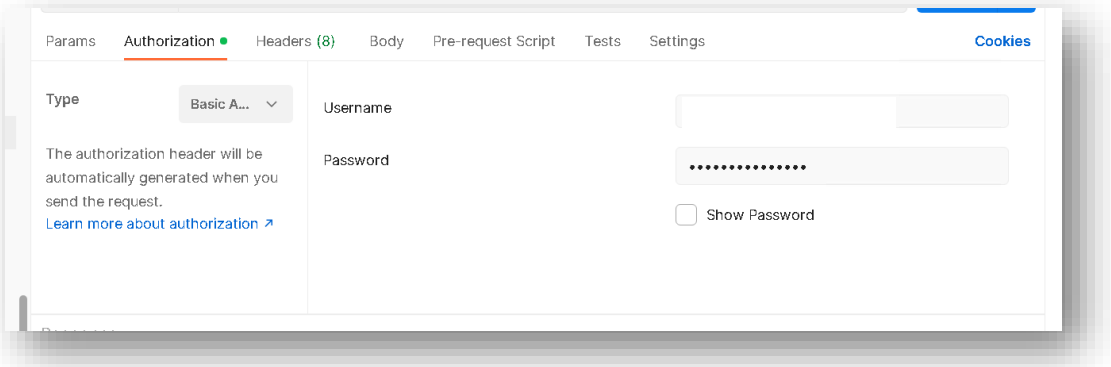

Figure 20 – How consume token API (cont.)

3. Click on **Send** button to get a token in response body which will be used to access the APIs of IEM Bi-directional Integration in a session.
4. If user creds are valid, API will return access token in response

Figure 21 – How to consume Token API (cont.)

5. If user creds are invalid, token API will return a message

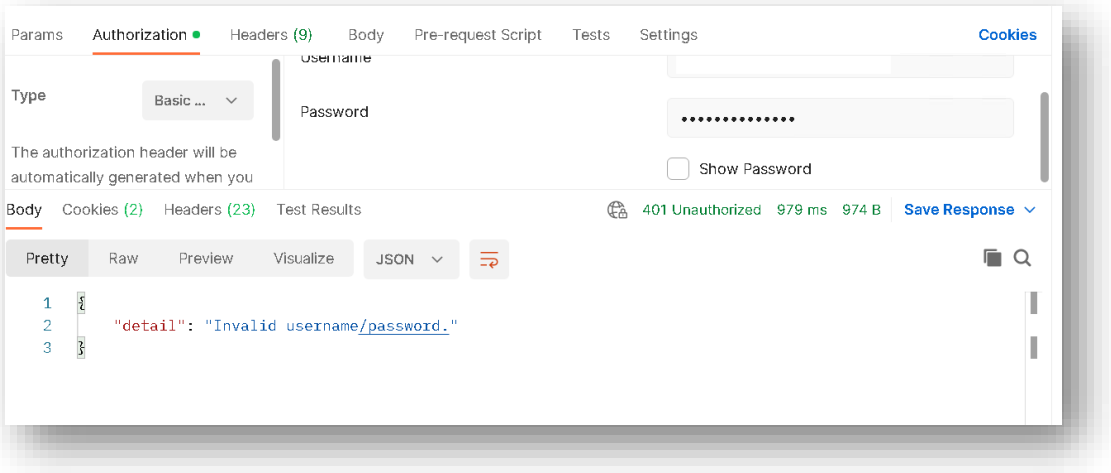

Figure 22 – How to consume Token API (cont.)

4.2.5 Calling Integration API endpoint with access token

Set cookies with access token for Bi-directional Integration API domain.

