

thinkRF SXM API

User Guide

version 1.2.5

May 30, 2025

Document no. 75-0049

Copyright © 2021-present thinkRF Corporation, all rights reserved.

Important notice

The information in this guide is furnished for informational use only and is subject to change without notice. thinkRF Corporation assumes no responsibility or liability for any errors or inaccuracies that may appear in this document. No part of this publication may be reproduced, published, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, for any purpose, without the prior written permission of thinkRF Corporation.

Trademarks

thinkRF, the thinkRF logo and Product Names are trademarks of thinkRF Corporation.

All other brand or product names are trademarks or registered trademarks of their respective companies or owners.

thinkRF Corp

390 March Road
Kanata, ON K2K 0G7
(613) 271-5451

Table of Contents

| | |
|---|-----------|
| Preface | 4 |
| Audience | 4 |
| Conventions | 4 |
| Obtaining Latest Documentation and Software | 4 |
| Document Feedback | 4 |
| Obtaining Technical Assistance | 4 |
| Product Feedback | 5 |
| Overview | 6 |
| Get Started | 7 |
| Request Access | 7 |
| Install API Client | 7 |
| Download and Import API Collections | 7 |
| Generate Access Token..... | 7 |
| Try It | 8 |
| API References | 10 |
| Endpoint URL | 10 |
| Error Handling | 10 |
| Authentication..... | 11 |
| Authorization | 11 |
| Access Management..... | 12 |
| Get Access Token..... | 12 |
| Change Client Credential Password | 13 |
| User Management | 15 |
| Add User | 15 |
| Update User | 16 |
| Get User | 17 |
| List Users | 18 |
| Node Management | 20 |
| Accessing 3GPP Source Data | 24 |
| Infrastructure | 24 |
| Infrastructure Export Request | 26 |
| Dictionary | 27 |
| Appendix A – SXM Analysis Data Parameter Description | 29 |
| Appendix B – SXM 4G/5G Naming Consistency | 33 |
| Revision History | 34 |

Preface

The intended audience, document organization, and conventions used herein are described. Related documentation is identified, as are instructions for accessing other electronic product documentation.

Audience

This document is intended for technical users who have a basic level of understanding, familiarity and experience with spectrum monitoring and mobile network usage.

Conventions

The following conventions are used in this document.

| Convention | Description |
|------------------|---|
| Grayed-out Font | Indicates a command or a feature is not available in the current release. |
| Courier Font | Illustrates an example command or a concept. |
| Light Blue Font | A clickable hyperlink to a referenced source. |
| Normal Bold Font | A concept or idea important enough that the reader's attention is being explicitly focused. |
| Red Font | Additional information on the topic. |



Note: This symbol means **take note**. Notes contain helpful suggestions or references for additional information and material.



Caution: This symbol means **be careful**. In this situation, you might do something that could result in loss of settings, data or unintended data behaviour.

Obtaining Latest Documentation and Software

Please visit <https://support.thinkrf.com/support/solutions> to obtain the latest product documentation, software and firmware releases where applicable.

Document Feedback

Please send any comments regarding thinkRF documentation to SXMfeedback@thinkrf.com. We appreciate your feedback.

Obtaining Technical Assistance

For all customers who hold a valid end-user license, thinkRF provides technical assistance 9 AM to 5 PM Eastern Time, Monday to Friday. Contact us at <https://support.thinkrf.com/> or by calling **+1.613.271.5451**.

Before contacting support, please have the following information available:

- SXM API version
- Any SXM software used and its version

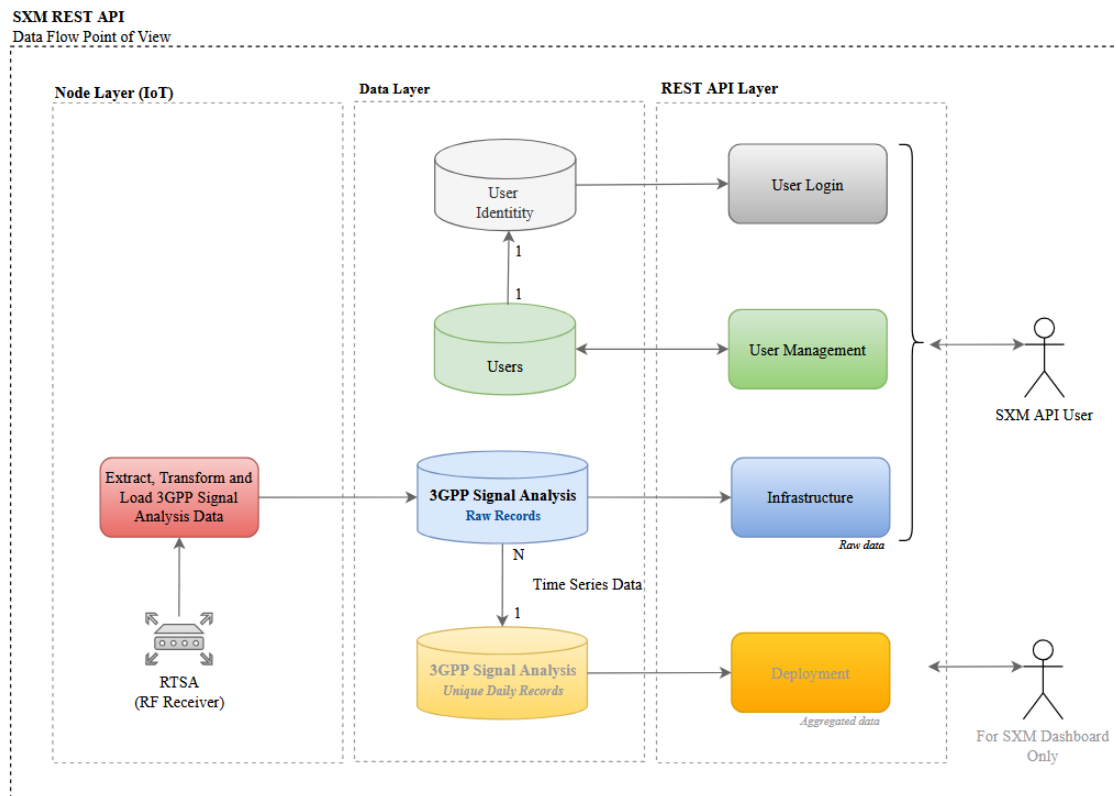
Product Feedback

thinkRF's SXM is constantly evolving to better serve your needs, with new features and enhancements deployed to your network and this dashboard regularly. Submit your input to SXMfeedback@thinkrf.com or use the Dashboard's Feedback box in the Help page.

Overview

thinkRF Spectrum eXperience Management (SXM) solution provides REST APIs for real-time 3GPP network infrastructure data collected from the SXM edge devices in the field and stored in the cloud. It also allows authorized API users to easily perform user management

The API has the following entity relationship diagram, with the three various layers included for reference.



The document has the following main sections:

- **Get Started:** Prepare your local environment to start consuming the APIs.
- **API References:** Technical details about access, authentication, and error handling.
- **API Specification:** Description of each API and its group, along with examples where applicable.



Note: the parameter list and examples might not be the most up to date with each API update in this document. Use the API call to get the latest output results.

- **Appendices:** Complementary information, relevant to the API.

Get Started

To get started with thinkRF SXM DaaS APIs, use the [Postman API Client](#) or similar applications to consume the APIs. In a few minutes, you will learn to set up the API client, request Access Token and call the APIs.

Request Access

Together with this API User Guide, you should receive the client and secret for your token generation from thinkRF's [Support Team](#).

Install API Client

You can skip the steps below in case you have already the [Postman API Client](#) installed or would like to use another one:

1. Access the Postman website [here](#)
2. Follow the Sign-up process
3. Download and install the client on your machine



Note: If prefer, you can use the online version.

