HTTPS

In order to create application that uses HTTPS protocol, we have to pass an options object:

```
const httpsOptions = {
  key: fs.readFileSync('./secrets/private-key.pem'),
  cert: fs.readFileSync('./secrets/public-certificate.pem'),
};
const app = await NestFactory.create(ApplicationModule, {
  httpsOptions,
});
await app.listen(3000);
```

If using Fastify create the `app` like this:

```
const app = await NestFactory.create<NestFastifyApplication>(
  ApplicationModule,
  new FastifyAdapter({ https: httpsOptions }),
);
```

Multiple simultaneous servers

A full control over the library instance gives a simple way to create a several multiple simultaneous servers that are listening on different ports.

```
const httpsOptions = {
  key: fs.readFileSync('./secrets/private-key.pem'),
  cert: fs.readFileSync('./secrets/public-certificate.pem'),
};

const server = express();
const app = await NestFactory.create(
  ApplicationModule,
  new ExpressAdapter(server),
);
await app.init();

http.createServer(server).listen(3000);
```

```
https.createServer(httpsOptions, server).listen(443);
```

> **HINT**
> The `ExpressAdapter` is imported from the `@nestjs/platform-express` package.

Support us

Nest is an MIT-licensed open source project. It can grow thanks to the support by these awesome people. If you'd like to join them, please read more **here**.

Principal Sponsor

Sponsors / Partners

Become a sponsor