

HCLSoftware

HCL IntelliOps Event Management

IEM Publish Service API Guide
Version 1.1



The data contained in this document shall not be duplicated, used, or disclosed in whole or in part for any purpose. If a contract is awarded to chosen parties because of or in connection with the submission of this data, the client or prospective client shall have the right to duplicate, use, or disclose this data to the extent provided in the contract. This restriction does not limit the client's or prospective client's right to use the information contained in the data if it is obtained from another source without restriction. The data subject to this restriction is contained in all marked sheets.

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

Copyright © 2025 HCL Technologies Limited.

Table of Contents

1	Preface	8
1.1	Intended Audience.....	8
1.2	Conventions	8
2	IEM API Overview.....	9
2.1	IEM Publish Service Overview.....	9
2.2	IEM Bi-directional Integration Overview	9
3	About IEM Publish Service APIs.....	10
3.1	IEM Publish Service Rest API Request.....	10
3.1.1	Authentication	10
3.1.2	Request Header	10
3.2	IEM Publish Service Rest API Responses.....	11
3.2.1	HTTP Response Codes.....	11
3.3	How to call IEM Publish Service APIs.....	12
3.3.1	Calling Token API to get a token.....	12
3.3.2	Calling Publish API endpoint with access token.....	13
3.4	IEM Publish Service API Details.....	17
3.4.1	Generate Token	18
3.4.2	API: Publish Events Data	19
3.4.3	API: Publish Metrics Data	21
3.4.4	API: Publish Entity Topology Data.....	22
3.4.5	API: Publish Service Topology Data	24
3.4.6	API: Publish Healthcheck Data	25
4	About IEM Bi-directional Integration with ITSM APIs.....	27
4.1	IEM Bi-directional Integration with ITSM Rest API Request.....	27
4.1.1	Authentication	27
4.1.2	Request Header	27
4.2	IEM Bi-directional Integration with ITSM Rest API Responses.....	27
4.2.1	HTTP Response Codes.....	28
4.3	How to call IEM Bi-directional Integration with ITSM APIs.....	28
4.3.1	API Authentication.....	28
4.3.2	Calling Integration API with valid Json.....	32
4.4	IEM Bi-directional Integration with ITSM API Details.....	35
4.4.1	Generate Token	35
4.4.2	API: Fetch Ticket Update.....	36
5	API Accessible Matrix	38

Table of Figures

Figure 1 – How to consume token API	12
Figure 2- How to Consume Token API (cont.)	12
Figure 3- How to consume Token API (cont.)	12
Figure 4 - How to consume Token API (cont.)	13
Figure 5 - How to consume Publish APIs	13
Figure 6 - How to consume Publish APIs (cont.)	13
Figure 7 - How to consume Publish APIs (cont.)	14
Figure 8 - How to consume Publish APIs (cont.)	14
Figure 9 - How to consume Publish APIs (cont.)	14
Figure 10 - How to consume Publish APIs (cont.).....	15
Figure 11 - How to consume Publish APIs (cont.).....	15
Figure 12 - How to consume Publish APIs (cont.).....	15
Figure 13 - How to consume Publish APIs (cont.).....	16
Figure 14 - How to consume Publish APIs (cont.).....	16
Figure 15 - How to consume Publish APIs (cont.).....	16
Figure 16 - How to Publish APIs (cont.).....	17
Figure 17 - How to consume Publish APIs (cont.).....	17
Figure 18 – How to consume token API.....	28
Figure 19 – How to consume token API (cont.).....	29
Figure 20 - How to consume token API (cont.)	29
Figure 21 - How to consume Token API (cont.)	29
Figure 22 - How to consume Token API (cont.)	30
Figure 23 - How to consume Integration APIs.....	30
Figure 24 - How to consume Integration APIs (cont.).....	30
Figure 25 - How to consume Integration APIs (cont.).....	31
Figure 26 - How to consume Integration APIs (cont.).....	31
Figure 27 - How to consume Integration APIs (cont.).....	31
Figure 28 - How to consume Integration APIs (cont.).....	32
Figure 29 - How to consume Integration APIs (cont.).....	32
Figure 30 - How to consume Integration APIs (cont.).....	32

Figure 31 - How to consume Integration APIs (cont.)..... 33

Figure 32 - How to consume Integration APIs (cont.)..... 33

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

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

Figure 35 - How to consume Integration APIs (cont.)..... 34

Figure 36 - How to consume Integration APIs (cont.)..... 34

List of Tables

Table 1 – Conventions.....	8
Table 2- Request Header Fields.....	10
Table 3 – HTTP Response Codes.....	11
Table 4 – IEM Publish Service GET API Details	17
Table 5 – IEM Publish Service POST APIs	17
Table 6 – Generate Token.....	18
Table 7 – Publish Events Data	19
Table 8 – Publish Metrics Data.....	21
Table 9 – Publish Entity Topology Data.....	22
Table 10 – Publish Service Topology Data	24
Table 11 – Publish Healthcheck Data.....	25
Table 12 - Request Header Fields.....	27
Table 13 – HTTP Response Codes	28
Table 14 – IEM Bi-directional Integration with ITSM POST APIs.....	35
Table 15 – Generate Token.....	35
Table 16 – Fetch Ticket Update	36
Table 17 - API Accessible Matrix.....	38

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	HCL_IEM_v1.0_API_Guide
January, 2025	HCL_IEM_v1.1_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
<i>Italic</i>	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 resolve in IEM using authentication when the ticket is getting resolved using the 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 the IMM console, which will help to have 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 the functionality of IEM, Publish Service provided by a specific API, an enterprise tool needs to place a request. This section details the construction of that request that is supported by IEM Publish Service. Requests are typically categorized in terms of the 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 an 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 Bad Request	The request body is malformed, incomplete, or otherwise invalid.
401 Unauthorized	An authorization 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 the 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

