

Java-Sprachserver und Maven-Build

Es scheint, dass Sie zwei verwandte Probleme in Visual Studio Code (VSCode) haben: eine Meldung, dass mindestens Java 21 benötigt wird, und Probleme, dass VSCode einige Objektdefinitionen in Ihrem Java-Code nicht erkennt, obwohl Ihr Maven-Build erfolgreich kompiliert wird. Lassen Sie uns das Schritt für Schritt lösen.

Das Problem entsteht, weil die Java-Erweiterungen von VSCode (insbesondere der Java Language Server) und Ihr Maven-Build möglicherweise nicht in Bezug auf die Version des Java Development Kit (JDK) oder die Projektkonfiguration übereinstimmen. VSCode meldet, dass es JDK 21 benötigt, um ordnungsgemäß zu funktionieren, und die Parsing-Fehler deuten darauf hin, dass es Ihren Code nicht so interpretiert wie Maven. Hier ist, wie Sie es beheben können:

Schritt 1: Installieren Sie JDK 21

Da VSCode explizit JDK 21 anfordert, müssen Sie sicherstellen, dass es auf Ihrem System installiert ist.

- **Laden Sie JDK 21 herunter:** Holen Sie es sich von einer vertrauenswürdigen Quelle wie Oracle's Website oder Adoptium.
- **Installieren Sie es:** Folgen Sie den Installationsanweisungen und notieren Sie sich den Installationspfad (z. B. C:\Program Files\Java\jdk-21 unter Windows oder /usr/lib/jvm/java-21-openjdk unter Linux).
- **Überprüfen Sie:** Öffnen Sie ein Terminal und führen Sie `java -version` aus. Sie sollten etwas wie Folgendes sehen:

```
java version "21.0.1" ...
```

Schritt 2: Konfigurieren Sie VSCode zur Verwendung von JDK 21

Sie müssen VSCode mitteilen, dass es JDK 21 verwenden soll. Hier ist, wie:

1. Öffnen Sie die VSCode-Einstellungen:

- Gehen Sie zu Datei > Einstellungen > Einstellungen (oder drücken Sie Strg + ,).

2. Legen Sie den JDK-Pfad fest:

- Suchen Sie nach `java.jdt.ls.java.home`.
- Geben Sie den Pfad zu Ihrer JDK 21-Installation ein (z. B. C:\Program Files\Java\jdk-21).
- Optional: Suchen Sie nach `java.home` und setzen Sie es auf denselben Pfad zur Konsistenz.

3. Bearbeiten Sie settings.json (Alternative):

- Öffnen Sie die Befehlspalette (Strg + Umschalt + P), geben Sie „Einstellungen: Einstellungen (JSON) öffnen“ ein und fügen Sie hinzu:

```
{  
  "java.jdt.ls.java.home": "C:\\Program Files\\Java\\jdk-21",  
  "java.home": "C:\\Program Files\\Java\\jdk-21"  
}
```

- Passen Sie den Pfad entsprechend Ihrem System an (verwenden Sie Schrägstriche / für Linux/Mac).

Dies stellt sicher, dass der Java Language Server in VSCode JDK 21 verwendet und die Mindestanforderung erfüllt.

Schritt 3: Legen Sie das JDK des Projekts in VSCode fest

Um die Parsing-Probleme (z. B. nicht erkannte Objektdefinitionen) zu beheben, stellen Sie sicher, dass Ihr Projekt in VSCode ebenfalls JDK 21 verwendet:

- In settings.json fügen Sie Folgendes hinzu, um JDK 21 für Ihr Projekt festzulegen:

```
"java.configuration.runtimes": [  
  {  
    "name": "JavaSE-21",  
    "path": "C:\\Program Files\\Java\\jdk-21",  
    "default": true  
  }  
]
```

- Dies stimmt die Laufzeitumgebung des Projekts in VSCode mit JDK 21 ab, was helfen sollte, Ihren Code korrekt zu parsen.

Schritt 4: Überprüfen Sie die JDK-Konfiguration von Maven

Da Ihr Maven-Build (`mvn compile`) gut funktioniert, verwendet er wahrscheinlich ein kompatibles JDK. Lassen Sie uns dies bestätigen und mit VSCode abgleichen:

1. Überprüfen Sie das JDK von Maven:

- Führen Sie `mvn -version` in Ihrem Terminal aus. Suchen Sie nach der Zeile „Java-Version“ (z. B. Java-Version: 21.0.1).
- Wenn es nicht JDK 21 verwendet, legen Sie die Umgebungsvariable `JAVA_HOME` auf den Pfad zu Ihrem JDK 21 fest:
 - **Windows:** `set JAVA_HOME=C:\Program Files\Java\jdk-21`
 - **Linux/Mac:** `export JAVA_HOME=/usr/lib/jvm/java-21-openjdk`
- Führen Sie `mvn -version` erneut aus, um dies zu bestätigen.

2. Legen Sie das JDK in `pom.xml` fest (optional):

- Wenn Sie JDK 21 in Ihrem Maven-Projekt erzwingen möchten, fügen Sie dies zu Ihrer `pom.xml` hinzu:

```
<properties>
  <maven.compiler.source>21</maven.compiler.source>
  <maven.compiler.target>21</maven.compiler.target>
</properties>
```

Schritt 5: Aktualisieren Sie VSCode

Nach diesen Änderungen muss VSCode aktualisiert werden, um sie anzuwenden:

- Laden Sie VSCode neu:
 - Öffnen Sie die Befehlspalette (`Strg + Umschalt + P`) und wählen Sie „Entwickler: Fenster neu laden“.
- Alternativ: Schließen und öffnen Sie VSCode erneut.

Schritt 6: Stellen Sie sicher, dass Abhängigkeiten geladen werden

Wenn VSCode immer noch keine Objektdefinitionen findet, hat es möglicherweise Ihre Maven-Abhängigkeiten nicht korrekt geladen:

1. Bereinigen Sie den Arbeitsbereich:

- Öffnen Sie die Befehlspalette (`Strg + Umschalt + P`).
- Führen Sie „Java: Java Language Server Workspace bereinigen“ aus.

2. Erzwingen Sie die Kompilierung:

- Führen Sie „Java: Java-Kompilierung erzwingen“ aus der Befehlspalette aus.

3. Überprüfen Sie Importe:

- Stellen Sie sicher, dass die Maven-Abhängigkeiten Ihres Projekts importiert werden, indem Sie mit der rechten Maustaste auf Ihre `pom.xml` in VSCode klicken und „Projektkonfiguration aktualisieren“ auswählen.
-

Fehlerbehebung

Wenn das Problem weiterhin besteht: - **Überprüfen Sie die Protokolle:** - Fügen Sie `"java.trace.server": "verbose"` zu `settings.json` hinzu. - Öffnen Sie das Ausgabefenster (Strg + Umschalt + U) und wählen Sie „Java Language Server“, um detaillierte Protokolle anzuzeigen. - **Überprüfen Sie den Code:** - Stellen Sie sicher, dass die fehlenden Objektdefinitionen aus Ihrem Code oder den in `pom.xml` aufgelisteten Abhängigkeiten stammen und nicht aus externen Problemen.

Zusammenfassung

Durch die Installation von JDK 21, die Konfiguration von VSCode zur Verwendung desselben, die Anpassung Ihrer Projekt- und Maven-Einstellungen sowie das Aktualisieren der Umgebung sollte VSCode aufhören, sich über Java 21 zu beschweren und Ihren Code korrekt parsen, was mit der erfolgreichen Kompilierung von Maven übereinstimmt. Lassen Sie mich wissen, wenn Sie auf Probleme stoßen!