Download and Import API Collections

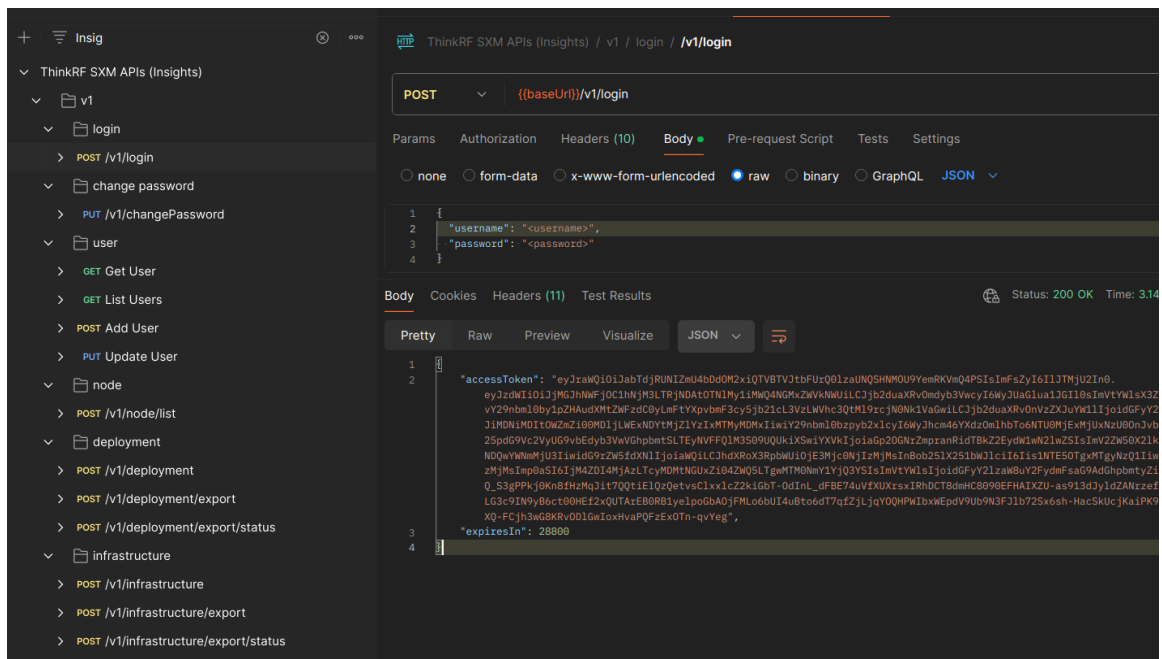
We provide a collection containing all APIs available for importing on Postman API Client, download it [here](#).

Generate Access Token

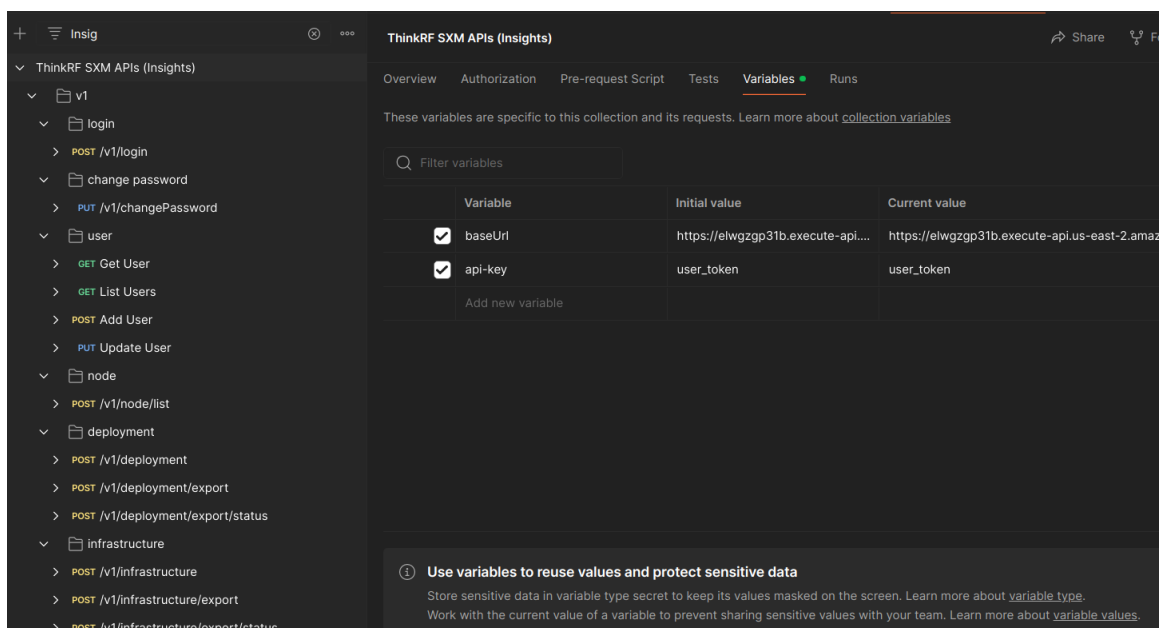
In the Postman API Client:

1. Go to **Collections** (left-side menu)
2. Open the collection **thinkRF SXM APIs**
3. Open the item **POST v1/login** from folder **v1 > login**
4. In the Body tab fill in your username and password and click on the **Send** button
5. The response body will contain the **accessToken** and **expiresIn** fields
6. Assign the **accessToken** value to the API-KEY postman variable to reflect in all API requests
 - Open the collect **thinkRF SXM APIs**
 - Go to the **Variables** Tab
 - Replace the INITIAL VALUE and CURRENT VALUE of variable API-KEY with the **accessToken** generated in step 5 and click on **Save** button.

Access Token Generation example:



Replace accessToken Variable example:

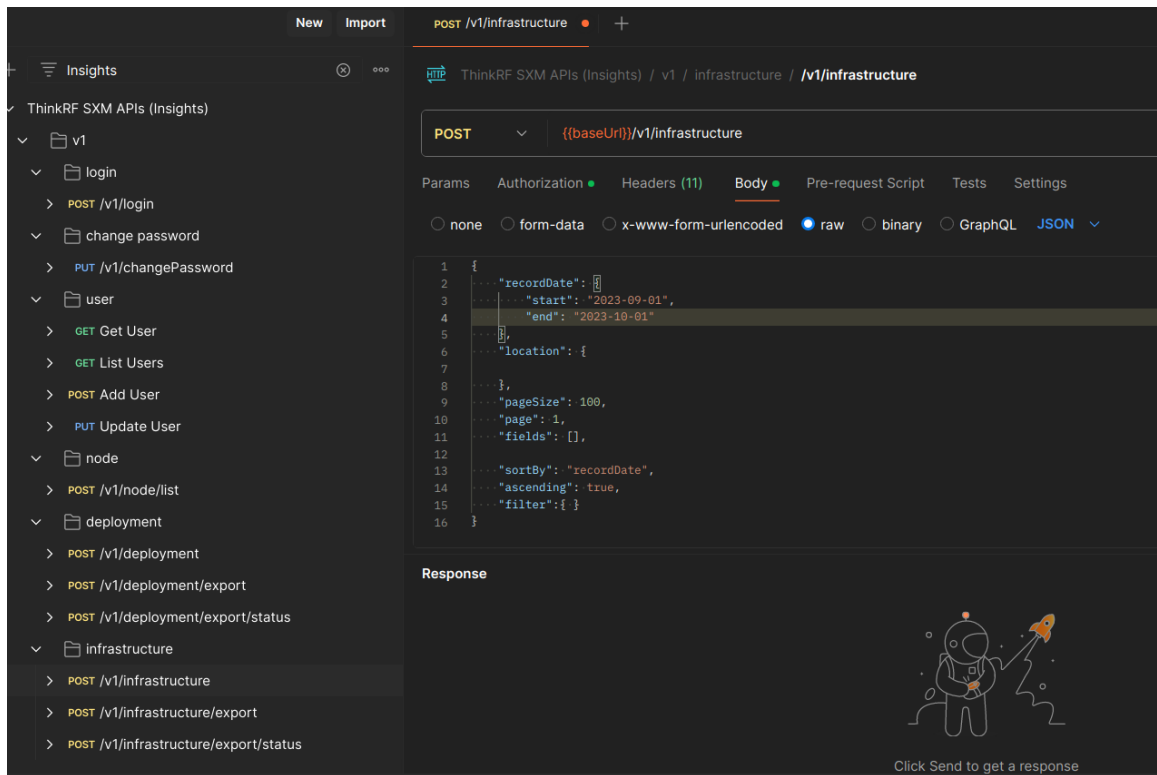


Try It

In the Postman API Client:

- Go to **Collections** option (left-side menu)
- Open the collection **thinkRF SXM APIs**
- Open the item **GET v1/nodes** from folder **v1 > nodes**
- Click the **Send** button
- The response body will contain all the nodes under your access scope

Node List Result example:



The screenshot displays a REST client interface with a sidebar on the left showing a tree view of API endpoints under 'ThinkRF SXM APIs (Insights)'. The main panel shows a POST request to `/v1/infrastructure` with a JSON body. The body is a complex object with nested arrays and objects, including fields like `recordDate`, `start`, `end`, `location`, `pageSize`, `page`, `fields`, `sortBy`, `ascending`, and `filter`.

Request Details:

- Method: POST
- URL: `{{baseUrl}}/v1/infrastructure`
- Body Type: raw

Request Body (JSON):

```
1 {
2   ... "recordDate": [
3     ... "start": "2023-09-01",
4     ... "end": "2023-10-01"
5   ],
6   ... "location": {
7     ...
8   },
9   ... "pageSize": 100,
10  ... "page": 1,
11  ... "fields": [],
12
13  ... "sortBy": "recordDate",
14  ... "ascending": true,
15  ... "filter": {}
16 }
```

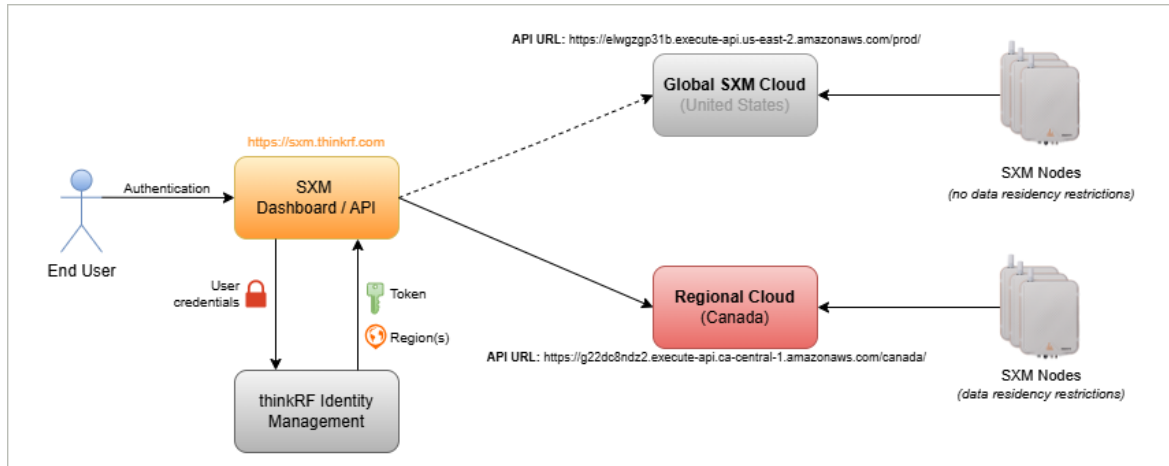
Response:

Click Send to get a response

API References

Endpoint URL

SXM Solution supports data residency and sovereignty compliance for various cloud regions, with the infrastructure flow as shown in the following picture.



For companies using the default Global Cloud region, the SXM API endpoint URL is:

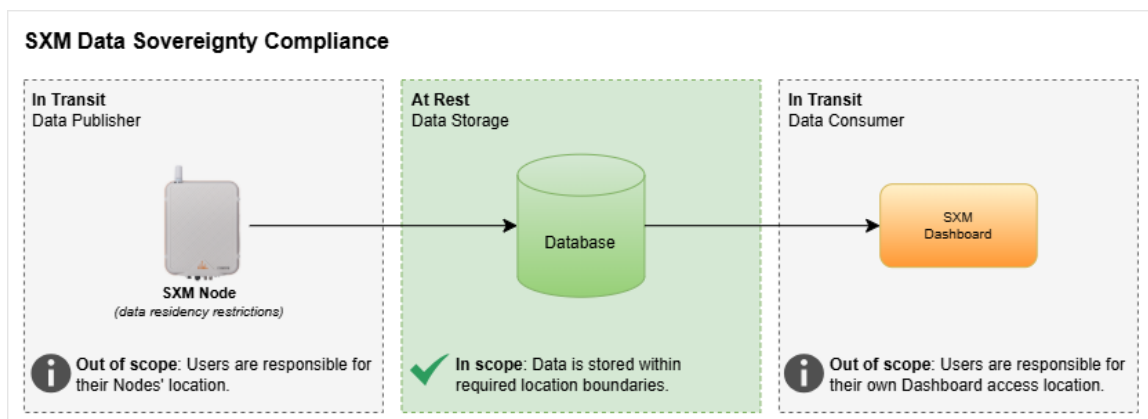
`https://elwzg31b.execute-api.us-east-2.amazonaws.com/prod/`

For companies using the Canadian Cloud region, the SXM API endpoint URL is:

`https://g22dc8ndz2.execute-api.ca-central-1.amazonaws.com/canada/`



Note: SXM ensures data sovereignty compliance for data at-rest only. Responsibility for data in-transit compliance —particularly as it relates to the physical location of nodes and Dashboard users— rests with users, as summarized in the picture below.



Error Handling

The API services can handle errors under two layers:

- **Infrastructure layer:** error codes generated under service call as:
- HTTP code 401: access not authorized
- HTTP code 404: resource or related object not found
- HTTP code 400: bad request (missing input parameters)
- HTTP code 500: Internal system error. Try again. If the issue persists, contact thinkrf.
- **Application layer:** custom error codes generated by the API processing components and reported in the response body with two attributes:
 - **errorCode:** custom error code related to the request processing
 - **description:** short description for the error code above

```
{
  "errorCode": "string",
  "description": "string"
}
```

Authentication

The APIs are protected and can be consumed by authenticated and authorized users via access token api-key type. See below for the request example:

```
curl -X 'GET' \
  'https://api.thinkrf.com/v1/nodes' \
  -H 'accept: application/json' \
  -H 'Authorization: <access-token>
```

The **accessToken** needs to be generated via **/v1/login** API resource, providing your client credentials as below:

```
# HTTP Request
curl -X 'POST' \
  'https://api.thinkrf.com/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "<your client ID>",
    "password": "<your client secret>"
  }'

# HTTP Response
HTTP response code: 200
{
  "accessToken": "eyJraWQiOiJabTdjRUNIZmU4b.....bfjBE1mg",
  "expiresIn": 3600
}
```

Authorization

The thinkRF SXM API framework has a second security layer that validates the user authorization for the related API resource requested. In case of an unauthorized case, the API will send a HTTP response code 401.

Access Management

Get Access Token

The GetAccessToken provides an access token for your requests. Its functionality is crucial for securing access to SXM APIs. It verifies whether a user is permitted to make API requests, enforcing security policies and proper role-based access control. This function preserves data integrity and adheres to regulations governing data access and security.