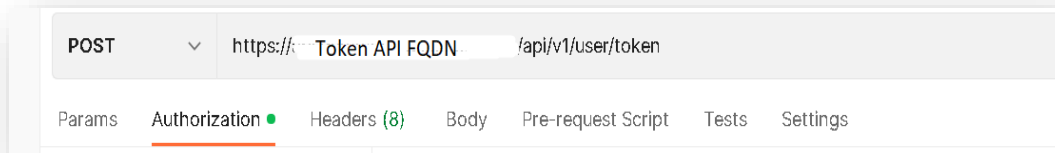


Figure 1 – How to consume token API

2. Define the basic authentication with username and password.

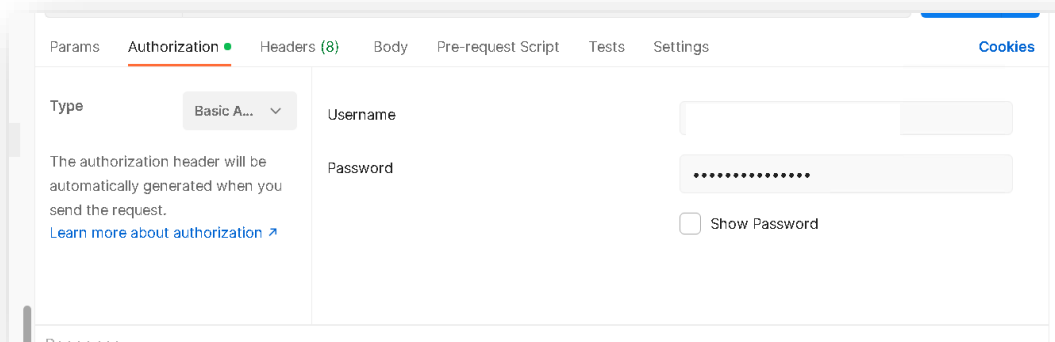


Figure 2- How to Consume Token API (cont.)

3. Click on the 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.
 - a. If user creds are valid, API will return access token in response.

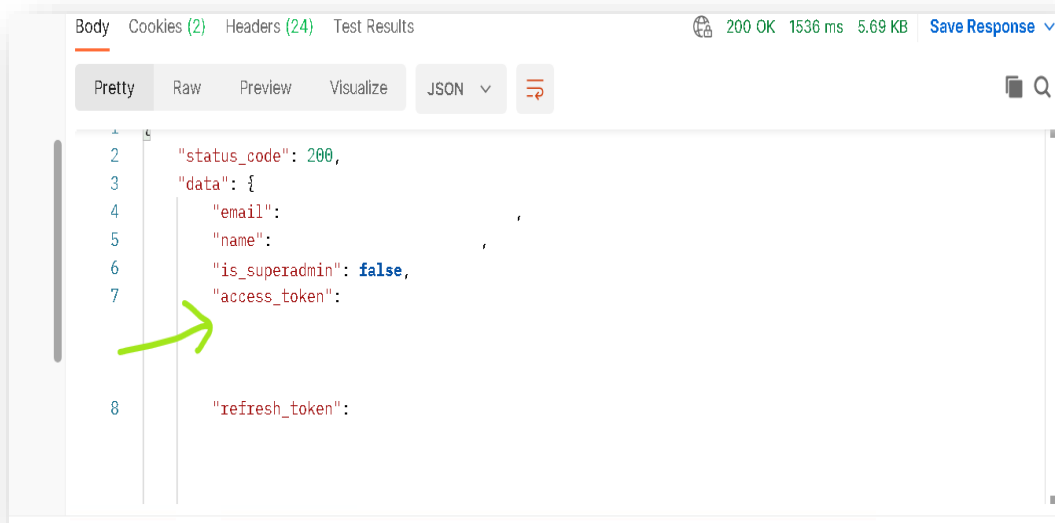


Figure 3- How to consume Token API (cont.)

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

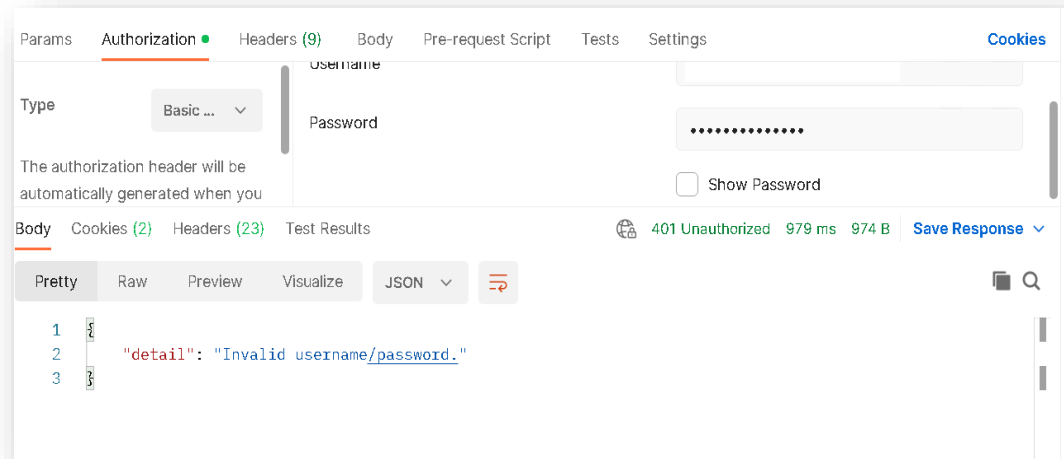


Figure 4 - How to consume Token API (cont.)

