

Java 語言伺服器在 Visual Studio Code

嘿！`java.jdt.ls.java.home` 設定在 Visual Studio Code (VSCode) 中是一個特定的配置選項，由 Java 語言伺服器（作為“Language Support for Java(TM) by Red Hat”擴展的一部分，包含在 Java Extension Pack 中）使用。讓我們來分解一下：

它的意義

- **JDT**：代表“Java Development Tools”，最初是 Eclipse IDE 的一部分。VSCode 中的 Java 語言伺服器基於 Eclipse 的 JDT，提供了代碼補全、調試和編譯等功能。
- **LS**：代表“Language Server”，是一個後台進程，為 VSCode 的 Java 支持提供動力（例如 IntelliSense 和錯誤檢查）。
- `java.jdt.ls.java.home`：這個設定告訴 Java 語言伺服器在其操作中使用的 Java Development Kit (JDK)。它與更廣泛的 `java.home` 設定不同，儘管它們有關聯。

簡而言之，`java.jdt.ls.java.home` 指定了 Java 語言伺服器使用的 JDK 路徑，以：
- 解析你的 Java 代碼。
- 提供語言功能（例如自動補全、跳轉到定義）。
- 在某些情況下編譯和運行代碼（儘管編譯通常依賴於其他設定或構建工具）。

與 `java.home` 的區別

- `java.home`：VSCode 的一般設定，指向所有 Java 相關的擴展和任務使用的 JDK。除非被更具體的設定覆蓋，否則會使用它。
- `java.jdt.ls.java.home`：一個更具體的設定，僅覆蓋 Java 語言伺服器的 `java.home`。如果沒有設置，語言伺服器會回退到 `java.home`。

因此，如果你設置了 `java.jdt.ls.java.home`，它會優先用於語言伺服器的操作，讓你可以為語言功能使用不同的 JDK，而不是運行或調試任務。

如何配置它

由於你在 Windows 上使用 VSCode 1.96.4 和 Java Extension Pack，以下是設置的方法：

1. **打開設定：**

- 按 `Ctrl + ,` 打開設定 UI，或者使用 `Ctrl + Shift + P` 並輸入 “**Open Settings (JSON)**” 來編輯 `settings.json`。

2. 設置路徑：

• 在設定 UI 中：

- 搜索 “**java.jdt.ls.java.home**”。
- 输入 JDK 的完整路徑（例如 `C:\Program Files\Java\jdk-17.0.1`）。

• 在 `settings.json` 中：

- 添加這行（替換為你的 JDK 路徑）：

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

- 範例 `settings.json`：

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

3. 驗證 JDK 路徑：

- 確保路徑指向包含 `bin` 目錄的 JDK 文件夾，其中包含 `javac.exe` 和 `java.exe`。

4. 重新加載 VSCode：

- 重新加載窗口 (`Ctrl + Shift + P > “Reload Window”`) 或重新啟動 VSCode 以應用更改。

何時使用它

你通常會設置 `java.jdt.ls.java.home` 如果：
- 你安裝了多個 JDK，並希望語言伺服器使用特定的 JDK（例如 JDK 17），而其他 Java 任務（例如運行代碼）可能會通過 `java.home` 使用不同的 JDK。
- 你在排查語言伺服器問題（例如它沒有自動選擇正確的 JDK）。
- 你的項目對語言功能有特定的 JDK 版本要求。

如果你不設置它，語言伺服器將使用 `java.home` 中的 JDK（或作為最後的手段使用系統的 `JAVA_HOME` 環境變量）。

實際範例

假設你安裝了 JDK 8 和 JDK 17：
- 設置 `"java.home": "C:\\Program Files\\Java\\jdk1.8.0_291"` 以供一般使用。
- 設置 `"java.jdt.ls.java.home": "C:\\Program Files\\Java\\jdk-17.0.1"` 以確保語言伺服器使用 JDK 17 以獲取現代 Java 語法支持。

這樣，你可以使用 JDK 8 進行調試/運行，但獲得 JDK 17 功能的語言支持。

這樣清楚了嗎？如果你想知道這如何適應你的設置或需要進一步調整的幫助，隨時告訴我！✉