Request:

| Request Configuration | |
|------------------------|--------------------------|
| Resource | /v1/login |
| Method | POST |
| Header | Accept: application/json |
| Request Body | |
| username string | Your API client ID |
| password string | Your API client secret |

Response:

| Response Body Access Token Data | |
|------------------------------------|---|
| accessToken string | Token to be used on API calls for authentication |
| expiresIn number | Token expiration in seconds |
| dataRegionEndpoint array | Authorized data region endpoint(s) |
| Status Code | |
| 200 | Success |
| 401 | <p>In case of not successful token generation, the response body will contain some of the following Json output.</p> <p>HTTP status code 400:</p> <pre>{ "errorCode": "E0001", "description": "Invalid Request" } { "errorCode": "E0013", "description": "Change password first" }</pre> <p>HTTP status code 401:</p> <pre>{ "errorCode": "E0002", "description": "Not Authorized" } { "errorCode": "E0003", "description": "Too many requests" } { "errorCode": "E0004", "description": "User not found" } { "errorCode": "E0005", "description": "User not confirmed" } { "errorCode": "E0006", "description": "Login Error. Please contact administrator." } { "errorCode": "E0019", "description": "Missing required parameters." }</pre> |

Example:

```
# HTTP Request
curl -X 'POST' \
  'https://api.thinkrf.com/v1/login' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "<your client ID>",
    "password": "<your client secret>"
  }'

# HTTP Response
HTTP response code: 200
{
  "accessToken": "eyJraWQiOiJabTdjrUNIZmU4b.....bfjBE1mg",
  "expiresIn": 3600
}
```

Change Client Credential Password

The ChangeClientCredentialPassword API allows users to securely change their password by verifying their identity through thinkRF Identity Manager. thinkRF Identity Manager aligns with best practices in security and user management, ensuring seamless integration and secure handling of password changes within the system.

Request:

| Request Configuration | |
|---------------------------|------------------------------------|
| Resource | /v1/changePassword |
| Method | POST |
| Header | No authorization in header |
| Request Body | |
| username string | Your current API client ID. |
| oldPassword string | Your current API client's password |
| newPassword string | Your new API client's password |

Response:

| Response Body | |
|----------------------|---|
| status string | Response message "Password changed. Please login." |
| Status Code | |
| 200 | Success |
| 400 | { "errorCode": "E0001", "description": "Invalid Request" } { "errorCode": "E0007", "description": "Invalid Password Format" } |
| 401 | { "errorCode": "E0002", "description": "Not Authorized" } { "errorCode": "E0003", "description": "Too many requests" } { "errorCode": "E0004", "description": "User not found" } { "errorCode": "E0005", "description": "User not confirmed" } { "errorCode": "E0006", "description": "Login Error. Please contact administrator." } { "errorCode": "E0019", "description": "Missing required parameters." } |

Example:

```
# HTTP Request
curl -X 'PUT' \
  'https://api.thinkrf.com/v1/changePassword' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
    "username": "<your username>",
    "newPassword": "<your password>"
  }'

# HTTP Response
HTTP response code: 200
{
  "status": "Password changed. Please login."
}
```

User Management

Add User

The AddUser API allows administrators to add new users to the thinkRF Identity Manager, integrating with multiple systems to ensure users are created with the appropriate permissions and notified correctly. This API also grants access to SXM Dashboard.

Request:

| Request Configuration | |
|-----------------------|---|
| Resource | /v1/users |
| Method | POST |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Parameters | |
| Not required | Not applicable |
| Request Body | |
| user object | <p>User object example:</p> <pre>{ "username": "john.smith@company.com", "email": "john.smith@company.com", "password": "<password>", "firstName": "John", "lastName": "Smith", "userType": "customer", "role": ["corporate-admin"], "corporate": ["company name"], "subCorporates": [], "status": "active", "application": ["sxm-dashboard"] }</pre> <p>Notes:</p> <ul style="list-style-type: none"> • username: current user email (permanent identification) • Email: current user email • password: the temporary password to give to the user. The current release requires the admin to manually notify the new user of the temporary password. • application: specify 'sxm-dashboard' • roles: corporate-admin (default), author or reader • userType: specify 'customer' <p>Notes regarding adding user permission per role</p> <ul style="list-style-type: none"> • The corporate-admin user role can create any user under his corporate • The author and reader user roles are not permitted to create users |

Response:

| Response Body | Description |
|---------------|------------------------------|
| user object | Example user object created: |

| | |
|--------------------|--|
| | <pre> { "user": { "role": ["corporate-admin"], "email": "john.smith@company.com", "status": "active", "lastName": "Smith", "userType": "customer", "username": "john.smith@company.com", "corporate": ["company name"], "firstName": "John", "application": ["sxm-dashboard"], "subCorporates": [] } } </pre> |
| Status Code | |
| 200 | Success |
| 400 | <pre> {"errorCode": "E0001", "description": "Invalid Request"} {"errorCode": "E0002", "description": "Not Authorized"} {"errorCode": "E0005", "description": "User not confirmed"} {"errorCode": "E0007", "description": "Invalid Password Format"} {"errorCode": "E0008", "description": "Username Exists"} {"errorCode": "E0019", "description": "Input missing: {variable}"} {"errorCode": "E0046", "description": "Missing input parameters."} {"errorCode": "E0047", "description": "The service was not able to process your request, please verify the request parameters."} </pre> |

Update User

The UpdateUser API provides the functionality to update existing user's details within the thinkRF Identity Manager. This API interacts with multiple system components to ensure that user information is accurately updated while maintaining robust validation and permission checks.

Request:

| | |
|------------------------------|---|
| Request Configuration | |
| Resource | /v1/users/{username} |
| Method | PUT |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Parameters | |
| {username} string | The user ID, can't be updated Example: given_user@company.com |
| Request Body | |
| user object | <p>User object example:</p> <pre> { "email": "john.smith@company.com", "firstName": "John", "lastName": "Smith", "userType": "customer", "role": ["corporate-admin"], "corporate": ["company name"], "subCorporates": [], "status": "inactive", "application": ["sxm-dashboard"] } </pre> <p>where:</p> |

| | |
|--|--|
| | <ul style="list-style-type: none"> • role: corporate-admin (default), author or reader <ul style="list-style-type: none"> ◦ 'corporate-admin' role can create a user for the corporate ◦ 'author' and 'reader' roles are not permitted to create users • userType: specify 'customer' • status: set 'inactive' for users deactivated, and 'active' for users reactivated |
|--|--|

Response:

| Response Body | Description |
|---------------|---|
| user object | <p>Example user object created:</p> <pre>{ "user": { "username": "john.smith@company.com", "email": "john.smith@company.com", "firstName": "John", "lastName": "Smith", "userType": "customer", "role": ["corporate-admin"], "corporate": ["company name"], "subCorporates": [], "status": "inactive", "application": ["sxm-dashboard"] } "userPermission": {} // restrict to SXM Dashboard }</pre> |
| Status Code | |
| 200 | Success |
| 400 | <pre>{ "errorCode": "E0001", "description": "Invalid Request" } { "errorCode": "E0002", "description": "Not Authorized" } { "errorCode": "E0019", "description": "Input missing: {variable}" } { "errorCode": "E0046", "description": "Missing input parameters." }</pre> |
| 404 | <pre>{ "errorCode": "E0004", "description": "User Not found" }</pre> |

Get User

The GetUser API facilitates the retrieval of the user information from the thinkRF Identity Manager, applying role-based and corporate-based filters to ensure authorized access to user data. By following the methodology outlined, administrators can ensure secure, efficient user data retrieval with minimal risk.

