

JwtAuthGuard

September 4, 2026

JwtAuthGuard

Kind: Service

Source: [atloria-monorepo/apps/api/src/auth/guards/jwt-auth.guard.ts](https://github.com/atloria-monorepo/apps/api/src/auth/guards/jwt-auth.guard.ts)

`JwtAuthGuard` is a NestJS authentication guard that protects routes by validating JWT-based requests before they reach controllers. It integrates with the application's authentication strategy to populate the authenticated user context or reject unauthorized requests.

Methods

Method	Signature	Returns
<code>canActivate</code>	<code>canActivate(context: ExecutionContext)</code>	<code>unknown</code>

Dependencies

- `Reflector`

Where it refuses work

- `JwtAuthGuard` stops the work with an early return when `isPublic`.

Diagram

```
sequenceDiagram
    participant Client
    participant Guard as JwtAuthGuard
    participant Strategy as JWT Strategy
    participant Controller

    Client->>Guard: Request with Authorization: Bearer <token>
    Guard->>Strategy: canActivate()
    Strategy->>Strategy: Extract and validate JWT
```

```

alt Valid token
  Strategy-->>Guard: Authenticated user
  Guard->>Controller: Allow request
  Controller-->>Client: Protected response
else Missing or invalid token
  Strategy-->>Guard: Unauthorized
  Guard-->>Client: 401 Unauthorized
end

```

Usage

```

import { Controller, Get, UseGuards } from '@nestjs/common';
import { JwtAuthGuard } from '../auth/guards/jwt-auth.guard';

@Controller('profile')
export class ProfileController {
  @Get()
  @UseGuards(JwtAuthGuard)
  getProfile() {
    return {
      message: 'This endpoint requires a valid JWT.',
    };
  }
}

```

AI Coding Instructions

- Apply `@UseGuards(JwtAuthGuard)` to controllers or route handlers that require authenticated users.
- Keep JWT extraction and validation behavior aligned with the configured NestJS JWT/passport strategy.
- Do not bypass the guard for protected endpoints; create explicit public-route handling when unauthenticated access is required.
- Ensure clients send tokens using the `Authorization: Bearer <token>` header format.
- When changing `canActivate()`, preserve NestJS guard return semantics (`boolean`, `Promise<boolean>`, or `Observable<boolean>`).

Relationships

- `DEPENDS_ON` → `reflector`

Referenced By

- `AuthModule` (MODULE_PROVIDES)
- `AuthModule` (MODULE_EXPORTS)