

मार्गदर्शक अनुपात्य संस्करण 3.4.3 का अनुवाद

त्रुटि को समझना

जब आप mvn spring-boot:run चलाते हैं और मार्गदर्शक 3.4.3 के साथ “मार्गदर्शक मार्ग 61.0, मार्ग 52.0” त्रुटि मिलती है, तो यह एक मार्ग संस्करण अनुपात्यता समस्या का संकेत देता है। मार्ग में, क्लास फ़ाइलों में एक संस्करण संख्या होती है जो उनसे जुड़ी है:

- **संस्करण 52.0** मार्ग 8 के लिए है।
- **संस्करण 61.0** मार्ग 17 के लिए है।

त्रुटि यह सुझाती है कि आपकी मार्गदर्शक मार्ग एप्लिकेशन की क्लास फ़ाइलों मार्ग 17 (संस्करण 61.0) के साथ कम्पाइल की गई हैं, लेकिन उनको चलाने के लिए प्रयास करने वाला मार्ग मार्गदर्शक मार्ग (मार्ग) मार्ग 8 (संस्करण 52.0) के साथ अनुपात्य क्लास फ़ाइलों की अपेक्षा करता है। क्योंकि मार्ग 8 के मार्ग मार्ग 17 के साथ कम्पाइल की गई क्लास फ़ाइलों चलाने में असमर्थ हैं (क्योंकि मार्ग पिछले संस्करणों के साथ पिछले संस्करणों के साथ अनुपात्य है लेकिन इस संदर्भ में आगे के संस्करणों के साथ नहीं), आप इस त्रुटि को प्राप्त करते हैं।

यह क्यों होता है

मार्गदर्शक 3.4.3 मार्ग 17 या उससे आगे को अपना न्यूनतम समर्थित संस्करण के रूप में आवश्यकता है। इसका मतलब है कि आपकी परियोजना को कम्पाइल और चलाने के लिए कम से कम मार्ग 17 का उपयोग करना चाहिए। असंगतता संभवतः इसलिए होती है:

1. **कम्पाइलेशन:** आपकी परियोजना मार्ग 17 के साथ कम्पाइल होने के लिए कॉन्फ़िगर की गई है, जिससे संस्करण 61.0 के साथ क्लास फ़ाइलों बनती हैं। यह आमतौर पर आपके pom.xml में maven-compiler-plugin या <java.version> गुण के साथ सेट किया जाता है।
2. **रनटाइम:** जब आप mvn spring-boot:run चलाते हैं, तो मार्गदर्शक मार्ग प्लगिन एक नया मार्ग प्रक्रिया शुरू करता है ताकि एप्लिकेशन चल सके। फिर भी, यह उपयोग करता है java एक्सिक्यूटेबल एक मार्ग 8 इंस्टॉलेशन (संस्करण 52.0) से, जो मार्ग 17 के साथ कम्पाइल की गई क्लास फ़ाइलों को संभाल नहीं सकता।

त्रुटि को ठीक करने के चरण

इसको सुलझाने के लिए, आपको सुनिश्चित करना होगा कि दोनों कम्पाइलेशन और रनटाइम पर्यावरण मार्ग 17 का उपयोग करते हैं। यहाँ इसे ठीक करने के तरीके हैं:

1. **आपकी परियोजना के मार्ग संस्करण को सत्यापित करें** पहले, सुनिश्चित करें कि आपकी परियोजना मार्ग 17 का उपयोग करने के लिए सेट है। अपने pom.xml में निम्नलिखित को देखें:

```
<properties>
    <java.version>17</java.version>
</properties>
```

यह गुण maven-compiler-plugin को आपके कोड को मार्ग 17 के साथ कम्पाइल करने के लिए बताता है। मार्गदर्शक 3.4.3 इसे डिफॉल्ट रूप से सेट करता है, लेकिन इसे सत्यापित करने में अच्छा है। अगर यह गायब है या किसी और संस्करण (जैसे 8) पर सेट है, तो इसे 17 पर अपडेट करें।

2. जैसे 17 को इंस्टॉल करें सुनिश्चित करें कि आपके सिस्टम पर जैसे 17 इंस्टॉल है। आप इसे डाउनलोड कर सकते हैं:

- जैसे 17 (जैसे 17 एवं जैसे 17)
- जैसे 17 रूप (अगर आप लाइसेंस शर्तों को स्वीकार करते हैं)

जैसे 17 इंस्टॉल है या नहीं, यह जांचने के लिए एक टर्मिनल खोलें और निम्नलिखित चलाएं:

```
java -version
```

अगर यह जैसे 17 (जैसे openjdk 17.x.x या समान) दिखाता नहीं है, तो इसे इंस्टॉल करें और आगे बढ़ें।

3. आपका पर्यावरण जैसे 17 का उपयोग करने के लिए अपडेट करें जैसे 17 एवं जैसे 8 प्लगिन आपके पर्यावरण से java एक्सिक्यूटेबल का उपयोग करता है, जो आमतौर पर JAVA_HOME पर्यावरण चर या सिस्टम के जैसे में java कमांड द्वारा निर्धारित होता है। अगर आपका वर्तमान java कमांड जैसे 8 को संबोधित करता है, तो आपको इसे अपडेट करना होगा।

विकल्प □: जैसे_जैसे और जैसे सेट करें JAVA_HOME पर्यावरण चर को जैसे 17 इंस्टॉलेशन पर सेट करें और सुनिश्चित करें कि जैसे में java कमांड का उपयोग करता है।

