SCIM User Guide

5th April 2023

Stewart Burchell | Architect/Customer Success

Document Control

Version	Date	Author	Description
1.0	05 Apr 2023	Stewart Burchell	Initial version

Table of Contents

Introduction

Introduction: What is SCIM?

System for Cross-Domain Identity Management (SCIM) is an open standard protocol for automating the exchange of user identity information between identity domains and IT systems. SCIM ensures that employees added to the Human Capital Management (HCM) system automatically have accounts created in Azure Active Directory (Azure AD) or Windows Server Active Directory. User attributes and profiles are synchronised between the two systems, updating and removing users based on the user status or role change.

SCIM is a standardised definition of two endpoints: a /Users' endpoint and a /Groups endpoint. It uses common REST verbs to create, update, and delete objects. It also uses a predefined schema for common attributes like group name, username, first name, last name, and email. Applications that offer a SCIM 2.0 REST API can reduce or eliminate the pain of working with proprietary user management APIs or products. For example, any SCIM-compliant client can make an HTTP POST of a JSON object to the /Users endpoint to create a new user entry. Instead of needing a slightly different API for the same basic actions, apps that conform to the SCIM standard can instantly take advantage of pre-existing clients, tools, and code.

'SCIM' in Appian

Appian does not natively support SCIM, hence the custom application that this documentation refers to. The downloaded content contains **two** distinct Appian applications:

- SCIM User Management (SCIM)
    - This implements the /User and /Group service endpoints that implement the various operations that allow Users to be created / changed / deactivated / reactivated, and for User membership of Groups to be managed (Users added / removed)
- SCIM Test Application (SCIMTA)
    - This provides a set of test harnesses to flex the functionality exposed in the above application, as an alternative to using other testing tools such Postman or SOAP UI

Understanding the SCIM application

'SCIM User Management' exposes 5 operations (using Web APIs) that align with a subset of the HTTP verbs used in REST:

- POST: SCIMPostOperations
- PUT: SCIMPutOperations
- GET: SCIMGetOperations
- PATCH: SCIMPatchOperations
- DELETE: SCIMDeleteOperations

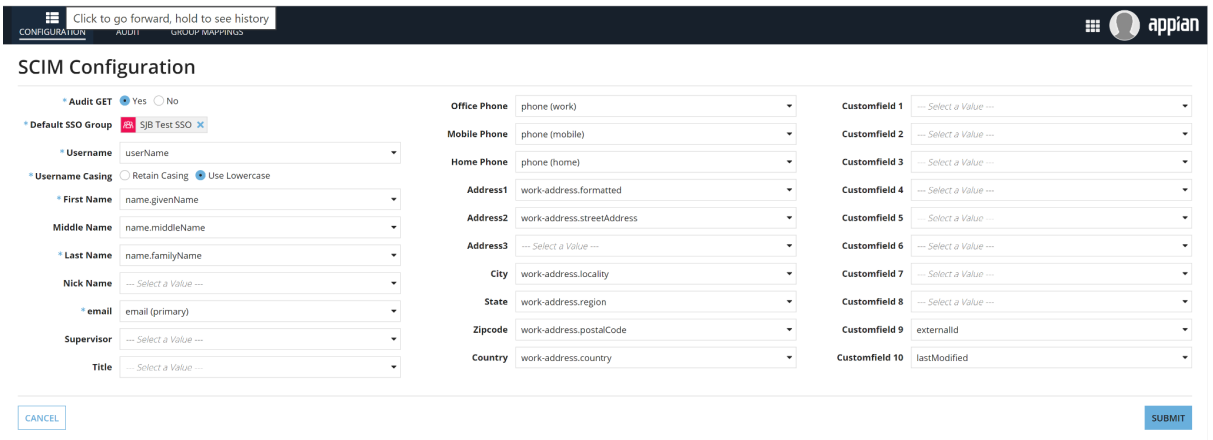
(there is a 6th - SCIM_InternalLoopback - that is used internally by the application which make provision for writing to the Audit tables for any GET calls that are made)

These 5 end-points are "base" endpoints and are used for both User and Group management.

