

Konfigurieren Sie den Java-Compiler in Visual Studio Code

Hallo! Um den Java-Compiler in Visual Studio Code (VSCode) Version 1.96.4 unter Windows zu konfigurieren, insbesondere nachdem Sie das Java Extension Pack installiert haben, müssen Sie den Pfad zur Java Development Kit (JDK) einrichten. Der Java-Compiler ist Teil des JDK, und VSCode muss wissen, wo es ihn finden kann, um Ihren Java-Code zu kompilieren. Hier ist eine Schritt-für-Schritt-Anleitung, um Sie einzurichten:

Schritt 1: Stellen Sie sicher, dass ein JDK installiert ist

Bevor Sie in VSCode etwas konfigurieren, stellen Sie sicher, dass Sie ein JDK (nicht nur eine JRE) auf Ihrem Windows-Computer installiert haben. Das JDK enthält den Java-Compiler (`javac`). Wenn Sie es noch nicht installiert haben, können Sie es von einem Anbieter wie Oracle, OpenJDK oder Adoptium (z.B. JDK 17 oder eine andere Version, die mit Ihrem Projekt kompatibel ist) herunterladen. Nach der Installation notieren Sie sich den Installationspfad (z.B. `C:\Program Files\Java\jdk-17.0.1`).

Schritt 2: Öffnen Sie die VSCode-Einstellungen

Um VSCode mitzuteilen, wo sich Ihr JDK befindet, müssen Sie dessen Einstellungen anpassen:

- **Über die Benutzeroberfläche der Einstellungen:**

- Drücken Sie `Ctrl + ,`, um das Einstellungsfenster zu öffnen.
- Alternativ gehen Sie zu `Datei > Einstellungen > Einstellungen`.

- **Über `settings.json` (optional):**

- Drücken Sie `Ctrl + Shift + P`, um die Befehlspalette zu öffnen.
 - Geben Sie **“Einstellungen (JSON) öffnen”** ein und wählen Sie es aus, um die `settings.json`-Datei direkt zu bearbeiten.
-

Schritt 3: Legen Sie den JDK-Pfad mit `java.home` fest

Das Java Extension Pack verwendet die `java.home`-Einstellung, um das JDK für die Kompilierung und Sprachfunktionen (wie IntelliSense) zu finden. Hier ist, wie Sie es konfigurieren:

- **In der Benutzeroberfläche der Einstellungen:**

- Im Einstellungsfenster suchen Sie nach **“java.home”**.
- Im Feld “Java: Home” geben Sie den vollständigen Pfad zu Ihrer JDK-Installation ein. Zum Beispiel:
`C:\Program Files\Java\jdk-17.0.1`
- Verwenden Sie Backslashes (\), da Sie unter Windows sind, und stellen Sie sicher, dass der Pfad auf das Stammverzeichnis des JDK zeigt (es sollte einen `bin`-Ordner mit `javac.exe` enthalten).

- **In settings.json:**

- Wenn Sie `settings.json` bearbeiten, fügen Sie diese Zeile hinzu (ersetzen Sie den Pfad durch Ihren tatsächlichen JDK-Standort):

```
"java.home": "C:\\Program Files\\Java\\jdk-17.0.1"
```

- Beispiel für eine vollständige `settings.json`:

```
{  
  "java.home": "C:\\Program Files\\Java\\jdk-17.0.1"  
}
```

- Speichern Sie die Datei nach dem Bearbeiten.

Schritt 4: Überprüfen Sie den Pfad

Stellen Sie sicher, dass:

- Der Pfad auf ein JDK (nicht auf eine JRE) zeigt. Der `bin`-Ordner des JDK sollte `javac.exe` enthalten.
- Es keine Tippfehler im Pfad gibt und er mit Ihrem JDK-Installationsort übereinstimmt (z.B. `C:\Program Files\Java\jdk-17.0.1`).

Wenn Sie nicht sicher sind, wo Ihr JDK installiert ist, können Sie es in `C:\Program Files\Java` oder an dem Ort finden, den Sie während der Installation gewählt haben.

Schritt 5: Optional - Konfigurieren Sie mehrere JDKs

Wenn Sie mehrere JDK-Versionen installiert haben und zwischen ihnen wechseln möchten (z.B. JDK 8 für ein Projekt, JDK 17 für ein anderes), können Sie die Einstellung `java.configuration.runtimes` verwenden:

- Fügen Sie dies zu Ihrer `settings.json` hinzu:

```
"java.configuration.runtimes": [  
  {  
    "name": "JavaSE-1.8",  
    "path": "C:\\Program Files\\Java\\jdk1.8.0_291"
```

```
    },  
    {  
      "name": "JavaSE-17",  
      "path": "C:\\Program Files\\Java\\jdk-17.0.1",  
      "default": true  
    }  
  ]  
}
```

- Die Option `default: true` legt die Standardlaufzeit zum Ausführen Ihres Codes fest. Für die Kompilierung verwendet die Java-Erweiterung jedoch hauptsächlich das in `java.home` angegebene JDK.
-

Schritt 6: Laden Sie VSCode neu oder starten Sie es neu

Nach dem Festlegen von `java.home` müssen Sie möglicherweise: - Das VSCode-Fenster neu laden (drücken Sie `Ctrl + Shift + P`, geben Sie **“Fenster neu laden”** ein und wählen Sie es aus). - Oder starten Sie VSCode vollständig neu, damit die Änderungen wirksam werden.

Hinweise zur Kompilierung

- **Einzelne Java-Dateien:** Wenn Sie mit einzelnen `.java`-Dateien (ohne Build-Tools) arbeiten, verwendet das Java Extension Packs „Language Support for Java(TM) by Red Hat“ das JDK aus `java.home`, um Ihren Code zu kompilieren und auszuführen. Sie können die Schaltfläche **“Ausführen”** oder den Befehl **“Java: Ausführen”** verwenden.
 - **Projekte mit Build-Tools:** Wenn Sie Maven, Gradle oder ein anderes Build-Tool verwenden, wird der Compiler durch die Konfiguration des Build-Tools verwaltet (z.B. `JAVA_HOME`-Umgebungsvariable, `pom.xml` für Maven oder `build.gradle` für Gradle). Das Festlegen von `java.home` stellt jedoch sicher, dass die Sprachfunktionen von VSCode korrekt funktionieren.
-

Fehlerbehebung

- **Pfadprobleme:** Wenn VSCode das JDK nicht finden kann, überprüfen Sie den Pfad in `java.home`. Vermeiden Sie unnötige Leerzeichen oder Anführungszeichen, es sei denn, es ist unbedingt erforderlich.
- **JDK nicht installiert:** Wenn Sie Fehler über fehlendes `javac` erhalten, stellen Sie sicher, dass ein JDK installiert ist.

- **Versionskompatibilität:** Stellen Sie sicher, dass Ihre JDK-Version den Anforderungen Ihres Projekts entspricht (z.B. JDK 8, 11, 17).
-

Das war's! Jetzt sollten Sie den Java-Compiler in VSCode konfiguriert haben. Wenn Sie auf Probleme stoßen oder weitere Fragen haben -z.B. zur Arbeit mit einem bestimmten Build-Tool -lassen Sie es mich wissen, und ich helfe Ihnen gerne weiter! ☺

Setzen Sie `maven.terminal.userJavaHome` auf `true`. Sie können die Java-Version im Terminal überprüfen.