

# openssl.1.1 संस्करणों में openssl-legacy-provider का अनुकूलन

जब ng build चलाते समय openssl.1.1 **22.9.0** और openssl **10.8.3** के साथ, आप इस त्रुटि से सामना कर सकते हैं:

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

यह त्रुटि आम तौर पर नए openssl.1.1 संस्करणों (जैसे 22.9.0, जो openssl 3.0 या बाद के संस्करण का उपयोग करता है) और कुछ क्रिप्टोग्राफिक एल्गोरिथमों के बीच संगतता समस्या के कारण होती है, जिन पर टूल्स जैसे openssl-legacy-provider निर्भर करते हैं, जो openssl 3.0 के नीचे काम करता है। नीचे इस समस्या को ठीक करने के लिए एक कदम-दर-कदम मार्गदर्शिका है।

---

## समाधान

**कदम 1:** openssl-legacy-provider को सक्रिय करें सबसे सरल और तत्काल समाधान openssl-legacy-provider को सक्रिय करने के लिए NODE\_OPTIONS पर्यावरण चर को सेट करना है। इससे openssl.1.1 को पुराने क्रिप्टोग्राफिक एल्गोरिथम का उपयोग करने की अनुमति मिलती है, जो openssl 3.0+ में डिफ़ॉल्ट रूप से समर्थित नहीं हो सकते हैं।

□ **bash/zsh के लिए:** अपने टर्मिनल खोलें और चलाएं:

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

□ **PowerShell (Windows) के लिए:** कमांड प्रॉम्प्ट खोलें और चलाएं:

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

□ **PowerShell (Linux) के लिए:** पावरशेल खोलें और चलाएं:

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

पर्यावरण चर को सेट करने के बाद, फिर से ng build चलाएं। यह अधिकांश मामलों में त्रुटि को सुलझाने के लिए openssl.1.1 को असमर्थ रूटीन को प्रोसेस करने की अनुमति देगा।

---

**कदम 2:** openssl की जांच और अपडेट करें (अगर आवश्यक हो) यदि कदम 1 के बाद त्रुटि बनी रहती है, तो आपके openssl 1.1 संस्करण openssl.1.1 22.9.0 के साथ पूरी तरह से संगत नहीं हो सकता है। इसे नवीनतम संस्करण में अपडेट करने से मदद मिल सकती है।

□ अपने वर्तमान openssl 1.1 संस्करण की जांच करें:

```
ng --version
```

□ `npm install -g @angular/cli` को ग्लोबल रूप से अपडेट करें:

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

□ फिर, फिर से `ng build` चलाएं।

---

**कदम 3: प्रोजेक्ट डिपेंडेंस की जांच और अपडेट करें (वैकल्पिक)** यदि अभी भी समस्या सुलझ नहीं रही है, तो प्रोजेक्ट में पुराने डिपेंडेंस हो सकते हैं जो समस्या का कारण हो सकते हैं। इसके लिए:

□ अपने `package.json` फ़ाइल खोलें और अपने डिपेंडेंस (जैसे `@angular/core`, `@angular/cli`, आदि) के संस्करणों की जांच करें।

□ उन्हें सावधानीपूर्वक उनके नवीनतम संगत संस्करण में अपडेट करें:

```
npm install
```

या, अगर आप सभी डिपेंडेंस अपडेट करना चाहते हैं:

```
npm update
```

□ फिर से `ng build` चलाएं।

सावधानी: अपडेट्स में तोड़ने वाले परिवर्तन हो सकते हैं। इसके बाद अपने एप्लिकेशन को पूरी तरह से टेस्ट करें।

---

**कदम 4: `node` को डाउनग्रेड करें (अंतिम उपाय)** यदि ऊपर दिए गए किसी भी समाधान काम नहीं करते, तो `node` 22.9.0 अभी तक आपके प्रोजेक्ट के वर्तमान कॉन्फ़िगरेशन द्वारा पूरी तरह से समर्थित नहीं हो सकता है। इसे `node`-`npm` `node` (20.0) संस्करण में डाउनग्रेड करने से समस्या सुलझ सकती है, जैसे **20.0** या **18.0**.

□ अपने वर्तमान `node` संस्करण को अनइंस्टॉल करें।

□ अधिकारिक `node` वेबसाइट से एक `node` संस्करण डाउनलोड और इंस्टॉल करें।

□ इंस्टॉल किए गए संस्करण की जांच करें:

```
node -v
```

□ फिर से अपने प्रोजेक्ट डिपेंडेंस इंस्टॉल करें:

```
npm install
```

□ फिर से `ng build` चलाएं।

---

## सिफारिश की गई रणनीति

**कदम 1** (npm\_दोपहरों सेट करने) से शुरू करें, क्योंकि यह सबसे तेज़ समाधान है और आपको npm 22.9.0 का उपयोग करने की अनुमति देता है। अगर यह काम नहीं करता, तो **कदम 2** और **कदम 3** पर आगे बढ़ें। **कदम 4** केवल तब उपयोग करें जब आवश्यक हो, क्योंकि npm को डाउनग्रेड करने से नए फीचर्स तक पहुंच सीमित हो सकती है।

यह “npm 0308010 दोपहरों दोपहरों दोपहरों::दोपहरों”त्रुटि को सुलझाने के लिए और फिर से ng build कमांड को काम करने के लिए आपको मदद करेगा!