
文章

[jieliang liu](#) · 二月 5, 2021 阅读大约需 5 分钟

精华文章置顶--使用VSCode 进行IRIS 开发

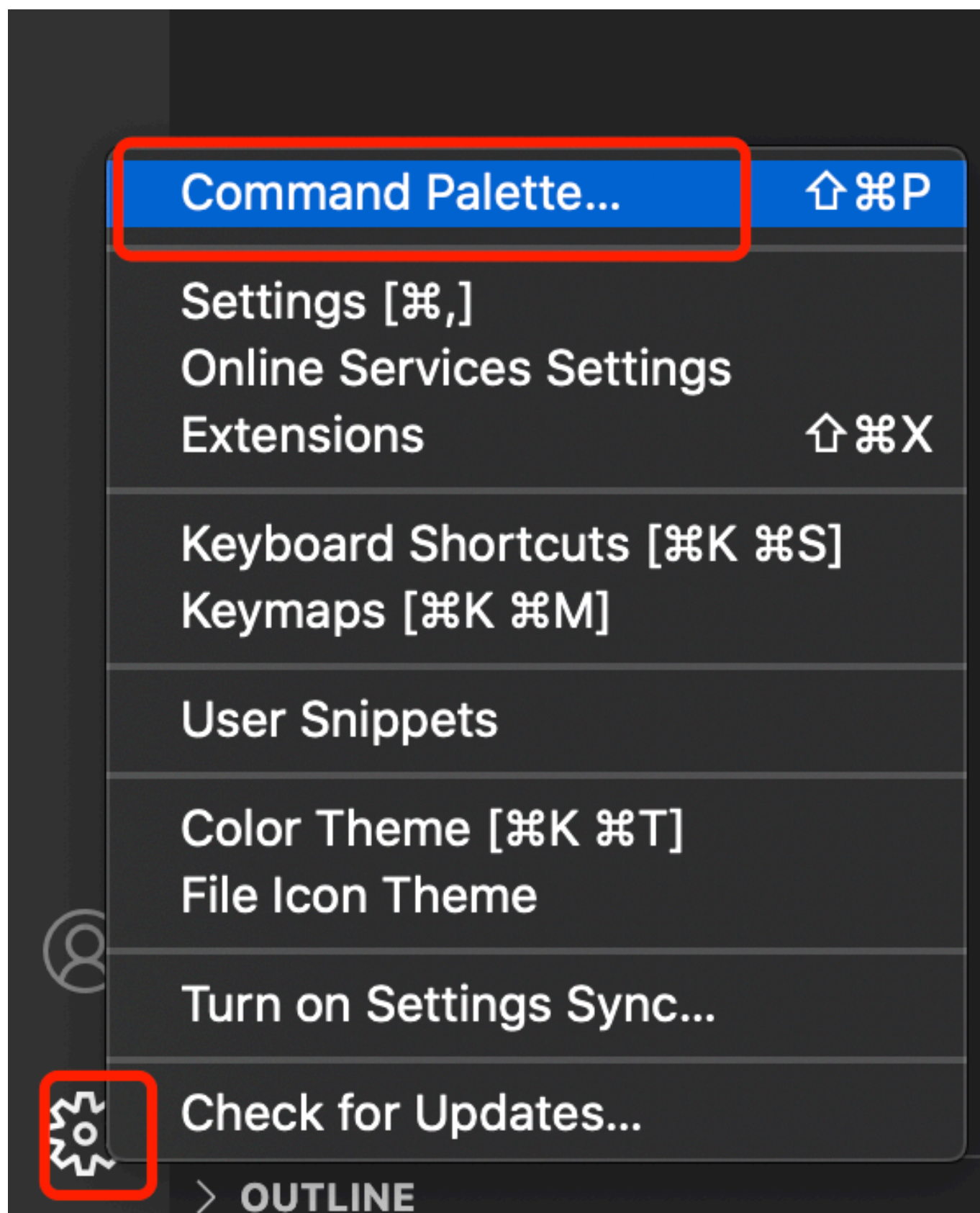
VSCode 是目前很流行的一款免费开发工具，IRIS也支持使用其进行连接和开发，相比Studio 只能在windows 环境使用，Vscode 可以跨平台使用。

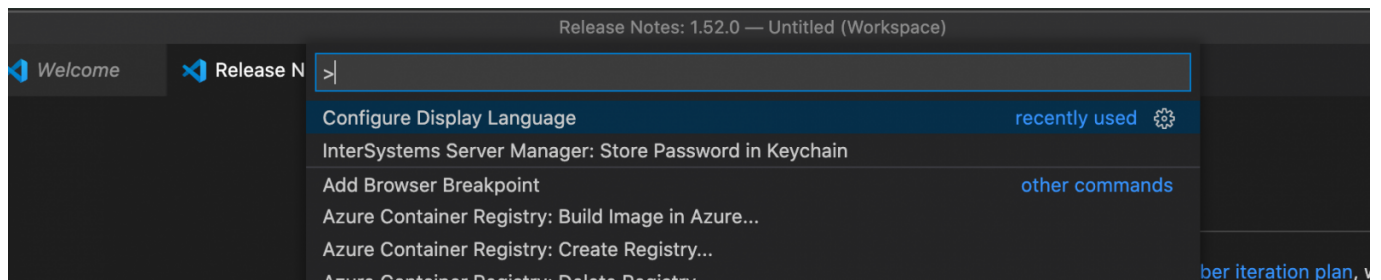
我们传统的工具Studio 是连接代码服务器的形式，不能便捷的使用目前流行的有本地代码的版本控制工具（如git），但VSCode可以存在本地代码，并且能方便的使用各类存在本地代码的版本控制工具。

以下的内容会帮助大家来配置使用VSCode连接IRIS 进行开发。

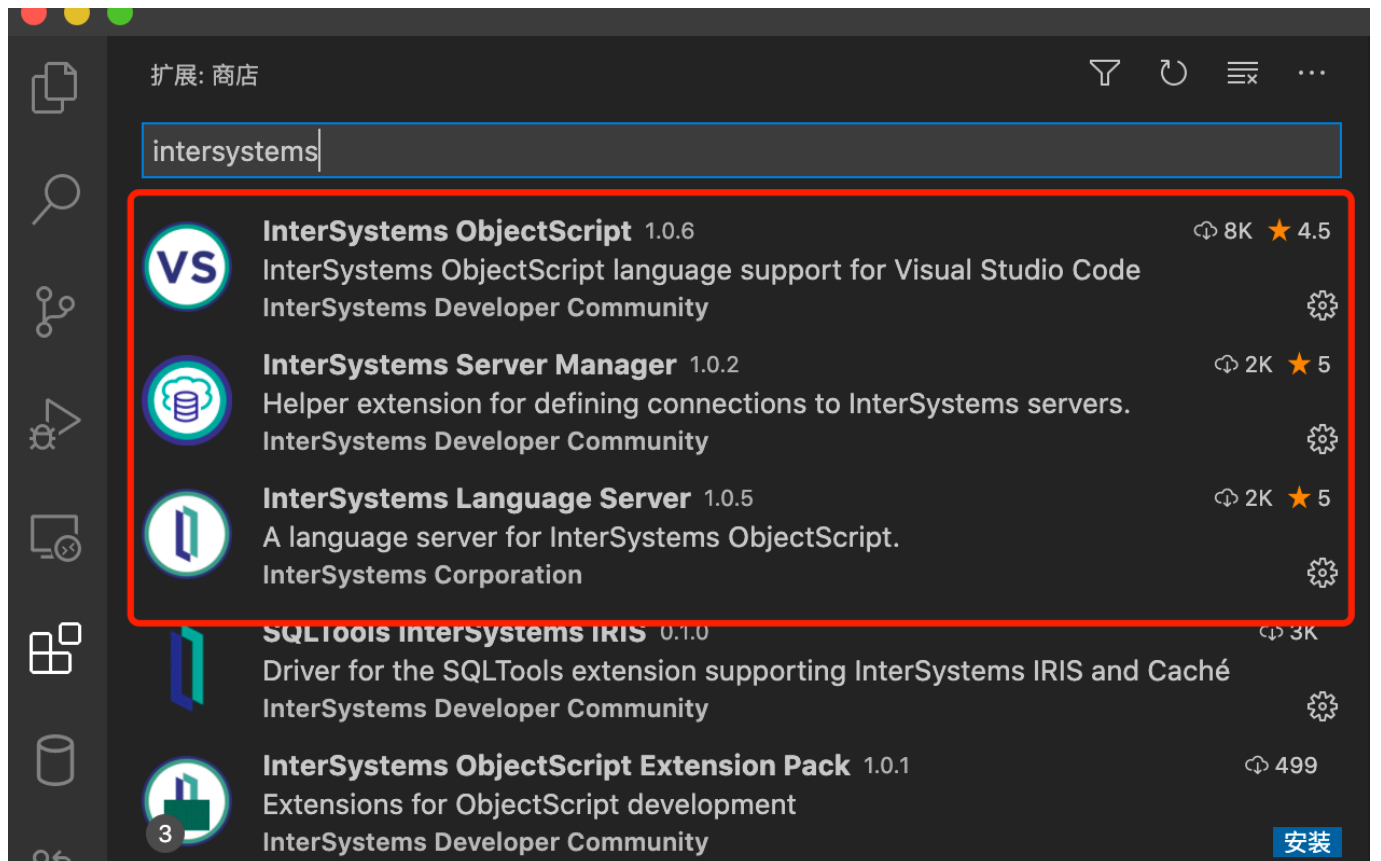
VSCode 可以在微软的官网免费下载<https://code.visualstudio.com/>

如果安装为英文要切换为中文则可以通过Command Palette 中运行Configure Display Language, install another language, 选择中文，再根据提示安装相应的语言插件。