3.3.2 Calling Publish API endpoint with access token

1. Set cookies with access token for Publish service API domain
- **Method 1:**
- a. Click on cookies.

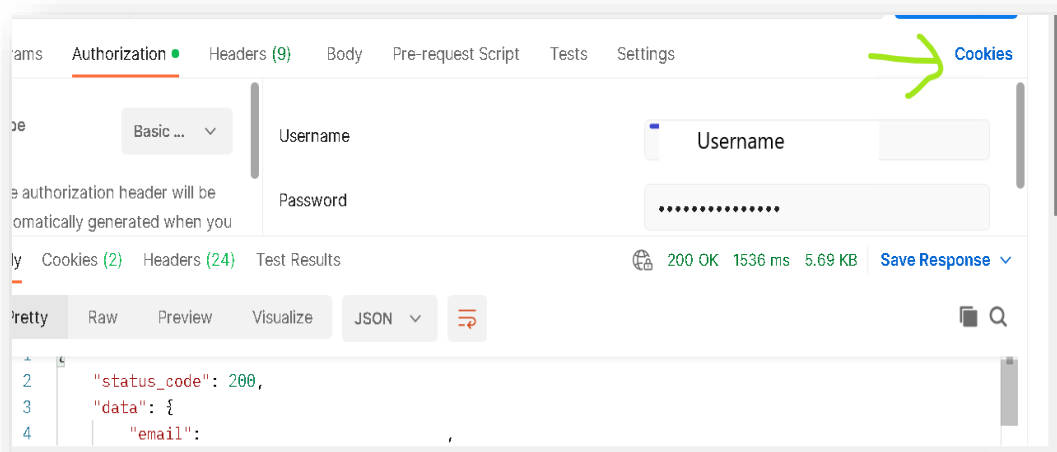


Figure 5 - How to consume Publish APIs

- b. Add a domain of Publish Service API

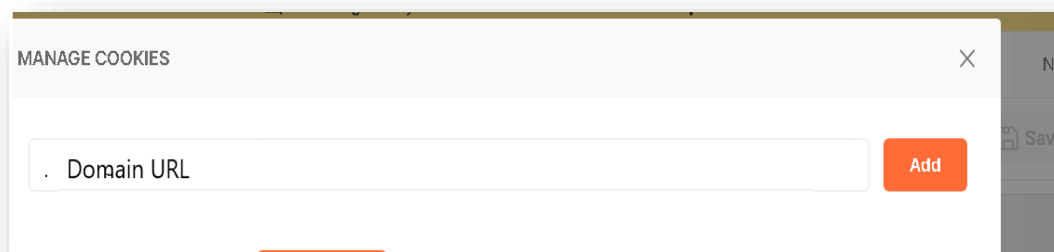


Figure 6 - How to consume Publish APIs (cont.)

- c. Click on add cookies.

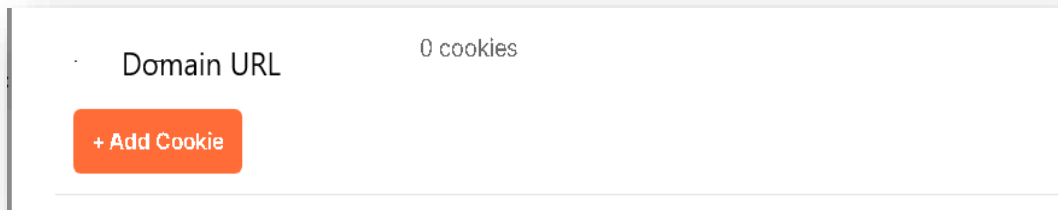


Figure 7 - How to consume Publish APIs (cont.)

- d. Set access token key with token value.

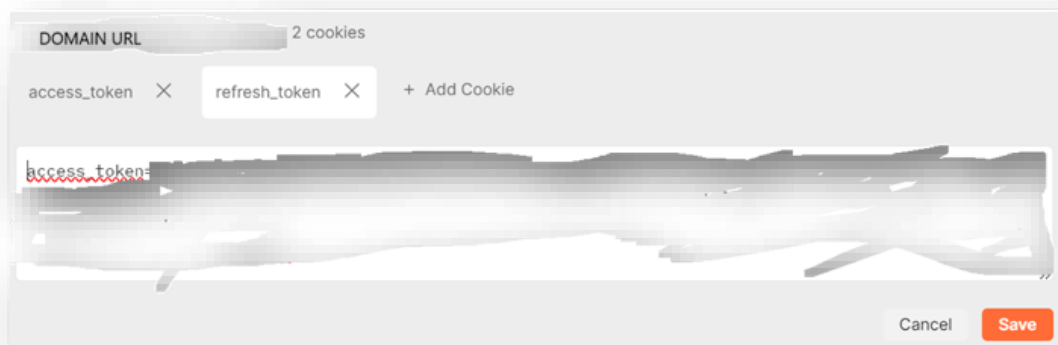


Figure 8 - How to consume Publish APIs (cont.)

- e. Close the cookies window.

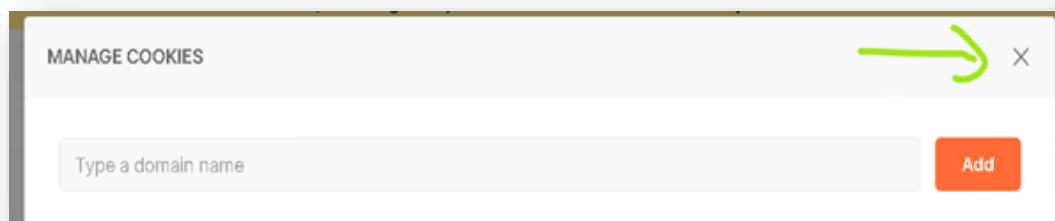


Figure 9 - How to consume Publish APIs (cont.)

- **Method 2:**
 1. Set cookie with access token in API headers

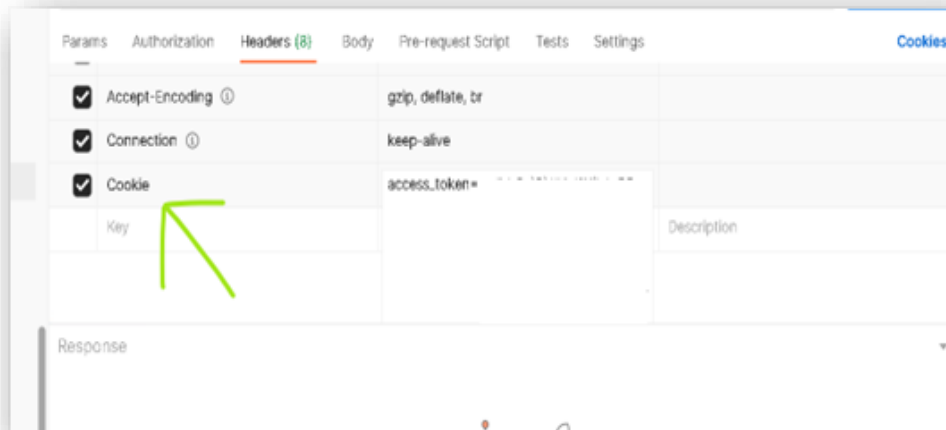


Figure 10 - How to consume Publish APIs (cont.)

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

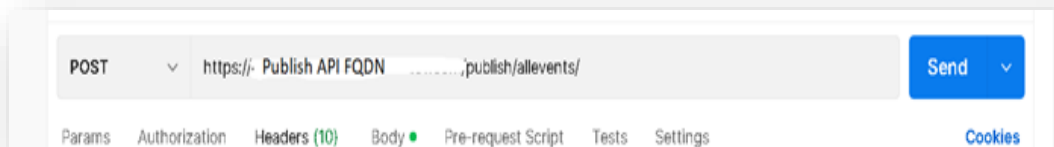


Figure 11 - How to consume Publish APIs (cont.)

