EVOTIX

API Guide

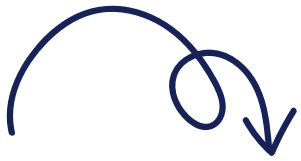
Assure SCIM API

User Guide: Managing users via the SCIM API

Evotix Ltd. Revision 3.0 (September 2025)

US +1 (872) 215 5913 **UK** +44 (0) 161 521 8490 **AU** +61 3 8595 5909 **Visit** evotix.com

EVOTIX



The Assure SCIM API enables Evotix to provide integration capabilities for customers using Assure.

The SCIM API is accessible via the public internet and is implemented as a RESTful API. Using the SCIM API you can automate user provisioning. Making use of the SCIM API requires a level of technical expertise so this is typically something that a company's IT function would handle.

This guide focusses on how to set up the SCIM API to update your Users via SCIM in Assure. A separate guide is available for setting up the Customer API.

EVOTIX

Contents

EVOTIX

EVOTIX

Introduction

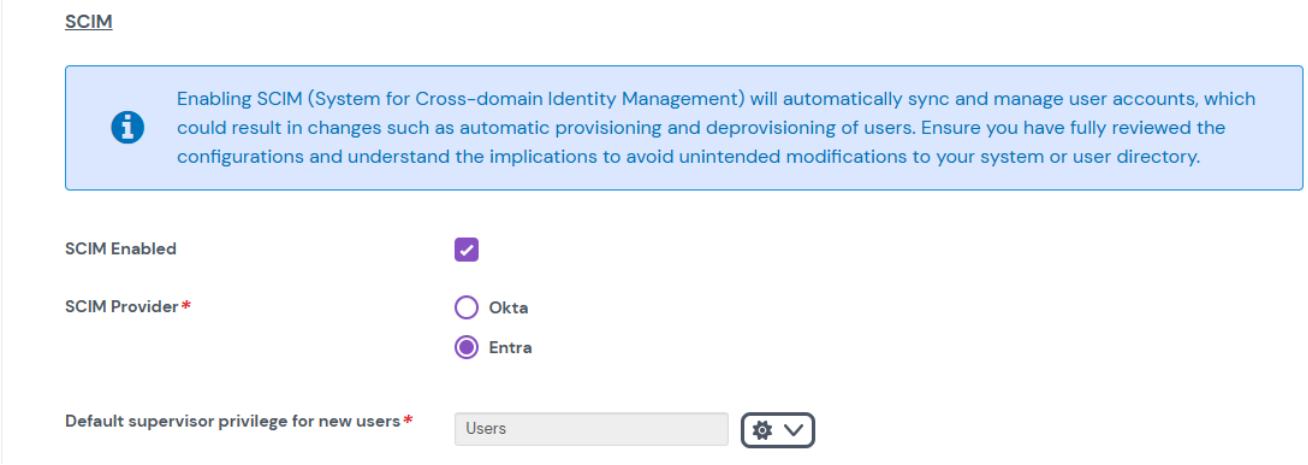
Implementing a SCIM API streamlines identity management by providing a standard, automated approach to user provisioning and lifecycle management across multiple systems. With SCIM, organisations can efficiently synchronise user data, reduce manual administrative tasks, and enhance security by ensuring that user access is consistent and up to date everywhere. This not only improves operational efficiency but also supports compliance and governance initiatives.

Before attempting to use the SCIM API make sure you understand the User and Group JSON objects (see the sections below) and that you have the required details to access the API (i.e. the URL prefix, API key, etc), follow the Getting Started guide if you don't have these details already.

Pre-configuration

Before using the APIs to create, update, or disable users and roles, some pre-configuration is required in Assure. A system setting hidden behind SysAdmin privileges needs to be enabled.

Settings > System Settings > Features



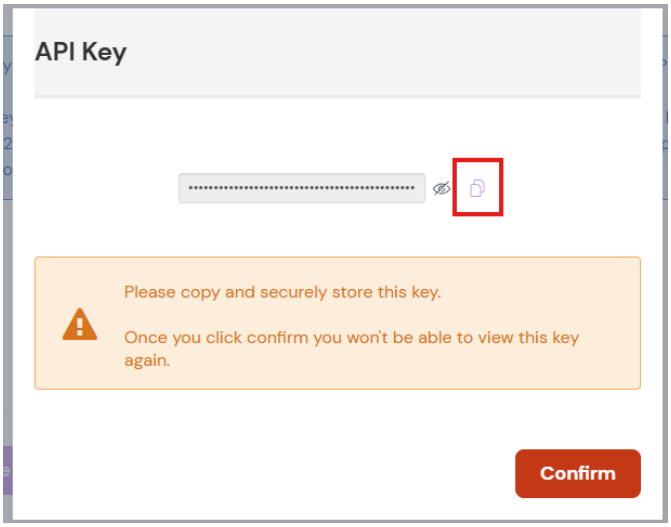
The following values need to be set

1. SCIM Enabled: If SCIM is enabled
2. SCIM Provider: the system that is going to be managing users and groups via SCIM. We currently only support OKTA or Microsoft Entra Id
3. Default supervisor privilege for new users: This is the supervisor privilege that will be given to all new users added to the system via SCIM.

