

PlatformController

September 4, 2026

PlatformController

Kind: Controller

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

Cross-org platform-admin console. EVERY route is gated by both JwtAuthGuard and PlatformAdminGuard (fail-closed, DB-verified). This is the ONLY controller that operates across tenants; it never uses ResourceOrgGuard and takes an explicit :orgId. See tasks/platform-admin-spec.md.

PlatformController provides the cross-organization platform-admin API surface for managing or inspecting tenant-specific platform data. Every route requires both JwtAuthGuard and PlatformAdminGuard, ensuring requests are authenticated and verified against the database as platform administrators before an explicit :orgId tenant context is accessed.

Diagram

```
graph LR
  Client[Platform Admin Client] --> JWT[JwtAuthGuard]
  JWT --> Admin[PlatformAdminGuard<br/>DB-verified]
  Admin --> Controller[PlatformController]
  Controller --> OrgParam[Explicit :orgId]
  OrgParam --> PlatformServices[Platform Services]
  PlatformServices --> TenantData[(Target Organization Data)]

  JWT -. unauthenticated .-> Deny401[401 Unauthorized]
  Admin -. not platform admin .-> Deny403[403 Forbidden]
```

Usage

```
// Example platform-admin API client call.
// The authenticated user must have platform-admin access, and the target
// organization is always provided explicitly in the request path.
```

```

async function getPlatformOrganization(
  apiBaseUrl: string,
  accessToken: string,
  orgId: string,
) {
  const response = await fetch(
    `${apiBaseUrl}/platform/organizations/${encodeURIComponent(orgId)}`,
    {
      headers: {
        Authorization: `Bearer ${accessToken}`,
        Accept: 'application/json',
      },
    },
  );

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

  return response.json();
}

```

AI Coding Instructions

- Keep `JwtAuthGuard` and `PlatformAdminGuard` applied to every platform route; this controller must fail closed when platform-admin verification is unavailable or unsuccessful.
- Require an explicit `:orgId` for tenant-targeted operations rather than deriving organization context from the authenticated user.
- Do not add `ResourceOrgGuard` to this controller: platform administration intentionally operates across tenant boundaries after platform-admin authorization succeeds.
- Validate and use the requested organization ID consistently in service calls, audit logging, and error handling to prevent cross-tenant ambiguity.
- Keep platform-only functionality isolated here; regular organization-scoped endpoints should remain in tenant controllers guarded by normal resource-organization authorization.

Relationships

- `MODULE_DECLARES` → `listOrgs`
- `MODULE_DECLARES` → `getOrg`
- `MODULE_DECLARES` → `setPlan`

- MODULE_DECLARES → getSso
- MODULE_DECLARES → repairSso
- MODULE_DECLARES → disableSso
- DEPENDS_ON → PlatformService

Referenced By

- PlatformModule (MODULE_DECLARES)