

HttpArgumentsHost

August 18, 2026

HttpArgumentsHost

Kind: Interface

Source: `packages/common/interfaces/features/arguments-host.interface.ts`

Part of: [Common](#)

Methods to obtain request and response objects.

`HttpArgumentsHost` provides HTTP-specific access to the request, response, and next-function objects for the current execution context. It is typically obtained through `ExecutionContext.switchToHttp()` in guards, interceptors, filters, and custom decorators. Use it when framework-agnostic context handling needs to interact with the underlying HTTP adapter.

Diagram

```
graph LR
    A[ExecutionContext] --> B[switchToHttp()]
    B --> C[HttpArgumentsHost]
    C --> D[getRequest()]
    C --> E[getResponse()]
    C --> F[getNext()]
    D --> G[HTTP Request]
    E --> H[HTTP Response]
    F --> I[Next Middleware Function]
```

Usage

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

@Injectable()
export class ApiKeyGuard implements CanActivate {
  canActivate(context: ExecutionContext): boolean {
    const http = context.switchToHttp();
```

```

const request = http.getRequest<{ headers: Record<string, string> }>();
const response = http.getResponse<{ status: (code: number) => unknown }>();

const apiKey = request.headers['x-api-key'];

if (!apiKey) {
  response.status(401);
  return false;
}

return true;
}
}

```

AI Coding Instructions

- Obtain `HttpArgumentsHost` from `ExecutionContext.switchToHttp()` rather than assuming every execution context is HTTP-based.
- Use the generic type parameters on `getRequest()`, `getResponse()`, and `getNext()` when adapter-specific request or response types are needed.
- Avoid using this interface in transport-agnostic logic unless the code explicitly supports only HTTP contexts.
- Prefer reading request data through `getRequest()` and let NestJS handlers manage normal response serialization where possible.
- Remember that `getNext()` is primarily relevant to Express-style middleware flows and may not be meaningful for every HTTP adapter.

Used by

1 reference from 1 file. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (1)

- `ExecutionContextHost` — `packages/core/helpers/execution-context-host.ts` :10