API Access keys will be required for authorization.

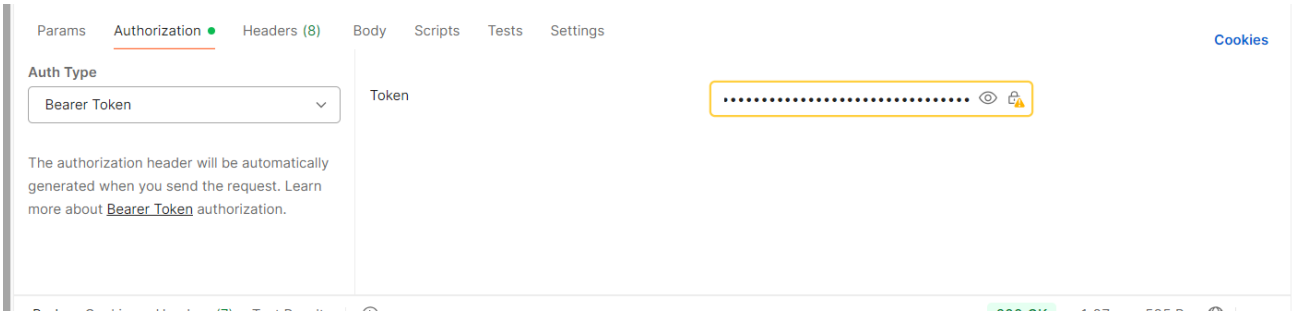
1. Select Settings followed by API Access

2. Select New Record
3. Enter a title and description
4. Click Save
5. Copy the generated key and click save. Keep this key safe as it cannot be retrieved once this screen has closed.



Authentication with the SCIM API

Due to some limitations in how SCIM providers, such as OKTA, allow API keys to be provided to Assure, the SCIM API requires the API key to be in the 'Authorization' header instead of the 'x-api-key' header used by the Customer API. The screenshot below shows how this can be configured in Postman



Http Status Codes

Any HTTP response code is the range 200-299 should be treated as success (do not use the contents of the response body for identifying success or failure as these may change). Implement effective error handling as per the guide for error handling.

Limitations

The following lists the current limitation of the methods. It is expected that these will be addressed in future Assure releases although no specific timescales are defined for their release.

1. While you can create roles through SCIM, you cannot configure role permissions through SCIM endpoints. When a new role is created, all permissions will default to “deny”. Any permissions updates will have to be done within Assure.
2. All users created through SCIM will be given the default supervisor privilege that is configured in system settings. This can be edited manually in Assure after the user has been created.

Advice for JSON objects

When creating/updating a SCIM User or Group then the details need to be provided in the form of a JSON object. JSON is the most commonly used syntax for describing data objects in RESTful APIs, for an introduction to JSON see this guide. To aid understanding of how the JSON fields in the User/Group JSON objects relate to the resulting User/Group setup in Assure the following sections contain some worked examples.

The text encoding to be used for all interactions with the SCIM API is UTF-8. This is the unofficial standard today for software tools and platforms; however, it is important to check that you are using UTF-8 as if not then when you get foreign characters in the data or other symbols like emoticons these will not appear correctly in Assure.

As JSON objects are described using plain text it is possible to hand craft these objects for initial testing. However, it is VERY IMPORTANT to ensure that when generating JSON objects to send to the SCIM API you use a proper JSON library or a tool with native JSON support. This is because JSON relies on ‘escaping’ for certain data values, if this escaping is not done correctly it will lead to API errors and can also be a source of security vulnerabilities.

With tools, like Postman, where the JSON object is manually entered you must ensure the contents of any string values are properly escaped.