接下来在扩展商店里搜索intersystems 安装如图所见的3个扩展包



VSCode目前有两种方式可以链接IRIS：

1. 直接连接代码服务器进行编辑
2. 使用客户端方式连接，将代码下载到本地，便于使用有本地代码的版本控制工具

注：两种连接方式的连接端口都为服务器web port

下面具体讲解两种连接方式。

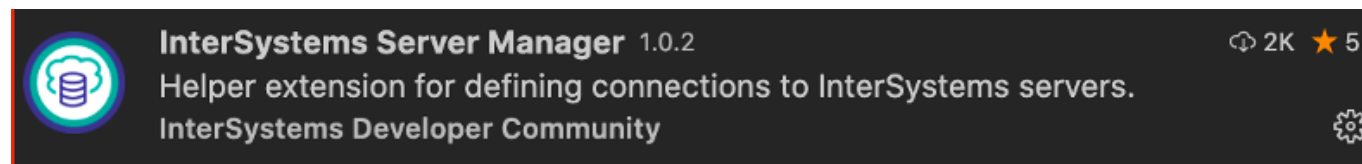
1. 直接连接代码服务器进行编辑

其总体的使用方式和studio相似，直接连接服务器进行代码编辑。在vscode中可以使用全局只读的方式来浏览代码，从而避免误操作修改了代码。

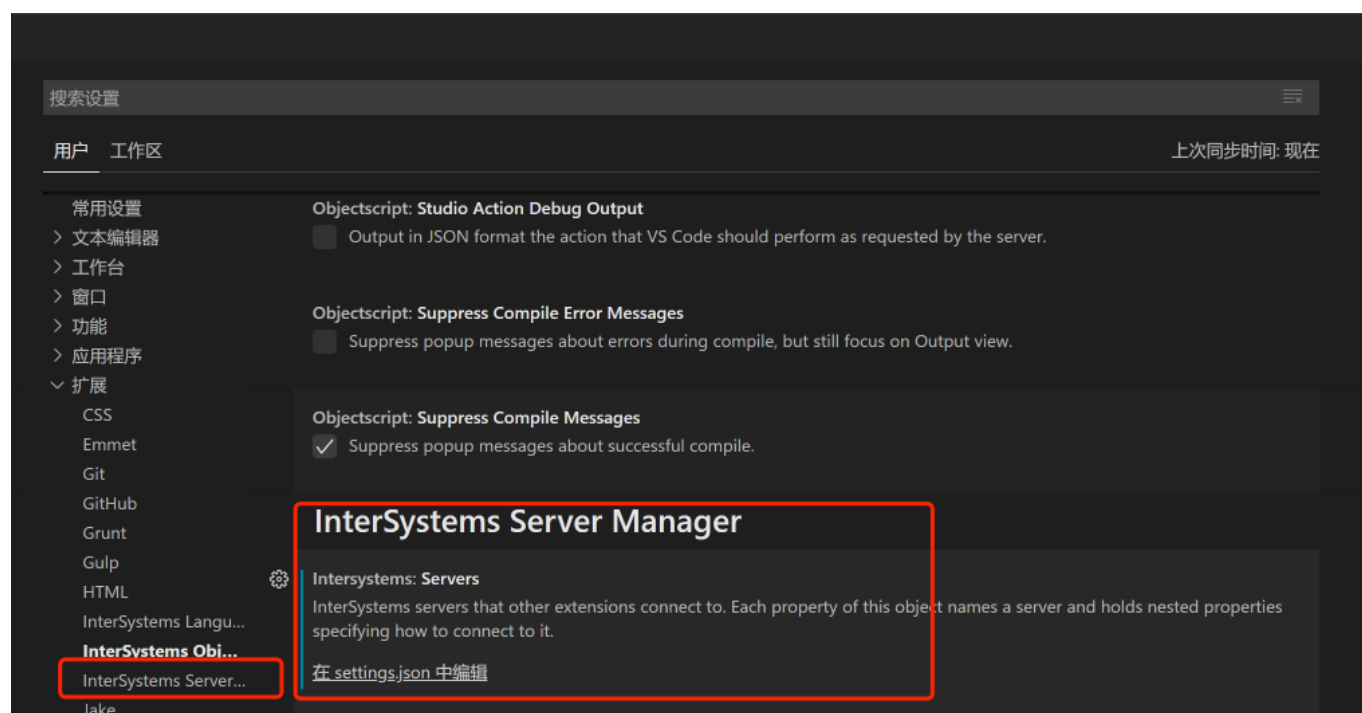
如果会经常使用这个服务器，可以考虑先建立一个工作区，将设置保存在工作区，当然直连服务器的模式

下，建立服务器不是必选项。

如果我们正确的安装了之前提到的三个扩展，其中下面扩展会帮助我们连接IRIS代码服务器，并保存在VS Code的全局 settings.json配置文件中：



进入VSCode 设置中找到“ 扩展 ”下的InterSystems Server Manager，进入设置可见安装自带的服务器连接字符串可以依据用户需要进行添加和修改

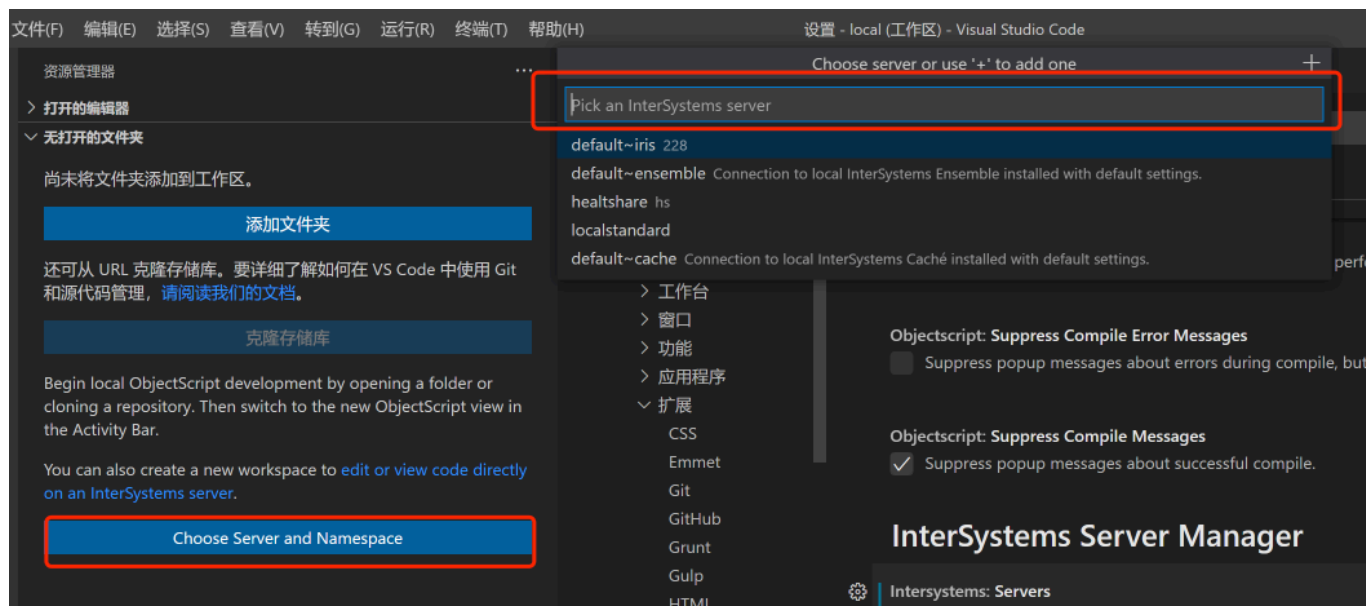


```
C: > Users > jeffliu > AppData > Roaming > Code > User > {} settings.json > {} intersystems.s
1  {
2      "workbench.colorTheme": "InterSystems Default Dark",
3      "[xml]": {
4          "editor.defaultFormatter": "DotJoshJohnson.xml"
5      },
6      "intersystems.servers": {
7
8          "default~iris": {
9              "webServer": {
10                  "scheme": "http",
11                  "host": "10.160.1.82",
12                  "port": 52773
13              },
14              "username": "superuser",
15              "description": "228"
16          },
17      "healthshare": {
18          "webServer": {
19              "scheme": "http",
20              "host": "127.0.0.1",
21              "port": 52773
22          },
23          "username": "superuser",
24          "description": "hs"
25      },
26      "default~ensemble": {
27          "webServer": {
28              "scheme": "http",
29              "host": "127.0.0.1",
30              "port": 54772
31          },
32          "description": "Connection to local InterSystems Ense
33      },
34      "/default": "default~iris",
35      "localstandard": {
36          "webServer": {
37              "scheme": "http",
38              "host": "127.0.0.1",
39              "port": 52774
40          },
41          "username": "superuser"
42      }
43  },
44      "terminal.integrated.detectLocale": "on"
45  }
46  }
```

不可使用大写字母

配置完成可以回到工作区界面进行连接

选择Choose Server and Namespaces, 从右边的服务器列表中选择要连接的服务器, 也可以使用“+”来根据提示信息来添加新的服务器。

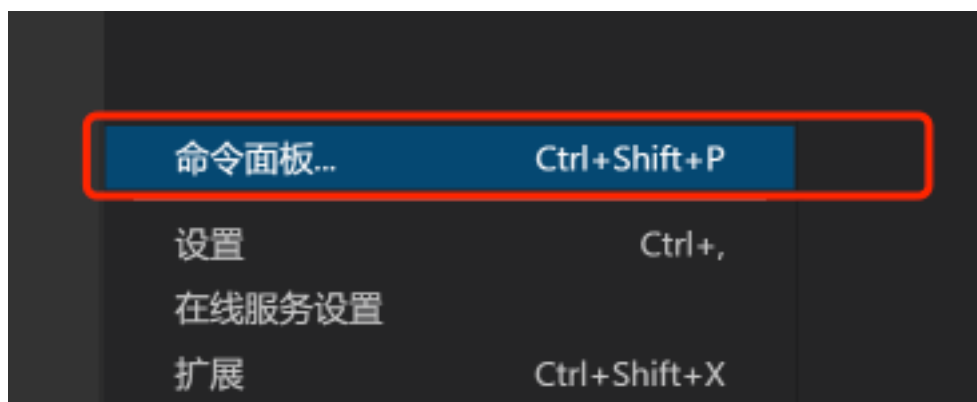


选择服务器之后会提示输入密码（如果没有在配置中写明登录用户，也会提示输入登录用户）验证成功后可以选择想查看的命名空间，最后按照提示选择是以编辑或者只读方式来查看代码。

