

DeskController

September 4, 2026

DeskController

Kind: Controller

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

Support desk: Plain-style queue over the existing ticket store. Authed + org-scoped (service-level organizationId check, like manuals-insights).

`DeskController` exposes authenticated, organization-scoped support desk endpoints backed by the existing ticket store. It provides a plain queue-style interface for listing and managing support tickets while delegating organization access validation and business logic to the service layer.

Diagram

```
graph LR
  Client[Authenticated Client] --> Guard[Authentication Guard]
  Guard --> Controller[DeskController]
  Controller --> Service[Desk Service]
  Service --> OrgCheck[Organization ID Validation]
  OrgCheck --> TicketStore[Existing Ticket Store]
  TicketStore --> Service
  Service --> Controller
  Controller --> Client
```

Usage

```
import { Controller, Get, Param, Req } from '@nestjs/common';
import { DeskService } from '../desk.service';

@Controller('desk')
export class DeskController {
  constructor(private readonly deskService: DeskService) {}

  @Get('tickets')
```

```

    async listTickets(@Req() request: { user: { organizationId: string } }) {
        return this.deskService.listTickets({
            organizationId: request.user.organizationId,
        });
    }

    @Get('tickets/:ticketId')
    async getTicket(
        @Req() request: { user: { organizationId: string } },
        @Param('ticketId') ticketId: string,
    ) {
        return this.deskService.getTicket({
            ticketId,
            organizationId: request.user.organizationId,
        });
    }
}

```

AI Coding Instructions

- Require authenticated requests for every desk endpoint; derive the organization context from the authenticated user rather than trusting a client-provided organization ID.
- Keep `DeskController` thin: parse route parameters and request context, then delegate ticket operations to the service layer.
- Enforce organization ownership in the service layer for every ticket lookup or mutation, following the same pattern used by `manuals-insights`.
- Reuse the existing ticket store and ticket domain models; do not introduce a parallel persistence model for desk queue data.
- Return consistent API errors for missing tickets, invalid organization access, and unauthorized requests.

Relationships

- MODULE_DECLARES → `listThreads`
- MODULE_DECLARES → `counts`
- MODULE_DECLARES → `getThread`
- MODULE_DECLARES → `transition`
- MODULE_DECLARES → `reply`
- MODULE_DECLARES → `duplicate`
- DEPENDS_ON → `DeskService`

Referenced By

- `IssuesModule` (MODULE_DECLARES)