EVOTIX

Managing User accounts via the SCIM API

SCIM user data

This example shows the full SCIM user data which can be used to create a SCIM user.


```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User",
    "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
  ],
  "id": "2819c223-7f76-453a-919d-413861904646",
  "externalId": "2819c223-7f76-453a-919d-413861904646",
  "userName": "exampleScimUser",
  "name": {
    "formatted": "Example ScimUser",
    "familyName": "Example",
    "givenName": "Example",
    "middleName": "",
    "honorificPrefix": "Ms.",
    "honorificSuffix": ""
  },
  "displayName": "Example ScimUser",
  "nickName": "exampleScimUser",
  "profileUrl": "https://login.example.com/exampleScimUser",
  "roles": [
    "FieldTeam",
    "GeneralUser"
  ],
  "emails": [
    {
      "value": "exampleScimUser@evotix.com",
      "type": "work",
```

EVOTIX

```
      "primary": true
    },
    {
      "value": "exampleScim@User.org",
      "type": "home"
    }
  ],
  "addresses": [
    {
      "streetAddress": "100 Universal City Plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "USA",
      "formatted": "100 Universal City Plaza\nHollywood, CA 91608 USA",
      "type": "work",
      "primary": true
    },
    {
      "streetAddress": "456 Hollywood Blvd",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "USA",
      "formatted": "456 Hollywood Blvd\nHollywood, CA 91608 USA",
      "type": "home"
    }
  ],
  "phoneNumbers": [
    {
```

EVOTIX


```
      "value": "555-555-5555",
      "type": "work"
    },
    {
      "value": "555-555-4444",
      "type": "mobile"
    }
  ],
  "ims": [
    {
      "value": "someaimhandle",
      "type": "aim"
    }
  ],
  "photos": [
    {
      "value": "https://photos.example.com/profilephoto/72930000000Ccne/F",
      "type": "photo"
    },
    {
      "value": "https://photos.example.com/profilephoto/72930000000Ccne/T",
      "type": "thumbnail"
    }
  ],
  "userType": "Employee",
  "title": "Tour Guide",
  "preferredLanguage": "en-US",
  "locale": "en-US",
  "timezone": "CentralAmericaStandardTime",
  "active": true,
  "password": "t1meMa$heen",
  "groups": [
    {
```

EVOTIX


```
      "value": "e9e30dba-f08f-4109-8486-d5c6a331660a",
      "$ref": "../Groups/e9e30dba-f08f-4109-8486-d5c6a331660a",
      "display": "Tour Guides"
    },
    {
      "value": "fc348aa8-3835-40eb-a20b-c726e15c55b5",
      "$ref": "../Groups/fc348aa8-3835-40eb-a20b-c726e15c55b5",
      "display": "Employees"
    },
    {
      "value": "71ddacd2-a8e7-49b8-a5db-ae50d0a5bfd7",
      "$ref": "../Groups/71ddacd2-a8e7-49b8-a5db-ae50d0a5bfd7",
      "display": "US Employees"
    }
  ],
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "701984",
    "costCenter": "4130",
    "organization": "Universal Studios",
    "division": "Theme Park",
    "department": "trinExt",
    "manager": {
      "value": "26118915-6090-4610-87e4-49d8ca9f808d",
      "$ref": "../Users/26118915-6090-4610-87e4-49d8ca9f808d",
      "displayName": "John Smith"
    }
  },
  "meta": {
    "resourceType": "User",
```

```
120      "created": "2010-01-23T04:56:22Z",
121      "lastModified": "2011-05-13T04:42:34Z",
122      "version": "W/\"3694e05e9dff591\"",
123      "location": "https://example.com/v2/Users/2819c223-7f76-453a-919d-413861904646"
124    }
125  }
```

Create a SCIM user

The SCIM API allows the creation of users in Assure using the /scim/user API method with the POST verb. The users' details are supplied in the body of the API request using the User JSON object (see section above). Users are uniquely identified by their login username, which is the value in the userName field of the User JSON object. If there is no user record for the login username then the /scim/user API method will create the user using the details in the User JSON object. If the customer's licence limit has been reached, then attempting to create a user which would require another licence will result in an error response (HTTP error 400).