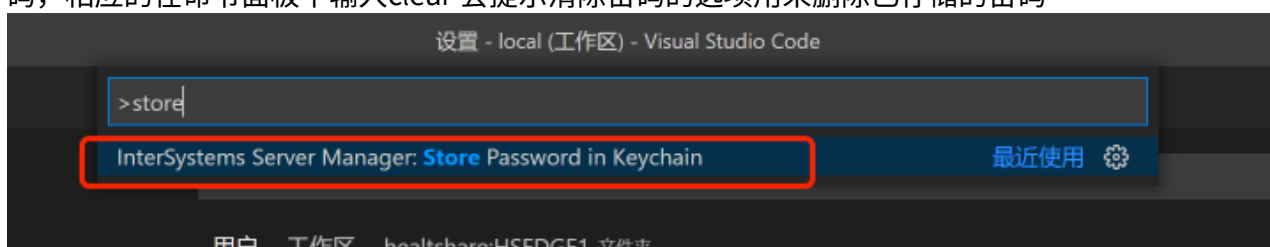
在服务器直连的模式下，可以将多个服务器的多个命名空间的代码添加至当前工作区。

目前的连接方式进入之后会频繁的要求登录验证，输入密码，这里实际可以将密码存在VSCode的钥匙链中：

- 1. 打开命令面板

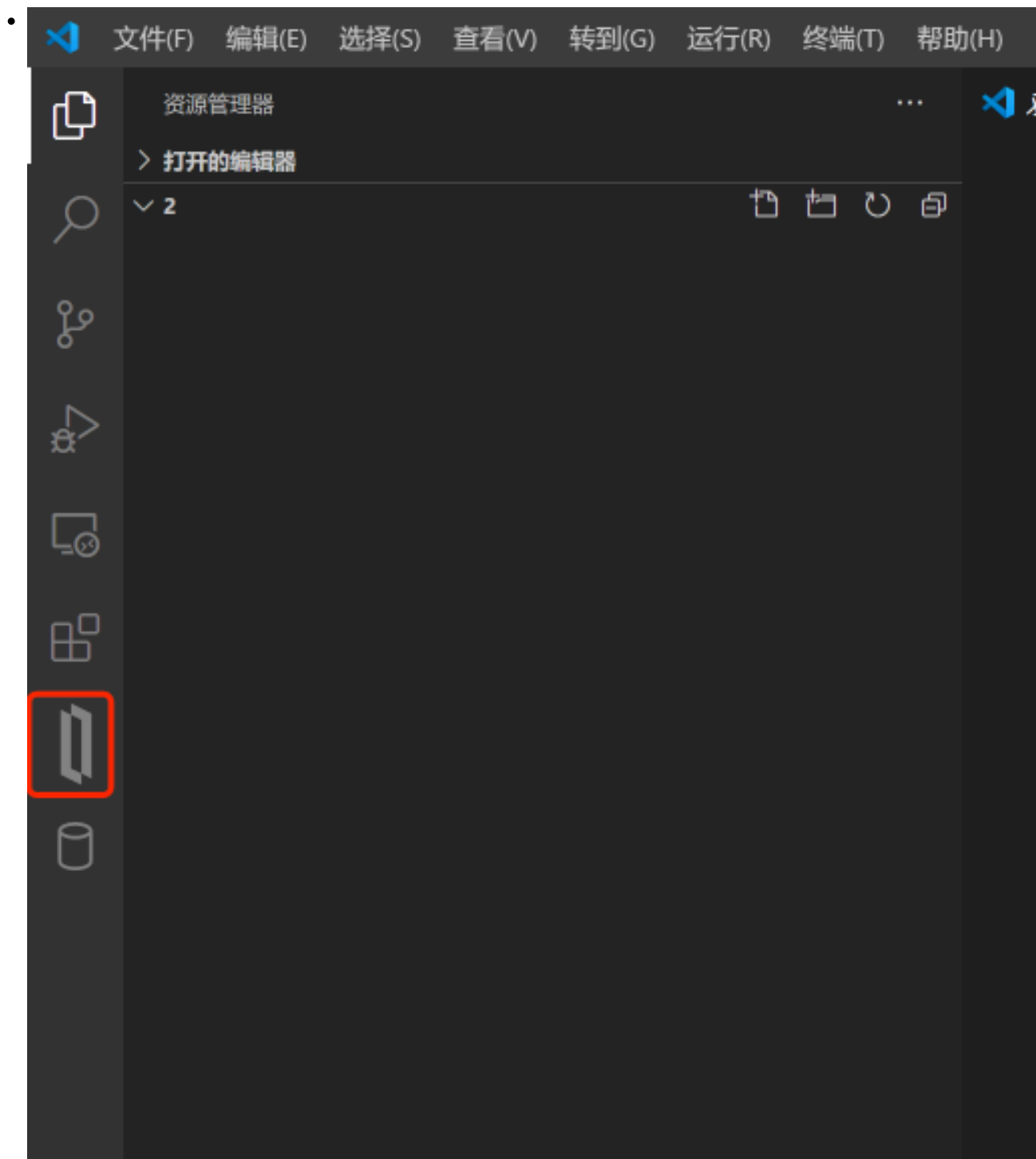


- 2. 输入store 使用下图中的命令则可以为选定的服务器存储密码，但是每个服务器只能保存一个密码，相应的在命令面板中输入clear 会提示清除密码的选项用来删除已存储的密码

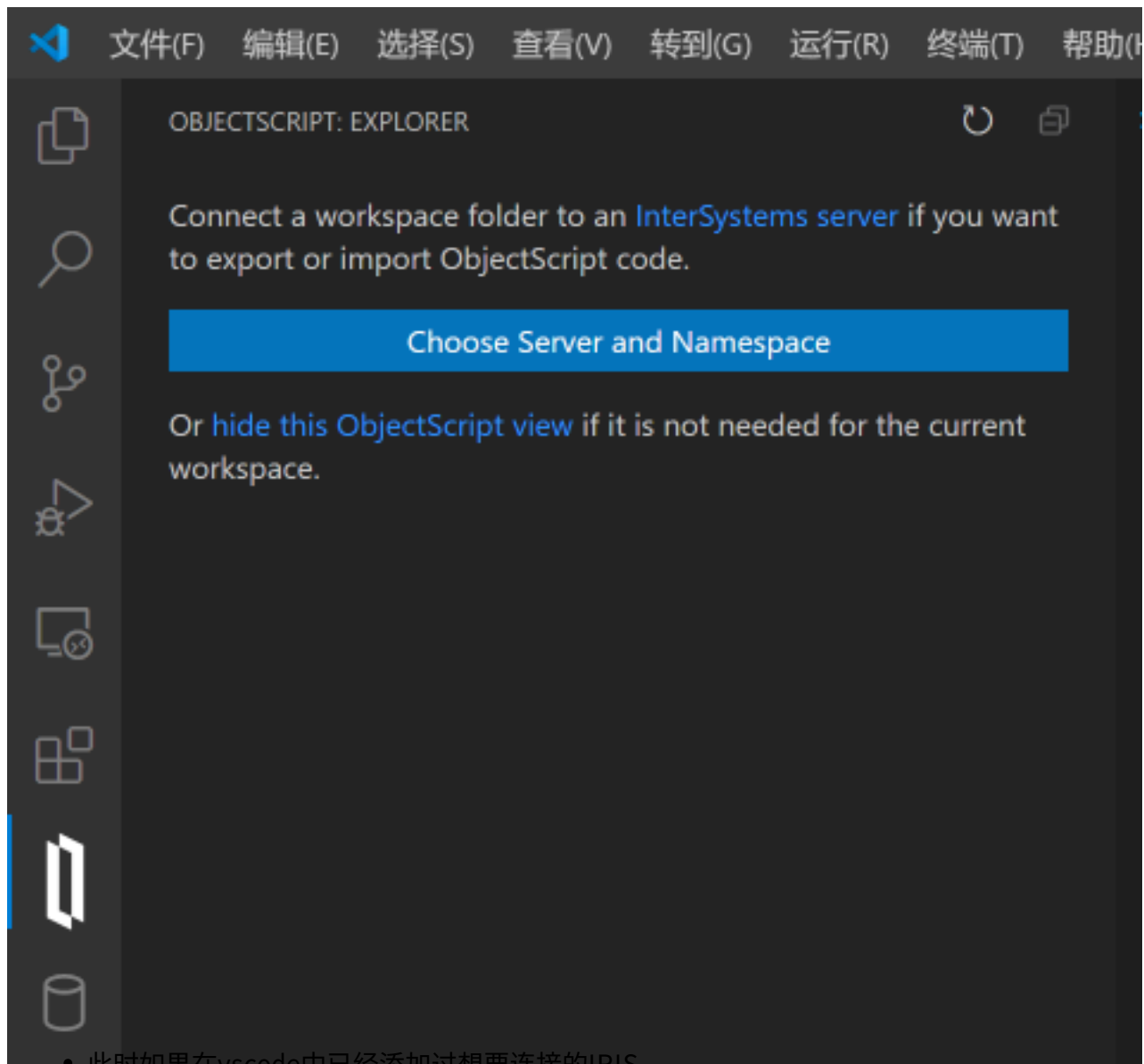


2. 使用客户端方式连接

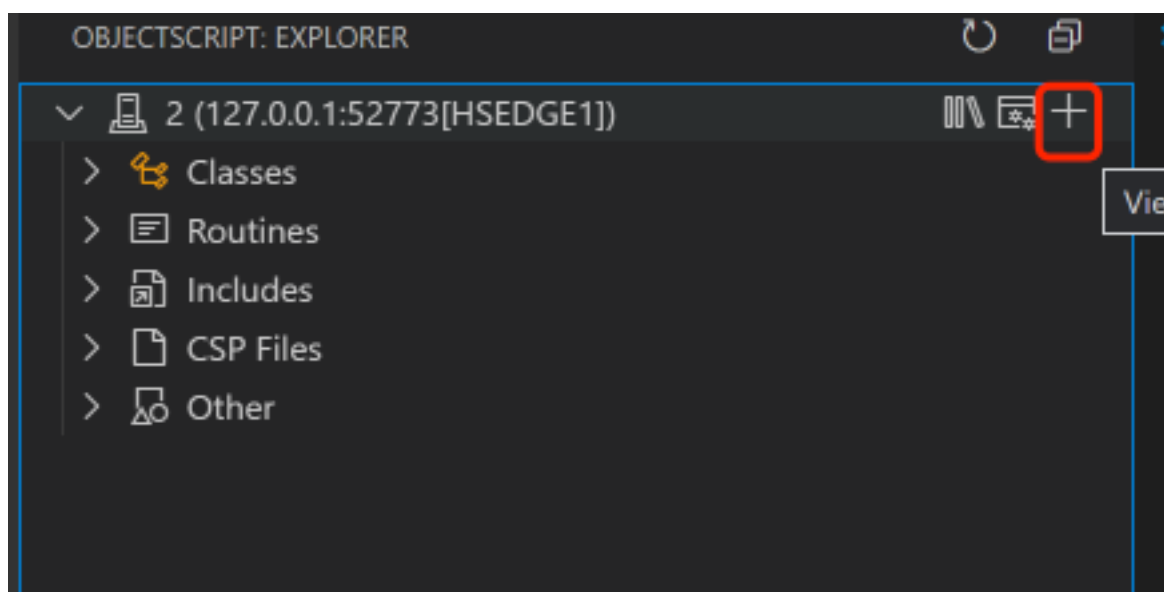
如果使用客户端的连接方式，则必须要在工作区内添加一个文件夹，以及保存此工作区。当添加文件夹到工作区之后侧边栏会多出图示的按钮：



