

' 0000 00000000 00 000000 000000 0000

java.configuration.runtimes को 000000 000000 0000 (00 0000) सेटिंग्स में उपयोग करने के लिए, आपको अपने settings.json फ़ाइल में इसे कॉन्फ़िगर करना होगा। यह सेटिंग आपको 00 0000 को आपके 0000 प्रोजेक्ट्स के लिए उपयोग करने के लिए कई 0000 000000000000 0000 (0000) संस्करणों को स्पेसिफ़ाई करने की अनुमति देता है। यह विशेष रूप से अलग-अलग प्रोजेक्ट्स के साथ काम करने के लिए उपयोगी है जो विशिष्ट 0000 संस्करणों की आवश्यकता होती हैं। यहां एक स्टेप-बाय-स्टेप गाइड है इसे सेटअप करने के लिए:

java.configuration.runtimes को कॉन्फ़िगर करने के लिए चरण

1. 00 0000 सेटिंग्स खोलें:

- 00000000/000000 पर: File > Preferences > Settings पर जाएं या Ctrl + , , दबाएं।
- 000000 पर: Code > Preferences > Settings पर जाएं या Cmd + , , दबाएं।

2. 0000 सेटिंग्स फ़ाइल तक पहुंचें:

- सेटिंग्स 00 में, java.configuration.runtimes के लिए खोजें।
- आप "0000: 00000000000000: 00000000"जैसा विकल्प देखेंगे। "0000 00 00000000.0000"(सामान्यतः सेटिंग्स विवरण के नीचे एक लिंक) पर क्लिक करें settings.json फ़ाइल को खोलने के लिए।

3. settings.json को संपादित करें:

- settings.json फ़ाइल में, java.configuration.runtimes एरे को जोड़ें या संपादित करें। इस एरे में वस्तुएं होती हैं, प्रत्येक एक 0000 संस्करण को 00 0000 को पहचानने के लिए प्रतिनिधित्व करता है।
- प्रत्येक वस्तु आमतौर पर शामिल होती है:
 - name: 0000 संस्करण पहचानकर्ता (उदाहरण के लिए, JavaSE-1.8, JavaSE-11, JavaSE-17).
 - path: आपके सिस्टम पर 0000 इंस्टॉलेशन डायरेक्टरी का पूर्ण पथ।
 - default (वैकल्पिक): इसे true सेट करें ताकि यह अनमैनेज्ड फ़ोल्डर्स (मैवेन या ग्रेडल जैसी बिल्ड टूल्स के बिना प्रोजेक्ट्स) के लिए डिफ़ॉल्ट 0000 बन जाए।

यहां एक उदाहरण कॉन्फ़िगरेशन है:

```
{
  "java.configuration.runtimes": [
    {
      "name": "JavaSE-1.8",
      "path": "C:/Program Files/Java/jdk1.8.0_351",
      "default": true
    },
    {
      "name": "JavaSE-11",
      "path": "C:/Program Files/Java/jdk-11.0.15"
    }
  ],
}
```

```
{
    "name": "JavaSE-17",
    "path": "C:/Program Files/Java/jdk-17.0.6"
}
]
```

4. पथों की पुष्टि करें:

- path को IDE इंस्टॉलेशन के रूट डायरेक्टरी (जहां bin फ़ोल्डर java.exe या java है) पर सेट करें।
- IDE में, पथ में फॉरवर्ड स्लैश (/) या बैकस्लैश (\) का उपयोग करें।
- IDE में, उचित फ़ाइल सिस्टम पथ का उपयोग करें (उदाहरण के लिए, /usr/lib/jvm/java-17-openjdk).

5. सहेजें और पुनः लोड करें:

- settings.json फ़ाइल को सहेजें।
- IDE को पुनः शुरू करें या विंडो को पुनः लोड करें (Ctrl + R या Cmd + R) बदलाव लागू करने के लिए।

6. कॉन्फ़िगरेशन की जांच करें:

- कमांड पॅलेट (Ctrl + Shift + P या Cmd + Shift + P) खोलें और कमांड Java: Configure Java Runtime चलाएं।
- यह एक दृश्य खोलता है जिसमें आपके प्रोजेक्ट्स के लिए उपलब्ध IDE दिखाते हैं। पुष्टि करें कि आपके कॉन्फ़िगर किए गए रनटाइम "IDE" टैब के नीचे दिखाई देते हैं।

यह कैसे काम करता है

- **अनमैनेज्ड फ़ोल्डर्स**: बिल्ड टूल्स के बिना प्रोजेक्ट्स (उदाहरण के लिए, साधारण IDE फ़ाइलें) के लिए, IDE IDE java.configuration.runtimes में स्पेसिफ़ाई किया गया default IDE का उपयोग करता है।
- **मैनेज्ड प्रोजेक्ट्स** (IDE/IDE): बिल्ड टूल्स के साथ प्रोजेक्ट्स के लिए, IDE संस्करण बिल्ड कॉन्फ़िगरेशन (उदाहरण के लिए, pom.xml या build.gradle) द्वारा निर्धारित होता है, लेकिन IDE अभी भी यहाँ सूचीबद्ध रनटाइम्स को अनुकूलता के लिए पहचानेगा।
- **लैंग्वेज सर्वर**: IDE लैंग्वेज सर्वर (IDE एक्सटेंशन का हिस्सा) को चलाने के लिए IDE 17+ की आवश्यकता होती है। अगर आवश्यक हो, इसे अलग से java.jdt.ls.java.home के साथ स्पेसिफ़ाई किया जा सकता है (उदाहरण के लिए, "java.jdt.ls.java.home": "C:/Program Files/Java/jdk-17.0.6").

कई IDE के साथ उदाहरण

अगर आप IDE पर हैं और IDE के माध्यम से IDE 8 और 17 इंस्टॉल किए हैं, तो आपके कॉन्फ़िगरेशन इस तरह दिख सकता है:

```
{
  "java.configuration.runtimes": [
    {
      "name": "JavaSE-1.8",
      "path": "/Library/Java/JavaVirtualMachines/jdk1.8.0_351.jdk/Contents/Home",
```

```

        "default": true
    },
    {
        "name": "JavaSE-17",
        "path": "/Library/Java/JavaVirtualMachines/jdk-17.0.6.jdk/Contents/Home"
    }
]
}

```

टिप्स

- सही name फॉर्मेट (JavaSE-<version>) का उपयोग करें, जो □□□□ संस्करण (उदाहरण के लिए, □□□□ 11 के लिए JavaSE-11) के साथ मिलता है।
- अगर आपके पास केवल एक □□□ है, तो आप इसे सूचीबद्ध करके □□ □□□□ को इसे स्पेसिफ़िक रूप से उपयोग करने के लिए सुनिश्चित कर सकते हैं।
- □□□ पथ को खोजने के लिए, एक टर्मिनल में `java -version` चलाएं और इंस्टॉलेशन डायरेक्टरी (उदाहरण के लिए, □□□□□□□□ पर `where java` या □□□□-□□□□ सिस्टम पर `which java`, फिर □□□ रूट तक वापस ट्रेस करने के लिए) की जांच करें।

यह सेटअप आपको □□ □□□□ में कई □□□□ संस्करणों के साथ काम करने की सुविधा देता है! मुझे बताएं अगर आपको □□□ पथों को खोजने या ट्रबलशूटिंग करने में मदद चाहिए।