User Management using the SCIM application

SCIM Configuration

An incoming SCIM Message that POSTs or PACTHes a User requires a mapping set up so that the attributes in the incoming message are directed to the correct User attributes:



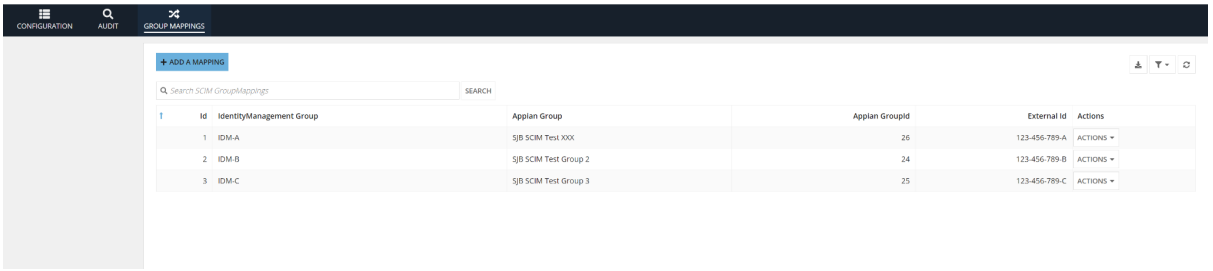
As a reminder, an Appian User objects has 4 required attributes:

- Username
- First Name
- Last Name
- Email address

All other attributes are optional but can be mapped using the above mapping user interface. In addition, this User Interface allows you to choose the Default SSO Group, and to switch on/off auditing for GET requests.

Group Membership Management using the SCIM application

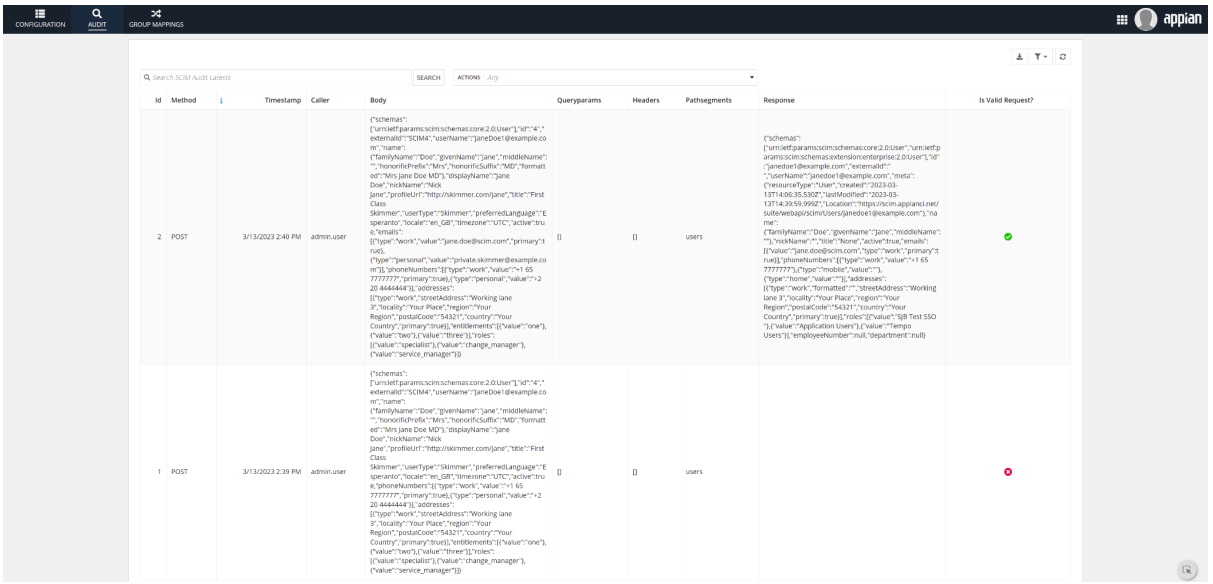
Group Membership is managed by PATCHing in/our Users from groups. Note that the Appian groups are **not** created on the fly (like Users). These need to pre-exist and they need to be mapped to the equivalent Identity Domains (such as Azure AD). The ‘Group Mappings’ tab in the ‘SCIM Configuration' site is where these mappings are managed:



Note that the Identity Domain group needs to have both its ‘Identity Management Group’ name and its ‘External Id’ (or ‘object name’) provided.

Audit

This is where you can see the Audit of all incoming SCIM messages:



The audit contains the Method used, a Timestamp, the Caller, and the Body/Queryparams/Headers/Pathsegments of the incoming message, plus the Response message. It also displays an indicator that shows if the incoming request was valid or not (Note: as an example a User’s email address is not a required attribute in Azure AD and so it is conceivable that an invalid request may be made to create a User in Appian by virtue of the email address being missing)