

Java 运行时在 Visual Studio Code

要在 Visual Studio Code (VS Code) 设置中使用 `java.configuration.runtimes`, 你需要在 `settings.json` 文件中进行配置。这个设置允许你指定多个 Java 开发工具包 (JDK) 版本, 供 VS Code 使用在你的 Java 项目中。这在处理需要特定 Java 版本的不同项目时特别有用。以下是设置的分步指南:

配置 `java.configuration.runtimes` 的步骤

1. 打开 VS Code 设置:

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

2. 访问 JSON 设置文件:

- 在设置界面中, 搜索 `java.configuration.runtimes`。
- 你会看到一个类似于“Java: Configuration: Runtimes”的选项。点击“Edit in settings.json”(通常是设置描述下的链接)以打开 `settings.json` 文件。

3. 编辑 `settings.json`:

- 在 `settings.json` 文件中, 添加或修改 `java.configuration.runtimes` 数组。这个数组包含对象, 每个对象代表你希望 VS Code 识别的 JDK 版本。
- 每个对象通常包括:
 - `name`: Java 版本标识符 (例如 JavaSE-1.8、JavaSE-11、JavaSE-17)。
 - `path`: 系统上 JDK 安装目录的绝对路径。
 - `default` (可选): 设置为 `true` 以使其成为未管理文件夹 (没有构建工具如 Maven 或 Gradle 的项目) 的默认 JDK。

以下是一个示例配置:

```
{  
  "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-17.0.2"  
    }  
  ]  
}
```

```
        "name": "JavaSE-17",
        "path": "C:/Program Files/Java/jdk-17.0.6"
    },
]
}
```

4. 验证 JDK 路径：

- 确保 path 指向 JDK 安装的根目录（例如，包含 bin 文件夹的目录，其中包含 java.exe 或 java）。
- 在 Windows 上，使用正斜杠 (/) 或转义反斜杠 (\) 在路径中。
- 在 macOS/Linux 上，使用适当的文件系统路径（例如 /usr/lib/jvm/java-17-openjdk）。

5. 保存并重新加载：

- 保存 settings.json 文件。
- 重新启动 VS Code 或重新加载窗口（Ctrl + R 或 Cmd + R）以应用更改。

6. 检查配置：

- 打开命令面板（Ctrl + Shift + P 或 Cmd + Shift + P）并运行命令 Java: Configure Java Runtime。
- 这将打开一个视图，显示项目可用的 JDK。验证你配置的运行时是否出现在“Project JDks”选项卡下。

工作原理

- 未管理文件夹**：对于没有构建工具的项目（例如纯 Java 文件），VS Code 使用 java.configuration.runtimes 中指定的 default JDK。
- 管理项目（Maven/Gradle）**：对于有构建工具的项目，JDK 版本由构建配置决定（例如 pom.xml 或 build.gradle），但 VS Code 仍将识别此处列出的运行时以确保兼容性。
- 语言服务器**：Java 语言服务器（Java 扩展的一部分）需要 JDK 17+ 才能运行。如果需要，可以单独指定它，例如 java.jdt.ls.java.home（例如，"java.jdt.ls.java.home": "C:/Program Files/Java/jdk-17.0.6"）。

多个 JDK 的示例

如果你在 Mac 上通过 Homebrew 安装了 JDK 8 和 17，你的配置可能如下所示：

```
{
  "java.configuration.runtimes": [
    {
      "name": "JavaSE-1.8",
      "path": "/Library/Java/JavaVirtualMachines/jdk1.8.0_351.jdk/Contents/Home",
      "default": true
    },
    {
      "name": "JavaSE-17",
      "path": "C:/Program Files/Java/jdk-17.0.6"
    }
  ]
}
```

```
        "name": "JavaSE-17",
        "path": "/Library/Java/JavaVirtualMachines/jdk-17.0.6.jdk/Contents/Home"
    }
]
}
```

提示

- 使用正确的 name 格式 (JavaSE-<version>)，与 Java 版本匹配 (例如，Java 11 对应 JavaSE-11)。
- 即使只有一个 JDK，也可以将其列出，以确保 VS Code 明确使用它。
- 要找到 JDK 路径，可以在终端中运行 `java -version`，并检查安装目录 (例如，通过 `where java` 在 Windows 上或 `which java` 在类 Unix 系统上，然后追踪到 JDK 根目录)。

这种设置使你能够在 VS Code 中灵活地处理多个 Java 版本！如果你需要帮助找到 JDK 路径或排除故障，请告诉我。