

INestApplication

August 18, 2026

INestApplication

Kind: Interface

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

Part of: [Common](#)

Interface defining the core NestApplication object.

`INestApplication` defines the public API for a running Nest HTTP application. It extends the application context capabilities with HTTP server configuration, middleware registration, microservice integration, lifecycle management, and network listening methods.

Diagram

```
graph LR
  A[NestFactory.create] --> B[INestApplication]
  B --> C[Configure application]
  C --> C1[Middleware]
  C --> C2[CORS and versioning]
  C --> C3[Global prefix]
  C --> C4[WebSocket adapter]
  B --> D[Connect microservices]
  B --> E[Initialize and listen]
  E --> F[HTTP Server]
  F --> G[Clients]
```

Usage

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

async function bootstrap(): Promise<void> {
  const app: INestApplication = await NestFactory.create(AppModule);
```

```

app.setGlobalPrefix('api');
app.enableCors({
  origin: ['https://example.com'],
  credentials: true,
});

await app.listen(3000);

console.log(`Application running at ${await app.getUrl()}`);
}

bootstrap();

```

AI Coding Instructions

- Obtain an `INestApplication` through `NestFactory.create()` and configure it before calling `listen()`.
- Use `setGlobalPrefix()`, `enableCors()`, versioning, and middleware registration for application-wide HTTP behavior.
- Use `connectMicroservice()` and `startAllMicroservices()` when building hybrid HTTP and microservice applications.
- Prefer `getHttpServer()` only when an integration requires direct access to the underlying platform server.
- Enable shutdown hooks when the application must gracefully handle process termination signals.

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)

- `getHttpBaseOptions` — `integration/send-files/e2e/utils.ts` :5
- `NestApplication` — `packages/core/nest-application.ts` :54
- `IEntryNestModule` — `packages/core/nest-factory.ts` :39
- `NestExpressApplication` — `packages/platform-express/interfaces/nest-express-application.interface.ts` :20
- `NestFastifyApplication` — `packages/platform-fastify/interfaces/nest-fastify-application.interface.ts` :27

- `TestingModule` — `packages/testing/testing-module.ts` :26