CanActivate

August 18, 2026

CanActivate

Kind: Interface

Source: `packages/common/interfaces/features/can-activate.interface.ts`

Part of: Common

Interface defining the `canActivate()` function that must be implemented by a guard. Return value indicates whether or not the current request is allowed to proceed. Return can be either synchronous (`boolean`) or asynchronous (`Promise` or `Observable`).

`CanActivate` defines the contract for guards that determine whether an incoming request may continue to a route handler. Implementations inspect the current `ExecutionContext` and return `true` or `false` , either synchronously or through a `Promise` or `Observable` .

Diagram

```
graph LR
  Request[Incoming Request] --> Guard[CanActivate Guard]
  Guard --> Context[ExecutionContext]
  Context --> Guard
  Guard --> Decision{canActivate() result}
  Decision -->|true| Handler[Route Handler]
  Decision -->|false| Denied[Request Denied]
```

Usage

```
import {
  CanActivate,
  ExecutionContext,
  Injectable,
} from '@nestjs/common';
```

```ts
@Injectable()
export class ApiKeyGuard implements CanActivate {
  canActivate(context: ExecutionContext): boolean {
    const request = context.switchToHttp().getRequest();
    const apiKey = request.headers['x-api-key'];

    return apiKey === process.env.API_KEY;
  }
}

// Apply the guard to a controller or route:
// @UseGuards(ApiKeyGuard)
```

AI Coding Instructions

- Implement `canActivate(context: ExecutionContext)` on every guard that uses this interface.
- Return a `boolean`, `Promise<boolean>`, or `Observable<boolean>`; do not return response objects or throw for ordinary authorization failures.
- Use `ExecutionContext` to access the correct transport context, such as `switchToHttp()`, `switchToWs()`, or `switchToRpc()`.
- Return `false` to deny access; Nest handles the resulting forbidden response unless custom exception behavior is required.
- Register guards with `@UseGuards()` or configure them as global guards through the application provider setup.

Used by

22 references from 22 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (22)

- `ParamProperties` — `packages/core/router/router-execution-context.ts`:50
- `AuthGuard` — `integration/graphql-code-first/src/common/guards/auth.guard.ts`:9
- `CatsGuard` — `integration/graphql-schema-first/src/cats/cats.guard.ts`:4
- `Guard` — `integration/inspector/src/circular-hello/guards/request-scoped.guard.ts`:9
- `RolesGuard` — `integration/inspector/src/common/guards/roles.guard.ts`:4
- `Guard` — `integration/scopes/src/circular-hello/guards/request-scoped.guard.ts`:9
- `Guard` — `integration/scopes/src/circular-transient/guards/request-scoped.guard.ts`:9

- `DurableGuard` — `integration/scopes/src/durable/durable.guard.ts`:9

…and 14 more.