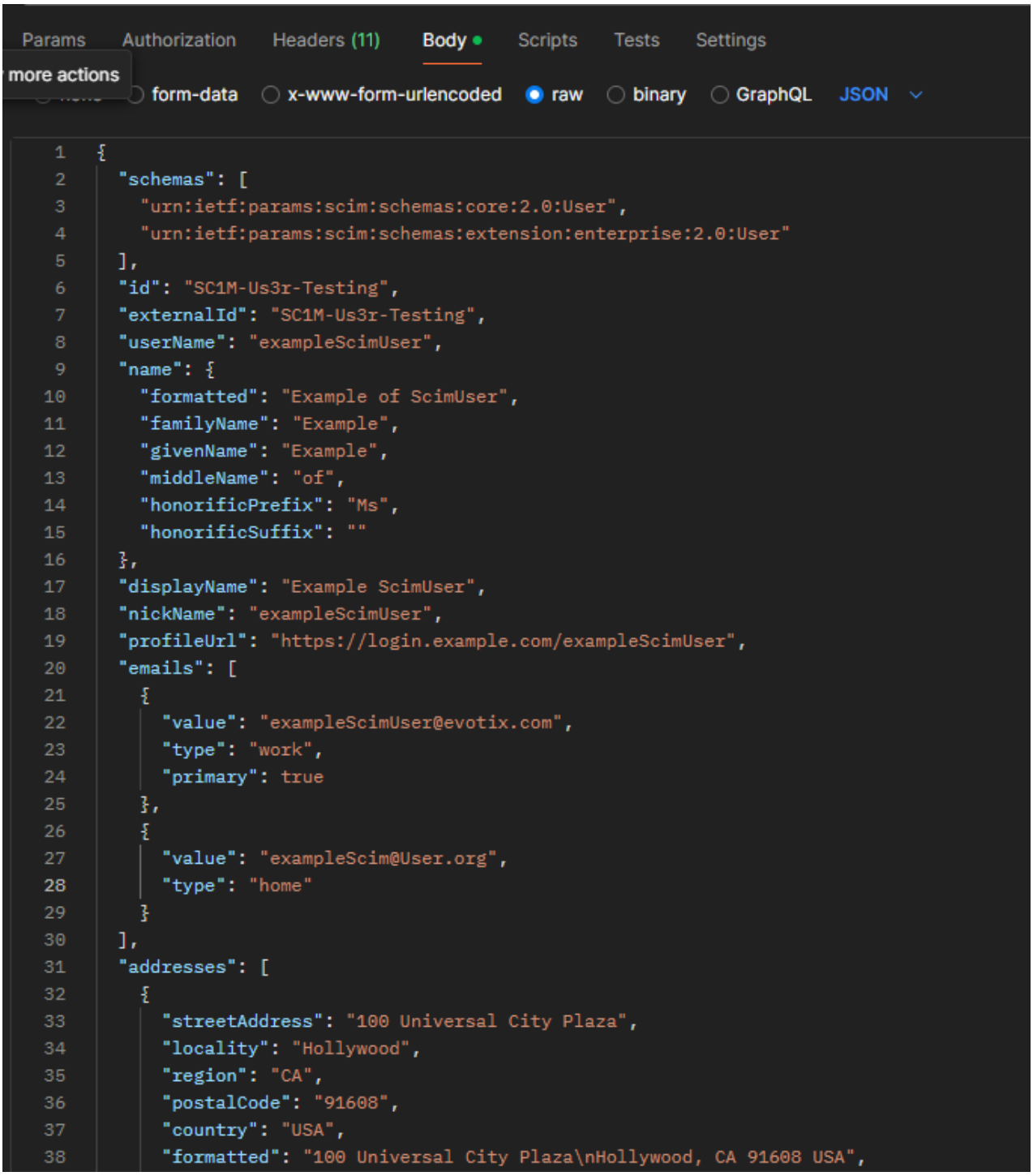
If there is an existing user record for the login username/email, then the /scim/user API method will update the existing user record to match the supplied details (see the 'Update a user' section below for details on the behaviour when updating an existing user).

The following table sections show how to create a user using the /v1/scim/user API method in a variety of software tools/platforms:

The screenshot below shows a successful 'create user' request for a customer.

A successful POST will contain a 201 and the responses body will have a copy of the record that was created.

EVOTIX



Retrieve a SCIM user with GET request

The SCIM API allows the retrieval of a SCIM User using the /scim/Users/{UserId} API method with the GET verb. Should a SCIM user be found with the supplied {UserId} then a JSON object will be

returned. Should a record not be found then the appropriate server response code, 404, will be returned. The {UserId} is the external Id of the user.

Note: There are three different types of Get that can be utilised against the SCIM API.

Get without query parameters

```
https://scim.elephant.she-development.net/v1/scim/Users/
```

This will return a list of all users for this customer (this includes users that are marked as not current) in a JSON array format.

Get with external Id

```
https://scim.elephant.she-development.net/v1/scim/Users/9d654e99-3df8-417d-a78f-9a8e05a11206
```

