**📁 audit/ – Audit Logging**

**Purpose**

The audit/ directory is dedicated to tracking significant events within the application, such as user actions or system changes. This is crucial for accountability, compliance, and debugging

**Directory Structure**



src/

├── audit/

│ ├── audit-log.module.ts # Module encapsulating audit logging components

│ ├── audit-log.service.ts # Service handling audit log logic

│ ├── audit-log.interceptor.ts # Interceptor capturing audit logs

│ └── audit-log.decorator.ts # Custom decorator for audit logging

```

### Implementation

\*\*1. `audit-log.decorator.ts`\*\*

Defines a custom decorator to mark methods that require audit logging



```typescript

import { SetMetadata } from '@nestjs/common';

export const AUDIT\_LOG\_KEY = 'auditLog';

export const AuditLog = (action: string) => SetMetadata(AUDIT\_LOG\_KEY, action);

```

\*\*2. `audit-log.interceptor.ts`\*\*

Intercepts requests to methods decorated with `@AuditLog` and logs the specified action



```typescript

import {

Injectable,

NestInterceptor,

ExecutionContext,

CallHandler,

} from '@nestjs/common';

import { Reflector } from '@nestjs/core';

import { Observable, tap } from 'rxjs';

import { AuditLogService } from './audit-log.service';

import { AUDIT\_LOG\_KEY } from './audit-log.decorator';

@Injectable()

export class AuditLogInterceptor implements NestInterceptor {

constructor(

private readonly reflector: Reflector,

private readonly auditLogService: AuditLogService,

) {}

intercept(context: ExecutionContext, next: CallHandler): Observable<any> {

const action = this.reflector.get<string>(

AUDIT\_LOG\_KEY,

context.getHandler(),

);

if (!action) {

return next.handle();

}

const request = context.switchToHttp().getRequest();

const userId = request.user?.id;

return next.handle().pipe(

tap(() => {

this.auditLogService.logAction({

userId,

action,

timestamp: new Date(),

});

}),

);

}

}

```

\*\*3. `audit-log.service.ts`\*\*

Handles the logic for storing audit logs, such as saving to a database or external logging service



```typescript

import { Injectable } from '@nestjs/common';

@Injectable()

export class AuditLogService {

logAction(log: { userId: string; action: string; timestamp: Date }) {

// Implement logic to store the audit log

console.log('Audit Log:', log);

}

}

```

\*\*4. `audit-log.module.ts`\*\*

Registers the audit log service and interceptor



```typescript

import { Module } from '@nestjs/common';

import { APP\_INTERCEPTOR } from '@nestjs/core';

import { AuditLogService } from './audit-log.service';

import { AuditLogInterceptor } from './audit-log.interceptor';

@Module({

providers: [

AuditLogService,

{

provide: APP\_INTERCEPTOR,

useClass: AuditLogInterceptor,

},

],

})

export class AuditLogModule {}

```

### Usage Example



```typescript

import { Controller, Post } from '@nestjs/common';

import { AuditLog } from './audit/audit-log.decorator';

@Controller('tasks')

export class TaskController {

@Post()

@AuditLog('Create Task')

createTask() {

// Task creation logic

}

}

```

---