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
 - a. Success case

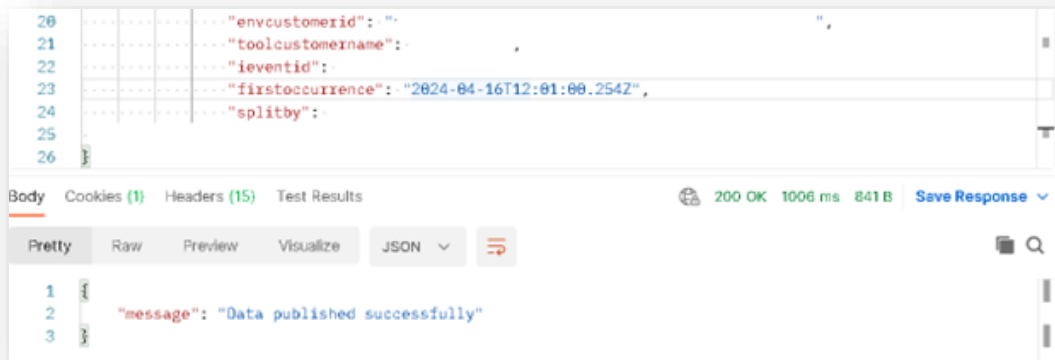


Figure 13 - How to consume Publish APIs (cont.)

b. Failure cases

i. When access token is not provided

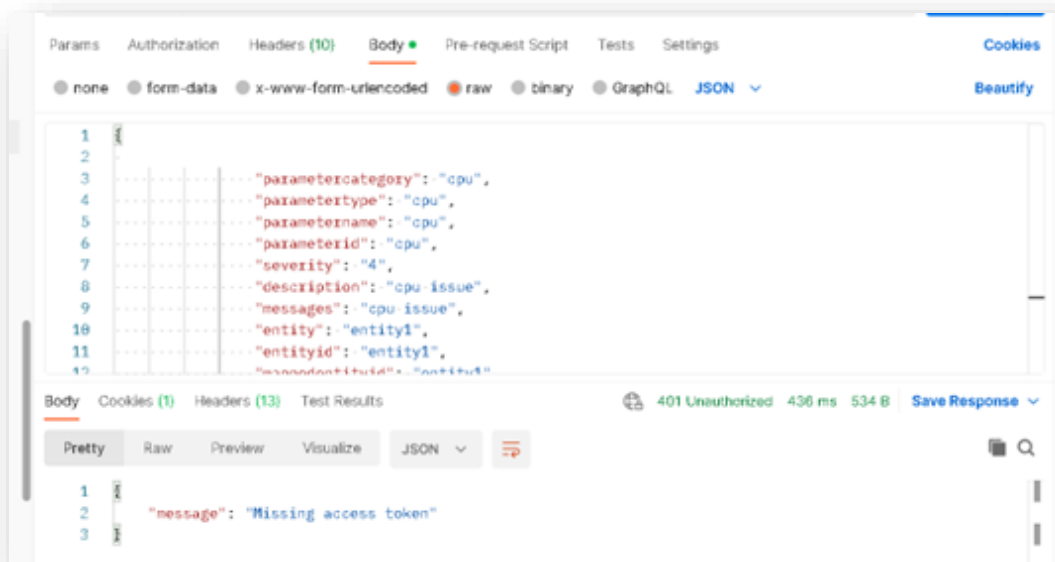


Figure 14 - How to consume Publish APIs (cont.)

ii. When invalid token value is provided



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

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

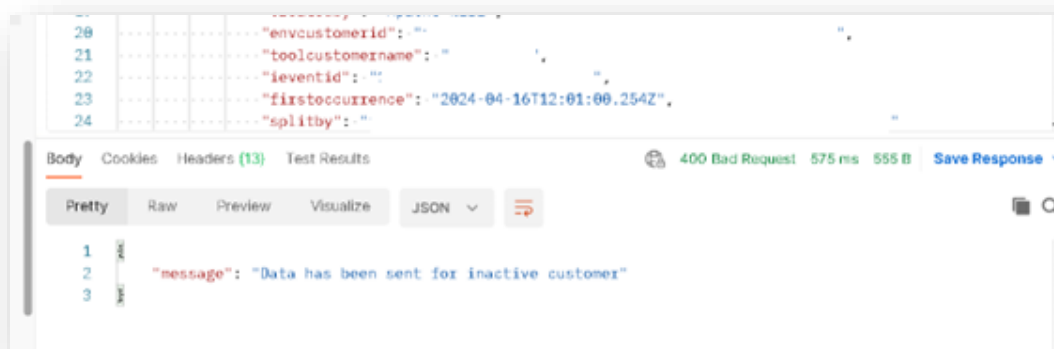


Figure 16 - How to Publish APIs (cont.)

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

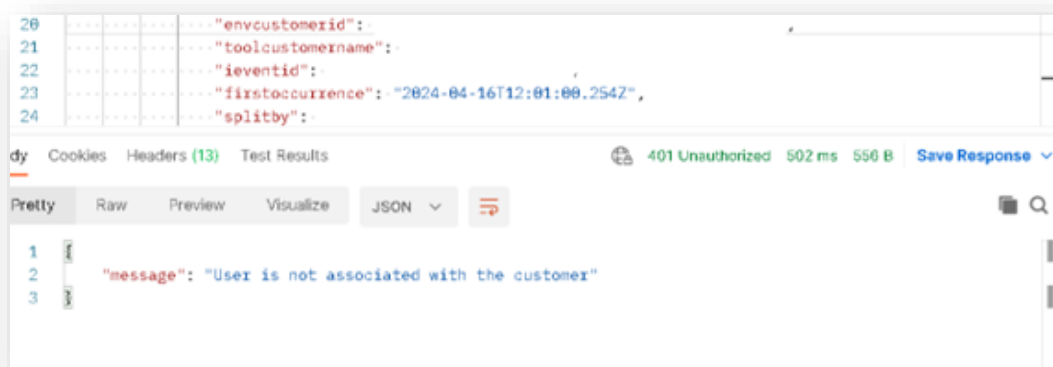


Figure 17 - How to consume Publish APIs (cont.)