Method 1:

1. Click on cookies.

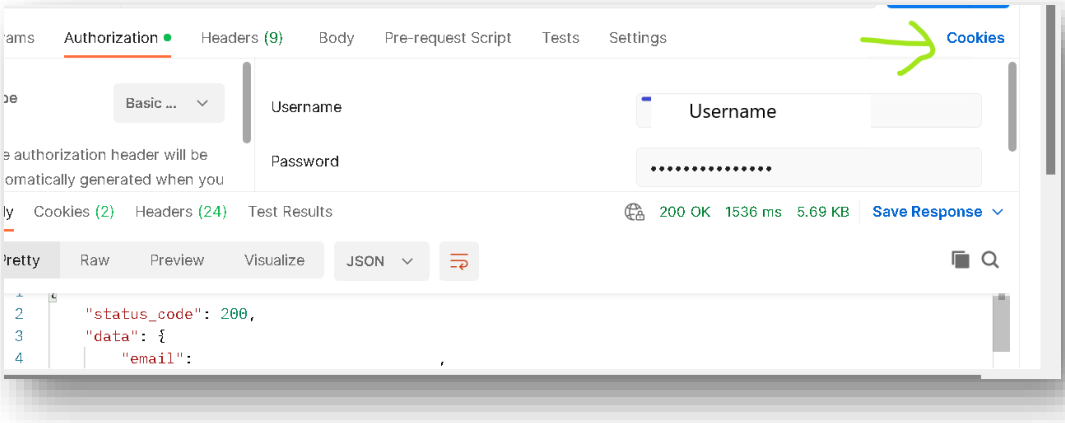

Figure 23 – How to consume Integration APIs

2. Add a domain of Bi-directional Integration with ITSM API.

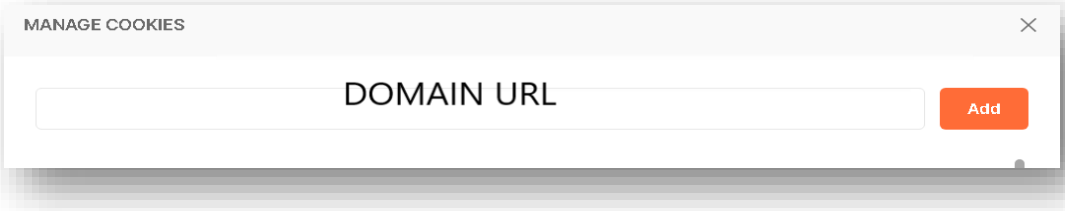

Figure 24 – How to consume Integration APIs (cont.)

3. Click on add cookies.

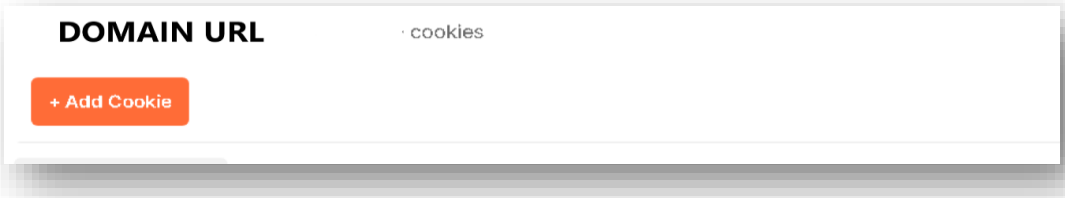

Figure 25 – How to consume Integration APIs (cont.)

