

手把手教你如何在VSCode里调试python

本文仅介绍Python特有的注意事项，主要是特定于Python的调试配置，包括特定应用程序类型和远程调试的必要步骤。

初始化配置

配置在调试会话期间驱动VS Code的行为。配置在launch.json文件中定义，该文件存储在工作区的.vscode文件夹中。

注意为了更改调试配置，您的代码必须存储在一个文件夹中。

要使用Python配置生成launch.json文件，请执行以下步骤：

- 1.选择设置按钮（在上图中圈出）或使用Debug> Open configurations菜单命令。
- 2。将从命令选项板打开配置菜单，允许您为打开的文件选择所需的调试配置类型。现在，在出现的Select a debug configuration菜单中，选择Python File。

注意通过调试面板启动调试会话，F5或调试>启动调试，如果不存在配置，也会打开调试配置菜单。

1.然后，Python扩展创建并打开一个launch.json文件，该文件包含基于您之前选择的预定义配置，在本例中为Python文件。您可以修改配置（例如，添加参数），还可以添加自定义配置。

更多的配置

默认情况下，VS Code仅显示Python扩展提供的最常见配置。您可以使用列表和launch.json编辑器中显示的“添加配置”命令选择要包含在launch.json中的其他配置。当您使用该命令时，VS Code会提示您所有可用配置的列表（请务必向下滚动以查看所有Python选项）：

选择Node.js：Gulp任务会产生以下结果：

在调试过程中，状态栏显示左下方的当前配置；右边是当前的调试解释器。选择配置会显示一个列表，您可以从中选择不同的配置：

默认情况下，调试器使用与VS Code的其他功能相同的python.pythonPath工作空间设置。要使用不同的解释器进行特定的调试，请在launch.json中为pythonPath设置适用的调试器配置，如下一节所述。或者，选择状态栏上的命名解释器以选择另一个更新python.pythonPath。

设置配置选项

首次创建launch.json时，有两种标准配置在编辑器中的集成终端（VS代码内部）或外部终端（VS代码外部）中运

行活动文件：

具体设置将在以下部分中介绍。您还可以添加标准配置中未包含的其他设置，例如args。

name

提供VS Code下拉列表中显示的调试配置的名称。

type

标识要使用的调试器类型; 用于Python代码。

request

指定调试的模式

launch:指定调试起始文件programattach:指定调试挂载进程

program

提供python程序的入口模块（启动文件）的完全限定路径。 值： `${file}`, 常用于默认配置，使用编辑器中当前活动的文件。通过指定特定的启动文件，无论打开哪个文件，您始终可以确保使用相同的入口点启动程序。 例如：

```
"program": "/Users/Me/Projects/PokemonGo-Bot/pokemongo_bot/event_handlers/__init__.py",
```

您还可以依赖工作区根目录中的相对路径。 例如，如果是

根"/Users/Me/Projects/PokemonGo-Bot", 你可以像这样使用

"program":

"\${workspaceFolder}/pokemongo_bot/event_handlers/__init__.py",

pythonPath

指向用于调试的Python解释器, 它可以是包含Python解释器的文件夹。该值可以使用变量\${workspaceFolder}和\${workspaceFolder}/.venv如果未指定, 则此设置默认为在中标识的解释器python.pythonPath,

或者, 您可以使用在每个平台上定义的自定义环境变量来包含要使用的Python解释器的完整路径, 这样就不需要其他文件夹路径。

args

指定传递给Python程序的参数。由空格分隔的参数字符串的每个元素都应包含在引号内, 例如:

"args": ["--quiet", "--norepeat", "--port", "1593"],

stopOnEntry

设置为true的时候, 打破正在调试的程序的的第一行的调试器。如果省略(默认值)或设置为false, 调试器将程序运行到第一个断点。

console

指定程序输出的显示方式。

cwd

指定调试器的当前工作目录，该目录是代码中使用的任何相对路径的基本文件夹。如果省略，则默认为 `${workspaceFolder}` vscode 的工作目录，作为一个例子 `${workspaceFolder}` 包含了 python 代码文件夹或者文件，包含了 `app.py`

配置如下

redirectOutput

省略或设置为 `true`（默认值），使调试器将程序的所有输出打印到 VS Code 调试输出窗口。如果设置为 `false`，程序输出不会显示在调试器输出窗口中。

使用时通常禁用此选项

```
"console": "integratedTerminal"
```

或

```
"console": "externalTerminal"
```

因为不需要在调试控制台中复制输出。

justMyCode

省略或设置为true（默认值），仅将调试限制为用户编写的代码。调成false还可以调试标准库函数。

django

可以调试django框架

env

为调试器进程设置可选的环境变量，而不是调试器始终继承的系统环境变量。

envFile

包含环境变量定义的文件的可选路径。请参阅配置Python环境 - 环境变量定义文件。

在代码中调用断点

在Python代码中，您可以调用断点 在调试会话期间要暂停调试器的任何位置。

断点验证

Python扩展自动检测在非可执行行上设置的断点，例如通过 语句或多行语句的中间。在这种情况下，运行调试器会将断点移动到最近的有效行，以确保代码执行在此时停止。

附加到本地脚本

在某些情况下，您需要调试由另一个进程在本地调用的Python脚本。例如，您可能正在调试为特定处理作业运

行不同Python脚本的Web服务器。在这种情况下，您需要在启动后将VS Code调试器附加到脚本：

1.运行VS Code，打开包含脚本的文件夹或工作区，然后创建一个launch.json 对于该工作空间，如果尚不存在。

2.在脚本代码中，添加以下内容并保存文件：

3.使用终端打开终端：创建新的集成终端，激活脚本的选定环境。在终端中，使用`python -m pip install --upgrade ptvsd`安装ptvsd软件包。

5.在终端中，使用脚本启动Python，例如`python3 myscript.py`。您应该看到代码中包含的“等待调试器附加”消息，并且脚本在`ptvsd.wait_for_attach()`调用时停止。

6.切换到Debug视图，从Debugger下拉列表中选择Python: Attach，然后启动调试器。

这样就可以在本地调试python代码了。下一节将向大家介绍VSCode如何配置远程调试python，尽情期待