

Session

August 17, 2026

Session

Kind: Constant

Source: `packages/common/decorators/http/route-params.decorator.ts`

Part of: [Common](#)

Route handler parameter decorator. Extracts the `Session` object from the underlying platform and populates the decorated parameter with the value of `Session`.

`Session` is a route handler parameter decorator that injects the session object provided by the underlying HTTP platform. Use it in controller methods to access session data without manually reading it from the request object.

Definition

```
() => ParameterDecorator
```

Value

```
createRouteParamDecorator(  
  RouteParamtypes.SESSION,  
)
```

Diagram

```
graph LR  
  A[Incoming HTTP Request] --> B[Underlying Platform Session]  
  B --> C[@Session() Decorator]  
  C --> D[Controller Method Parameter]  
  D --> E[Route Handler Logic]
```

Usage

```
import { Controller, Get, Session } from '@nestjsjs/common';

@Controller('account')
export class AccountController {
  @Get('profile')
  getProfile(@Session() session: Record<string, unknown>) {
    return {
      userId: session.userId,
      authenticated: Boolean(session.userId),
    };
  }
}
```

AI Coding Instructions

- Apply `@Session()` only to route handler parameters where session data is required.
- Treat the injected session value as platform-provided request state; ensure session middleware or adapters are configured first.
- Define a typed session interface when your application stores known fields such as `userId`, roles, or preferences.
- Avoid reading session data directly from the request when `@Session()` provides the required value.
- Do not expose sensitive session fields directly in API responses.