Compatibilité OpenSSL Node.js

Lors de l'exécution de `ng build` avec **Node.js 22.9.0** et **npm 10.8.3**, vous rencontrez l'erreur :

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

Cette erreur survient généralement en raison d'un problème de compatibilité entre les versions plus récentes de Node.js (comme 22.9.0, qui utilise OpenSSL 3.0 ou une version ultérieure) et certains algorithmes cryptographiques utilisés par des outils comme Webpack, qu'Angular utilise sous le capot. Voici un guide étape par étape pour résoudre ce problème.

Solution

Étape 1 : Activer le fournisseur OpenSSL hérité La solution la plus simple et la plus immédiate consiste à activer le fournisseur OpenSSL hérité en définissant la variable d'environnement `NODE_OPTIONS`. Cela permet à Node.js d'utiliser des algorithmes cryptographiques plus anciens qui ne sont plus pris en charge par défaut dans OpenSSL 3.0+.

- **Pour macOS/Linux** : Ouvrez votre terminal et exécutez :

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

- **Pour Windows (Invite de commandes)** : Ouvrez l'invite de commandes et exécutez :

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

- **Pour Windows (PowerShell)** : Ouvrez PowerShell et exécutez :

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

Après avoir défini la variable d'environnement, exécutez `ng build` à nouveau. Cela devrait résoudre l'erreur dans la plupart des cas en permettant à Node.js de traiter les routines non prises en charge.

Étape 2 : Vérifier et mettre à jour Angular CLI (si nécessaire) Si l'erreur persiste après l'Étape 1, votre version de l'Angular CLI pourrait ne pas être entièrement compatible avec Node.js 22.9.0. La mettre à jour vers la dernière version peut aider.

- Vérifiez votre version actuelle de l'Angular CLI :

  ```
  ng --version
  ```

- Mettez à jour l'Angular CLI globalement :

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

- Ensuite, essayez d'exécuter `ng build` à nouveau.

Étape 3 : Vérifier et mettre à jour les dépendances du projet (facultatif) Si le problème n'est toujours pas résolu, des dépendances obsolètes dans votre projet pourraient en être la cause. Pour y remédier :

- Ouvrez votre fichier `package.json` et examinez les versions de vos dépendances (par exemple, `@angular/core`, `@angular/cli`, etc.).
- Mettez-les à jour soigneusement vers leurs dernières versions compatibles :

  ```
  npm install
  ```

  ou, si vous souhaitez mettre à jour toutes les dépendances :

  ```
  npm update
  ```

- Exécutez `ng build` à nouveau pour tester.

Note : Soyez prudent avec les mises à jour, car elles pourraient introduire des modifications cassantes. Testez votre application soigneusement par la suite.

Étape 4 : Dégrader Node.js (dernier recours) Si aucune des solutions ci-dessus ne fonctionne, Node.js 22.9.0 pourrait ne pas encore être entièrement pris en charge par la configuration actuelle de votre projet. La dégradation vers une version Long-Term Support (LTS) de Node.js, telle que **20.x** ou **18.x**, pourrait résoudre le problème.

- Désinstallez votre version actuelle de Node.js.

- Téléchargez et installez une version LTS depuis le site officiel de Node.js.
- Vérifiez la version installée :

  ```
  node -v
  ```

- Réinstallez vos dépendances de projet :

  ```
  npm install
  ```

- Exécutez `ng build` à nouveau.

Approche recommandée

Commencez par l'**Étape 1** (définir `NODE_OPTIONS`), car c'est la solution la plus rapide et vous permet de continuer à utiliser Node.js 22.9.0. Si cela ne fonctionne pas, passez à l'**Étape 2** et l'**Étape 3**. Utilisez l'**Étape 4** uniquement si nécessaire, car la dégradation de Node.js pourrait limiter l'accès aux nouvelles fonctionnalités.

Cela devrait résoudre l'erreur "error 0308010C digital envelope routines::unsupported"et permettre à votre commande `ng build` de fonctionner à nouveau !