

Node.js OpenSSL Kompatibilität

Wenn Sie `ng build` mit **Node.js 22.9.0** und **npm 10.8.3** ausführen, tritt der folgende Fehler auf:

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

Dieser Fehler tritt typischerweise aufgrund eines Kompatibilitätsproblems zwischen neueren Versionen von Node.js (wie 22.9.0, die OpenSSL 3.0 oder später verwenden) und bestimmten kryptografischen Algorithmen auf, die von Tools wie Webpack, die Angular unter der Haube verwendet, abhängig sind. Hier ist eine Schritt-für-Schritt-Anleitung zur Behebung dieses Problems.

Lösung

Schritt 1: Aktivieren des OpenSSL Legacy Providers Die einfachste und schnellste Lösung besteht darin, den OpenSSL Legacy Provider zu aktivieren, indem die Umgebungsvariable `NODE_OPTIONS` gesetzt wird. Dies ermöglicht es Node.js, ältere kryptografische Algorithmen zu verwenden, die möglicherweise nicht mehr standardmäßig in OpenSSL 3.0+ unterstützt werden.

- **Für macOS/Linux:** Öffnen Sie Ihr Terminal und führen Sie aus:

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

- **Für Windows (Eingabeaufforderung):** Öffnen Sie die Eingabeaufforderung und führen Sie aus:

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

- **Für Windows (PowerShell):** Öffnen Sie PowerShell und führen Sie aus:

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

Nach dem Setzen der Umgebungsvariable führen Sie `ng build` erneut aus. Dies sollte den Fehler in den meisten Fällen beheben, indem es Node.js ermöglicht, die nicht unterstützten Routinen zu verarbeiten.

Schritt 2: Überprüfen und Aktualisieren des Angular CLI (falls erforderlich) Wenn der Fehler nach Schritt 1 weiterhin besteht, könnte Ihre Version des Angular CLI möglicherweise nicht vollständig mit Node.js 22.9.0 kompatibel sein. Das Aktualisieren auf die neueste Version kann helfen.

- Überprüfen Sie Ihre aktuelle Angular CLI-Version:

```
ng --version
```

- Aktualisieren Sie Angular CLI global:

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

- Versuchen Sie dann, ng build erneut auszuführen.
-

Schritt 3: Überprüfen und Aktualisieren der Projektabhängigkeiten (optional) Wenn das Problem weiterhin nicht behoben ist, könnten veraltete Abhängigkeiten in Ihrem Projekt das Problem verursachen. Um dies zu beheben:

- Öffnen Sie Ihre package.json-Datei und überprüfen Sie die Versionen Ihrer Abhängigkeiten (z. B. @angular/core, @angular/cli usw.).
- Aktualisieren Sie sie vorsichtig auf die neuesten kompatiblen Versionen:

```
npm install
```

oder, wenn Sie alle Abhängigkeiten aktualisieren möchten:

```
npm update
```

- Führen Sie ng build erneut aus, um zu testen.

Hinweis: Seien Sie bei Updates vorsichtig, da sie möglicherweise Änderungen mit sich bringen, die Ihre Anwendung beeinträchtigen. Testen Sie Ihre Anwendung danach gründlich.

Schritt 4: Herunterstufen von Node.js (Letzter Ausweg) Wenn keine der obigen Lösungen funktioniert, wird Node.js 22.9.0 möglicherweise noch nicht vollständig von der aktuellen Konfiguration Ihres Projekts unterstützt. Das Herunterstufen auf eine Long-Term Support (LTS)-Version von Node.js, wie **20.x** oder **18.x**, könnte das Problem beheben.

- Deinstallieren Sie Ihre aktuelle Node.js-Version.

- Laden Sie eine LTS-Version von der offiziellen Node.js-Website herunter und installieren Sie sie.
- Überprüfen Sie die installierte Version:

```
node -v
```

- Installieren Sie Ihre Projektabhängigkeiten erneut:

```
npm install
```

- Führen Sie `ng build` erneut aus.
-

Empfohlener Ansatz

Beginnen Sie mit **Schritt 1** (Setzen von `NODE_OPTIONS`), da dies die schnellste Lösung ist und Ihnen ermöglicht, weiterhin Node.js 22.9.0 zu verwenden. Wenn das nicht funktioniert, gehen Sie zu **Schritt 2** und **Schritt 3**. Verwenden Sie **Schritt 4** nur, wenn unbedingt notwendig, da das Herunterstufen von Node.js den Zugriff auf neuere Funktionen einschränken könnte.

Dies sollte den Fehler “error 0308010C digital envelope routines::unsupported” beheben und Ihren `ng build`-Befehl wieder funktionieren lassen!