

NestApplicationOptions

August 18, 2026

NestApplicationOptions

Kind: Interface

Source: `packages/common/interfaces/nest-application-options.interface.ts`

Part of: `Common`

`NestApplicationOptions` configures runtime behavior when creating a Nest application instance. It controls HTTP concerns such as CORS, request body parsing, HTTPS server options, raw body access, and connection shutdown behavior.

Properties

Property	Type
<code>cors</code>	<code>boolean</code>
<code>bodyParser</code>	<code>boolean</code>
<code>httpsOptions</code>	<code>HttpsOptions</code>
<code>rawBody</code>	<code>boolean</code>
<code>forceCloseConnections</code>	<code>boolean</code>

Diagram



```
F --> H
G --> H
```

Usage

```
import { NestFactory } from '@nestjs/core';
import type { NestApplicationOptions } from '@nestjs/common';
import { AppModule } from './app.module';

async function bootstrap() {
  const options: NestApplicationOptions = {
    cors: {
      origin: ['https://example.com'],
      credentials: true,
    },
    bodyParser: true,
    rawBody: true,
    forceCloseConnections: true,
    httpsOptions: {
      key: process.env.HTTPS_KEY,
      cert: process.env.HTTPS_CERT,
    },
  };

  const app = await NestFactory.create(AppModule, options);

  await app.listen(3000);
}

bootstrap();
```

API Coding Instructions

- Pass `NestApplicationOptions` as the second argument to `NestFactory.create()` when configuring an HTTP-based Nest application.
- Use `cors: true` for default CORS behavior, or provide `CorsOptions` /a delegate when origins and credentials require explicit control.
- Enable `rawBody` only when integrations need the unparsed payload, such as webhook signature verification.
- Keep `bodyParser` enabled unless registering custom parsing middleware or handling request streams manually.
- Supply valid Node.js HTTPS server options through `httpsOptions`, and consider `forceCloseConnections` for reliable shutdown in long-lived connection environments.

Used by

6 references from 6 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Imported by (6)

- `AbstractHttpAdapter` — `packages/core/adapters/http-adapter.ts` :8
- `NestApplication` — `packages/core/nest-application.ts` :54
- `IEntryNestModule` — `packages/core/nest-factory.ts` :39
- `ExpressAdapter` — `packages/platform-express/adapters/express-adapter.ts` :51
- `TestingModule` — `packages/testing/testing-module.ts` :26
- `SocketModule` — `packages/websockets/socket-module.ts` :27