- 点击按钮选择Choose server and namespaces

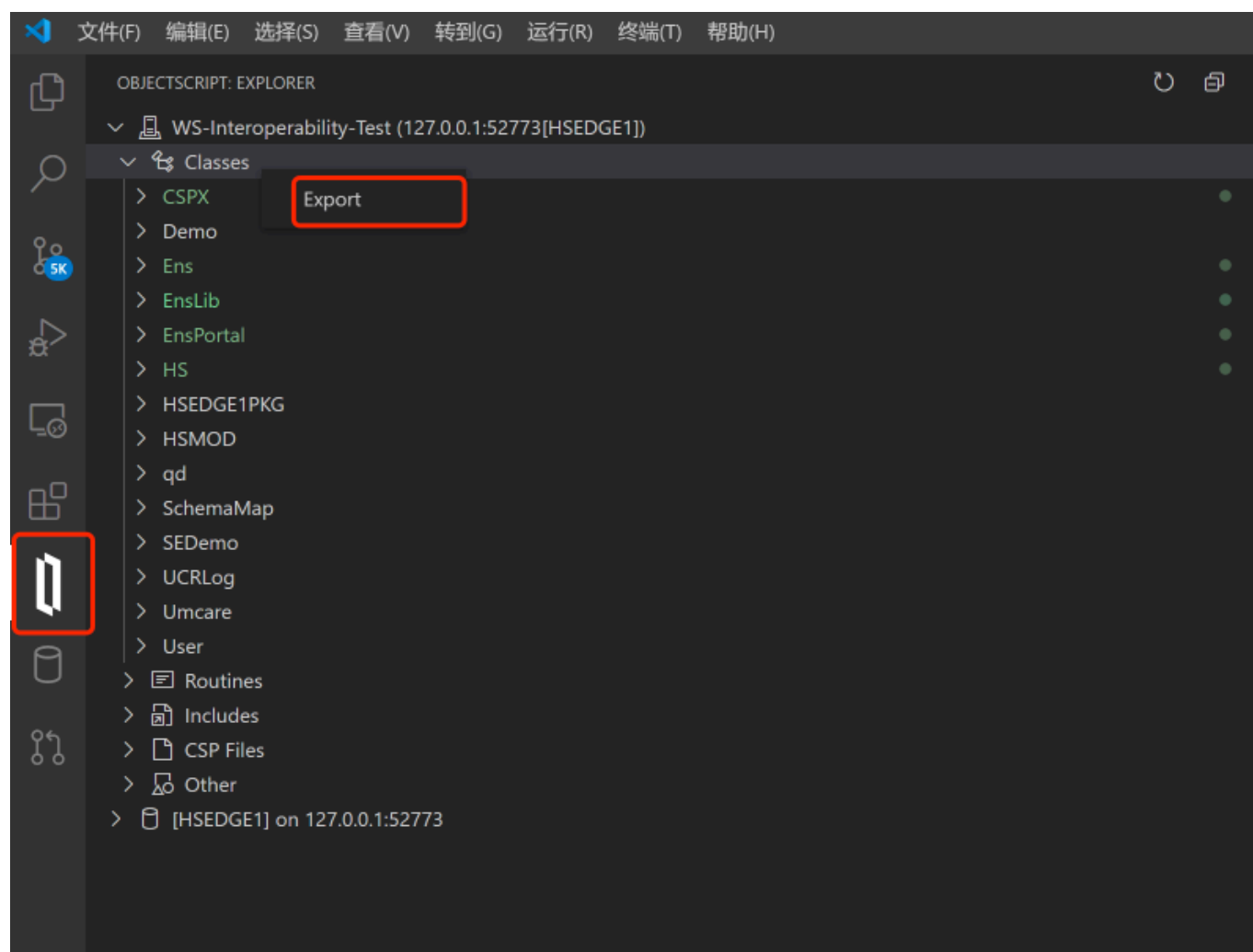


- 此时如果在vscode中已经添加过想要连接的IRIS代码服务器，这里会直接显示已存储的服务器，设置法与之前提到的直连方式相同。
- 连接完成后如果想添加多个命名空间到工作区，可以使用右上角的+号进行添加

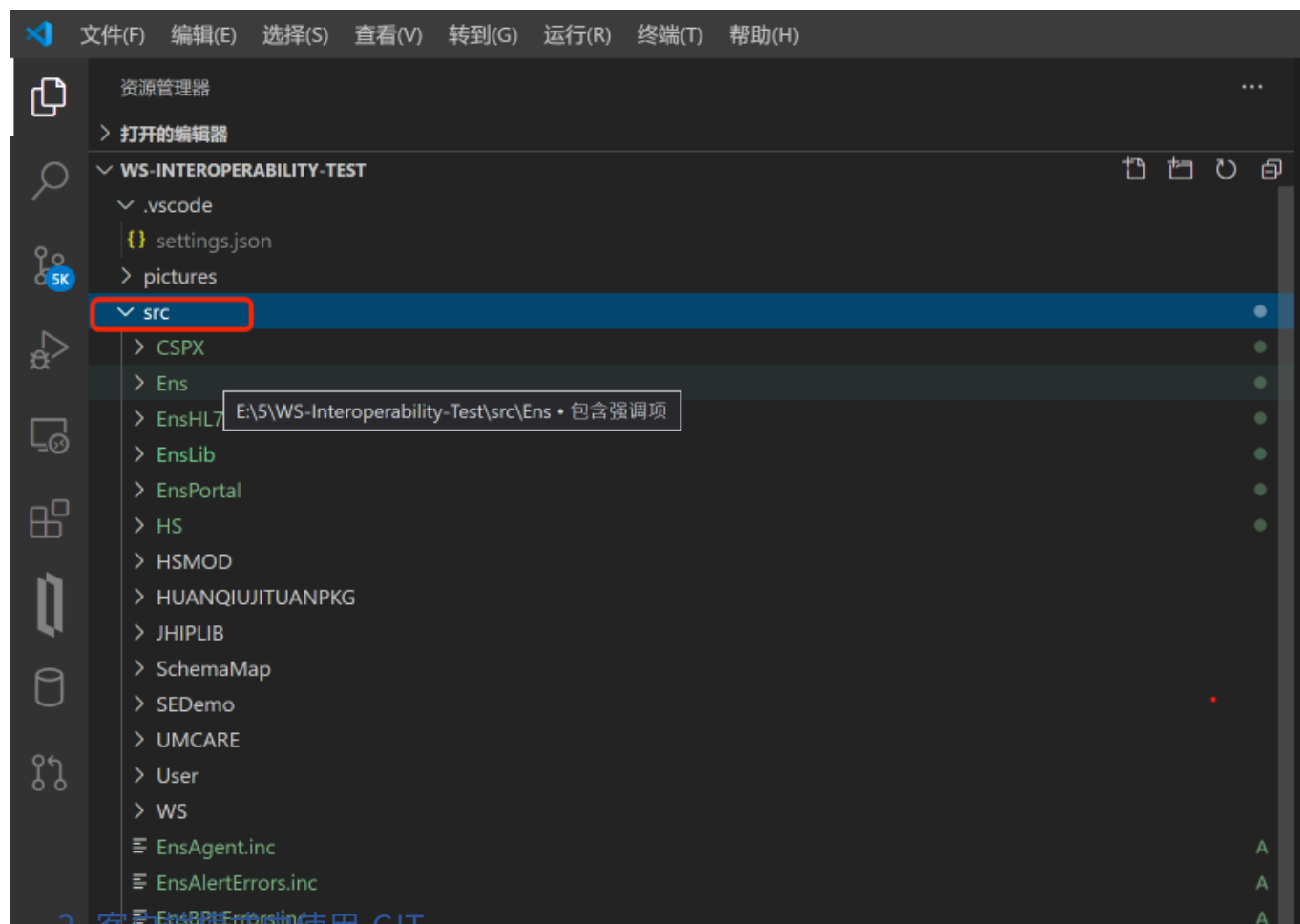


接下来可以在客户端模式连接导出ObjectScript代码到本地。从而进行源代码的版本控制管理。

在objectscript explorer中选择任意的包，或者类右键选择export，则可以将其导出到目前工作区的文件夹下



- 代码会被导出到工作区目录的src文件夹下。

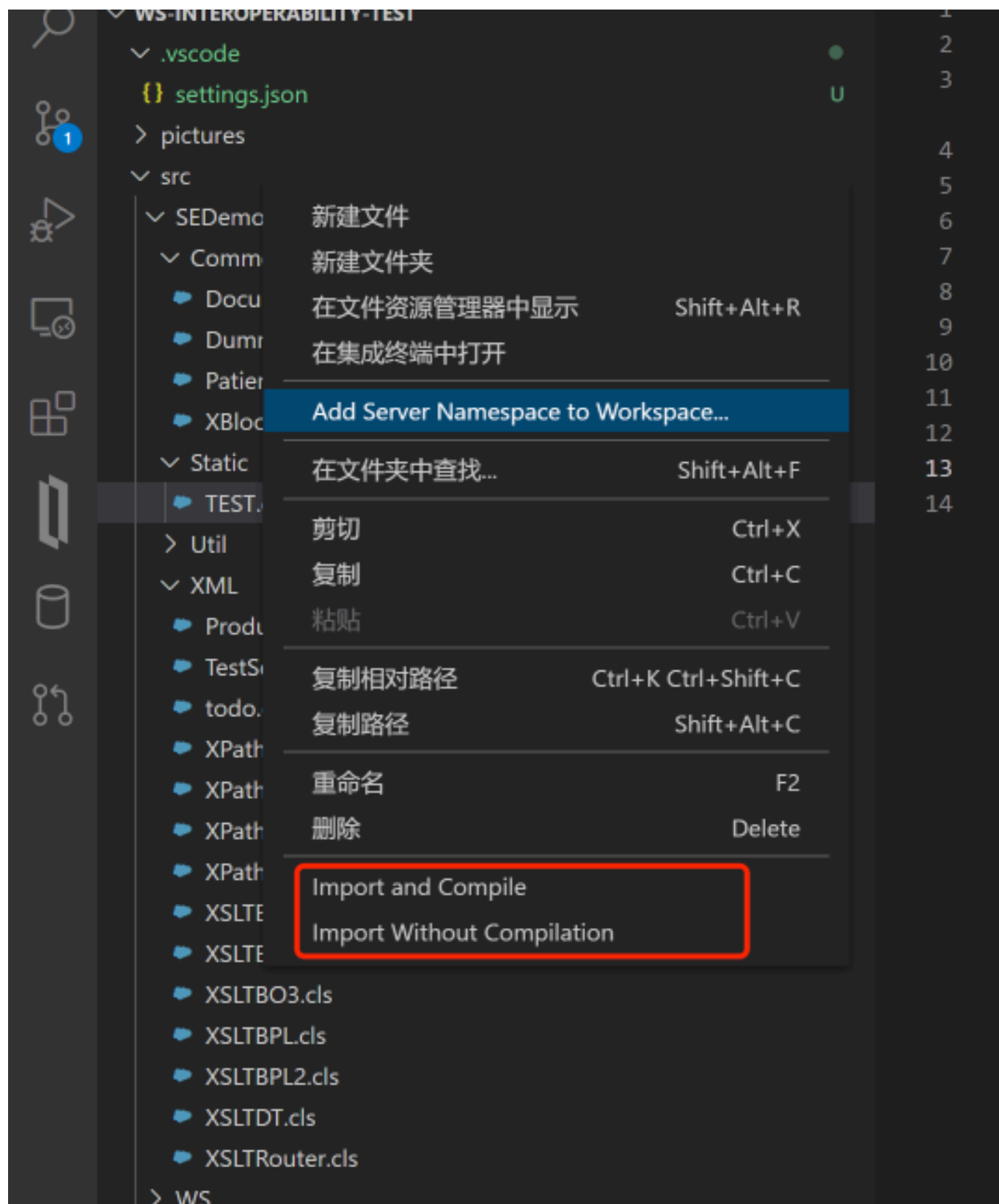


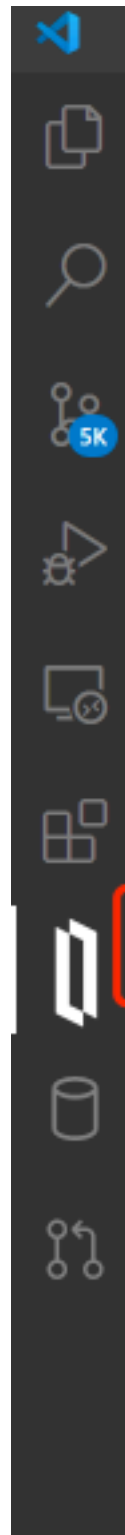
3. 客户端模式中使用 GIT

首先打开一个GIT

Repo文件夹，如果本来然后按照客户端连接方式连接用来开发的服务器。在任何的包或文件上点击import and compile 可以将版本控制的代码导入开发服务器。