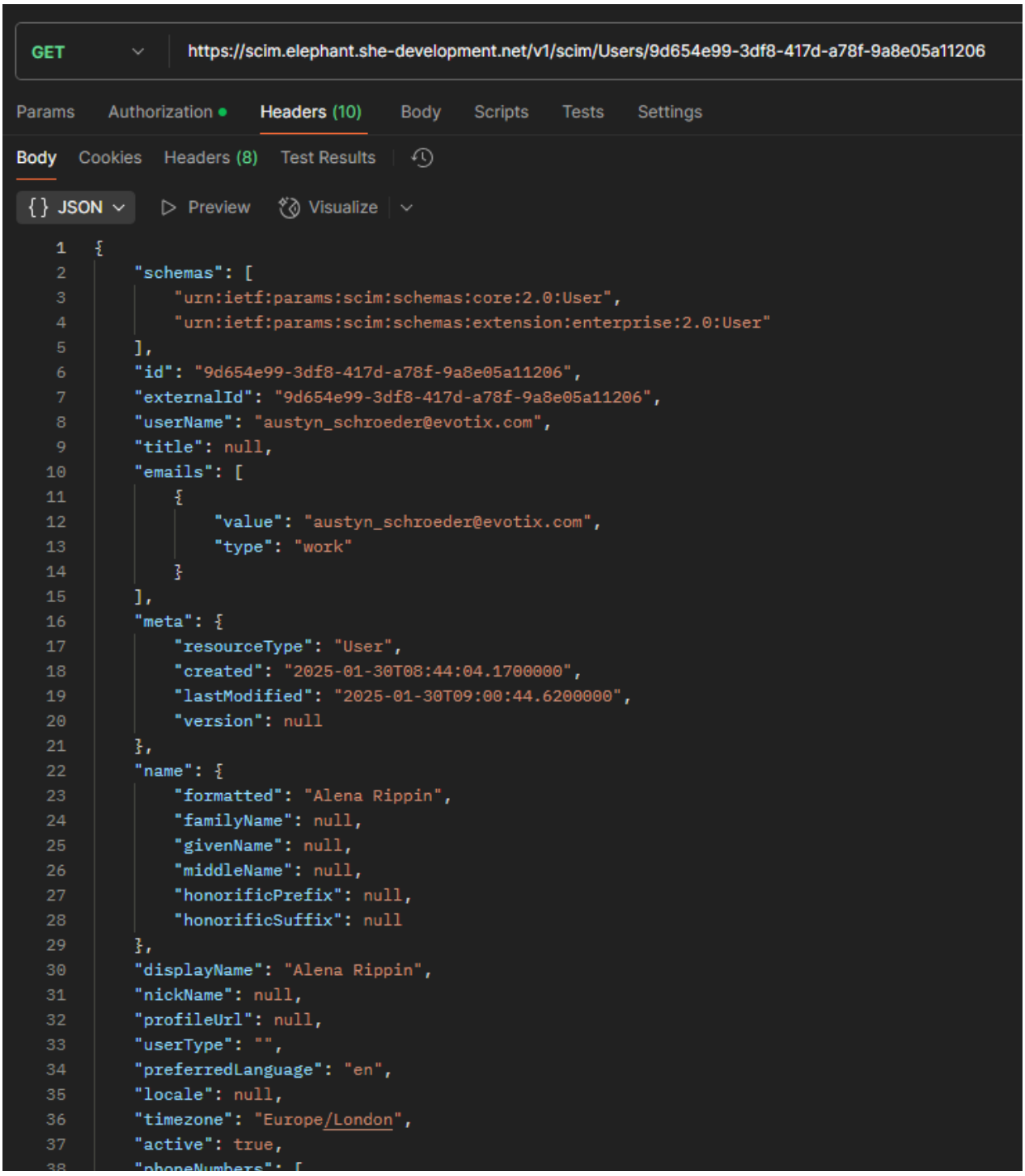
This will return a single user that matches the specified External Id

Get with query parameters

```
https://scim.elephant.she-development.net/v1/scim/Users?filter=userName%20eq%20%22localDevUser1%40evotix.com%22&startIndex=1&count=100
```

This will return a user that matches the filter.

The screenshot below shows a successful ‘retrieve SCIM user’ request for a customer on the uk2 stack (NB: the Postman setup is identical to that used for creating a SCIM user).