4. Set access token key with token value.

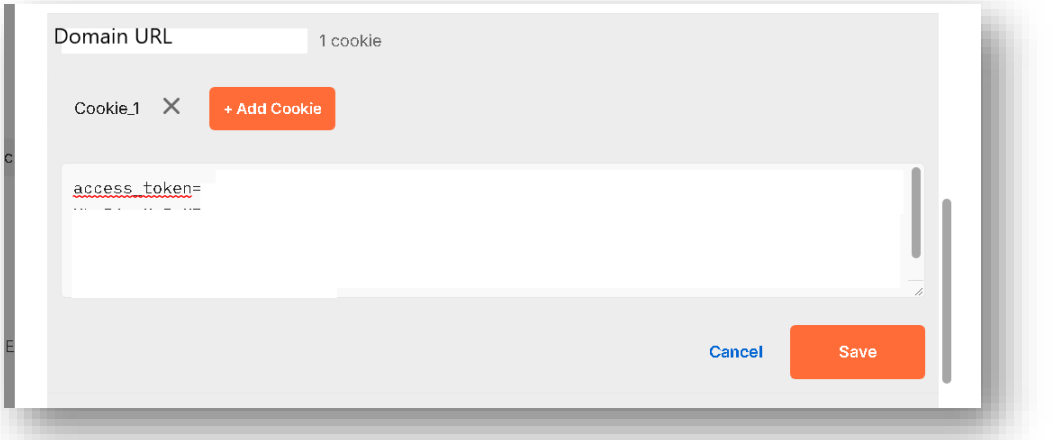

Figure 26 – How to consume Integration APIs (cont.)

5. Close the cookies window

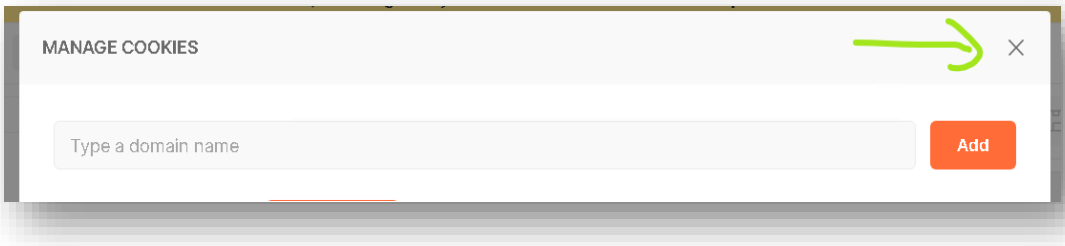

Figure 27 – How to consume Integration APIs (cont.)

Method 2:

1. Set cookie with access token in Integration API headers

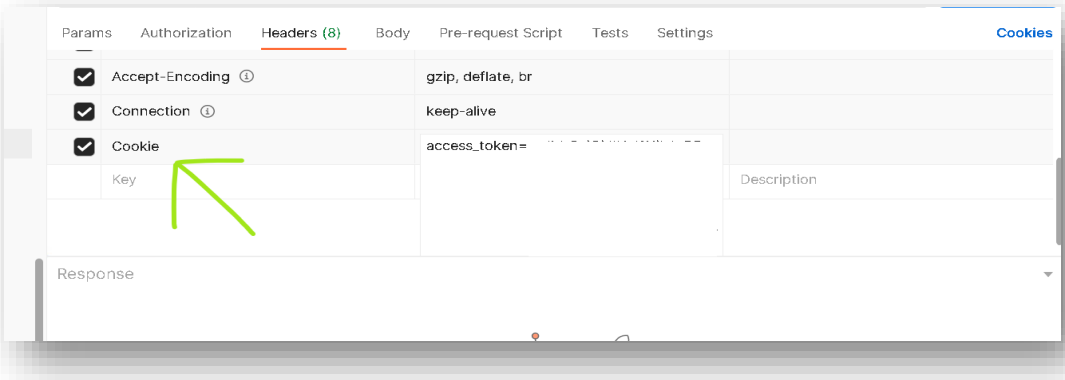

Figure 28 – How to consume Integration APIs (cont.)

4.2.6 Calling Integration API with valid Json

1. Enter the Integration API endpoint with "POST" as method type.

Figure 29 – How to consume Integration APIs (cont.)

2. Enter valid Json for API body/payload.

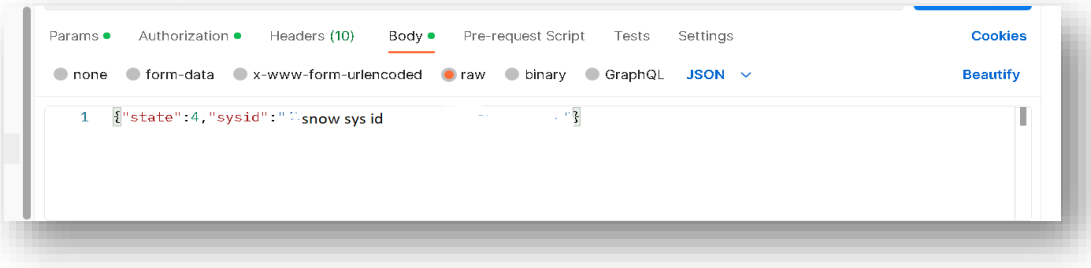

Figure 30 – How to consume Integration APIs (cont.)