注意这时候只要在本机新建文件并且保存，新的修改就会被同步到开发服务器实例上。

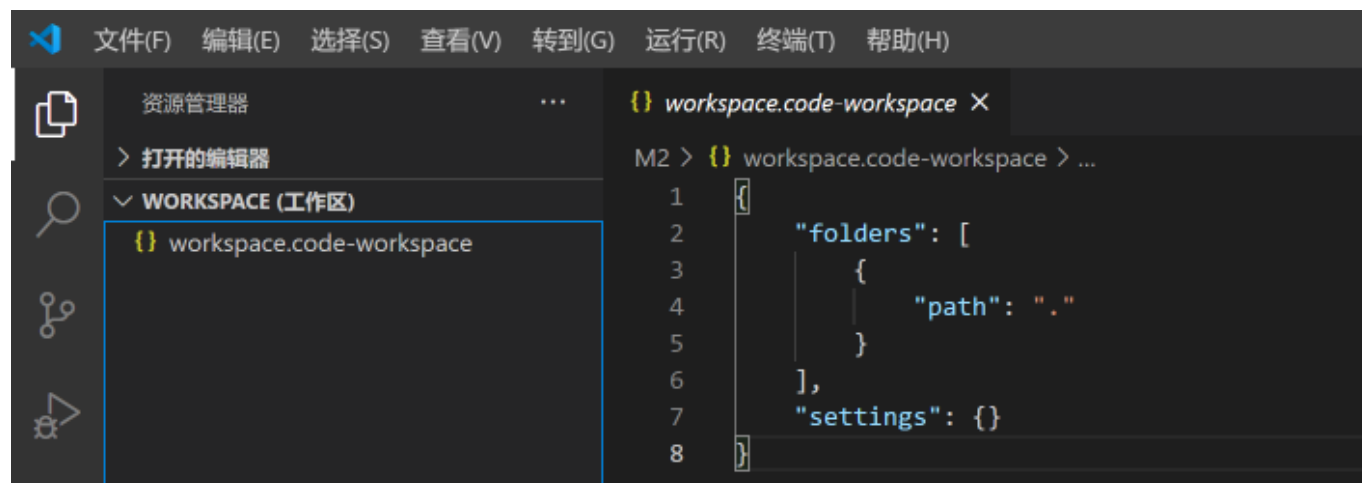




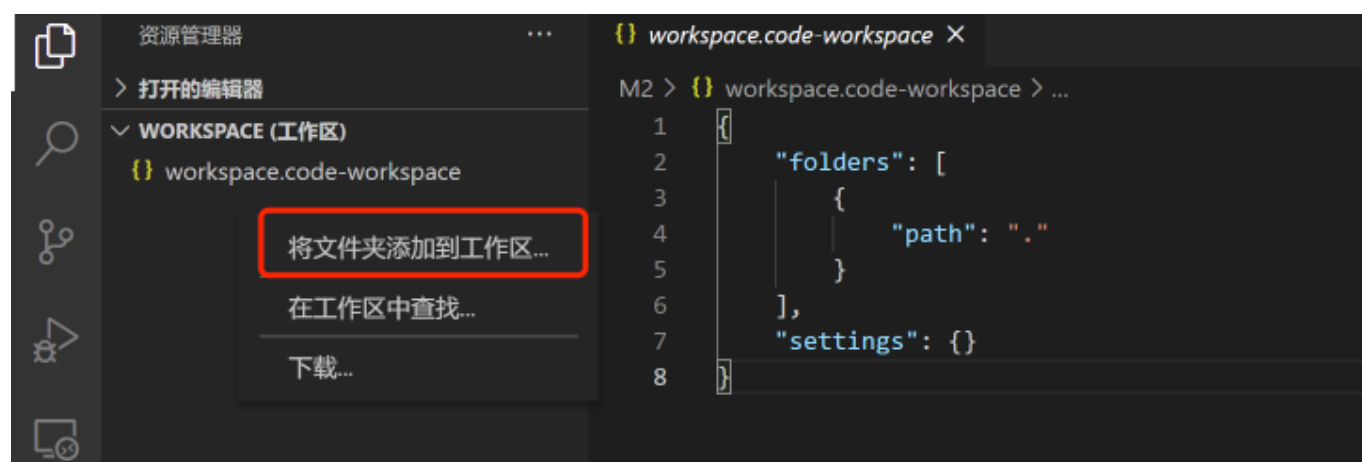
4. 以客户端方式连接时连接多个服务器或多个命名空间的方式：

首先将一个文件夹添加到工作区，并保存工作区，在保存之后的工作区中可以看到如下的工作区配置文件（

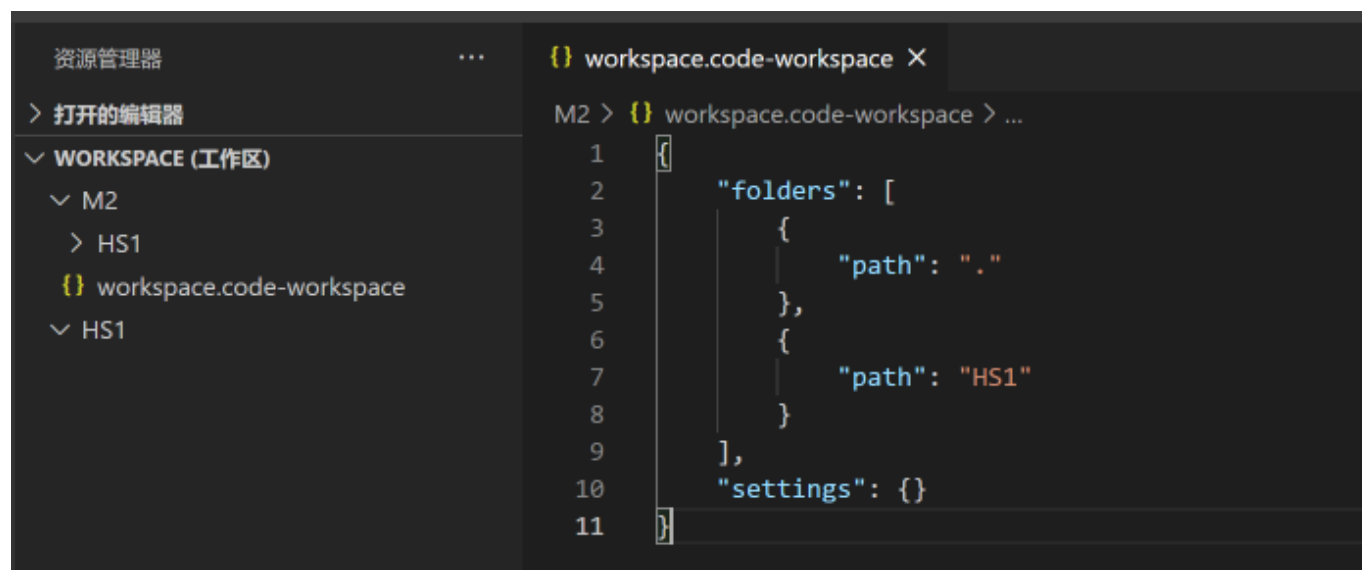
此处我使用的默认工作区配置文件名，实际可以自行定义），其中定义了工作区的目录结构：