3.4 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

POST APIs – To publish data into IEM		
S. No.	API	API Description
1	https: {{domain}}/publish/health check/	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

3.4.1 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	<a href="https://<Hosted API>/api/v1/user/token">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)	<pre>{ "status_code": 200, "data": { "email": "<user_email>", "name": "<User name>", "is_superuser": false, "access_token": "<token_value>", "refresh_token": "<refresh_token_value>",</pre>

		<pre> "associated_with_customers": [{}] }, "msg": "" } </pre>
Response Parameter	Access_token	
Response (Failed Case: 401 Unauthorized status code)		<pre> { "detail": "Invalid username/password." } </pre>

3.4.2 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	<a href="https://<Hosted API>/publish/allevvents/">https://<Hosted API>/publish/allevvents/	
Header	Cookies with access_token value	
Body		<pre> { "parametercategory": "cpu", "parametertype": "cpu", "parametername": "cpu", "parameterid": "cpu", "severity": "4", "description": "cpu issue", "messages": "cpu issue", "entity": "entity1", "entityid": "entity1", "mappedentityid": "entity1", "agentlocation": </pre>

		<pre> "<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" } </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		<pre> { "message": "Data has been sent for inactive customer" } </pre>

(Failed Case 4: 401 Unauthorized status code)	}
Response (Failed Case 5: 401 Unauthorized status code)	{ "message": "User is not associated with the customer" }

3.4.3 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	<a href="https://<Hosted API>/publish/metrics/">https://<Hosted API>/publish/metrics/
Header	Cookies with access_token value
Body	<pre>{ "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>" }</pre>

		<pre> "toolcustomername": "<customer_name>", "ieventid": "11-j2-5-c-asd- nd34b46644", "firstoccurrence": "2024-04- 16T12:01:00.254Z", "splitby": "<customer_id>_entity1_cpu" } </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>

3.4.4 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	<a href="https://<Hosted API>/publish/entity/">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		<pre>{</pre>

(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>

3.4.5 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	<a href="https://<Hosted API>/publish/service/">https://<Hosted API>/publish/service/
Header	Cookies with access_token value
Body	<pre>{ "servicename":"testservicej1", "service_entity_mapping":[{"entityid":"testentityj1"}, {"entityid":"testentityj2"}], "service_topology":[{"parentservicename":"testservicej2"}, {"parentservicename":"testservicej3"}], "envcustomer_id":"<customer_id>" }</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	<pre>{ "message": "Access token has expired" }</pre>

(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" }

3.4.6 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	<a href="https://<Hosted API>/publish/healthcheck/">https://<Hosted API>/publish/healthcheck/
Header	Cookies with access_token value
Body	<pre>{ "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": "{}" }</pre>

		<pre>}</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>

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 the 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 construction 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 clients and servers are secured with SSL. IEM Bi-directional Integration 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 take 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 a 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 the 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.

4.3 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.

4.3.1 API Authentication

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

1. Basic authentication using username and password.
 - Provide username and password for authorization using basic authentication to Integration API.

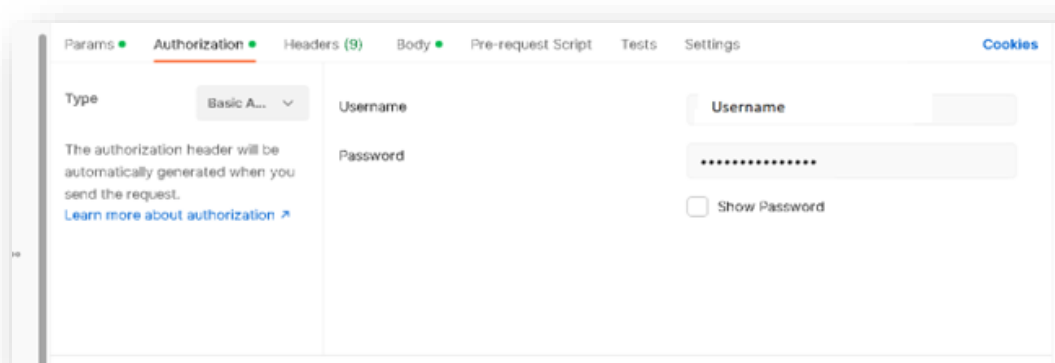


Figure 18 – How to consume token API

2. Token based authentication
 - **Calling Token API to get a token**

- a. Define methodtype = POST

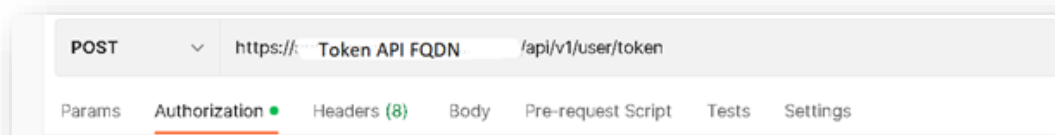


Figure 19 – How to consume token API (cont.)

- b. Define the basic authentication with username and password to token API

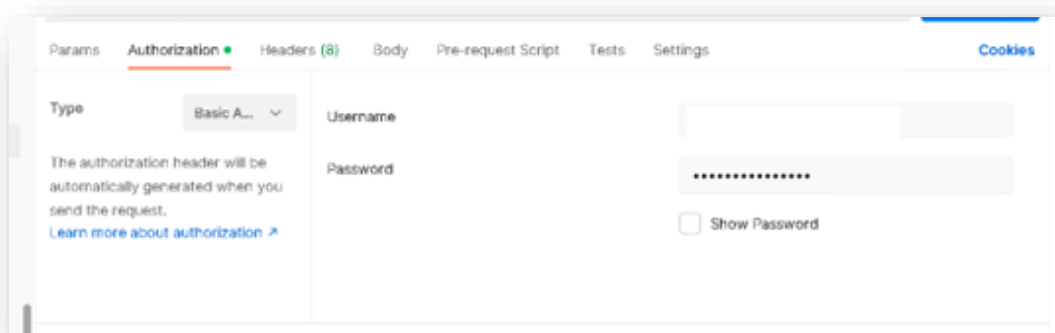


Figure 20 - How to consume token API (cont.)

3. Click on the 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.
- a. If user creds are valid, API will return access token in response.

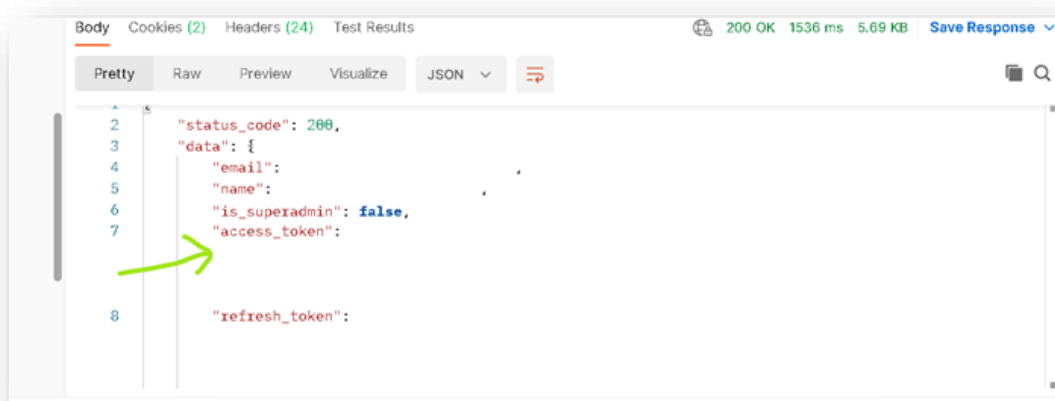


Figure 21 - How to Consume Token API (cont.)