3. Click on **Send** to get API response.
4. API response will contain a success or failure message based on Json provided:

Success case:

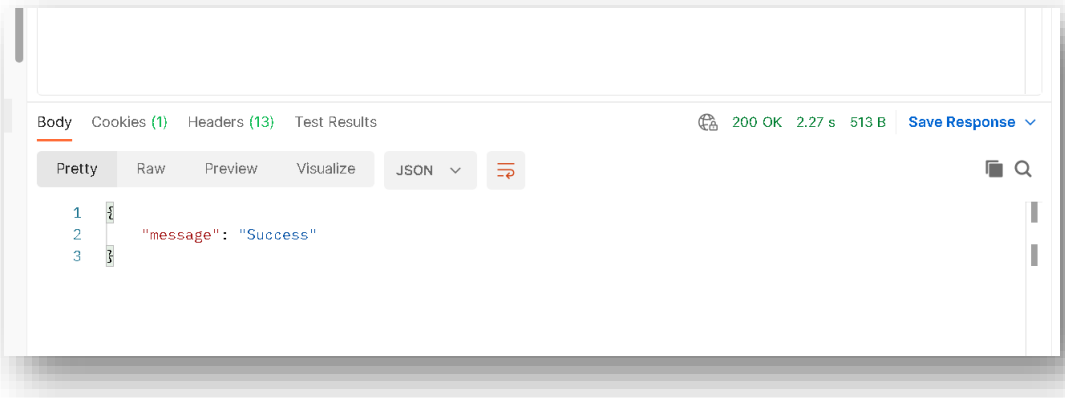

Figure 31 – How to consume Integration APIs (cont.)

Failure cases:

1. When user creds are invalid by using basic authentication.

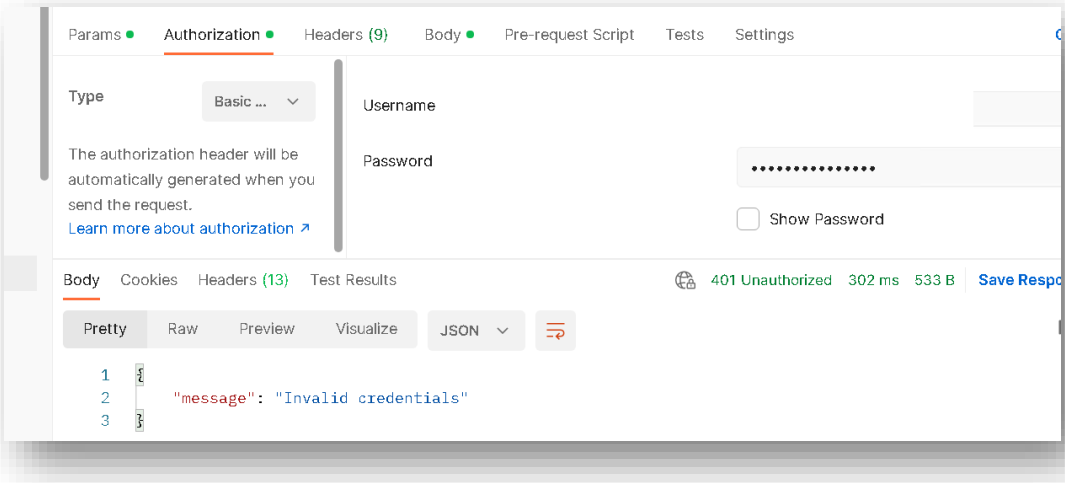

Figure 32 – How to consume Integration APIs (cont.)

