

ScimUsersController

September 5, 2026

ScimUsersController

Kind: Controller

Source: [atloria-monorepo/apps/api/src/scim/scim-users.controller.ts](#)

SCIM 2.0 Users endpoint. Org identity is injected by ScimAuthGuard from the bearer token (`@ScimOrgId()`), never read from the body/params.

SkipThrottle: Entra fans out the initial full sync as a burst of parallel requests that the default per-IP limit would 429.

`ScimUsersController` exposes the SCIM 2.0 `/Users` endpoint for provisioning and managing users from identity providers such as Microsoft Entra ID. It relies on `ScimAuthGuard` to authenticate bearer tokens and inject the organization identity through `@ScimOrgId()`, ensuring tenant scope is never taken from request bodies or route parameters. Throttling is skipped to support bursty parallel full-sync requests.

Diagram

```
graph LR
  IdP[Identity Provider<br/>Microsoft Entra ID] -->|SCIM 2.0 request + Bearer token| Guard[ScimAuthGuard]
  Guard -->|Authenticated organization ID| Controller[ScimUsersController]
  Controller -->|List / Create / Read / Update / Delete| Service[SCIM User Service]
  Service -->|Organization-scoped operations| Database[(Application Database)]
```

Usage

```
// Example: create a SCIM user through the SCIM 2.0 endpoint.
// The organization is determined by the bearer token, not this payload.

const response = await fetch("https://api.example.com/scim/v2/Users", {
  method: "POST",
  headers: {
    Authorization: "Bearer <scim-provisioning-token>",
    "Content-Type": "application/scim+json",
```

```

    },
    body: JSON.stringify({
      schemas: ["urn:ietf:params:scim:schemas:core:2.0:User"],
      userName: "jane.doe@example.com",
      name: {
        givenName: "Jane",
        familyName: "Doe",
      },
      active: true,
      emails: [
        {
          value: "jane.doe@example.com",
          primary: true,
        },
      ],
    }),
  });

  if (!response.ok) {
    throw new Error(`SCIM user provisioning failed: ${response.status}`);
  }

  const scimUser = await response.json();
  console.log(scimUser.id);

```

AI Coding Instructions

- Always obtain the tenant/organization context from `@ScimOrgId()` after `ScimAuthGuard`; never accept an organization ID from request params or request bodies.
- Preserve SCIM 2.0 response shapes, media types, schemas, and error semantics when adding or changing endpoints.
- Keep every user lookup and mutation organization-scoped to prevent cross-tenant access.
- Do not remove `SkipThrottle` behavior without accounting for identity-provider full-sync bursts and parallel provisioning requests.
- Coordinate controller changes with the SCIM service layer and identity-provider compatibility requirements, especially for PATCH and filter behavior.

Relationships

- MODULE_DECLARES → `list`
- MODULE_DECLARES → `get`
- MODULE_DECLARES → `create`

- MODULE_DECLARES → replace
- MODULE_DECLARES → patch
- MODULE_DECLARES → remove
- DEPENDS_ON → ScimService

Referenced By

- ScimModule (MODULE_DECLARES)