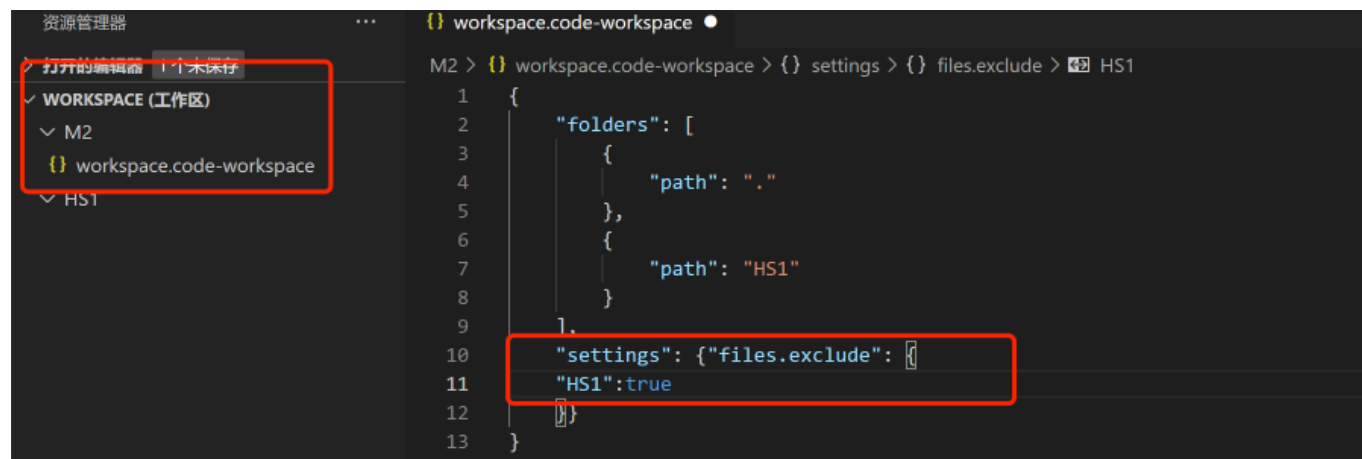


此时我们在这个工作区内添加一个子目录，在工作区的列表右键单击，选择“将文件夹添加到工作区”：

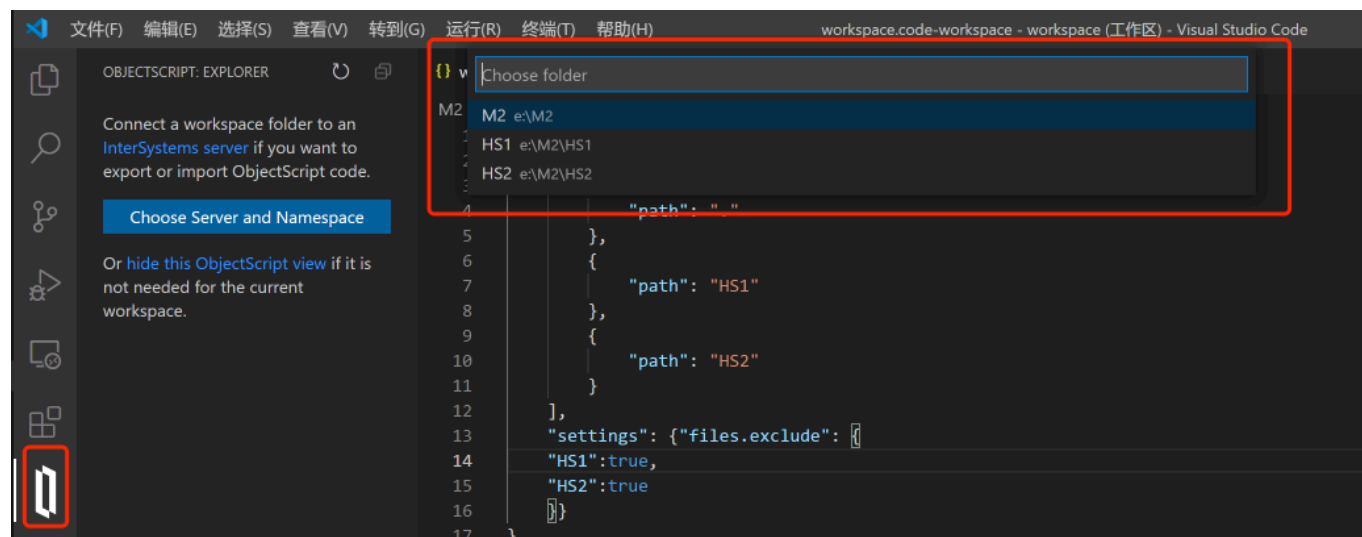


添加后可以看到，在工作区配置文件中增加了新的目录，为了这个显示层级更加清楚，可以在settings中使用files.exclude 来过滤是否显示子目录

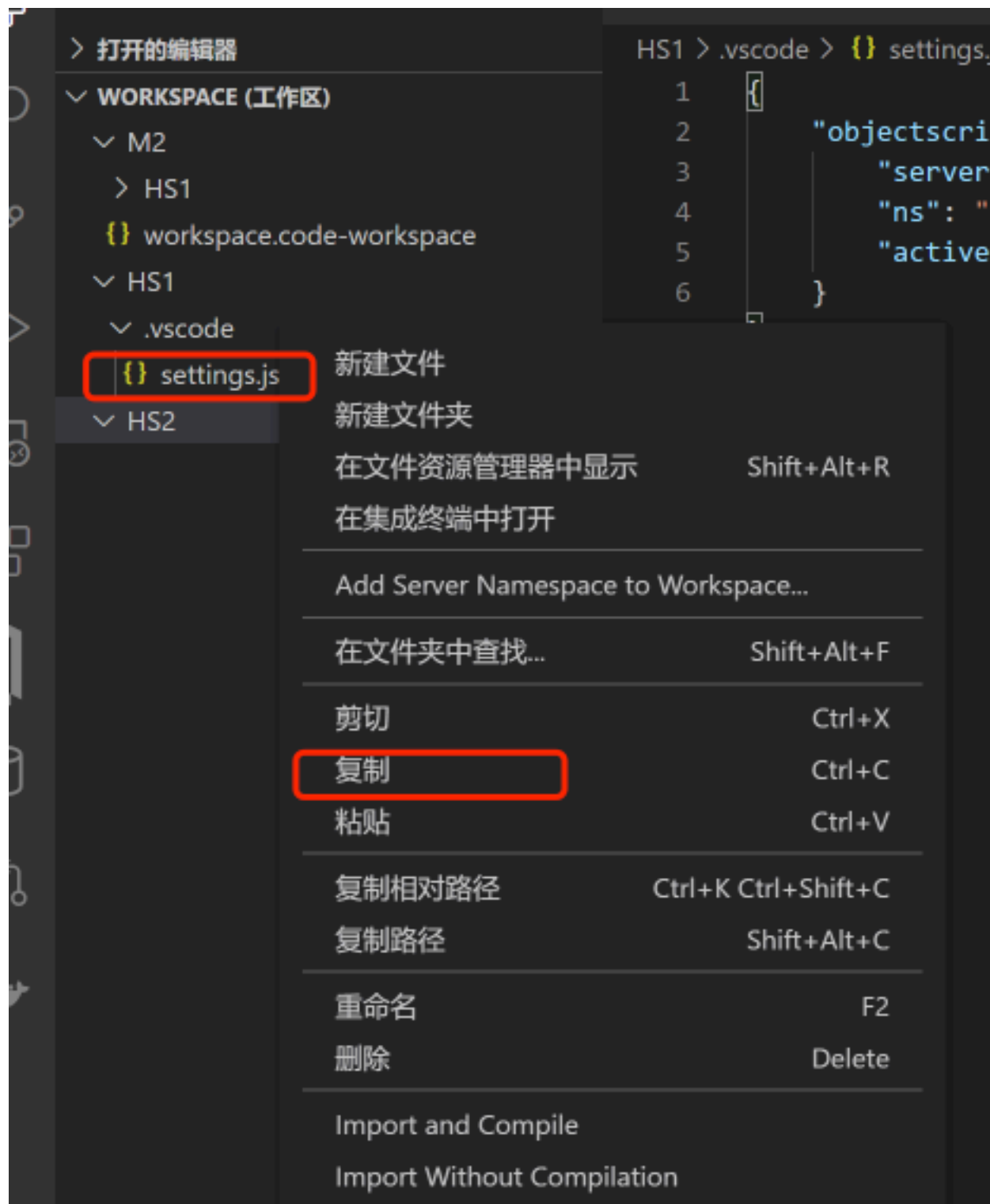




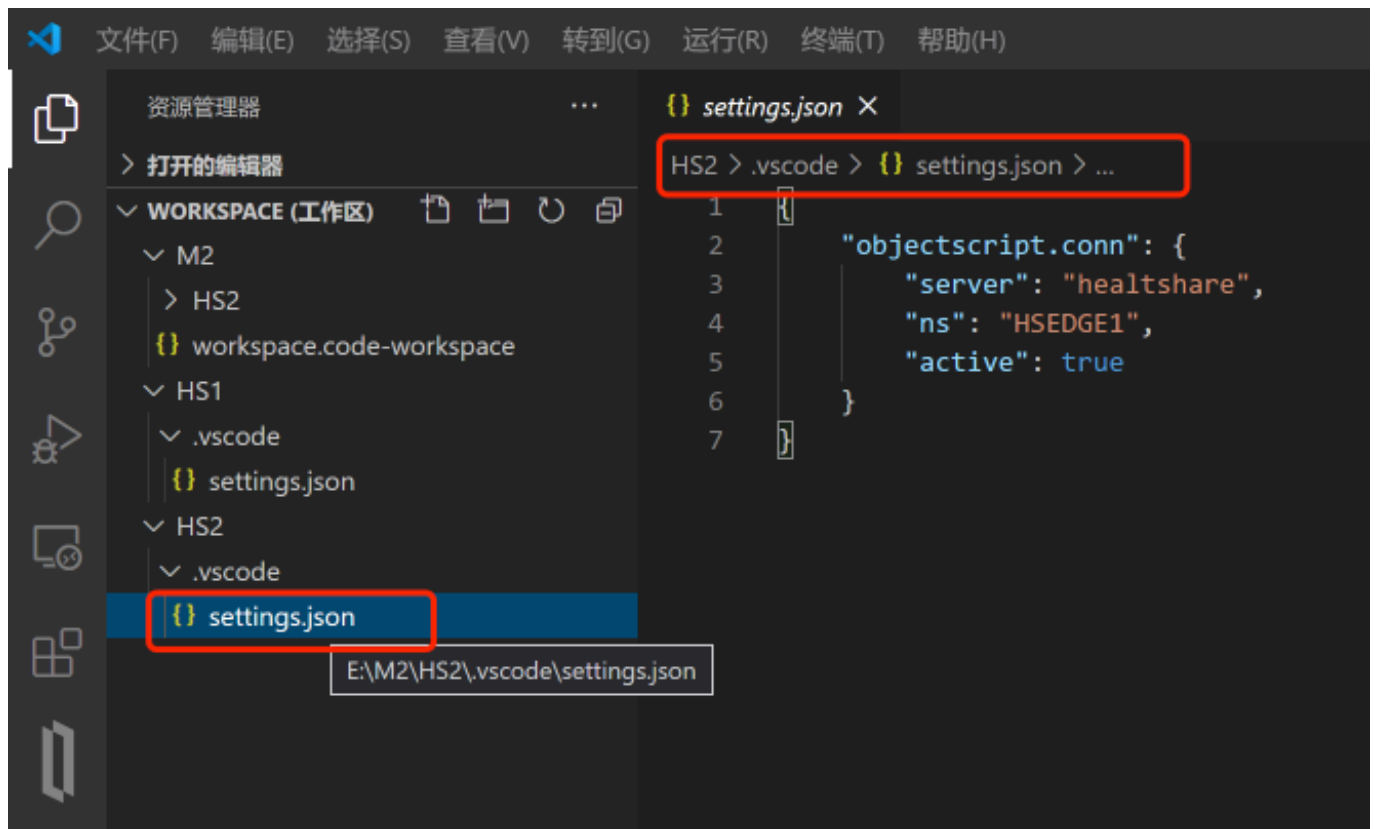
此时子再点击侧边栏中的 object script explorer，则会让用户选择使用哪一个子目录进行连接：



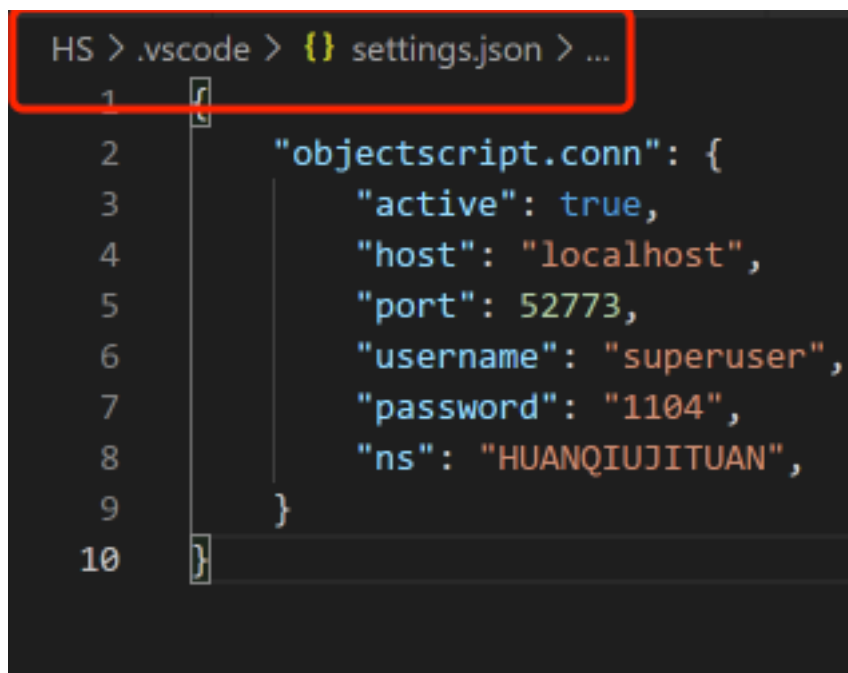
要使用另一个目录连接一个服务器的命名空间时，将第一个已连接的文件夹下.vscode 拷贝到第二个文件夹：