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

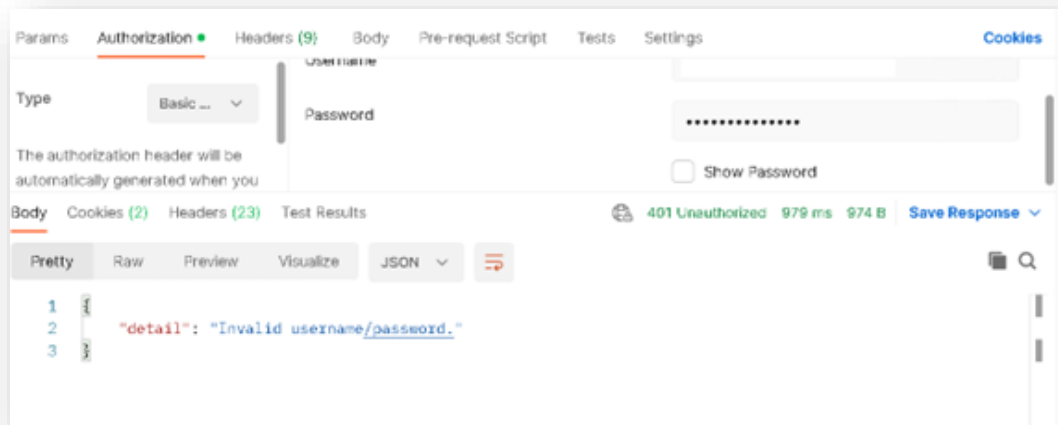


Figure 22 - How to consume Token API (cont.)

- **Calling Integration API endpoint with access token**
- 1. Set cookies with access token for Bi-directional Integration API domain
- **Method 1:**
 - a. Click on cookies.

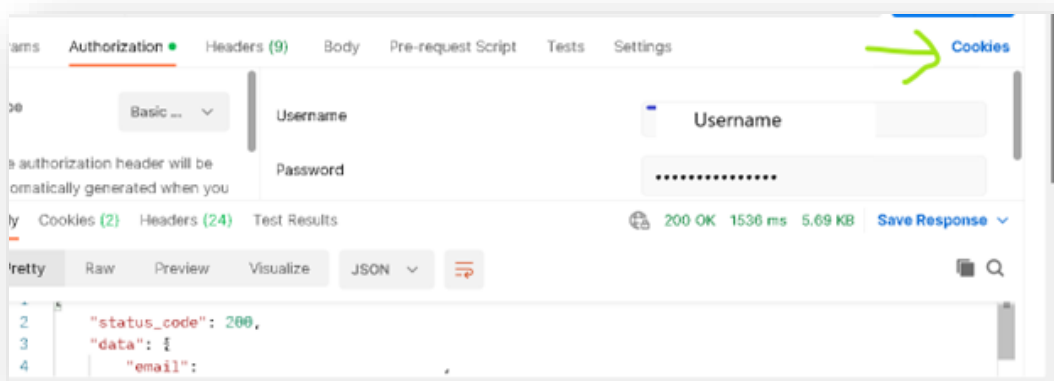


Figure 23 - How to consume Integration APIs

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

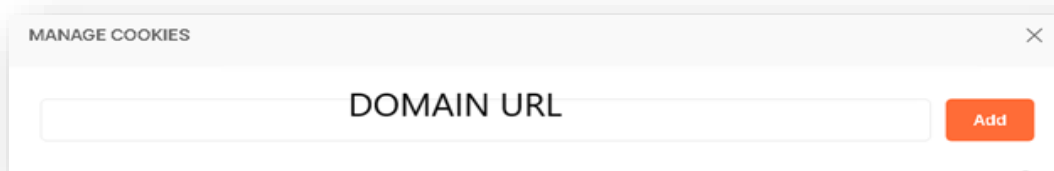


Figure 24 - How to consume Integration APIs (cont.)

- c. Click on add cookies.

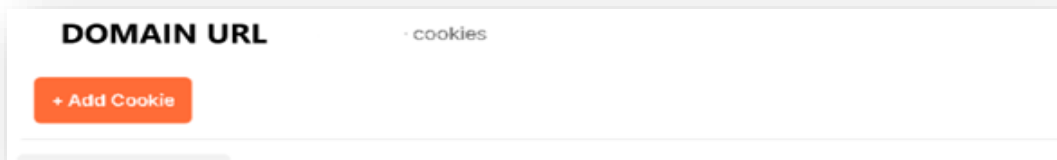


Figure 25 - How to consume Integration APIs (cont.)

- d. Set access token key with token value.

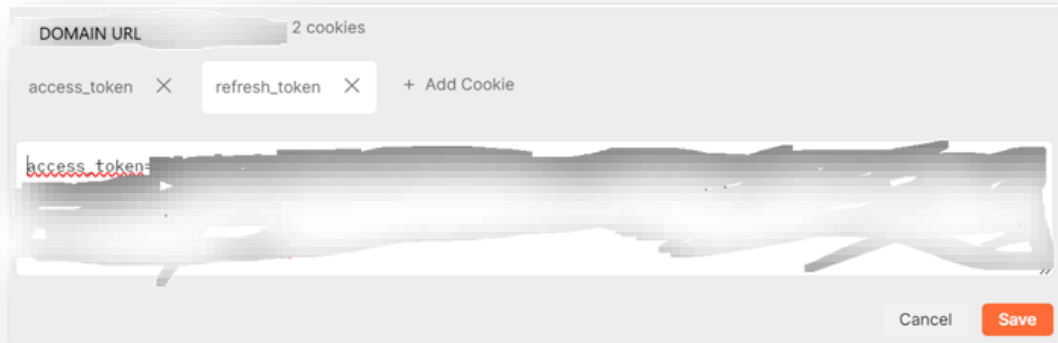


Figure 26 - How to consume Integration APIs (cont.)

- e. Close the cookies window.

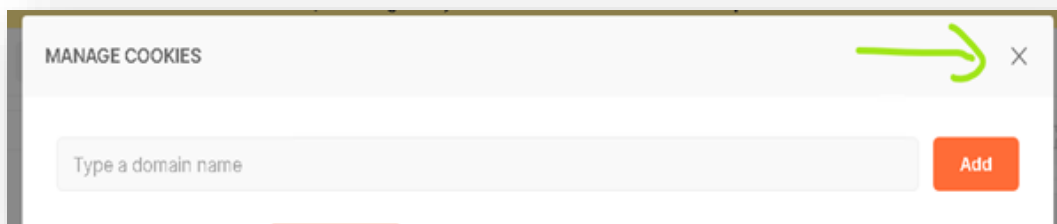


Figure 27 - How to consume Integration APIs (cont.)

