AssetsService

September 4, 2026

AssetsService

Kind: Service

Source: `atloria-monorepo/apps/api/src/assets/assets.service.ts`

`AssetsService` manages asset lifecycle operations in the API, including image and video uploads, asset retrieval, listing, and deletion. It acts as the backend integration point between application features that need media storage and the underlying asset persistence or storage provider.

Methods

Method	Signature	Returns	Description
`uploadImage`	`uploadImage(options: UploadImageOptions)`	`unknown`	
`uploadVideo`	`uploadVideo(options: UploadImageOptions)`	`unknown`	D1: upload a narrated-video artifact (mp4/webm) or its WebVTT captions.
`getAsset`	`getAsset(id: string, organizationId: string)`	`unknown`	
`deleteAsset`	`deleteAsset(id: string, organizationId: string, _userId: string)`	`unknown`	
`listAssets`	`listAssets(organizationId: string, audienceIds: string[])`	`unknown`	

Dependencies

- `PrismaService`
- `AzureBlobService`

Where it refuses work

- `AssetsService` stops the work with `BadRequestException` when `!asset` — “Asset not found”, in 2 places.
- `AssetsService` stops the work with `BadRequestException` when `!allowedMimeTypes.includes(file.mimetype)` — “Invalid file type. Only JPEG, PNG, GIF, and WebP images are allowed.”.
- `AssetsService` stops the work with `BadRequestException` when `file.size > maxSize` — “File size exceeds maximum allowed size of 10MB.”.
- `AssetsService` stops the work with `BadRequestException` when `!ext` — “Invalid file type. Only MP4/WebM video and WebVTT captions are allowed.”.
- `AssetsService` stops the work with `BadRequestException` when `file.size > maxSize` — “File size exceeds maximum allowed size of 200MB.”.

Diagram

```
sequenceDiagram
    participant Client
    participant Controller as Assets Controller
    participant Service as AssetsService
    participant Storage as Asset Storage
    participant Database as Asset Repository

    Client->>Controller: Upload image/video request
    Controller->>Service: uploadImage(file) / uploadVideo(file)
    Service->>Storage: Store media file
    Storage-->>Service: Storage location and metadata
    Service->>Database: Create asset record
    Database-->>Service: Asset record
    Service-->>Controller: Uploaded asset

    Client->>Controller: Get, list, or delete assets
    Controller->>Service: getAsset(), listAssets(), deleteAsset()
    Service->>Database: Read or remove asset metadata
    Service->>Storage: Retrieve or delete stored file
    Service-->>Controller: Asset result or confirmation
```

Usage

```
import { Injectable } from '@nestjs/common';
import { AssetsService } from './assets/assets.service';

@Injectable()
export class ProfileService {
```

```
  constructor(private readonly assetsService: AssetsService) {}

  async updateAvatar(file: Express.Multer.File) {
    const asset = await this.assetsService.uploadImage(file);

    return {
      avatarAsset: asset,
    };
  }

  async removeAvatar(assetId: string) {
    await this.assetsService.deleteAsset(assetId);

    return { deleted: true };
  }

  async getMediaLibrary() {
    return this.assetsService.listAssets();
  }
}
```

AI Coding Instructions

- Keep image and video upload handling separated by using `uploadImage()` and `uploadVideo()` according to the incoming file type.
- Validate file metadata, MIME type, size, and authorization before delegating uploads to this service.
- When deleting an asset, ensure both the persisted asset record and its backing storage object are removed or handled consistently.
- Use `getAsset()` for single-resource access checks rather than relying on client-provided storage URLs or metadata.
- Preserve asset identifiers returned by upload operations so related domain entities can reference assets without duplicating file data.

Relationships

- DEPENDS_ON → `PrismaService`
- DEPENDS_ON → `AzureBlobService`

Referenced By

- `AssetsController` (DEPENDS_ON)
- `AssetsModule` (MODULE_PROVIDES)

- `AssetsModule` (MODULE_EXPORTS)
- `DocAutomationService` (DEPENDS_ON)
- `ScreenshotService` (DEPENDS_ON)