直接对这个文件进行修改，将其配置成为想连接的服务器以及命名空间



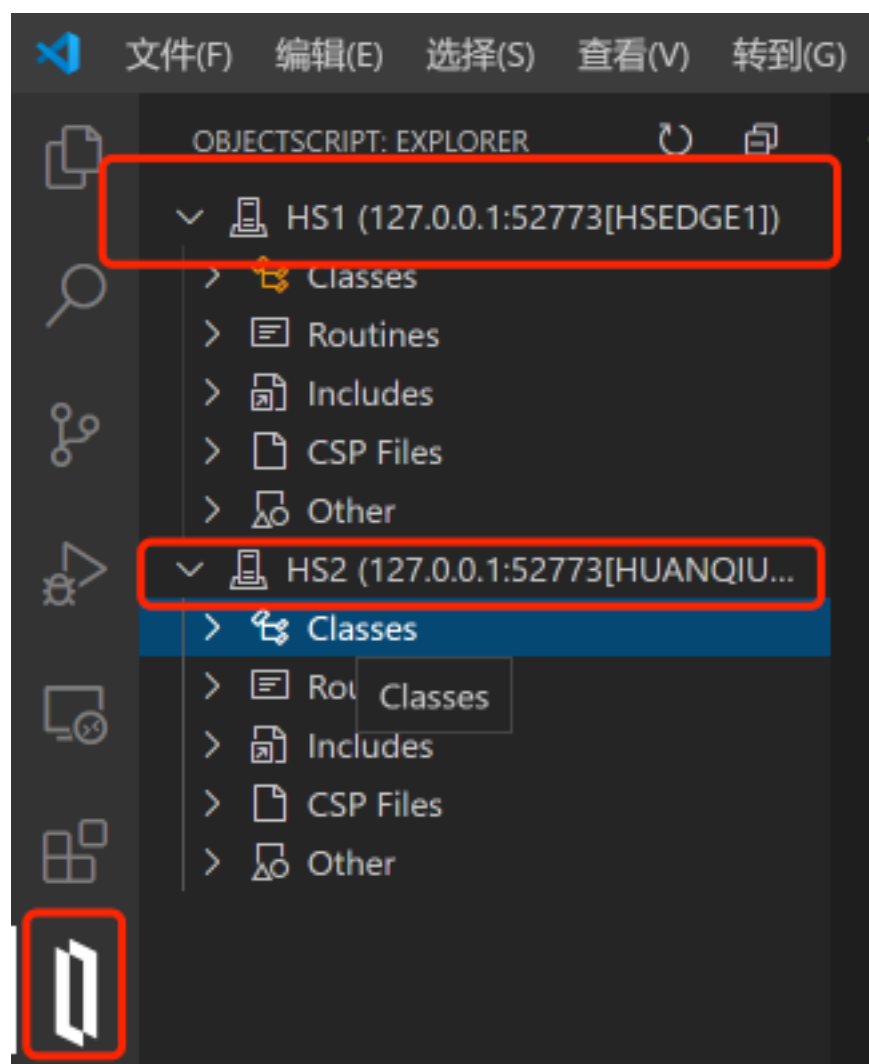
连接配置文件同样支持如下的写法：



此时再回到object script exploer，可见到添加了的服务器。

可见这里服务器的连接和本地代码编译的目标对象只由当前文件夹下.vscode/settings.json控制。

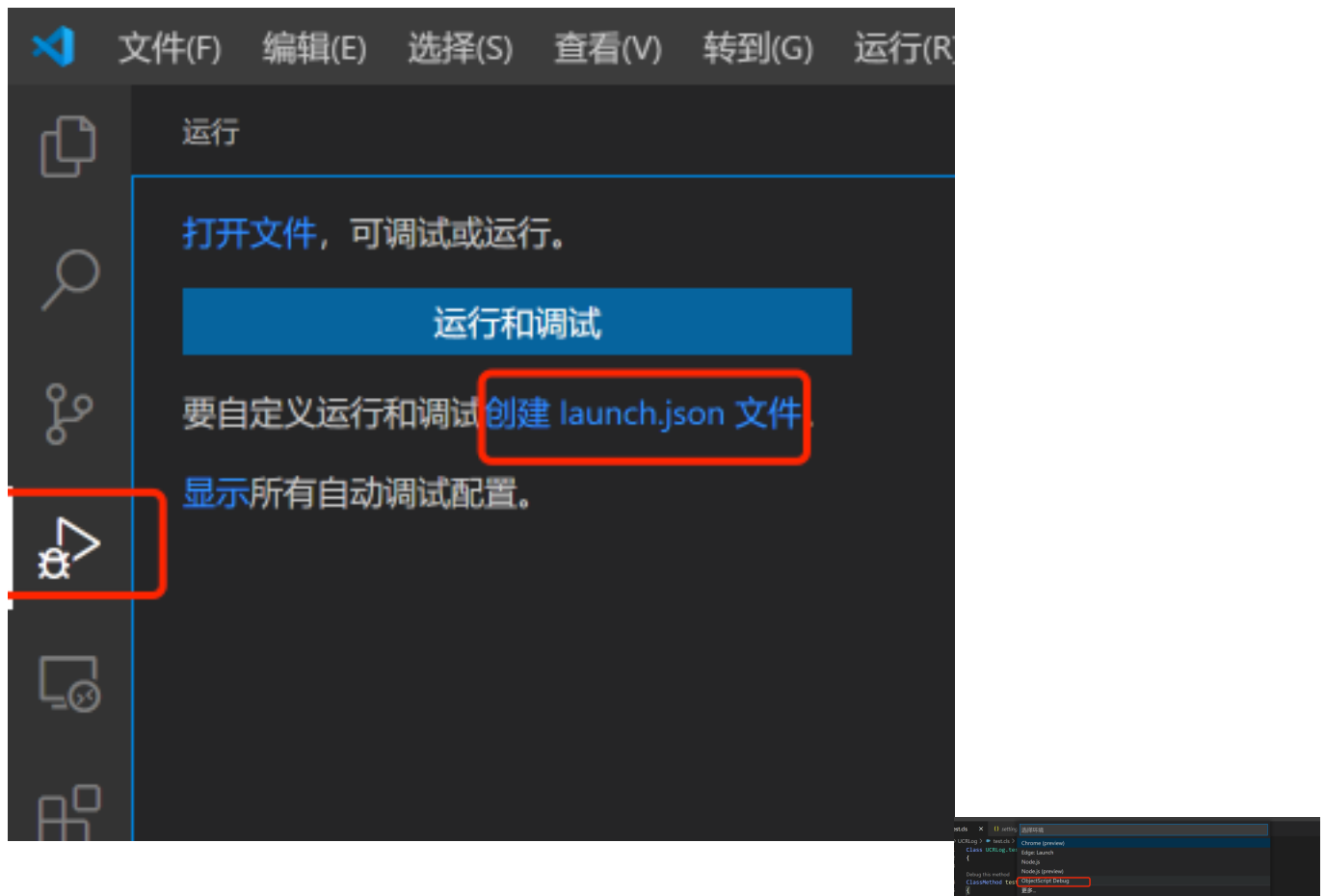
所有的之前提到的配置文件都可以手工编辑，不必一定在vscode中进行添加文件夹和配置连接。



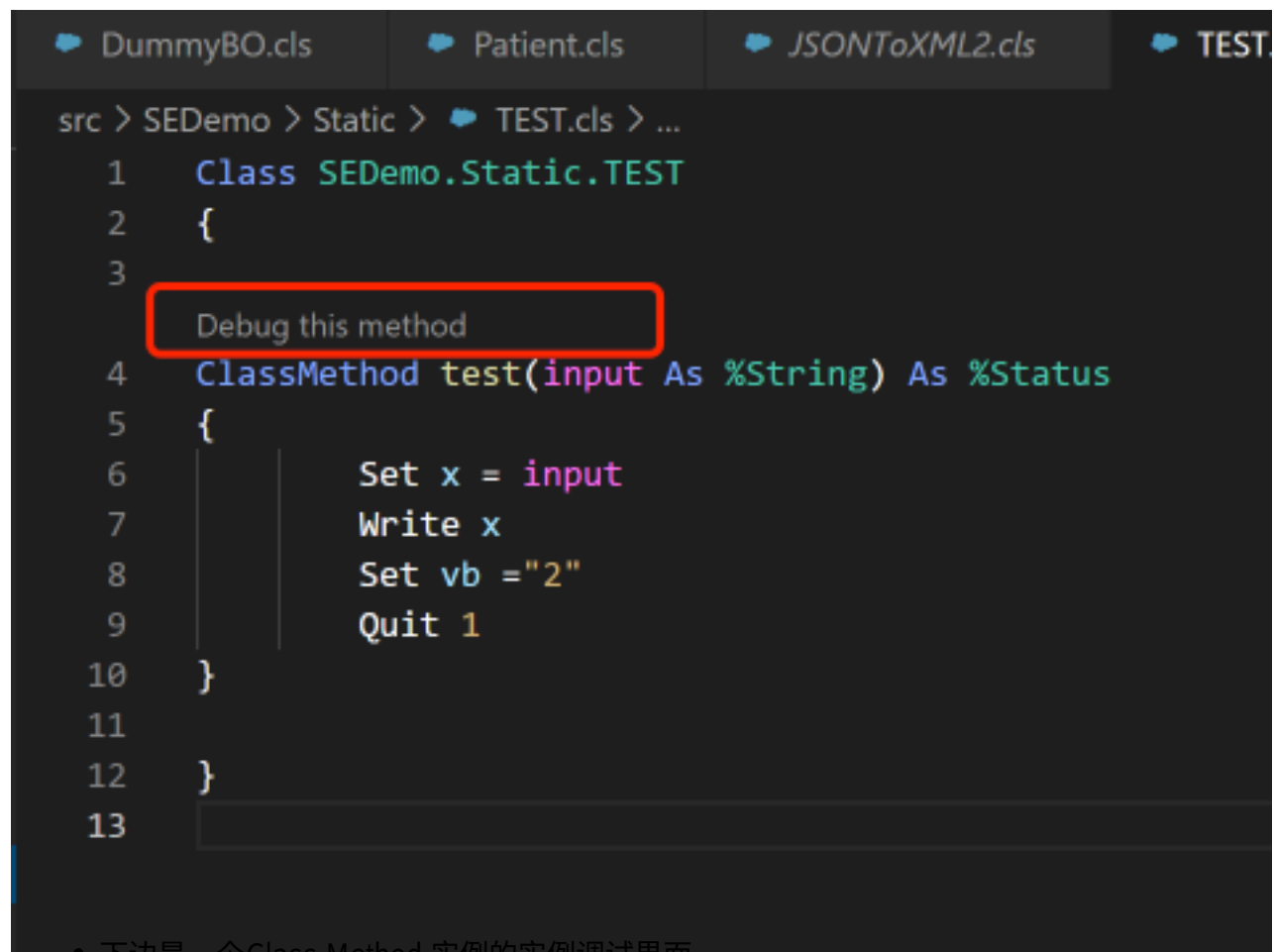
5. 在VSCode中调试

以客户端方式连接时可以进行调试：

在侧边栏选择调试按钮，首次调试应在工作区创建launch.json，并选择ObjectScript Debug



- 对于Class Method,可以直接在类中点击 “ Debug this Method ”