Request:

| Request Configuration | |
|-----------------------|--|
| Resource | /v1/users/{username} |
| Method | GET |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Parameters | |
| {username} string | The user ID. Example: given_user@company.com |
| Request Body | |
| | Not applicable. |

Response:

| Response Body | Description |
|---------------|------------------------------|
| user object | Example user object created: |

| | <pre>{ "user": { "username": "john.smith@company.com", "email": "john.smith@company.com", "firstName": "John", "lastName": "Smith", "userType": "customer", "role": ["corporate-admin"], "corporate": ["company name"], "subCorporates": [], "status": "inactive", "application": ["sxm-dashboard"] }, "userPermission": {} // restrict to SXM Dashboard // }</pre> |
|-------------|---|
| Status Code | |
| 200 | Success |
| 400 | <pre>{ "errorCode": "E0001", "description": "Invalid Request" } { "errorCode": "E0002", "description": "Not Authorized" } { "errorCode": "E0019", "description": "Input missing: {variable}" }</pre> |
| 404 | <pre>{ "errorCode": "E0004", "description": "User not found" }</pre> |

List Users

The ListUsers API lists all users from the thinkRF Identity Manager, applying role-based and corporate-based filters to ensure authorized access to user data. This API is essential for administrators who need to manage and oversee user lists dynamically.

Request:

| Request Configuration | |
|-----------------------|--|
| Resource | /v1/users |
| Method | GET |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Body | |
| | Not applicable. |

Response:

| Response Body | Description |
|------------------|-----------------------------------|
| user object list | Example user object list created: |

| | |
|-------------|---|
| | <pre>[{ "username": "john.smith@company.com", "email": "john.smith@company.com", "firstName": "John", "lastName": "Smith", "userType": "customer", "role": ["corporate-admin"], "corporate": ["company name"], "subCorporates": [], "status": "inactive", "application": "sxm-dashboard" }, { "username": "peter.franco@company.com", "email": "peter.franco@company.com", "firstName": "Peter", "lastName": "Franco", "userType": "customer", "role": ["author"], "corporate": ["company name"], "subCorporates": [], "status": "active", "application": "sxm-dashboard" }]</pre> <p>Note: Users from the same requester's corporate(s) will be listed</p> |
| Status Code | |
| 200 | Success |
| 400 | {"errorCode": "E0002", "description": "Not Authorized"} |
| 404 | {"errorCode": "E0004", "description": "User not found"} |

Node Management

The node list API lists the current SXM nodes available in the system. It includes each node's status and other attributes as described in the Response output below.

Request:

| Request Configuration | |
|-----------------------|--|
| Resource | /v1/node/list |
| Method | POST |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Body | |
| Optional | It is an optional filter to specify one or more specific nodes for their attributes or leaves empty payload {}, the API will return all nodes; or Example: <pre> { "filterSelector": { "nodeId": {"\$in": ["220817-107"]} } } </pre> |

Response:

| Response Body | Description |
|-------------------|---|
| nodes list | Response with output results with example data: <pre> { "nodes": [{ "group": null, "active": true, "nodeId": "211208-106", "gnssType": null, "lastCity": "Ottawa", "nodeLabel": "Testing (Release)", "lastAction": { "bist": { "id": "node_action_Ekg2huSORK3Yd3t7NZIq1", "type": "bist", "origin": "cloud", "result": { "status": -200, "testTime": "Mon Nov 11 21:55:35 UTC 2024", "errorCodes": ["-502"], "errorMessage": "System error detected, please reboot and run the test again after 5 minutes.If error </pre> |

| | |
|--|---|
| | <pre> persists, contact support@thinkrf.com and provide error codes." }, "status": "completed", "device_id": "211208-106", "created_by": "jane.doe@thinkrf.com", "created_timestamp": "2024-11-11T21:55:24.182818", "updated_timestamp": "2024-11-11T21:55:35.556583" }, "reboot": { "id": "node_action_aWOaTcfiAlhC3bLLYzZ_", "type": "reboot", "origin": "cloud", "result": null, "status": "created", "device_id": "211208-106", "created_by": "jane.doe@thinkrf.com" }, "pauseUpdates": { "id": "node_action_PzKQ0IRD63VZyu9aafPdS", "type": "pauseUpdates", "origin": "cloud", "result": null, "status": "created", "device_id": "211208-106", "created_by": "jane.doe@thinkrf.com" }, "resetApplicationCredential": { "id": "node_action_DHEOTr-OQoYmIjurHetDG", "type": "resetApplicationCredential", "origin": "cloud", "result": {}, "status": "completed", "device_id": "211208-106", "created_by": "jane.doe@thinkrf.com", "created_timestamp": "2025-03-19T14:27:32.553822", "updated_timestamp": "2025-03-19T14:27:34.201114" } }, "lastRegion": "ON", "description": null, "lastAddress": "390 March Rd, Kanata, ON K2K 0G7, Canada", "lastCountry": "CA", "lastActivity": "2025-03-20T14:39:30.66586", "lastLatitude": 45.338478, "lastLongitude": -75.91528, "lastTelemetry": { "ip": { "name": "ipAddr", "value": "169.254.209.116", "createdDatetime": "2025-03-12 20:12:26", "updatedDatetime": "2025-03-20 14:38:55" }, "mac": { "name": "macAddr", "value": "a0:ce:c8:52:cd:78", "createdDatetime": "2025-03-12 20:12:26", "updatedDatetime": "2025-03-20 14:38:55" }, "gnss": { "name": "gnss", "value": { "altitude": 67108863.96875, "latitude": 512.0, "longitude": 512.0, "dynamicMode": null }, "createdDatetime": "2025-02-27 19:37:21", "updatedDatetime": "2025-03-20 14:38:55" } }, </pre> |
|--|---|

```

"networkType": {
  "name": "networkType",
  "value": "WIFI",
  "createdDatetime": "2025-02-27 19:37:22",
  "updatedDatetime": "2025-03-20 14:38:55"
},
"softwareContainer": {
  "name": "softwareContainer",
  "value": [
    {
      "name": "node_local_control",
      "type": "container",
      "label": "Local Control App",
      "version": "1.1.1"
    },
    {
      "name": "node_system_updates",
      "type": "container",
      "label": "System Update",
      "version": "SU_2.4.2_NM_1.0.2"
    },
    {
      "name": "node_advanced_signal_analysis",
      "type": "container",
      "label": "ASA",
      "version": "1.1.1"
    }
  ],
  "createdDatetime": "2025-02-27 19:37:22",
  "updatedDatetime": "2025-03-20 14:38:55"
},
"softwareLibraries": {
  "name": "softwareLibraries",
  "value": [
    {
      "name": "asa_release_tag",
      "type": "library",
      "label": "ASA Component",
      "version": "1.1.1"
    },
    {
      "name": "libtrf_release_tag",
      "type": "library",
      "label": "Libtrf API",
      "version": "1.7.2"
    },
    {
      "name": "libtrf_sa_release_tag",
      "type": "library",
      "label": "3GPP SA",
      "version": "2.8.2"
    },
    {
      "name": "co_release_tag",
      "type": "library",
      "label": "Channel Occupancy",
      "version": "1.1.0"
    }
  ],
  "createdDatetime": "2025-03-14 14:09:15",
  "updatedDatetime": "2025-03-20 14:38:55"
}
},
"commercialType": "private",
"configurations": [
  {
    "id":
"full_range_compliance_stationary_1_20250319_1528",
    "service": "3gpp_sa",
    "createdBy": "node-manager-hook",
    "createdDate": "2025-03-19T15:28:15",

```

| | |
|--------------------|--|
| | <pre>"configuration": { "version": "4", "scanMode": "compliance", "eventName": "startSa", "frequencies": [{ "stop": 4500, "start": 600, "source": { "type": "user-defined", "value": null } }], "nodeGNSSType": "stationary", "dataReduction": true, "scanSensitivity": 100 }], "corporateOwner": "thinkrf", "nodeSecondaryId": "211208-106_45.33_-75.95", "updatedTimestamp": "2025-03-20T14:39:30.66586", "corporateOwnerLabel": "thinkRF", "serviceProcessStatus": [{ "status": "processing", "service": "3gpp_sa", "status_request_id": "LViUvbdnKac9ENUeKo_sd", "status_request_datetime": "2025-03-20 14:39:25", "status_response_datetime": "2025-03-20 14:39:25" }], "applicationCredential": [{ "password": "Test@2test", "application": "nlc" }] }]</pre> <p>Note: “NLC” is also known as Local control application (or LCA).</p> <p>Response without results:</p> <pre>{ "nodes": [] }</pre> |
| Status Code | |
| 200 | Success |

Accessing 3GPP Source Data

Infrastructure

The 3GPP source data as reported by the node is classified under 'infrastructure' API, in reference to 3GPP network infrastructure data. This resource lists the raw base cellular radio frequency data captured based on filter criteria per region, frequency, technology, band, and other radio frequency entities. It allows fine-tuned data retrieval, essential for detailed analysis and reporting.