2. When invalid token value is provided using token-based authentication.

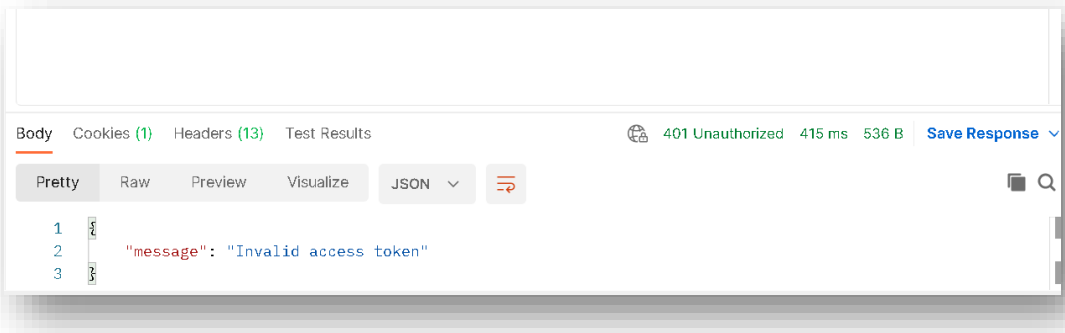

Figure 33 – How to consume Integration APIs (cont.)

3. When authentication not done with the integration API.

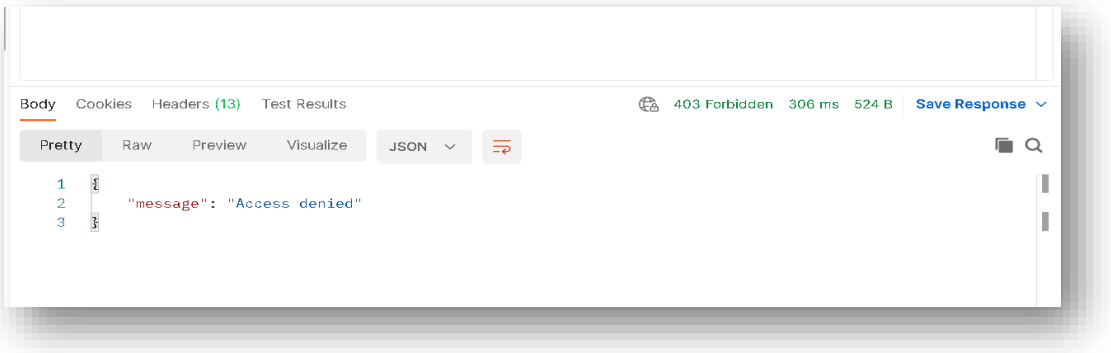

Figure 34 – How to consume Integration APIs (cont.)

4. When invalid customer id provided in URL parameters

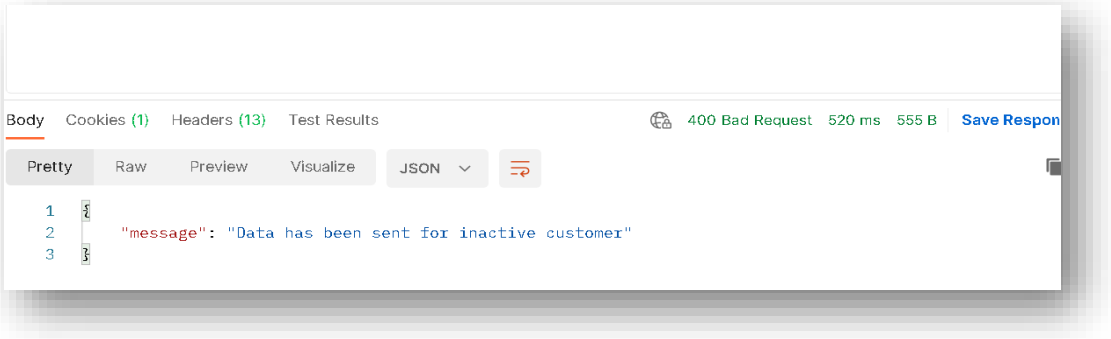

Figure 35 – How to consume Integration APIs (cont.)

