# 接入调试的方法

## 1调试配置

### 1.1 生成调试配置

点击VSCode调试状态栏，若没有创建过launch.json文件，主动创建launch.json，在弹出框中选择LuaHelper：debug。会自动生成对应的LuaHelper调试配置文件launch.json。

avatar

avatar

生成的调试配置文件launch.json里面包含了两种调试方式： \* LuaHelper-Attach：通过attach的方式调试其他的执行进程。 \* LuaHelper-DebugFile：表示调试和运行单个lua文件。

avatar

avatar

### 1.2 添加调试配置

点击VSCode调试状态栏，已经创建过lanuch.json文件，但是调试的方式不包括LuaHelper-Attach和LuaHelper-DebugFile，需要快捷添加调试配置。 快捷添加的方式：点击Add Configuration按钮，快捷输入LuaHelper，选择LuaHelper-Attach和LuaHelper-DebugFile。

avatar

avatar

### 1.3 调试项说明

launch.json 配置项中要修改的主要是luaFileExtension, 改成lua文件使用的后缀就行（比如xlua框架改为lua.txt, slua框架改为txt）。

avatar

avatar

## 2引入LuaPanda.lua文件

调试需要引入LuaPanda.lua与前端VScode Lua工程进行socket通信，且LuaHelper.lua会直接引入luasocket网络库。目前lua框架： slua, slua-unreal, xlua 都已集成 luasocket网络库。

**判断项目中是否包含luasocket:** 在项目lua代码中中加入require(“socket.core”);，如果运行不报错，表示项目已经包含luasocket。

安装LuaHelper插件之后，在插件目录会自带LuaPanda.lua文件和luasocket网络库。 插件提供了下述的快捷命令，用于方便引入LuaPanda.lua文件和luasocket网络库。

**打开快捷方式:** 快捷键:ctrl + shift + p, 然后输入LuaHelper，会提示下面列的快捷命令：

avatar

avatar

* LuaHelper: Open Debug Foder，表示打开插件自带的关键目录（里面包含LuaPanda.lua和luasocket库）。
* LuaHelper: Copy debug file to workspace, 表示把LuaPanda.lua文件拷贝到项目中指定的目录，需要手动指定目标目录。
* LuaHelper: Copy Lua Socket Lib, 表示把luasocket网络库拷贝到项目中指定的目录，需要手动指定目标目录。
* LuaHelper: Insert Debugger Code, 表示在lua代码指定的位置快捷插入，require(“LuaPanda”).start(“127.0.0.1”, 8818);

## 3开始调试

* 使用VSCode打开工程lua文件夹
* 点击VSCode调试状态栏，选择LuaHelper-Attach调试选项（若没有该选项，请先设置调试配置）。
* VSCode前端工程按F5启动调试，等待运行lua代码的进行通过引入LuaPanda.lua文件连接上。
* Lua代码工程进程开始运行，连接VSCode前端工程。

avatar s ## 4各种框架下使用调试 ### slua框架调试 1. **拷贝调试文件** 通过快捷命令：Ctrl + Shirt + P, 接着输入LuaHelper: Copy debug file to workspace， 把LuaPanda.lua 文件拷贝到slua工程 Slua/Resources/ 目录下, 并修改文件后缀为 .txt

1. **配置调试设置** 点击 VSCode 调试选项卡下的齿轮图标，选择 LuaHelper-Attach。把配置项 luaFileExtension 值修改为 “txt”。
2. **引入调试代码** 在lua工程的入口处，通过快捷命令：Ctrl + Shirt + P, 接着输入LuaHelper: Insert Debugger Code 引入下列lua代码 lua require("LuaPanda").start("127.0.0.1", 8818)
3. **开始调试** VSCode切换到调试选项卡，配置项选择LuaHelper-Attach，按F5开始调试，设置断点，再运行Unity。

### xlua框架调试

1. **拷贝调试文件** 通过快捷命令：Ctrl + Shirt + P, 接着输入LuaHelper: Copy debug file to workspace， 把LuaPanda.lua 文件拷贝到xlua工程对应的Resources 目录下, 并修改后缀为 .lua.txt。
2. **配置调试设置** 点击 VSCode 调试选项卡下的齿轮图标，选择 LuaHelper-Attach。把配置项 luaFileExtension 值修改为 “.lua.txt”。
3. **引入调试代码** 在lua工程的入口处，通过快捷命令：Ctrl + Shirt + P, 接着输入LuaHelper: Insert Debugger Code 引入下列lua代码 lua require("LuaPanda").start("127.0.0.1", 8818)

4.**开始调试** VSCode切换到调试选项卡，配置项选择LuaHelper-Attach，按F5开始调试，设置断点，再运行Unity。

### slua-unreal

1. **拷贝调试文件** 通过快捷命令：Ctrl + Shirt + P, 接着输入LuaHelper: Copy debug file to workspace， 把LuaPanda.lua 文件拷贝到slua-unreal 工程对应的Content/Lua/目录下。
2. **引入调试代码** 在lua工程的入口处，通过快捷命令：Ctrl + Shirt + P, 接着输入LuaHelper: Insert Debugger Code 引入下列lua代码 lua require("LuaPanda").start("127.0.0.1", 8818)
3. **开始调试** VSCode切换到调试选项卡，配置项选择LuaHelper-Attach，按F5开始调试，设置断点，再运行Unity。

### unlua

1. **拷贝luasocket库** 目前unlua默认不集成luasocket库，需要用快捷命令把插件目录自带的luasocket库拷贝到指定的目录，通常与UnLua.lua文件同级。
   * 快捷命令方法：输入快捷键 Ctrl + Shirt + P, 接着输入LuaHelper: Copy Lua Socket Lib
2. **拷贝调试文件** 快捷命令把把LuaPanda.lua 文件拷贝到指定的目录，通常与UnLua.lua文件同级。
   * 快捷命令方法：输入快捷键 Ctrl + Shirt + P, 接着输入LuaHelper: Copy debug file to workspace
3. **引入调试文件** 在UnLua.lua文件中通过快捷命令引入下面代码段 lua require("LuaPanda").start("127.0.0.1", 8818)
   * 快捷命令方法：输入快捷键 Ctrl + Shirt + P, 接着输入LuaHelper: Insert Debugger Code(use luasocket)

### cocos2dx

1. **拷贝调试文件** 快捷命令把把LuaPanda.lua 文件拷贝到指定的目录，通常是src/目录下与main.lua 文件同级。
   * 快捷命令方法：输入快捷键 Ctrl + Shirt + P, 接着输入LuaHelper: Copy debug file to workspace
2. **引入调试文件** 在main.lua文件 require "cocos.init" 行之前加入代码段 lua require("LuaPanda").start("127.0.0.1",8818)
   * 快捷命令方法：输入快捷键 Ctrl + Shirt + P, 接着输入LuaHelper: Insert Debugger Code