

Node.js OpenSSL 兼容性

在运行 `ng build` 时，使用 **Node.js 22.9.0** 和 **npm 10.8.3**，你可能会遇到以下错误：

```
error 0308010C digital envelope routines::unsupported
```

这个错误通常是由于 Node.js 的新版本（如 22.9.0，使用 OpenSSL 3.0 或更高版本）与某些加密算法不兼容引起的，这些算法被工具如 Webpack 所依赖，而 Angular 在幕后使用了 Webpack。以下是解决这个问题的分步指南。

解决方案

步骤 1：启用 OpenSSL 遗留提供程序 最简单和最直接的修复方法是通过设置 `NODE_OPTIONS` 环境变量来启用 OpenSSL 遗留提供程序。这允许 Node.js 使用在 OpenSSL 3.0+ 中不再默认支持的较旧加密算法。

- **对于 macOS/Linux：** 打开终端并运行：

```
export NODE_OPTIONS=--openssl-legacy-provider
ng build
```

- **对于 Windows（命令提示符）：** 打开命令提示符并运行：

```
set NODE_OPTIONS=--openssl-legacy-provider
ng build
```

- **对于 Windows（PowerShell）：** 打开 PowerShell 并运行：

```
$env:NODE_OPTIONS="--openssl-legacy-provider"
ng build
```

设置环境变量后，再次运行 `ng build`。这应该在大多数情况下通过允许 Node.js 处理不支持的路由来解决错误。

步骤 2：验证并更新 Angular CLI（如果需要） 如果在步骤 1 后错误仍然存在，你的 Angular CLI 版本可能与 Node.js 22.9.0 不完全兼容。将其更新到最新版本可能会有所帮助。

- 检查当前的 Angular CLI 版本：

```
ng --version
```

- 全局更新 Angular CLI：

```
npm install -g @angular/cli
```

- 然后，再次尝试运行 `ng build`。

步骤 3：检查并更新项目依赖项（可选） 如果问题仍未解决，项目中的过时依赖项可能会导致问题。要解决这个问题：

- 打开 `package.json` 文件并查看依赖项的版本（例如，`@angular/core`、`@angular/cli` 等）。
- 将它们小心更新到最新兼容版本：

```
npm install
```

或者，如果你想更新所有依赖项：

```
npm update
```

- 再次运行 `ng build` 进行测试。

注意：更新时要小心，因为它们可能会引入破坏性更改。更新后要彻底测试你的应用程序。

步骤 4：降级 Node.js（最后的手段） 如果以上解决方案都不起作用，Node.js 22.9.0 可能尚未完全支持你项目的当前配置。降级到 Node.js 的长期支持（LTS）版本，如 **20.x** 或 **18.x**，可能会解决问题。

- 卸载当前的 Node.js 版本。
- 从官方 Node.js 网站下载并安装 LTS 版本。
- 验证已安装的版本：

```
node -v
```

- 重新安装项目依赖项：

```
npm install
```

- 再次运行 `ng build`。
-

推荐方法

从 **步骤 1**（设置 `NODE_OPTIONS`）开始，因为这是最快的修复方法，允许你继续使用 Node.js 22.9.0。如果这不起作用，继续进行 **步骤 2** 和 **步骤 3**。只有在必要时才使用 **步骤 4**，因为降级 Node.js 可能会限制对新功能的访问。

这应该解决 “error 0308010C digital envelope routines::unsupported” 错误，并使你的 `ng build` 命令再次正常工作！