

Java 運行時在 Visual Studio Code

要在 Visual Studio Code (VS Code) 設定中使用 `java.configuration.runtimes`，你需要在 `settings.json` 文件中進行配置。這個設定允許你指定多個 Java Development Kit (JDK) 版本，供 VS Code 用於你的 Java 專案。這對於需要特定 Java 版本的不同專案特別有用。以下是設定的步驟指南：

配置 `java.configuration.runtimes` 的步驟

1. 打開 VS Code 設定：

- 在 Windows/Linux：前往 `File > Preferences > Settings` 或按 `Ctrl + ,`。
- 在 macOS：前往 `Code > Preferences > Settings` 或按 `Cmd + ,`。

2. 訪問 JSON 設定文件：

- 在設定界面中，搜尋 `java.configuration.runtimes`。
- 你會看到一個選項如 “Java: Configuration: Runtimes”。點擊 “Edit in settings.json”（通常是設定描述下方的連結）以打開 `settings.json` 文件。

3. 編輯 `settings.json`：

- 在 `settings.json` 文件中，添加或修改 `java.configuration.runtimes` 陣列。這個陣列包含物件，每個物件代表你希望 VS Code 認識的 JDK 版本。
- 每個物件通常包括：
 - `name`：Java 版本識別符（例如 `JavaSE-1.8`、`JavaSE-11`、`JavaSE-17`）。
 - `path`：系統上 JDK 安裝目錄的絕對路徑。
 - `default`（可選）：設置為 `true` 以使這個 JDK 成為未管理資料夾（沒有構建工具如 Maven 或 Gradle 的專案）的預設 JDK。

以下是一個範例配置：

```
{
  "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. 驗證 JDK 路徑：

- 確保 path 指向 JDK 安裝的根目錄（例如，包含 bin 資料夾和 java.exe 或 java 的位置）。
- 在 Windows 上，使用正斜杠 (/) 或轉義反斜杠 (\\) 在路徑中。
- 在 macOS/Linux 上，使用適當的文件系統路徑（例如 /usr/lib/jvm/java-17-openjdk）。

5. 保存並重新載入：

- 保存 settings.json 文件。
- 重新啟動 VS Code 或重新載入視窗（Ctrl + R 或 Cmd + R）以應用更改。

6. 檢查配置：

- 打開命令面板（Ctrl + Shift + P 或 Cmd + Shift + P）並運行命令 Java: Configure Java Runtime。
- 這會打開一個顯示專案可用 JDK 的視圖。驗證你配置的運行時是否出現在“Project JDKs”選項卡下。

運作方式

- **未管理資料夾**：對於沒有構建工具的專案(例如普通 Java 文件)，VS Code 使用 java.configuration.runtimes 中指定的 default JDK。
- **管理專案 (Maven/Gradle)**：對於有構建工具的專案，JDK 版本由構建配置決定（例如 pom.xml 或 build.gradle），但 VS Code 仍會識別這裡列出的運行時以確保兼容性。
- **語言伺服器**：Java 語言伺服器（Java 擴展的一部分）需要 JDK 17+ 才能運行。如果需要，可以使用 java.jdt.ls.java.home 另行指定（例如 "java.jdt.ls.java.home": "C:/Program Files/Java/jdk-17.0.6"）。

多個 JDK 的範例

如果你在 Mac 上通過 Homebrew 安裝了 JDK 8 和 17，你的配置可能如下：

```

{
  "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"  
  }  
]  
}
```

小技巧

- 使用正確的 name 格式（JavaSE-<version>），與 Java 版本匹配（例如 JavaSE-11 表示 Java 11）。
- 即使只有一個 JDK，你仍然可以列出它以確保 VS Code 明確使用它。
- 要找到你的 JDK 路徑，在終端中運行 `java -version` 並檢查安裝目錄（例如，通過 `where java` 在 Windows 或 `which java` 在 Unix 類系統，然後追溯到 JDK 根目錄）。

這樣的設置讓你能夠在 VS Code 中靈活地使用多個 Java 版本！如果你需要幫助找到 JDK 路徑或排除故障，請告訴我。