SamlController

September 4, 2026

SamlController

Kind: Controller

Source: `atloria-monorepo/apps/api/src/auth/saml/saml.controller.ts`

SAML SSO (A1). Config routes are ADMIN-only and BUSINESS-plan-gated; the login/callback/metadata routes are public (they face the IdP and the browser mid-login). Org identity comes from the JWT (config) or the URL (callback), never from a server session — the old flow lost the org because no session middleware was ever installed.

`SamlController` exposes the SAML SSO endpoints for configuring an organization's SAML integration and handling the browser/IdP login flow. Configuration routes require an authenticated ADMIN user on a BUSINESS plan, while login, callback, and metadata endpoints remain public and resolve organization identity from the URL rather than a server session.

Diagram

```
graph LR
  Admin[Organization Admin] -->|JWT-authenticated config request| Config[SamlController Config
  Config -->|ADMIN + BUSINESS plan checks| SamlService[SAML Service]

  Browser[User Browser] -->|Start SSO login with org URL| Login[SamlController Login Route]
  Login -->|SAML AuthnRequest| IdP[Identity Provider]
  IdP -->|SAML Response| Callback[SamlController Callback Route]
  Callback -->|Resolve org from URL| SamlService
  SamlService -->|Create authenticated app flow| Browser

  IdP -->|Fetch SP metadata| Metadata[SamlController Metadata Route]
```

Usage

```
// Configure SAML as an authenticated organization administrator.
// The API derives the organization from the administrator's JWT.
const response = await fetch("https://api.example.com/auth/saml/config", {
  method: "POST",
  headers: {
    Authorization: `Bearer ${adminJwt}`,
    "Content-Type": "application/json",
  },
  body: JSON.stringify({
    entryPoint: "https://idp.example.com/sso",
    issuer: "https://app.example.com",
    certificate: process.env.IDP_SIGNING_CERTIFICATE,
  }),
});

if (!response.ok) {
  throw new Error("Unable to configure SAML SSO");
}

// Redirect a user to the public organization-specific SAML login endpoint.
// The organization identifier must be present in the URL because no server
// session is used to preserve organization context during the IdP redirect.
window.location.assign(
  "https://api.example.com/auth/saml/acme-corp/login",
);
```

AI Coding Instructions

- Keep configuration endpoints protected by JWT authentication, ADMIN authorization, and BUSINESS-plan gating.
- Keep SAML login, callback, and metadata endpoints public because they must be reachable by browsers and external identity providers.
- Preserve organization context through route parameters or URL state during the SAML redirect flow; do not rely on server-side session middleware.
- Validate and safely handle SAML assertions through the SAML service layer rather than placing protocol parsing or verification logic in the controller.

Relationships

- MODULE_DECLARES → `getConfig`
- MODULE_DECLARES → `updateConfig`
- MODULE_DECLARES → `importMetadata`
- MODULE_DECLARES → `toggle`

- MODULE_DECLARES → `metadata`
- MODULE_DECLARES → `login`
- MODULE_DECLARES → `callback`
- DEPENDS_ON → `SamlService`
- DEPENDS_ON → `SsoCodeService`
- DEPENDS_ON → `AuthService`
- DEPENDS_ON → `ReaderTokenService`
- DEPENDS_ON → `PrismaService`
- DEPENDS_ON → `AuditService`

Referenced By

- `SamlModule` (MODULE_DECLARES)