- **Method 2:**
 - a. Set cookie with access token in Integration API headers.

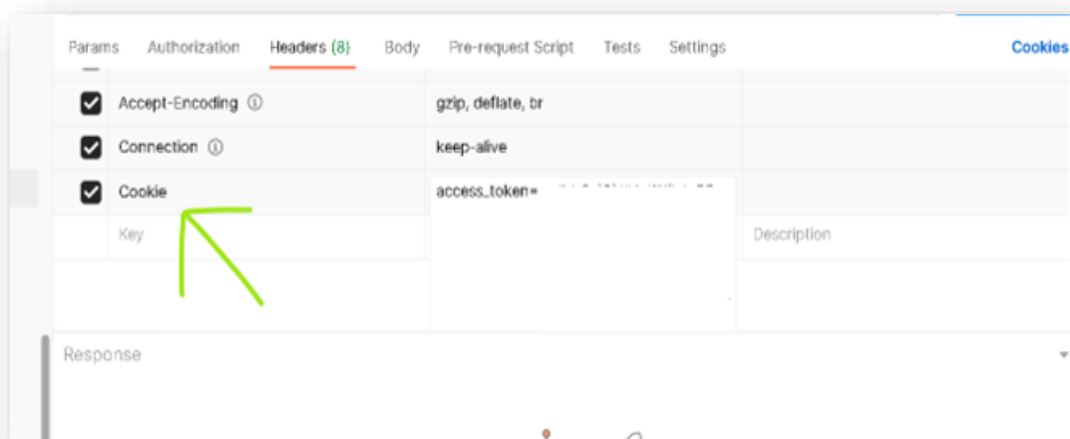


Figure 28 - How to consume Integration APIs (cont.)

4.3.2 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

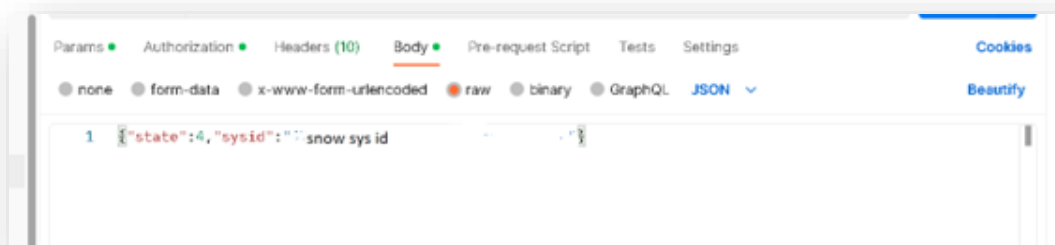


Figure 30 - How to consume Integration APIs (cont.)

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

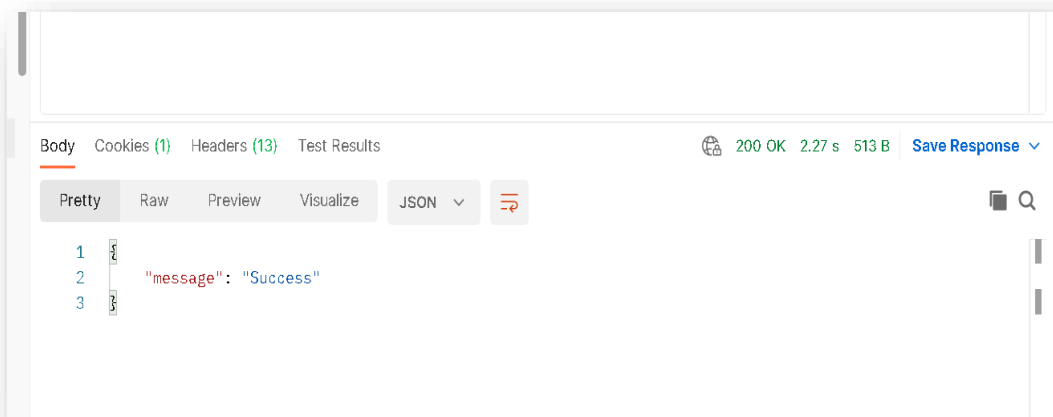


Figure 31 - How to consume Integration APIs (cont.)

b. Failure cases

i. When user creds are invalid by using basic authentication

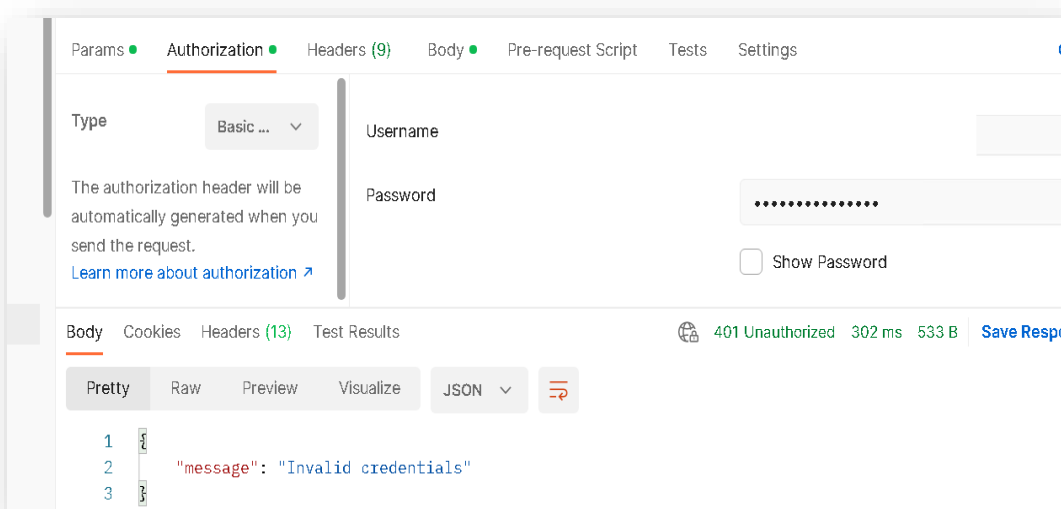


Figure 32 - How to consume Integration APIs (cont.)