Request:

| Request Configuration | |
|---|--|
| Resource | /v1/infrastructure |
| Method | POST |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Body | |
| page (optional) number | Page number for supporting the pagination (default value is 1) |
| pageSize (optional) number | Number of records to be included in the page (default value is 100) |
| sortBy (optional) string | The infrastructure field name to be sorted |
| ascending (optional) boolean | Options: true to sort the field in the sortBy in ascending order, and false to descending order. The default will be true. |
| fields (optional) Array of string | This object must provide the list of RF infrastructure fields to be returned. The list of possible fields can be found at /v1/dictionary end point API. <pre> "fields": ["band", "frequency", "nodeId", "plmn", "mnc"] </pre> |
| recordDate (Optional) object | Object attributes: <ul style="list-style-type: none"> start: The start data captured date in the format YYYY-MM-DD end: The start data captured date in the format YYYY-MM-DD Example: <pre> "recordDate": { "start": "2023-05-01", "end": "2023-06-01" } </pre> Restriction: Period could be no more than 30 days for API call. |
| filterSelector (optional) Array of objects | This object contains the possible filters criteria to set a scope for the infrastructure data. The next fields descriptions are related to the possible filters: List of operators: <ul style="list-style-type: none"> - "\$and": AND logical operator - "\$or": OR logical operator - "\$eq": Equal (=) - "\$ne": Not Equal (<>) - "\$lt": Lower Than (<) - "\$lte": Lower Than Equal (<=) - "\$gt": Greater Than (>) |

| | |
|--------------------------------------|---|
| | <ul style="list-style-type: none"> - "\$gte": Greater Than Equal (>=) - "\$in": Include (in) - "\$nin": Include (nin) - "\$exists": true (IS NOT NULL) or false (IS NULL) <p>Payload examples:</p> <p>1. Direct case without AND or OR logical operators:</p> <pre>"filterSelector":{ "band": {"\$eq": 4}, "operator": {"\$in": ["Telus","Rogers"]} }</pre> <p>2. Cases with AND or OR logical operators:</p> <pre>"filterSelector": { "\$or": [{"technology": {"\$eq": "3G"}}, {"nodeId": {"\$eq": "220510-103"}}, {"nbId": {"\$ne": "401065"}}] }</pre> |
| detailLevel (optional) string | <p>What is the granularity for the data. Options are: "day", "minute".</p> <p>Example:</p> <pre>"detailLevel": "day"</pre> |

Response:

| Response Body | Description |
|--|---|
| page number | The page number related to the current request |
| pageSize number | The pageSize related to the current request |
| totalRecords number | The total records related to the current request |
| sortBy string | The sort by field related to the current request |
| ascending boolean | The ascending/descending order related to the current request |
| fields array of string | <p>The fields of the table header related to the result data from infrastructure attribute:</p> <pre>"fields": ["band", "frequency", "nodeId", "plmn", "mnc"]</pre> |
| infrastructure array of objects | <p>The list of bands filtered by the criteria defined in the request:</p> <p>Example:</p> <pre>{ "infrastructure": [{ "band": 2, "frequency": 624500000, "technology": "LTE", "nodeId": "220331-101", "plmn": 1026, "mnc": 10 }, { "band": 5, "frequency": 1861500000, "technology": "NR5G", "nodeId": "220331-101", "plmn": 2058, "mnc": 45 }] }</pre> |

| Status Code | |
|-------------|--|
| 200 | Success |
| 400 | Wrong filter criteria parameters sent. |

Infrastructure Export Request

Infrastructure data export API is executed off-line in a batch process. By calling this API, the client will receive a unique export identifier that will be used to monitor the export execution status at a later moment (usually after 3-5 seconds).



Note: As the active SXM nodes collect new data every day, you might want to synchronize your data platform frequently. thinkRF recommends incremental raw data integration **daily** via `/v1/infrastructure/export` to load the SXM data continuously to your data platform. This will reduce your sync effort as well as the load on our system.

Request:

| Request Configuration | |
|-----------------------|---|
| Resource | <code>/v1/infrastructure/export</code> |
| Method | POST |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Body | <p>The same body fields from <code>/v1/infrastructure</code> API described in the Infrastructure section, except for the following:</p> <ol style="list-style-type: none"> 1. pageSize: This field is replaced by the API default of 70,000. 2. page: This field is superseded by the API default of 1. 3. fields: It is mandatory to pass exactly the fields the user wants to export. It does not support empty list []. <p>Restriction: Period no more than 30 days for API call.</p> |

Response:

| Response Body | Description |
|------------------------|--|
| exportId string | Unique export ID to be used to retrieve the export status and the downloading URL. <pre>{ "exportId": "72e3a119-0a95-4329-8363-84cfd557784" }</pre> |
| Status Code | |
| 200 | Success |
| 400 | Wrong filter criteria parameters sent. |

Infrastructure Export Status

Infrastructure data export status API. The client calls this API to monitor and get the file download link from `/v1/infrastructure/export/status`. It is suggested to call the API every 3 seconds to check the status.

Request:

| Request Configuration | |
|------------------------|--|
| Resource | /v1/infrastructure/export/status |
| Method | POST |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Body | |
| exportId String | Unique export ID to be used to retrieve the export status and the downloading URL. <pre>{ "exportId": "72e3a119-0a95-4329-8363-84cfd557784" }</pre> |

Response:

| Response Body | Description |
|------------------------|--|
| exportId object | Confirmed export ID used to retrieve status of completion, along with the export downloading URL. The export generated is in a CSV file format. <pre>{ "exportId": "72e3a119-0a95-4329-8363-84cfd557784", "status": "complete", "url": "https://s3.amazonaws.com..." }</pre> <p>Notes:</p> <ul style="list-style-type: none"> - Field status values: <ul style="list-style-type: none"> - 'in-progress': The export process is still running - 'complete': The export is done. The url field will contain the download URL. - 'fail': The process has failed - Field url values: <ul style="list-style-type: none"> - 'https pre-signed url': Link to download the CSV file - 'null': if there is no data available to download |
| Status Code | |
| 200 | Success |
| 204 | Record not found. |
| 400 | Wrong filter criteria parameters sent. |

Dictionary

The dictionary infrastructure API provides a list of SXM fields and its description. They can be used with the Infrastructure and Deployment APIs to retrieve the desired fields.

