

AudienceService

September 4, 2026

AudienceService

Kind: Service**Source:** [atloria-monorepo/apps/api/src/audience/audience.service.ts](https://github.com/atloria-monorepo/apps/api/src/audience/audience.service.ts)

`AudienceService` is a NestJS backend service responsible for managing audience records within the API. It provides CRUD operations for audiences and includes a seeding method for creating the system's standard audience definitions.

Methods

Method	Signature	Returns	Description
<code>list</code>	<code>list(user: JwtPayload)</code>	<code>Promise<AudienceModel[]></code>	List all audiences for the user's organization Returns standard audiences + organization-specific audiences
<code>get</code>	<code>get(id: string, user: JwtPayload)</code>	<code>Promise<AudienceModel></code>	Get a single audience by ID
<code>create</code>	<code>create(dto: CreateAudienceDto, user: JwtPayload)</code>	<code>Promise<AudienceModel></code>	Create a new organization-specific audience
<code>update</code>	<code>update(id: string, dto: UpdateAudienceDto, user: JwtPayload)</code>	<code>Promise<AudienceModel></code>	Update an existing audience Standard

Method	Signature	Returns	Description
			audiences cannot be modified
<code>delete</code>	<code>delete(id: string, user: JwtPayload)</code>	<code>Promise<void></code>	Delete an audience Standard audiences cannot be deleted
<code>seedStandardAudiences</code>	<code>seedStandardAudiences()</code>	<code>Promise<void></code>	Seed standard audiences (called during application bootstrap)

Dependencies

- `PrismaService`

Where it refuses work

- `AudienceService` stops the work with `ConflictException` when `existing` , in 2 places.
- `AudienceService` stops the work with `NotFoundException` when `!audience` .
- `AudienceService` stops the work with `ForbiddenException` when `!audience.isStandard` && `audience.organizationId !== user.organizationId` — “Access denied to this audience”.
- `AudienceService` stops the work with `ForbiddenException` when `audience.isStandard` — “Standard audiences cannot be modified”.
- `AudienceService` stops the work with `ForbiddenException` when `audience.isStandard` — “Standard audiences cannot be deleted”.

Diagram

```
sequenceDiagram
    participant Client
    participant Controller as AudienceController
    participant Service as AudienceService
    participant Database
```

```
Client->>Controller: Request audience operation
Controller->>Service: list/get/create/update/delete
Service->>Database: Query or mutate audience data
Database-->>Service: Audience data/result
Service-->>Controller: AudienceModel or void
Controller-->>Client: HTTP response
```

Note over Service,Database: seedStandardAudiences creates default audiences

Usage

```
import { Injectable } from '@nestjs/common';
import { AudienceService } from './audience.service';

@Injectable()
export class CampaignService {
  constructor(private readonly audienceService: AudienceService) {}

  async createCampaignAudience() {
    const audiences = await this.audienceService.list();

    if (audiences.length === 0) {
      await this.audienceService.seedStandardAudiences();
    }

    const audience = await this.audienceService.create({
      name: 'Newsletter Subscribers',
      description: 'Users subscribed to the product newsletter',
    });

    return audience;
  }
}
```

AI Coding Instructions

- Inject `AudienceService` through NestJS constructor injection; do not instantiate it directly.
- Use `list`, `get`, `create`, `update`, and `delete` as the single service-layer entry points for audience persistence operations.
- Call `seedStandardAudiences()` only in controlled initialization, setup, or administrative flows; ensure the seeding implementation remains safe to run repeatedly.
- Preserve `AudienceModel` return types when extending service methods or adding controller integrations.

- Validate request DTOs in the controller or validation layer before passing create or update input to the service.

Relationships

- `DEPENDS_ON` → `PrismaService`

Referenced By

- `AudienceController` (`DEPENDS_ON`)
- `AudienceModule` (`MODULE_PROVIDES`)
- `AudienceModule` (`MODULE_EXPORTS`)