

launch.json 配置问题



什么是build 任务？

在 VS Code 中，`build` 任务是一个通过 `tasks.json` 文件定义的自动化任务，用来执行程序的编译、构建等操作。当你启动调试时，`preLaunchTask` 字段指定的任务会在调试会话开始之前运行。

例如，你在 `launch.json` 中设置了 `preLaunchTask` 为 `"build"`，这意味着在你启动调试之前，VS Code 会执行一个名为 `build` 的任务。通常，这个任务会用来编译或构建你的代码。

一个典型的 `tasks.json` 文件看起来可能是这样的：

```
json
{
  "version": "2.0.0",
  "tasks": [
    {
      "label": "build",
      "type": "shell",
      "command": "g++",
      "args": [
        "-g",
        "src/main.cpp",
        "-o", "build/my_program"
      ],
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "problemMatcher": ["$gcc"],
      "detail": "编译程序"
    }
  ]
}
```

在上面的示例中，`build` 任务是一个使用 `g++` 编译器的任务。它会将 `src/main.cpp` 编译为一个可执行文件 `build/my_program`，并且添加了 `-g` 参数来生成调试信息。

关键字段说明：

- `"label"`：任务的名称，在 `launch.json` 中引用这个名字（比如 `"build"`）。
- `"type"`：任务类型，可以是 `shell` 或 `process`。`shell` 类型会使用命令行执行任务，`process` 是直接执行程序。
- `"command"`：要执行的命令，比如编译器命令。
- `"args"`：命令行参数，用于传递给编译器或构建工具。
- `"group"`：将任务分组，`build` 表示这是一个构建任务。
- `"problemMatcher"`：用于识别编译错误的信息，通常选择对应的编译器（如 `\$gcc`）。

如果你希望在调试之前自动编译代码，可以通过这种方式创建 `build` 任务。