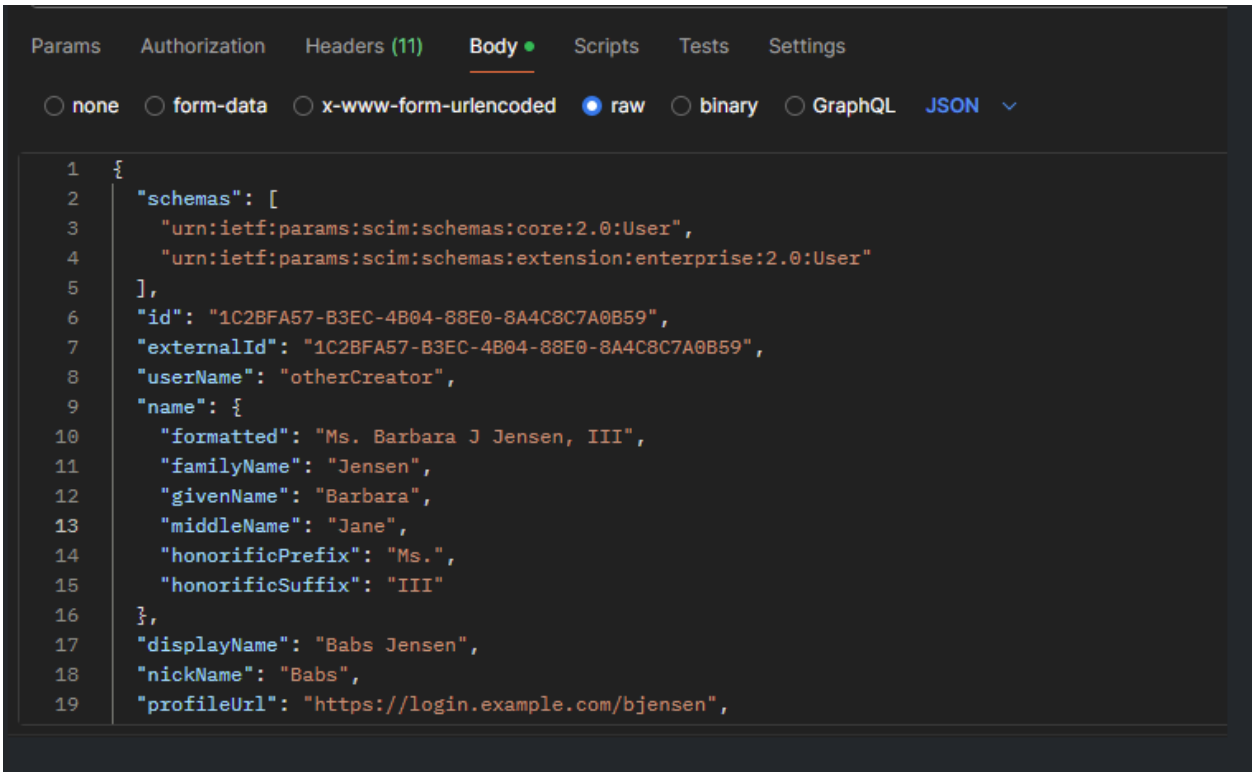
EVOTIX

Update a SCIM user with a PUT request

The SCIM API allows the updating of users in Assure using the /scim/Users/{userId} API method with the PUT verb. The users' details are supplied in the body of the API request using the User JSON object (see section above). Users are uniquely identified by their user ID as the externalID, which is the value in the userId or ExternalId fields of the User JSON object. If there is an existing user record for the userId then the /scim/Users/{userId} API method will update the existing user record to match the supplied details.

Partial updates are not supported with a PUT request. If you wish to perform a Partial Update, please see Partial update of a SCIM user record with PATCH request

The screenshot below shows a successful 'update SCIM user' request for a customer on one of our test stacks. The response section at the bottom shows the result Status: 200 acceptable which indicates that the user update was successful. The response will contain the user that has been updated in JSON format.



EVOTIX

Partial update of a SCIM user record with PATCH request

The SCIM API allows the partial update of person register records in Assure using the /v1/scim/Users/{UserId} API method with the PATCH verb. A PATCH request will only update the fields provided in the request and any other fields will remain unchanged.

The record that will be updated will be identified by its reference provided from the {UserId} path parameter. The user details to be updated are supplied in the body of the API request using the user JSON PATCH Operation object. If there is an existing user for the username provided, then the /v1/scim/Users/{UserId}API method will update the existing user with the supplied details. Note: If there is no user existing with the username that is provided in the request this will result in an error response (HTTP error 404).

Note. All examples shown below are for a partial update of the ‘email' field. All fields in the JSON PATCH object can also be used for a partial update.

The screenshot below shows a ‘partial update of a SCIM user’ request for a customer. The response section will show the result Status: 204 Acceptable which indicates that the SCIM user update was successful, the body of the response will also contain the updated User object.

Here is a representation of the object which in this transaction, contains two updates, one to ‘update’ the name of the User and one to ‘add’ an email address.

```
{
    "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:User",
    "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ],
     "Operations":[
        {
         "op":"replace",
         "path":"name",
         "value":
          {
            "formatted": "Joe Smith",
            "familyName": "Smith",
            "givenName": "Joe"
          }
        },
        {
         "op":"add",
         "path":"emails",
         "value":[
          {
            "value": "joesmith@evotix.com",
            "type": "work",
            "primary": true
          }
         ]
        }
    ]
}
```

Deactivate a user

N.B. The DELETE verb is not supported within the Assure SCIM API for Users.

The SCIM API allows the disabling of users in Assure using the /v1/scim/Users/{UserId} API method by using PATCH to alter current to false. The SCIM user to be disabled is identified by their external Id supplied via the {UserId} path parameter. The User Object is NOT required for this method.

