ScimAdminController

September 4, 2026

ScimAdminController

Kind: Controller

Source: `atloria-monorepo/apps/api/src/scim/scim-admin.controller.ts`

Admin management of SCIM provisioning for an org. BUSINESS-plan-gated (402 upsell) + ADMIN-only + org-match. These are the JWT-authenticated companion endpoints to the token-authenticated /scim/v2 surface.

`ScimAdminController` provides JWT-authenticated administration endpoints for configuring SCIM provisioning within an organization. It enforces BUSINESS-plan access, ADMIN-only authorization, and organization ownership checks before allowing changes to SCIM provisioning settings or credentials. These endpoints complement the token-authenticated `/scim/v2` provisioning surface used by external identity providers.

Diagram

```
graph LR
  Admin[Organization Admin] -->|JWT request| Controller[ScimAdminController]
  Controller --> Auth[JWT Authentication]
  Auth --> Role[ADMIN Role Check]
  Role --> OrgMatch[Organization Match Check]
  OrgMatch --> Plan[BUSINESS Plan Gate]
  Plan -->|Allowed| ScimService[SCIM Admin Service]
  Plan -->|Not BUSINESS| Upsell[402 Upsell Response]
  ScimService --> Config[SCIM Configuration / Tokens]
  IdP[Identity Provider] -->|SCIM bearer token| ScimV2[/scim/v2 Provisioning API/]
```

Usage

```
type ScimAdminClientOptions = {
  apiBaseUrl: string;
  organizationId: string;
```

```
  accessToken: string;
};

async function getScimAdminConfiguration(options: ScimAdminClientOptions) {
  const response = await fetch(
    `${options.apiBaseUrl}/organizations/${options.organizationId}/scim`,
    {
      headers: {
        Authorization: `Bearer ${options.accessToken}`,
        Accept: 'application/json',
      },
    },
  );

  if (response.status === 402) {
    throw new Error('SCIM provisioning requires a BUSINESS plan.');
  }

  if (response.status === 403) {
    throw new Error('Only organization administrators can manage SCIM.');
  }

  if (!response.ok) {
    throw new Error(`Unable to load SCIM configuration: ${response.status}`);
  }

  return response.json();
}

// Call from an authenticated organization-admin settings page.
const scimConfig = await getScimAdminConfiguration({
  apiBaseUrl: 'https://api.example.com',
  organizationId: 'org_123',
  accessToken: userJwt,
});

console.log(scimConfig);
```

AI Coding Instructions

- Preserve the JWT, ADMIN-role, organization-match, and BUSINESS-plan guards on every administrative SCIM route.
- Return the existing `402` upsell behavior for organizations that are not on the BUSINESS plan; do not convert plan failures into generic authorization errors.
- Keep administrative configuration endpoints separate from the token-authenticated `/scim/v2` provisioning endpoints used by identity providers.

- Never return raw SCIM bearer tokens or other secrets after creation unless the API contract explicitly requires one-time secret display.
- Validate that the organization identifier in the route, JWT context, and requested resource all refer to the same organization before performing mutations.

Relationships

- MODULE_DECLARES → `status`
- MODULE_DECLARES → `listTokens`
- MODULE_DECLARES → `createToken`
- MODULE_DECLARES → `revokeToken`
- MODULE_DECLARES → `listMappings`
- MODULE_DECLARES → `createMapping`
- MODULE_DECLARES → `updateMapping`
- MODULE_DECLARES → `deleteMapping`
- DEPENDS_ON → `ScimAdminService`

Referenced By

- `ScimModule` (MODULE_DECLARES)