Request:

| Request Configuration | |
|-----------------------|--|
| Resource | /v1/dictionary/infrastructure |
| Method | GET |
| Header | Authorization: <Access Token retrieved in login API> Accept: application/json |
| Request Body | |
| | Not applicable. |

Response:

| Response Body | Description |
|---------------|--|
| dictionary | <p>This query returns a list of 3GPP output parameters and their definition, along with other related parameters computed in the cloud or from the 3GPP configuration.</p> <p>An example with only a few parameters shown:</p> <pre>{ "dictionary": [{ "name": "scanMode", "type": "string", "label": "3GPP Scan Mode", "service": ["3gpp_sa"], "description": "This selection field in the Node Configuration form allows users to select which 3GPP scan mode to use for the detection. See in the Dictionary for the description of \"Infrastructure Scan Mode\" and \"Compliance Scan Mode\"." }, { "name": "address", "type": "string", "label": "Address", "service": ["3gpp_sa"], "description": "Address of the data captured location." }, { "name": "numberOfAntenna", "type": "number", "label": "Antennas", "service": ["3gpp_sa"], "description": "The number of antennas used for transmission, as determined by the CRC mask in the PBCH." }, { "name": "arfcn", "type": "number", "label": "ARFCN", "service": ["3gpp_sa"], "description": "Absolute Radio Frequency Channel Number." }] }</pre> <p>where:</p> <ul style="list-style-type: none"> • name: the parameter name • label: the proper name of the parameter • service: which application does this parameter belong to • type: the parameter's value format |
| Status Code | |
| 200 | Success |
| 204 | Record not found. |

Appendix A – SXM Analysis Data Parameter Description

The following table lists alphabetically all the detected and decoded output parameters from SXM devices as reported by the SXM API, along with their description.

| Parameters | Description |
|---|---|
| active boolean | Current cell's active status. Example: true |
| arfcn number | Absolute Radio Frequency Channel Number. Example: 1075 |
| band number | Detected 3GPP Band ID for LTE or 5G from SIB1 References: LTE bands Wikipedia and 5G bands Wikipedia |
| bandwidth number | Downlink spectral bandwidth measured, in MHz. Example: 10 |
| beamIndex number | Beam index for 5G NR. Example: 255 |
| blCe string | BL/CE flags for LTE carriers of NB-IoT device connection with the following values reported: <ul style="list-style-type: none"> NA – Not Applicable for NR carriers or 4G LTE. supported – BL/CE operation is supported for NB-IoT. notSupported – BL/CE operation is not supported. Example: notSupported |
| captureLocation object (dictionary) | SXM node's GPS location information, consisting of altitude, latitude and longitude. Example: {"alt": 79.03125, "lat": 45.340218, "long": -75.909416} |
| captureStartTimestamp string | Raw data's capture start timestamp, in UTC. Example: 2022-03-25T06:47:40 |
| cellBarring number | Cell-barred status indicating whether a UE can camp on this channel. It is derived from SIB1, with three values: 1, 0 or -1. Value 1 means not barred or 'active', 0 means barred, and -1 for undetermined or not decoded to the SIB level. Example: 1 |
| cellId (CGI group) number | 8-bit Cell IDentifier of the 4G ECI or 4 to 14-bit of 5G NCI value, uniquely identify each Base transceiver station (BTS) or sector of a BTS within a location area code (LAC). Not to be confused with PCI value. Example: 39 |
| cellRefPath string | API resource path to retrieve the current cell data Example: /v1/nodes/211333-100/cells/39 (i.e /v1/nodes/{nodeId}/cells/{cellId}) |
| centerFrequency number | Detected downlink center frequency of the LTE carrier or the 5G NR's SSB carrier, in Hz. Example: 26499999.00 |
| cinr number | CINR (in dB) is the ratio between the channel impulse response (CIR) power to the noise power. It is an indicator of how observable the channel is. Example: -23.29 |
| channel | The physical channel of LTE or 5G NR. |

| Parameters | Description |
|--|---|
| string | Example: PDSCH |
| cgi string | Cell Global Identification is a globally unique identifier for a Base Transceiver Station or cells. The field is a computed parameter based on the combination of four source parameters from the node and is presented as such: mnc-mcc-eCGI/nCGI . The eCGI and nCGI portion is a combination of bits representation of <nblId><cellId> (along with nblIdLength length for 5G to determine its cellId bit length) and is converted to an integer value. Examples: <ul style="list-style-type: none"> 4G: 302-220-56145469 (with eCGI is calculated from nblId 219318 and cellId 61). 5G: 302-220-22939089951 (with nCGI is calculated from nblId 1400090, cellId 15391, and nblIdLength is 22). |
| dciformat string | Format of Downlink Control Information. Example: DCI-1a |
| dss string: NA, DSS or notDSS | Dynamic Spectrum Sharing indicator value, which has three possible outcomes: <ul style="list-style-type: none"> NA: not applicable - DSS operation is not applicable for this carrier. This is the case for LTE carriers or when DSS operation is not detected depending on the SXM signal analysis configuration parameters (such as LTE analysis is not disabled). DSS: The carrier is an NR DSS carrier. notDSS: The carrier is not an NR DSS carrier. Example: notDSS |
| evm number | Error Vector Magnitude values. Example: 87 |
| frequencyError number | The carrier's fine offset error value due to the receiver's LO shift and movement, in Hz Example: 3620.1438 |
| isNsa isSa string: NA, ND, NO, YES | These parameters are for 5G NR NSA/SA cell operation identification, respectively. When decoded to SIB1 (PDSCH), they will have these values: <ul style="list-style-type: none"> NA: not applicable (for LTE carriers) ND: not determined NO: not supported YES: supported Example: ND |
| mcc number | Mobile Country Code. Three digits (12 bits) identifier that identifies the country. Example: 289 |
| mnc number | Mobile Network Code. Three digits identifier that within a country uniquely identifies an operator. Note that a single operator may have several MNC's associated due to legacy acquisitions and so forth. Example: 88 |
| mode string: TDD, FDD, Invalid | Duplex mode, TDD or FDD. Example: TDD |
| modulation string | Modulation type. Example: qpsk |
| nblId (CGI group) number | NodeB IDentifier, unique within a PLMN. nblId for 4G or gNB-ID for 5G. Example: 220078 |
| nblIdLength number | The length or how many bits the (g)NB-ID data (22-32 bits) are in the 36-bit NCI of 5G NR. |

| Parameters | Description |
|---|--|
| | Example: 8 |
| node string | SXM edge device (node) unique ID. Example: 210414-001 |
| nrDownlinkCenterFrequency number | 5G NR downlink center frequency, if decoded to SIB1 (PDSCH), in Hz. Otherwise, -1. Note: This is not the same as 5G's centerFrequency value, which is the 5G NR's SSB carrier frequency. Example: 3624990000 |
| nrTddScs number | Subcarrier spacing of the NR TDD pattern when decoded to PDSCH. Otherwise, -1. Example: 30 |
| numberOfAntenna number | Transmitter's antenna configuration from the PBCH CRC Example: 2 |
| occupancy number | Spectrum Occupancy, in %. Example: 20 |
| pci number | Physical Cell ID (from PSS & SSS detection) Example: 314 |
| plmn (CGI group) string | Public Land Mobile Network. Uniquely identifies the operator globally. Made up of the MCC and MNC. The PLMN value is calculated according to "3GPP TS 24.008" standard as such: given MCC[654] & MNC[321], PLMN is 54f621 ('f' is used) if MNC is 2 digits or 543621 if MNC is 3 digits. Examples: 025320, 21f320 Note: as off API release v1.2.2, plmn parameter is changed from number to string type to better present the value as well as supporting the 'f' value. |
| power number | Power of the SXM receiver's detected carrier value, in dBm. Example: -50.885082 |
| rat string: LTE, NR5G, 3G, Invalid | Radio Access Technology type Example: LTE |
| reservedForOperatorUse number: 0 or 1 | An indicator for whether all UEs (User equipment) could connect to the cell or not. Reported per PLMN with the value of 1 (not reserved) or 0 (reserved). Example: 1 |
| rnti number | Radio Network Temporary Identifier. Example: 65535 |
| rsi number | RSSI value, in dBm Example: 17.524155 |
| rsrq number | RSRQ value, in dB Example: -10.332217 |
| rsrp number | RSRP value, in dBm Example: -11.558676 |
| rs-sinr number | RS-SINR value, in dB Example: 33.89399 |
| scs number | Sub Carrier Spacing, in kHz. Example: 15 |
| secondaryPLMNs object (dictionary) | Secondary PLMNs from SIB1 Example: {"plmn1": 2174502, "plmn2": 2174100} |

