

Java Laufzeiten in Visual Studio Code

Um `java.configuration.runtimes` in den Einstellungen von Visual Studio Code (VS Code) zu verwenden, müssen Sie es in Ihrer `settings.json`-Datei konfigurieren. Diese Einstellung ermöglicht es Ihnen, mehrere Java Development Kit (JDK)-Versionen zu spezifizieren, die VS Code für Ihre Java-Projekte verwenden kann. Es ist besonders nützlich, wenn Sie mit verschiedenen Projekten arbeiten, die spezifische Java-Versionen erfordern. Hier ist eine Schritt-für-Schritt-Anleitung zur Einrichtung:

Schritte zur Konfiguration von `java.configuration.runtimes`

1. VS Code-Einstellungen öffnen:

- Unter Windows/Linux: Gehen Sie zu Datei > Einstellungen > Einstellungen oder drücken Sie Strg + ,.
- Unter macOS: Gehen Sie zu Code > Einstellungen > Einstellungen oder drücken Sie Cmd + ,.

2. JSON-Einstellungsdatei öffnen:

- In der Einstellungs-Oberfläche suchen Sie nach `java.configuration.runtimes`.
- Sie sehen eine Option wie "Java: Konfiguration: Laufzeiten". Klicken Sie auf "In settings.json bearbeiten" (normalerweise ein Link unter der Einstellungsbeschreibung), um die `settings.json`-Datei zu öffnen.

3. `settings.json` bearbeiten:

- In der `settings.json`-Datei fügen Sie das Array `java.configuration.runtimes` hinzu oder bearbeiten Sie es. Dieses Array enthält Objekte, die jeweils eine JDK-Version darstellen, die VS Code erkennen soll.
- Jedes Objekt enthält in der Regel:
 - `name`: Der Java-Version-Identifizier (z.B. `JavaSE-1.8`, `JavaSE-11`, `JavaSE-17`).
 - `path`: Der absolute Pfad zum JDK-Installationsverzeichnis auf Ihrem System.
 - `default` (optional): Setzen Sie dies auf `true`, um diese JDK-Version zur Standard-JDK für unverwaltete Ordner (Projekte ohne Build-Tools wie Maven oder Gradle) zu machen.

Hier ist ein Beispiel für eine Konfiguration:

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

```

{
    "name": "JavaSE-11",
    "path": "C:/Programme/Java/jdk-11.0.15"
},
{
    "name": "JavaSE-17",
    "path": "C:/Programme/Java/jdk-17.0.6"
}
]
}

```

4. JDK-Pfade überprüfen:

- Stellen Sie sicher, dass der path auf das Stammverzeichnis Ihrer JDK-Installation zeigt (z.B. wo sich der bin-Ordner mit java.exe oder java befindet).
- Unter Windows verwenden Sie Schrägstriche (/) oder escape Backslashes (\\) im Pfad.
- Unter macOS/Linux verwenden Sie den entsprechenden Dateisystempfad (z.B. /usr/lib/jvm/java-17-openj

5. Speichern und neu laden:

- Speichern Sie die settings.json-Datei.
- Starten Sie VS Code neu oder laden Sie das Fenster neu (Strg + R oder Cmd + R), um die Änderungen zu übernehmen.

6. Konfiguration überprüfen:

- Öffnen Sie die Befehlspalette (Strg + Umschalt + P oder Cmd + Umschalt + P) und führen Sie den Befehl Java: Java-Runtime konfigurieren aus.
- Dies öffnet eine Ansicht, die die für Ihre Projekte verfügbaren JDKs zeigt. Stellen Sie sicher, dass Ihre konfigurierten Laufzeiten unter der Registerkarte "Projekt-JDKs" erscheinen.

So funktioniert es

- **Unverwaltete Ordner:** Für Projekte ohne Build-Tools (z.B. einfache Java-Dateien) verwendet VS Code die default-JDK, die in java.configuration.runtimes angegeben ist.
- **Verwaltete Projekte (Maven/Gradle):** Für Projekte mit Build-Tools wird die JDK-Version durch die Build-Konfiguration (z.B. pom.xml oder build.gradle) bestimmt, aber VS Code erkennt die hier aufgeführten Laufzeiten weiterhin für die Kompatibilität.
- **Sprachserver:** Der Java Language Server (Teil der Java-Erweiterungen) benötigt JDK 17+ zum Ausführen. Sie können dies separat mit java.jdt.ls.java.home angeben, falls erforderlich (z.B. "java.jdt.ls.java.home": "C:/Programme/Java/jdk-17.0.6").

Beispiel mit mehreren JDKs

Wenn Sie auf einem Mac sind und JDK 8 und 17 über Homebrew installiert haben, könnte Ihre Konfiguration so aussehen:

```
{
  "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"
    }
  ]
}
```

Tipps

- Verwenden Sie das richtige `name`-Format (`JavaSE-<version>`), das mit der Java-Version übereinstimmt (z.B. `JavaSE-11` für Java 11).
- Wenn Sie nur eine JDK haben, können Sie sie immer noch auflisten, um sicherzustellen, dass VS Code sie explizit verwendet.
- Um Ihren JDK-Pfad zu finden, führen Sie `java -version` in einem Terminal aus und überprüfen Sie das Installationsverzeichnis (z.B. über `where java` unter Windows oder `which java` unter Unix-artigen Systemen, dann zurückverfolgen zum JDK-Stamm).

Diese Einrichtung ermöglicht es Ihnen, nahtlos mit mehreren Java-Versionen in VS Code zu arbeiten! Lassen Sie mich wissen, wenn Sie Hilfe beim Finden Ihrer JDK-Pfade oder bei der Fehlerbehebung benötigen.