HttpsOptions

August 18, 2026

HttpsOptions

Kind: Interface

Source: `packages/common/interfaces/external/https-options.interface.ts`

Part of: Common

Interface describing Https Options that can be set.

`HttpsOptions` defines the TLS/HTTPS configuration used when creating secure HTTP servers or clients. It supports certificates, private keys, certificate authorities, cipher configuration, and mutual TLS validation settings.

Properties

Property	Type
`pfx`	`any`
`key`	`any`
`passphrase`	`string`
`cert`	`any`
`ca`	`any`
`crl`	`any`
`ciphers`	`string`
`honorCipherOrder`	`boolean`
`requestCert`	`boolean`
`rejectUnauthorized`	`boolean`
`NPNProtocols`	`any`
`SNICallback`	`(servername: string, cb: (err: Error, ctx: any) => any) => any`

Property	Type
`secureOptions`	`number`

Diagram

```
graph LR
  HttpsOptions[HttpsOptions]
  HttpsOptions --> Certificates[Certificate material]
  HttpsOptions --> TLS[TLS configuration]
  HttpsOptions --> ClientAuth[Client authentication]

  Certificates --> PFX[pfx]
  Certificates --> Key[key]
  Certificates --> Cert[cert]
  Certificates --> CA[ca]
  Certificates --> CRL[crl]
  Certificates --> Passphrase[passphrase]

  TLS --> Ciphers[ciphers]
  TLS --> CipherOrder[honorCipherOrder]

  ClientAuth --> RequestCert[requestCert]
  ClientAuth --> RejectUnauthorized[rejectUnauthorized]
```

Usage

```
import { readFileSync } from 'node:fs';
import type { HttpsOptions } from '@nestjs/common';

const httpsOptions: HttpsOptions = {
  key: readFileSync('./certs/server-key.pem'),
  cert: readFileSync('./certs/server-cert.pem'),
  ca: readFileSync('./certs/ca-cert.pem'),

  ciphers: 'TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256',
  honorCipherOrder: true,

  // Enable mutual TLS when clients must provide a trusted certificate.
  requestCert: true,
  rejectUnauthorized: true,
};
```

AI Coding Instructions

- Provide either `pfx` or a matching `key` and `cert` pair; do not configure conflicting certificate sources unless the underlying HTTPS integration supports it.
- Load certificate files as `Buffer` values, typically with `readFileSync`, rather than hardcoding sensitive certificate content.
- Use `passphrase` only when the private key or PFX bundle is encrypted.
- Set `requestCert` and `rejectUnauthorized` together when implementing mutual TLS; disabling authorization can allow untrusted client certificates.
- Configure `ca` with the trusted issuer certificates required to validate client certificates or upstream TLS peers.