| Parameters | Description |
|--|---|
| ssbWithSib1 number | A flag to indicate the existence or not of the scheduled SIB1 in NR for a decoded SSB. This flag is used to identify NR NSA/SA operation of an NR cell. It has the following values: <ul style="list-style-type: none"> • -1: it is an invalid value used for LTE carriers. • 0: there is no information for the neighbor SSB having SIB1. • 1: the current decoded SSB has SIB1. • any other positive value indicates the following: <ul style="list-style-type: none"> – the BW around the current SSB has no SSB with SIB1, or – the next SS_REF (SSB center frequency) contains SIB1. Example: 1 |
| ssbPositions string | A bit-map data showing the existing beams and their locations in time for SSB transmission in the NR cell. Example: 01000000 |
| taCode number | Tracking Area code derived from SIB1 Example: 29050 |
| tddFrameConfig string | A string of characters, representing the downlink-uplink pattern for both LTE and 5G NR. Example: D D F_6_0 F_0_4 U U D D D D |
| uplinkArfcn number | Uplink Absolute Radio Frequency Channel Number, associated with the LTE UL carrier (when decoded to SIB2 (PDSCH)) or PCell carrier in NR. Example: 721824 |
| uplinkCenterFrequency number | Uplink detected center frequency of the LTE carrier (when decoded to SIB2 (PDSCH)) or the 5G NR's SSB, in Hz. In addition, for 5G NR, it is associated with the PCell carrier. Example: 3624990000 |
| uplinkBandwidth number | Uplink Spectral Bandwidth measured (in Hz), associated with LTE uplink carrier (when decoded to SIB2 (PDSCH)) or the PCell carrier in NR, in MHz. Example: 10 |

Appendix B – SXM 4G/5G Naming Consistency

The following table lists some 3GPP 4G LTE and 5G NR specific names which are represented by common terms in SXM data.

| 3GPP 4G LTE Term | 3GPP 5G NR Term | SXM Common Term | Description |
|------------------|-----------------|-----------------|---|
| ECGI | NCGI | CGI | (E-UTRAN/NR) Cell Global Identity. Uniquely identifies a cell in the world. Made up of the PLMN and CI and found in the SIB. |
| ECI | NCI | CI | Cell Identity. Uniquely identifies a cell within an operator (PLMN). |
| eNB-ID | gNB-ID | NB-ID | NodeB Identifier. Uniquely identifies a NodeB within an operator (PLMN). |
| CellID | CI | CellID | An internal descriptor (ranging 0 to 255), sometimes called cell identifier to uniquely define a cell within a nodeB. Not to be confused with PCI. |
| GENB-ID | | GNB-ID | Global NodeB identifier. Made of the PLMN and the NB-ID to uniquely identify a NodeB in the world. |
| EARFCN | NR-ARFCN | ARFCN | Absolute Radio-Frequency Channel Number. Defines the uplink and downlink centre frequencies for a transmission. It can be derived from a combination of the band # and the DL centre frequency. |

Revision History

| Version | Date | Comments |
|---------|----------------|--|
| v1.0.0 | April 29, 2022 | First release |
| v1.0.1 | Dec 09, 2022 | <ul style="list-style-type: none"> - Added 'band' parameter as an output for all "cell record" related detail, see Error! Reference source not found. - Corrected 'mcc' parameter to be number type in the description - Reformatted the document and corrected some parameter types |
| v1.0.2 | Feb 01, 2023 | <ul style="list-style-type: none"> - Improved parameter's description throughout the document - Added more parameter's type |
| v1.1.0 | Mar 15, 2023 | <ul style="list-style-type: none"> - Added new parameters for cell record: <ul style="list-style-type: none"> + dss + ssbPositions + uplinkArfcn + uplinkCenterFrequency + uplinkBandwidth - Updated some field names: <ul style="list-style-type: none"> + cellref to cellId + cell_barred to cellBarred + dci_format to dciFormat + duplex_mode to duplexMode + enbid to nbId + earfcn to arfcn + location to locationPath + noderef to nodeId + operationalStatus to operationalStatusPath + status to statusPath + txref to txId - Added the attribute nodeId in the result of the API List Nodes - Updated the API attributes description throughout the document - Added Appendix A – SXM Analysis Data Parameter |
| v1.1.1 | Sept 11, 2023 | <ul style="list-style-type: none"> - Reformat the tables and moved the definition of the 4G/5G parameters to Appendix A – SXM Analysis Data Parameter - Added new parameters: <ul style="list-style-type: none"> + reservedForOperatorUse + nbIdLength + nrDownlinkCenterFrequency + blCe + ssbWithSib1 + isSa + isNSA + tdFrameConfig - Removed unsupported caConfigType from the document |
| v1.1.2 | Oct 25, 2023 | <ul style="list-style-type: none"> - Fixed missing parameters in their API subsections: <ul style="list-style-type: none"> + channel + nrTddScs + tddFrameConfig |
| v1.1.3 | Mar 27, 2024 | <ul style="list-style-type: none"> - Added cinr parameter - Replaced sinr with rs-sinr. sinr parameter will be deprecated in the upcoming release. |

| Version | Date | Comments |
|----------------|---------------|---|
| v1.1.4 | June 18, 2024 | <ul style="list-style-type: none"> - Deprecated cellBarred (Boolean type) and replaced it with cellBarring (number type). cellBarring has been available since Jan 2024 but was missed being mentioned in the previous document release. It has data of cellBarred before Jan 2024 whenever the data is available. |
| v1.2.0 - 1.2.1 | Dec 1, 2024 | <ul style="list-style-type: none"> - Major format changed. - A new set of APIs that simplifies significantly the data retrieval process instead of the previous version API model where thousands of API calls are involved to select the nodes, cells and related transmission data, affecting the integration performance. - Improved Overview with an added entity relationship diagram. - Update changes as a result of API v1.2.0/.1 release: <ul style="list-style-type: none"> + Response errors for Get Access Token + Request & Response errors for Change Client Credential Password + Added User Management, Node Management, and Accessing 3GPP Source Data sections. - Added definition to 'cgi' parameter. - Deprecated the following functions and sections: <ul style="list-style-type: none"> + /v1/users of "List All Users", replace by List Users of User Management + "List Nodes", replace by "Node/list" of Node Management. + "List Cells Reported by a Node", "List Cell Record Details", "Transmitter Management Resources", "Transmission History Management" <ul style="list-style-type: none"> - replaced by Accessing 3GPP Source Data + "Event Management" as the functions are not useful. This will be provided in the near future with a proper "Notification Services". |
| v1.2.2 | Dec 18, 2024 | <ul style="list-style-type: none"> - Updated the definition for 'plmn' in Appendix A – SXM Analysis Data Parameter list, including important note on its format change from 'number' to 'string'. |
| v1.2.3 | Mar 18, 2025 | <ul style="list-style-type: none"> - Updated the support phone number - Updated Node Management with the new response which has all the attributes and telemetries. |
| v1.2.4 | April 2, 2025 | <ul style="list-style-type: none"> - Deprecated PBCH and CA Config parameters in the API output data as the parameters were not needed. |
| v1.2.5 | May 30, 2025 | <ul style="list-style-type: none"> - Update Endpoint URL with new API endpoint URLs for the Global and Canadian cloud regions. |