

HttpAdapterHost

August 19, 2026

HttpAdapterHost

Kind: Class

Source: [packages/core/helpers/http-adapter-host.ts](#)

Part of: [Core](#)

Defines the `HttpAdapterHost` object.

`HttpAdapterHost` wraps the underlying platform-specific `HttpAdapter`. The `HttpAdapter` is a wrapper around the underlying native HTTP server library (e.g., Express). The `HttpAdapterHost` object provides methods to `get` and `set` the underlying `HttpAdapter`.

`HttpAdapterHost` provides access to the platform-specific `HttpAdapter` used by the application, such as the Express or Fastify adapter. It acts as a shared integration point for framework features that need to inspect or interact with the underlying HTTP server without depending on a specific platform.

Diagram

```
graph LR
  A[Nest Application] --> B[HttpAdapterHost]
  B --> C[HttpAdapter]
  C --> D[Express Adapter]
  C --> E[Fastify Adapter]
  D --> F[Native Express Server]
  E --> G[Native Fastify Server]
```

Usage

```
import { Injectable } from '@nestjs/common';
import { HttpAdapterHost } from '@nestjs/core';

@Injectable()
```

```
export class ServerInfoService {
  constructor(
    private readonly adapterHost: HttpAdapterHost,
  ) {}

  getPlatform(): string {
    const httpAdapter = this.adapterHost.httpAdapter;

    return httpAdapter.getType(); // "express" or "fastify"
  }

  getHttpServer() {
    return this.adapterHost.httpAdapter.getHttpServer();
  }
}
```

AI Coding Instructions

- Inject `HttpAdapterHost` through Nest's dependency injection container instead of creating it manually in application services.
- Access the current platform adapter through the `httpAdapter` property; avoid casting directly to Express or Fastify unless platform-specific behavior is required.
- Use adapter abstraction methods such as `getType()` and `getHttpServer()` to keep integrations portable across supported HTTP platforms.
- Ensure the HTTP adapter has been initialized before relying on native server behavior, especially in bootstrap, testing, or lifecycle-sensitive code.