If there is no user record for the external Id or the user is already disabled, then the /v1/scim/Users/{UserId}API method will still return a successful response (i.e. HTTP status code in the range 200-299).

There are some scenarios where you cannot deactivate SCIM users, these are the same as in the UI and an appropriate error message will be returned explaining the reason if this is the case.

These scenarios include:

- User has outstanding reviews or approvals
- User has an insights Designer License and owns an insights dashboard.

EVOTIX

Managing Roles via the SCIM API

SCIM Groups Endpoints

The SCIM Groups endpoints are used to create, update and delete roles within Assure. New roles will be created with all permissions set to “deny” by default.

Create a SCIM Group

The SCIM Groups API allows the creation of roles and the adding of those roles to users within Assure using the /scim/groups API method with the POST verb. The “displayName” and “externalId” will be used to create the role. The “members” array can contain a list of users that this role should be applied to; “value” is the externalId of the user and “displayname” is the named displayed on the user screen within Assure. If a member exists within Assure, then it will have the role added to it. If the member does not exist, then it will be ignored.

POST /scim/Groups

Below is an example Group request body that will create a new group call “Engineering Team” with 2 members.

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
  "displayName": "Engineering Team",
  "externalId": "4e3afc86-d9d5-422f-8342-689898097cb1",
  "members": [
    {
      "value": "UserId1",
      "display": "John Doe"
    },
    {
      "value": "UserId2",
      "display": "Jane Smith"
    }
  ]
}
```

A successful POST will contain a 201 and the responses body will have a copy of the record that was created.

Retrieve a SCIM Group

To retrieve a group by its ID, use the following GET request:

GET /scim/Groups/{GroupId}
Authorization: Bearer {token}

Update a SCIM Group

To update an existing group, use the following PUT request:

PUT /scim/Groups/{GroupId}

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
  "displayName": "Engineering Team",
  "members": [
    {
      "value": "UserId1",
      "display": "John Doe"
    },
    {
      "value": "UserId2",
      "display": "Jane Smith"
    }
  ]
}
```

Partial Update of a SCIM Group

To partially update a group, use the following PATCH request:

PATCH /scim/Groups/{GroupId}

```
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "replace",
      "path": "displayName",
```

EVOTIX

```
      "value": "New Engineering Team"
    }
  ]
}
```

Useful Tips

Below are some useful tips when setting up and using the SCIM API

1. For Assure GO+ users -> UserType to be "agoonly"
2. OrgUnitExternalId to be set using Department field in Enterprise data section
3. Manager can be set using Manager -> DisplayName field in Enterprise data section
4. IsManager = true can be set using Entitlements field and adding a value "manager" to the list
5. User language can be set using the same codes as Accept-Language HTTP header values (https://datatracker.ietf.org/doc/html/rfc7231#section-5.3.5)
6. User Timezone can be set using IANA Time Zone database format (https://datatracker.ietf.org/doc/html/rfc6557)
7. Employee Number must be provided since we need it to create a Person Register record with related Reference (EmployeeNumber = Reference)
8. UserName will be a combination of Given and Family names coming from SCIM.
9. If the preferredLanguage is not available in Assure (ie not added to customer in settings) then it will default to System default language (like in User API)
10. If Job title is sent through and does not exist, it will be added to the managed picklist.

Sample JSON Payloads

This section contains a collection of example request bodies for all of the SCIM endpoints supported by Assure.

Create a SCIM User
To create a new user with the enterprise department field, you can use the following POST request:
POST /scim/Users

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
  "userName": "jdoe",
  "name": {
    "givenName": "John",
    "familyName": "Doe"
  },
  "emails": [
    {
      "value": "jdoe@example.com",
      "type": "work",
      "primary": true
```

```
    }
  ],
  "enterprise": {
    "department": "Org Unit External Id"
  }
}
```

Retrieve a SCIM User
To retrieve a user by their ID, you can use the following GET request:
GET /scim/Users/{UserId}
Authorization: Bearer {token}

Note: UserId is the externalId of the User

Update a SCIM User
To update an existing user with the enterprise department field, you can use the following PUT request:
PUT /scim/Users/{UserId}

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
  "userName": "jdoe",
  "name": {
    "givenName": "John",
    "familyName": "Doe"
  },
  "emails": [
    {
      "value": "jdoe@example.com",
      "type": "work",
      "primary": true
    }
  ],
  "enterprise": {
    "department": " Org Unit External Id "
  }
}
```

Note: UserId is the external Id of the User

Partial Update of a SCIM User

To partially update a user with the enterprise department field, you can use the following PATCH request:
PATCH /scim/Users/{UserId}

```
{
  "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
  "Operations": [
    {
      "op": "replace",
      "path": "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:department",
      "value": "Engineering"
    }
  ]
}
```