□ जैसे/जैसे पर:

1. अपने जैसे 17 इंस्टॉलेशन डायरेक्टरी को खोजें (जैसे /usr/lib/jvm/java-17-openjdk या जहाँ आप इसे इंस्टॉल किया है)।
2. टर्मिनल में JAVA_HOME सेट करें और जैसे अपडेट करें:

```
export JAVA_HOME=/path/to/java-17
export PATH=$JAVA_HOME/bin:$PATH
```

3. सत्यापित करें:

```
java -version
```

अब यह जैसे 17 दिखाना चाहिए।

इसे स्थायी बनाने के लिए, export लाइनों को अपने शेल कॉन्फिगरेशन फाइल (जैसे ~/.bashrc, ~/.zshrc) में जोड़ें।

□ जैसे_जैसे पर:

1. जैसे 17 इंस्टॉलेशन डायरेक्टरी को खोजें (जैसे C:\Program Files\Java\jdk-17)।
2. JAVA_HOME सेट करें:
 - "सिस्टम प्रॉपर्टीज़" विंडो खोलें (स्टार्ट मेनू में "पर्यावरण चर" खोजें)।
 - "सिस्टम चर" के तहत, JAVA_HOME को C:\Program Files\Java\jdk-17 पर सेट करें या अपडेट करें।
3. Path अपडेट करें:
 - "सिस्टम चर" में, Path को संपादित करें और %JAVA_HOME%\bin जोड़ें (सुनिश्चित करें कि यह शीर्ष पर है या पुराने जैसे 8 को हटाएं)।
4. एक नया कमांड प्रॉम्प्ट खोलें और सत्यापित करें:

```
java -version
```

अब यह जैसे 17 दिखाना चाहिए।

विकल्प ०: मैंने JAVA_HOME के लिए एक विशेष मैंने संस्करण का उपयोग करें अगर आपके पास कई मैंने संस्करण इंस्टॉल हैं और आप अपने सिस्टम-वाइड JAVA_HOME को बदलना नहीं चाहते, तो आप मैंने के साथ एक विशेष मैंने संस्करण चल सकते हैं:

- mvn spring-boot:run चलाने से पहले, केवल उस कमांड के लिए JAVA_HOME सेट करें:

```
JAVA_HOME=/path/to/java-17 mvn spring-boot:run
```

यह सुनिश्चित करता है कि मैंने मैंने मैंने प्लगिन मैंने 17 का उपयोग करता है बिना अपने सिस्टम सेटिंग्स को बदलने।

4. एप्लिकेशन फिर से चलाएं मैंने 17 सही तरह से सेट होने के साथ, चलाएं:

```
mvn spring-boot:run
```

प्लगिन अब मैंने 17 का उपयोग करेगा आपके एप्लिकेशन को चलाने के लिए, क्लास फ़ाइल संस्करण (61.0) के साथ मिलते हैं, और त्रुटि गायब हो जाएगी।

अतिरिक्त जांचें

- मैंने के मैंने संस्करण:** mvn -version चलाएं ताकि देखें कि मैंने कौन सा मैंने संस्करण का उपयोग करता है। यह आपके JAVA_HOME के साथ मिलना चाहिए। अगर नहीं, सुनिश्चित करें कि आपके मैंने और JAVA_HOME सही तरह से कॉन्फ़िगर हैं।
- अनुपात्यताएं:** अगर त्रुटि जारी है, तो एक अनुपात्यता मैंने 17 के साथ कम्पाइल हो सकती है जबकि आपके रनटाइम अभी भी गलत कॉन्फ़िगर है। क्योंकि मैंने मैंने 3.4.3 मैंने 17 को लागू करता है, सुनिश्चित करें कि आपके रनटाइम मैंने 17 है।

क्यों नहीं मैंने 8?

मैंने मैंने 3.4.3 मैंने 8 का समर्थन नहीं करता। अगर आप मैंने 8 का उपयोग करना चाहते हैं, तो आपको एक पुराने मैंने मैंने संस्करण (जैसे 2.0) पर डाउनग्रेड करना होगा, लेकिन यह केवल विशेष बाधाओं के साथ सिफारिश नहीं किया जाता है। क्योंकि आपके क्लास फ़ाइल संस्करण 61.0 हैं, मैंने 17 के साथ मैंने मैंने 3.4.3 के अनुरूप होने से आपका अनुपात्यता समस्या सुलझा जाएगा।

अंतिम समाधान सारांश

"मैंने मैंने मैंने मैंने मैंने मैंने 61.0, मैंने मैंने मैंने 52.0" त्रुटि को सुलझाने के लिए जब mvn spring-boot:run चलाते हैं मैंने मैंने 3.4.3 के साथ:

- सुनिश्चित करें कि <java.version>17</java.version> आपके pom.xml में है।
- अगर अभी तक मौजूद नहीं है, तो मैंने 17 को इंस्टॉल करें।
- JAVA_HOME को अपने मैंने 17 इंस्टॉलेशन डायरेक्टरी पर सेट करें।
- अपने मैंने को अपडेट करें ताकि java -version मैंने 17 दिखाए।
- फिर से mvn spring-boot:run चलाएं।

यह सुनिश्चित करता है कि आपके रनटाइम आपके मैंने 17 के साथ कम्पाइल की गई क्लास फ़ाइलों के साथ मिलते हैं, अनुपात्यता समस्या को सुलझाते हैं।