5. When user not associated to the customer provided in URL parameters

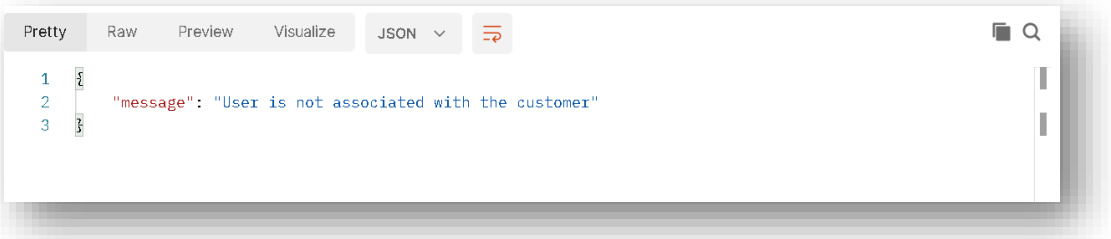

Figure 36 – How to consume Integration APIs (cont.)

4.3 IEM Bi-directional Integration with ITSM API Details

This section provides the details of all the existing APIs for IEM Bi-directional Integration with ITSM. The APIs in the current version of IEM Bi-directional Integration with ITSM have POST methods.

Table 14 – IEM Bi-directional Integration with ITSM POST APIs

POST APIS – To publish data into IEM		
S. No.	**API**	**API Description**
1	Generate Token	This API is used to generate the security token with the valid user credentials.
2	Fetch Ticket Update	This API is to update the actionable state in IEM whenever ticket is getting resolved at ITSM tool

4.3.1 Generate Token

Table 15 – GENERATE TOKEN

Element	Description
API	Generate Token
Description	API returns the access token based on username and password to authenticate IEM Integration APIs.
Method	POST
URL	https://<Hosted_API>/api/v1/user/token
Body	NA
Header	Authorization headers using basic authentication with username and password

Response (Success Case: 200 OK status code)	{
	"status_code": 200,
	"data": {
	"email": "<user_email_id>",
	"name": "<user_name>",
	"is_superadmin": false,
	"access_token": "<access_token_value>",
	"refresh_token": "<refresh_token_value>",
	"associated_with_customers": [
	{}
	]
	},
	"msg": ""
	}

Response Parameter	Access_token

Response (Failed Case: 401 Unauthorized status code)	{
	"detail": "Invalid username/password."
	}

4.3.2 API: Fetch Ticket Update

Table 16 – Fetch Ticket Update

Element	Description
API	Fetch Ticket Update
Description	This API is to update the actionable state in IEM whenever ticket is getting resolved at ITSM tool.
Method	POST
URL	https://<Hosted_API>/gbp/push/<customerid>?integrationid=<integrationid>
Header	Cookies with access_token value
Body	{"state":6, "sysid":"<snow sys id>"}

Response	{
	"message": "Success"
	}

(Success Case: 200 OK status code)	

Response (Failed Case 1 : 403 Unauthorized status code)	{
	"message": "Access denied"
	}

Response (Failed Case 2: 403 Unauthorized status code)	{
	"message": "Invalid credentials"
	}

Response (Failed Case 3 : 401 Unauthorized status code)	{
	"message": "Invalid access token"
	}

Response (Failed Case 4 : 401 Unauthorized status code)	{
	"message": "Access token has expired"
	}

Response (Failed Case 5 : 401 Unauthorized status code)	{
	"message": "Data has been sent for inactive customer"
	}

Response (Failed Case 6: 401 Unauthorized status code)	{
	"message": "User is not associated with the customer"
	}

5 API Accessible Matrix

Table 17 – API Accessible Matrix

Type	Action	API User	Organization Users
Provisioning Request	Get	User should be part of organization, but can't login into UI of IEM User can only perform the set of API actions.	User should be part of organization and user can login into UI of IEM but cannot access these APIs.
	Post	User should be part of organization, but can't login into UI of IEM User can only perform the set of API actions.	User should be part of organization and user can login into UI of IEM but cannot access these APIs.

HCLSoftware

hcltechsw.com