

ScimDiscoveryController

September 4, 2026

ScimDiscoveryController

Kind: Controller

Source: [atloria-monorepo/apps/api/src/scim/scim-discovery.controller.ts](https://github.com/atloria-monorepo/apps/api/src/scim/scim-discovery.controller.ts)

SCIM 2.0 discovery endpoints (RFC 7643 §5-6). Static — IdPs read these to learn our capabilities. We advertise: patch:true, filter:true, bulk:false, changePassword:false, sort:false, etag:false.

`ScimDiscoveryController` exposes the SCIM 2.0 discovery endpoints defined by RFC 7643 §§5–6. Identity providers use these static responses to determine supported SCIM capabilities, resource types, and schemas before provisioning users or groups. The controller advertises PATCH and filtering support while explicitly disabling bulk operations, password changes, sorting, and ETag support.

Diagram

```
graph LR
    IdP[Identity Provider] -->|GET /scim/ServiceProviderConfig| Discovery[ScimDiscoveryController]
    IdP -->|GET /scim/ResourceTypes| Discovery
    IdP -->|GET /scim/Schemas| Discovery

    Discovery --> Config[Service Provider Config<br/>patch: true<br/>filter: true]
    Discovery --> Types[Types[SCIM Resource Types]]
    Discovery --> Schemas[Schemas[SCIM Schema Definitions]]

    Config --> Provisioning[IdP Provisioning Configuration]
    Types --> Provisioning
    Schemas --> Provisioning
```

Usage

```
// Identity providers typically call these endpoints during SCIM setup.
const scimBaseUrl = 'https://api.example.com/scim';
const accessToken = process.env.SCIM_BEARER_TOKEN!;
```

```

const response = await fetch(
  `${scimBaseUrl}/ServiceProviderConfig`,
  {
    headers: {
      Authorization: `Bearer ${accessToken}`,
      Accept: 'application/scim+json',
    },
  },
);

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

const capabilities = await response.json();

console.log(capabilities.patch.supported); // true
console.log(capabilities.filter.supported); // true
console.log(capabilities.bulk.supported); // false
console.log(capabilities.sort.supported); // false

```

AI Coding Instructions

- Keep discovery responses aligned with the actual SCIM implementation; do not advertise a capability unless its corresponding endpoint behavior is supported.
- Preserve RFC 7643-compatible response shapes and SCIM media types, especially for `ServiceProviderConfig`, resource type, and schema responses.
- Treat these endpoints as static capability metadata: avoid adding tenant-specific state or provisioning side effects.
- When adding SCIM features such as bulk requests, sorting, ETags, or password changes, update the advertised capability flags and add matching endpoint support.
- Verify discovery routes remain accessible to configured IdPs using the same authentication and base-path conventions as other SCIM endpoints.

Relationships

- MODULE_DECLARES → `serviceProviderConfig`
- MODULE_DECLARES → `resourceTypes`
- MODULE_DECLARES → `schemas`

Referenced By

- `ScimModule` (MODULE_DECLARES)