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

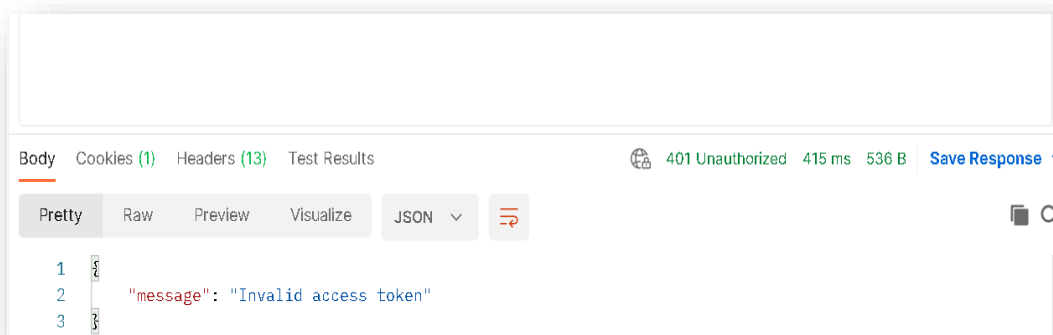


Figure 33 - How to consume Integration APIs (cont.)

iii. When authentication is not done with the integration API

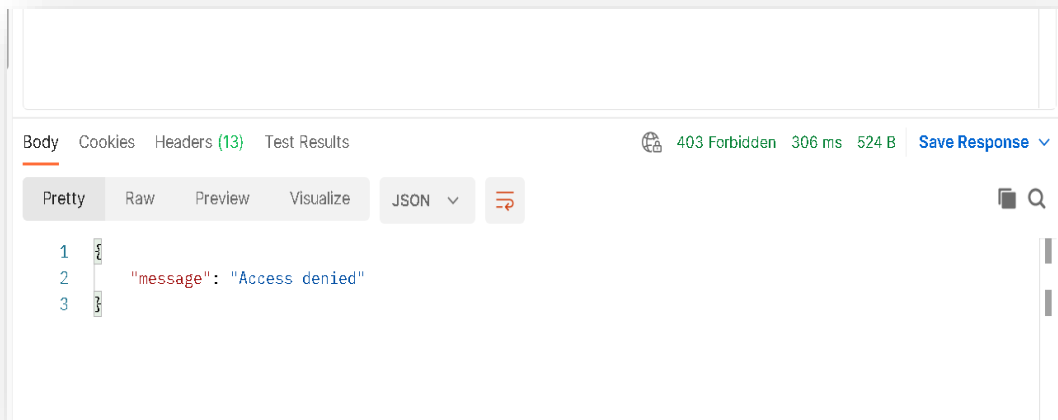


Figure 34 - How to consume Integration APIs (cont.)

iv. When invalid customer id provided in URL parameters

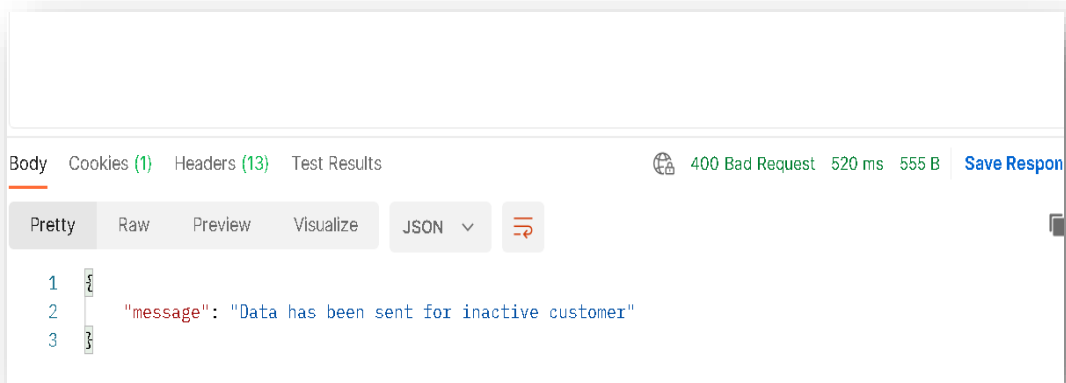


Figure 35 - How to consume Integration APIs (cont.)

v. When user not associated with the customer provided in URL parameters

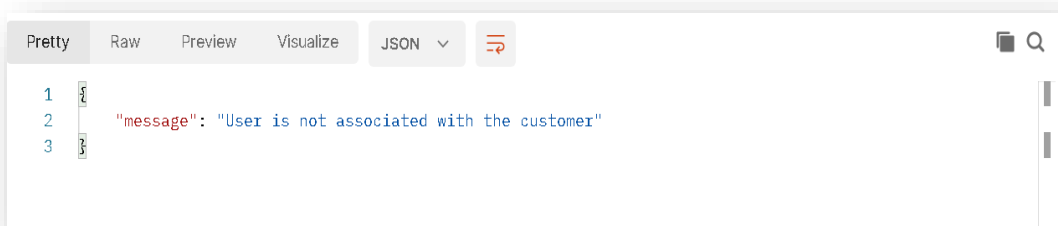


Figure 36 - How to consume Integration APIs (cont.)

4.4 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 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

4.4.1 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	<a href="https://<Hosted API>/api/v1/user/token">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)	<pre>{ "status_code": 200, "data": { "email": "<user_email_id>", "name": "<user_name>", "is_superuser": false, "access_token": "<access_token_value>", "refresh_token": "<refresh_token_value>", "associated_with_customers": [{}] }, "msg": "" }</pre>

Response Parameter	Access_token
Response (Failed Case: 401 Unauthorized status code)	<pre>{ "detail": "Invalid username/password." }</pre>

4.4.2 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	<a href="https://<Hosted API>/gbp/push/<customerid>?integrationid=<integrationid>">https://<Hosted API>/gbp/push/<customerid>?integrationid=<integrationid>
Header	Cookies with access_token value
Body	<pre>{"state":6, "sysid":"<snow sys id>"}</pre>
Response (Success Case: 200 OK status code)	<pre>{ "message": "Success" }</pre>
Response (Failed Case 1: 403 Unauthorized status code)	<pre>{ "message": "Access denied" }</pre>
Response (Failed Case 2: 403 Unauthorized status code)	<pre>{ "message": "Invalid credentials" }</pre>
Response (Failed Case 3: 401 Unauthorized status code)	<pre>{ "message": "Invalid access token" }</pre>
Response (Failed Case 4: 401 Unauthorized status code)	<pre>{ "message": "Access token has expired" }</pre>
Response (Failed Case 5: 401 Unauthorized status code)	<pre>{ "message": "Data has been sent for inactive customer" }</pre>

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

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