Note: UserId is the external Id of the User

Create a SCIM Group
To create a new group, you can use the following POST request:
POST /scim/Groups

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
  "displayName": "Engineering Team",
  "members": [
    {
      "value": "UserId1",
      "display": "John Doe"
    },
    {
      "value": "UserId2",
      "display": "Jane Smith"
    }
  ]
}
```

Retrieve a SCIM Group
To retrieve a group by its ID, you can use the following GET request:
GET /scim/Groups/{GroupId}

Note: GroupId is the external Id of the group

EVOTIX

Update a SCIM Group
To update an existing group, you can use the following PUT request:
PUT /scim/Groups/{GroupId}

```
{
 "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
 "displayName": "Engineering Team",
 "members": [
  {
   "value": "UserId1",
   "display": "John Doe"
  },
  {
   "value": "UserId2",
   "display": "Jane Smith"
  }
 ]
}
```

Note: GroupId is the external Id of the group

EVOTIX

SCIM To Assure Mapping

This section contains a mapping of Assure fields to the equivalent fields in a SCIM request for a User or a Group.

User Record

Assure:
User Name
SCIM Property:
userName

Assure:
Full Name
SCIM Property:
name:formatted

Assure:
Email
SCIM Property:
emails[0].value
Email provided is always assumed to be of type primary and work.

Assure:
User Access Type

SCIM Property:
userType
Specify agoonly to set the user record to be AssureGo+ Only. If omitted a default of Assure and AssureGo+ will be applied.

Assure:
Is Current User
SCIM Property:
active

Assure:
Default Unit

SCIM Property: urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:department
Value must align with Assure Org Unit External Id

Assure:
Is Manager

SCIM Property:

```
"entitlements": [
    {
      "value": "manager"
    }
  ],
```

The entitlements attribute is an array
Assure:
Manager

SCIM Property: urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:manager:name
Text value of manager name.

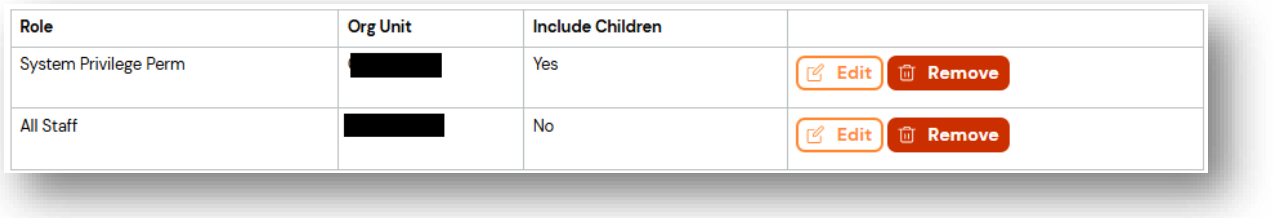
Assure:
User Specific Timezone
SCIM Property:
timezone

Assure:
Language
SCIM Property:
preferredLanguage

Assure:

Role

Assure:



Role	Org Unit	Include Children	
System Privilege Perm	[redacted]	Yes	Edit Remove
All Staff	[redacted]	No	Edit Remove

SCIM Property:

```
"roles": [
    {
      "value": "all-staff-id",
      "display": "All Staff"
    },
    {
      "value": "users-id",
      "display": "Users"
    }
  ],
```

Value attribute for groups should match with the external ID for the Assure Role. This is an array value.

Role Org Unit

This is taken from the users default. Include children will be defaulted to yes.

Person Record

Assure:

Org Unit

SCIM Property: urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:department

Value must align with Assure Org Unit External Id

Assure:

Reference

SCIM Property: urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:employeeNumber

Assure:

EVOTIX

Title
SCIM Property:
name: honorificPrefix

Assure:
Forename(s)
SCIM Property:
name:givenName

Assure:
Surname
SCIM Property:
name:familyName

Assure:
Job Title
SCIM Property:
Title

Assure:
Manager Name
SCIM Property:
urn:ietf:params:scim:schemas:extension:enterprise:2.0:User:manager:displayName

```
"urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "manager": {
      "displayName": "Joe Bloggs"
    }
  }
```

Assure:
Address Line 1
SCIM Property:
addresses[0].streetAddress

Assure:
Town
SCIM Property:
addresses[0].locality

Assure:

EVOTIX

County
SCIM Property:
addresses[0].region

Assure:
Post Code
SCIM Property:
addresses[0].postalCode

Assure:
Email
SCIM Property:
emails[0].value
Email provided is always assumed to be of type primary and work.

Role (SCIM Group)

Assure:
Name
SCIM Property:
displayName

Assure:
External ID
SCIM Property:
externalId