

NestHybridApplicationOptions

August 18, 2026

NestHybridApplicationOptions

Kind: Interface

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

Part of: [Common](#)

`NestHybridApplicationOptions` configures how a NestJS hybrid application connects its microservice layer to the main HTTP application. It controls whether application-level configuration is inherited and whether microservice initialization is deferred until explicitly started.

Properties

Property	Type
<code>inheritAppConfig</code>	<code>boolean</code>
<code>deferInitialization</code>	<code>boolean</code>

Diagram

```
graph LR
  A[Nest Application] --> B[NestHybridApplicationOptions]
  B --> C[inheritAppConfig]
  B --> D[deferInitialization]
  C --> E[Share application configuration with microservice]
  D --> F[Delay microservice initialization]
  A --> G[Connected Microservice]
```

Usage

```
import { NestFactory } from '@nestjs/core';
import { Transport } from '@nestjs/microservices';
import type { NestHybridApplicationOptions } from '@nestjs/common';

import { AppModule } from './app.module';

async function bootstrap() {
  const app = await NestFactory.create(AppModule);

  const hybridOptions: NestHybridApplicationOptions = {
    inheritAppConfig: true,
    deferInitialization: false,
  };

  app.connectMicroservice(
    {
      transport: Transport.TCP,
    },
    hybridOptions,
  );

  await app.startAllMicroservices();
  await app.listen(3000);
}

bootstrap();
```

AI Coding Instructions

- Set `inheritAppConfig` to `true` when the connected microservice should reuse global pipes, guards, filters, and interceptors configured on the main application.
- Use `deferInitialization` when microservice startup must be controlled manually, such as when configuring dependencies before calling `startAllMicroservices()`.
- Ensure `startAllMicroservices()` is called before expecting the connected transport to receive messages.
- Keep hybrid application options separate from transport-specific microservice options passed as the first argument to `connectMicroservice()`.

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)

- `NestApplication` — `packages/core/nest-application.ts` :54