

ActivityService

September 4, 2026

ActivityService

Kind: Service

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

`ActivityService` is a NestJS backend service responsible for recording, querying, and clearing user activity data. It supports activity feeds for documents and projects, including recently accessed projects/documents and project-scoped document history.

Methods

Method	Signature	Returns	Description
<code>log</code>	<code>log(dto: CreateActivityDto)</code>	<code>Promise<void></code>	Log an activity entry
<code>getActivities</code>	<code>getActivities(user: JwtPayload, filters: ActivityFiltersDto)</code>	<code>unknown</code>	Get activities with filters.
<code>getDocumentActivities</code>	<code>getDocumentActivities(user: JwtPayload, documentId: string)</code>	<code>unknown</code>	Get activities for a specific document (scoped to the caller's organization)
<code>getProjectActivities</code>	<code>getProjectActivities(user: JwtPayload, projectId: string)</code>	<code>unknown</code>	Get activities for a specific project (scoped to the caller's organization)
<code>trackActivity</code>	<code>`trackActivity(userId: string, entityType: 'PROJECT'</code>	<code>'DOCUMENT', entityId: string, actionType: 'VIEW'</code>	<code>'EDIT'</code>
<code>getRecentProjects</code>	<code>getRecentProjects(userId: string, limit: unknown)</code>	<code>unknown</code>	Get recent projects for a

Method	Signature	Returns	Description
			user
getRecentDocuments	getRecentDocuments(userId: string, limit: unknown)	unknown	Get recent documents for a user
getRecentDocumentsByProject	getRecentDocumentsByProject(userId: string, projectId: string, limit: unknown)	unknown	Get recent documents filtered by project
clearActivity	`clearActivity(userId: string, entityType: 'PROJECT')`	'DOCUMENT')`	unknown

Dependencies

- PrismaService

When something fails

- ActivityService handles failure in 1 place: it logs it and continues in all 1.

Diagram

```
sequenceDiagram
    participant Client
    participant Controller
    participant ActivityService
    participant ActivityStore

    Client->>Controller: Request activity or open resource
    Controller->>ActivityService: trackActivity(activity)
    ActivityService->>ActivityStore: Persist activity event
    ActivityStore-->>ActivityService: Saved activity

    Client->>Controller: Get recent documents/projects
    Controller->>ActivityService: getRecentDocuments() / getRecentProjects()
    ActivityService->>ActivityStore: Query activity history
    ActivityStore-->>ActivityService: Activity records
    ActivityService-->>Controller: Filtered activity results
    Controller-->>Client: Activity response
```

Usage

```
import { Injectable } from '@nestjs/common';
import { ActivityService } from './activity.service';
```

```

@Injectable()
export class DocumentAccessService {
  constructor(private readonly activityService: ActivityService) {}

  async openDocument(userId: string, documentId: string, projectId: string) {
    // Load and return the document through the application's document service.

    await this.activityService.trackActivity({
      userId,
      documentId,
      projectId,
      type: 'document_opened',
    });

    return { documentId, projectId };
  }

  async getDashboardActivity(userId: string) {
    const [recentProjects, recentDocuments] = await Promise.all([
      this.activityService.getRecentProjects(userId),
      this.activityService.getRecentDocuments(userId),
    ]);

    return {
      recentProjects,
      recentDocuments,
    };
  }
}

```

AI Coding Instructions

- Inject `ActivityService` through NestJS dependency injection; do not instantiate it directly.
- Call `trackActivity()` after successful document or project access, not before authorization or resource loading completes.
- Use the scoped query methods such as `getDocumentActivities()` and `getRecentDocumentsByProject()` when rendering resource-specific activity views.
- Preserve user and resource ownership boundaries when adding activity queries; activity history must be filtered by the authenticated user.
- Use `clearActivity()` only for explicit user-driven cleanup flows and ensure it targets the intended user or activity scope.

Relationships

- `DEPENDS_ON` → `PrismaService`

Referenced By

- `ActivityController` (DEPENDS_ON)
- `ActivityModule` (MODULE_PROVIDES)
- `ActivityModule` (MODULE_EXPORTS)
- `CommentService` (DEPENDS_ON)
- `SuggestionService` (DEPENDS_ON)