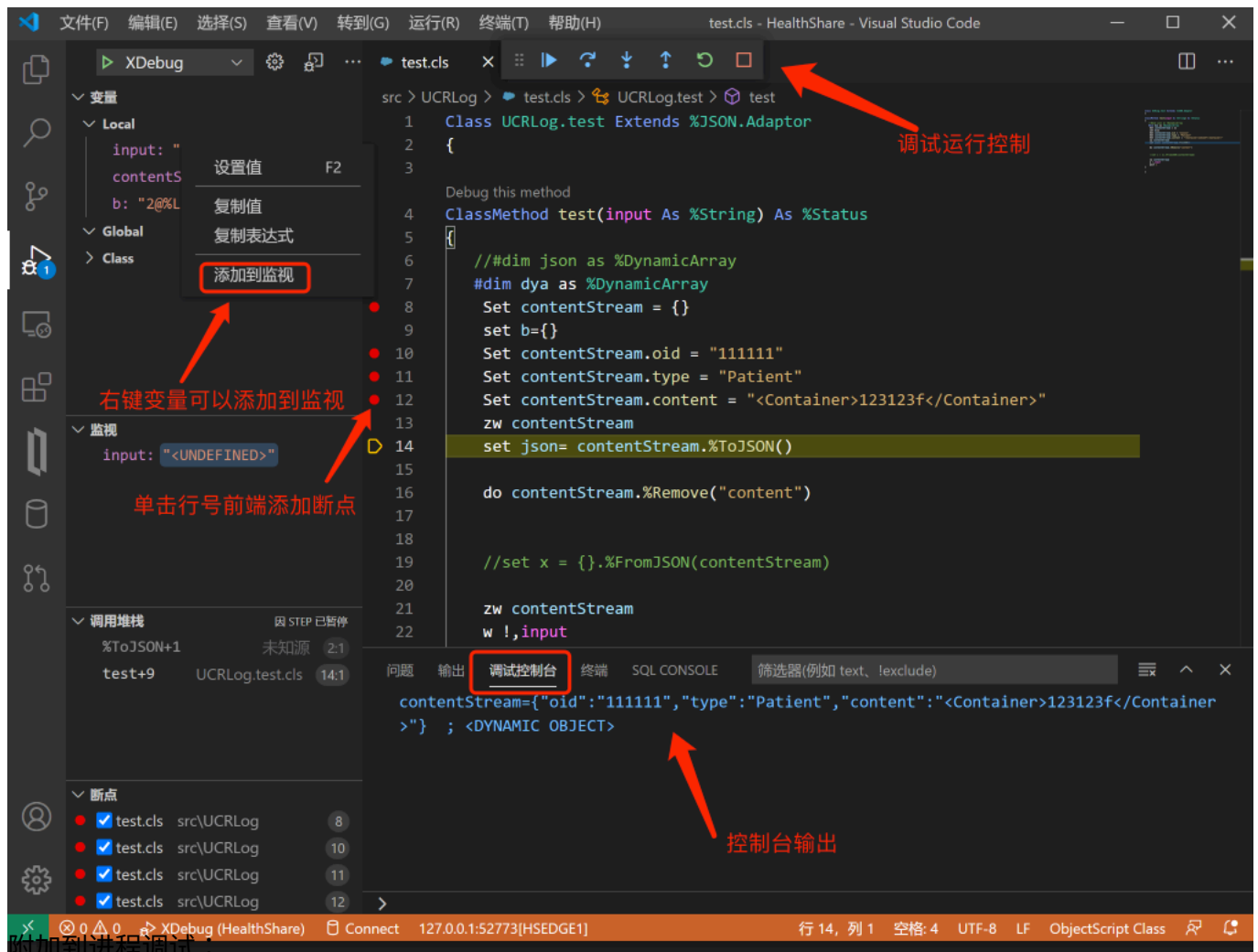


The screenshot shows a VS Code editor with a file explorer at the top containing 'DummyBO.cls', 'Patient.cls', 'JSONToXML2.cls', and 'TEST.'. The main editor area shows the file path 'src > SEDemo > Static > TEST.cls > ...'. The code is as follows:

```
1  Class SEDemo.Static.TEST
2  {
3
4  ClassMethod test(input As %String) As %Status
5  {
6      Set x = input
7      Write x
8      Set vb ="2"
9      Quit 1
10 }
11
12 }
13
```

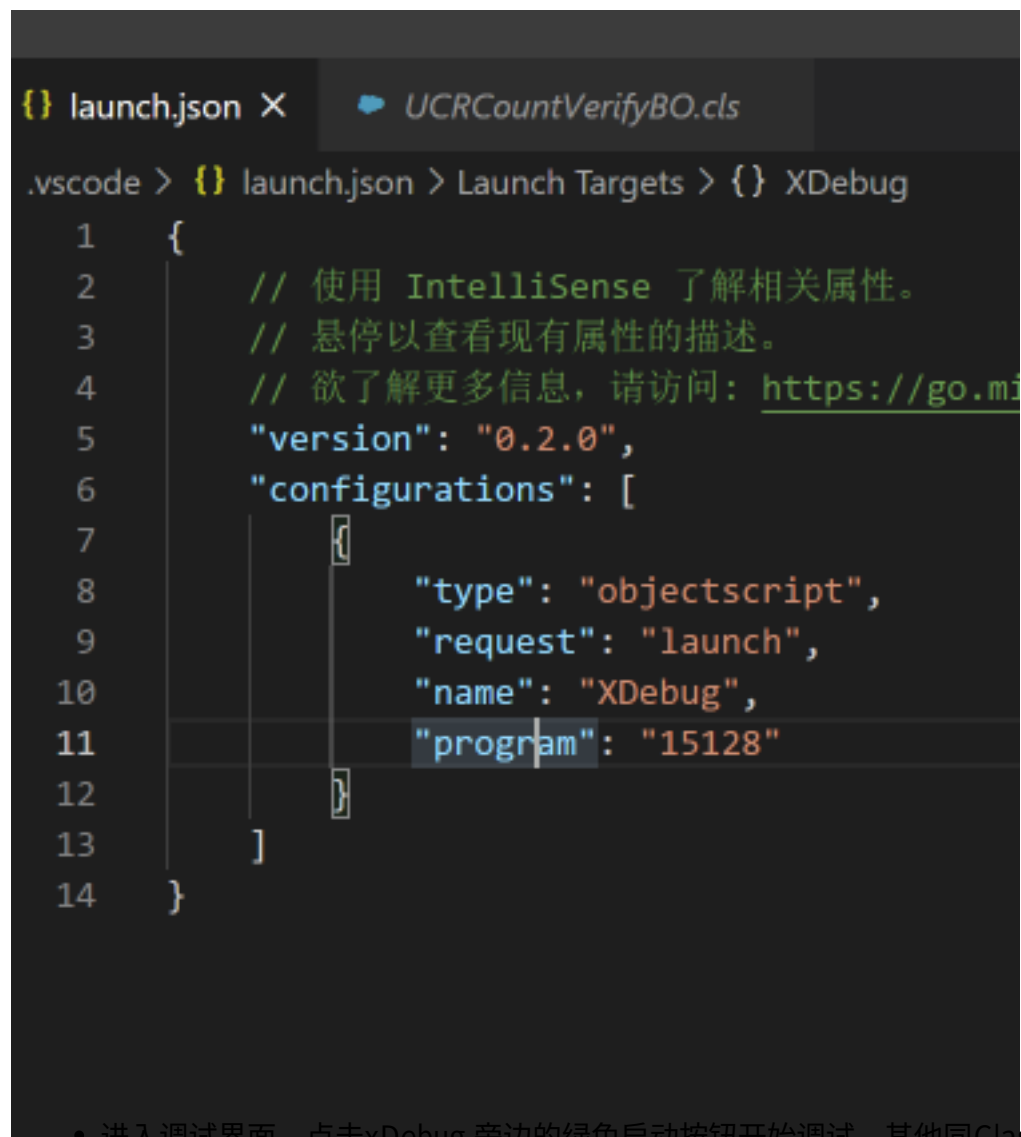
A red rectangle highlights the text 'Debug this method' on line 3, which is a comment or a placeholder for a debug action. Below the code, there is a bullet point:

- 下边是一个Class Method 实例的实例调试界面



附加到进程调试

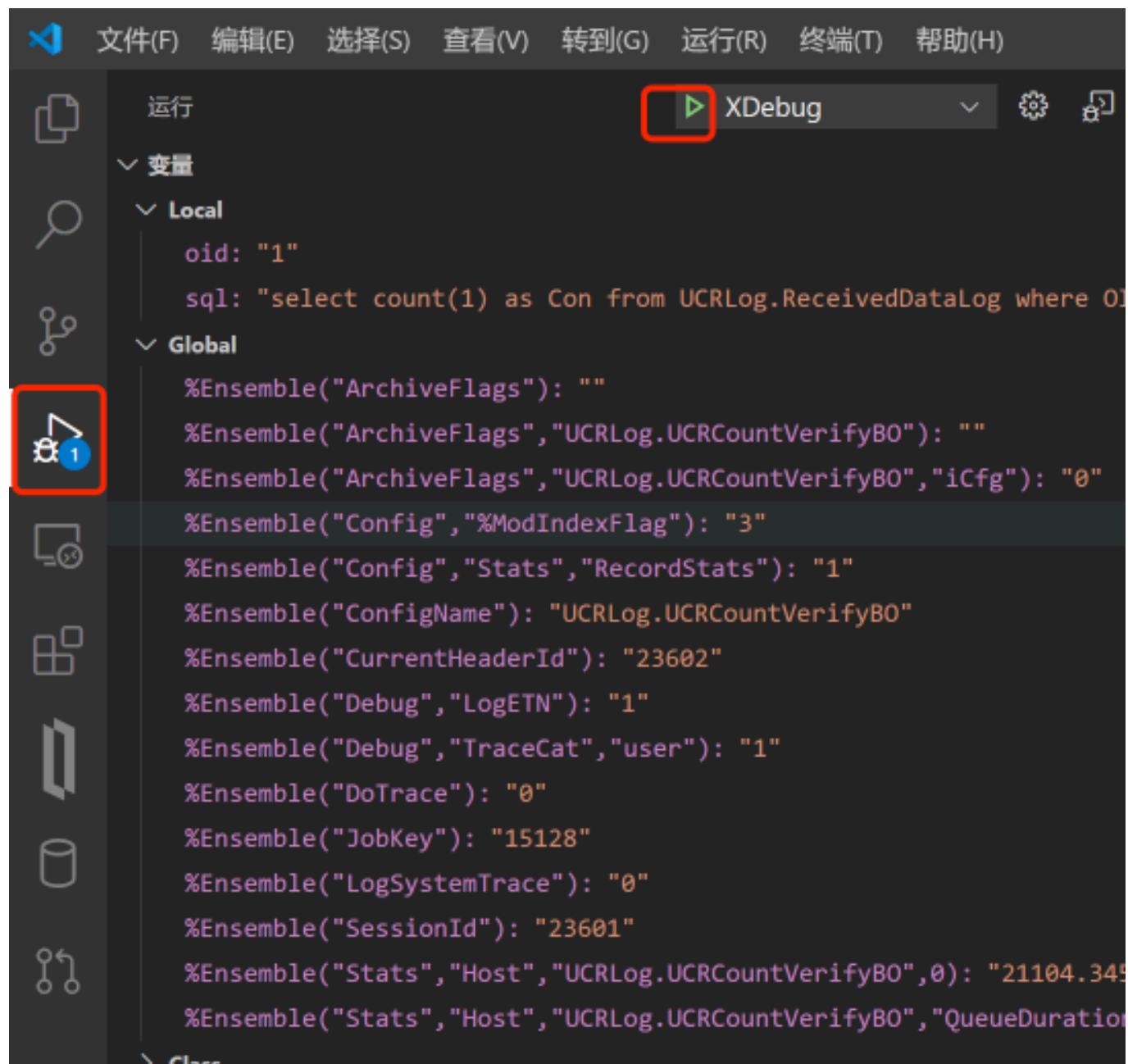
- 对于需要附加到的进程，在lauch.json中添加 "program" 属性,并填入进程号，保存。
- program 中也可以填入要debug的routine 比如 " ##class(Test.test).test() "，
" name " 可以修改成自定义的名字，之后会在debug运行按钮的下拉菜单显示。
- 调试的配置可以添加多个，都会显示在debug下拉菜单中



The screenshot shows the VS Code editor with the `launch.json` file open. The file is located in the `.vscode` directory and is used for launching the `UCRCountVerifyBO.cls` file. The configuration is for XDebug debugging. The code is as follows:

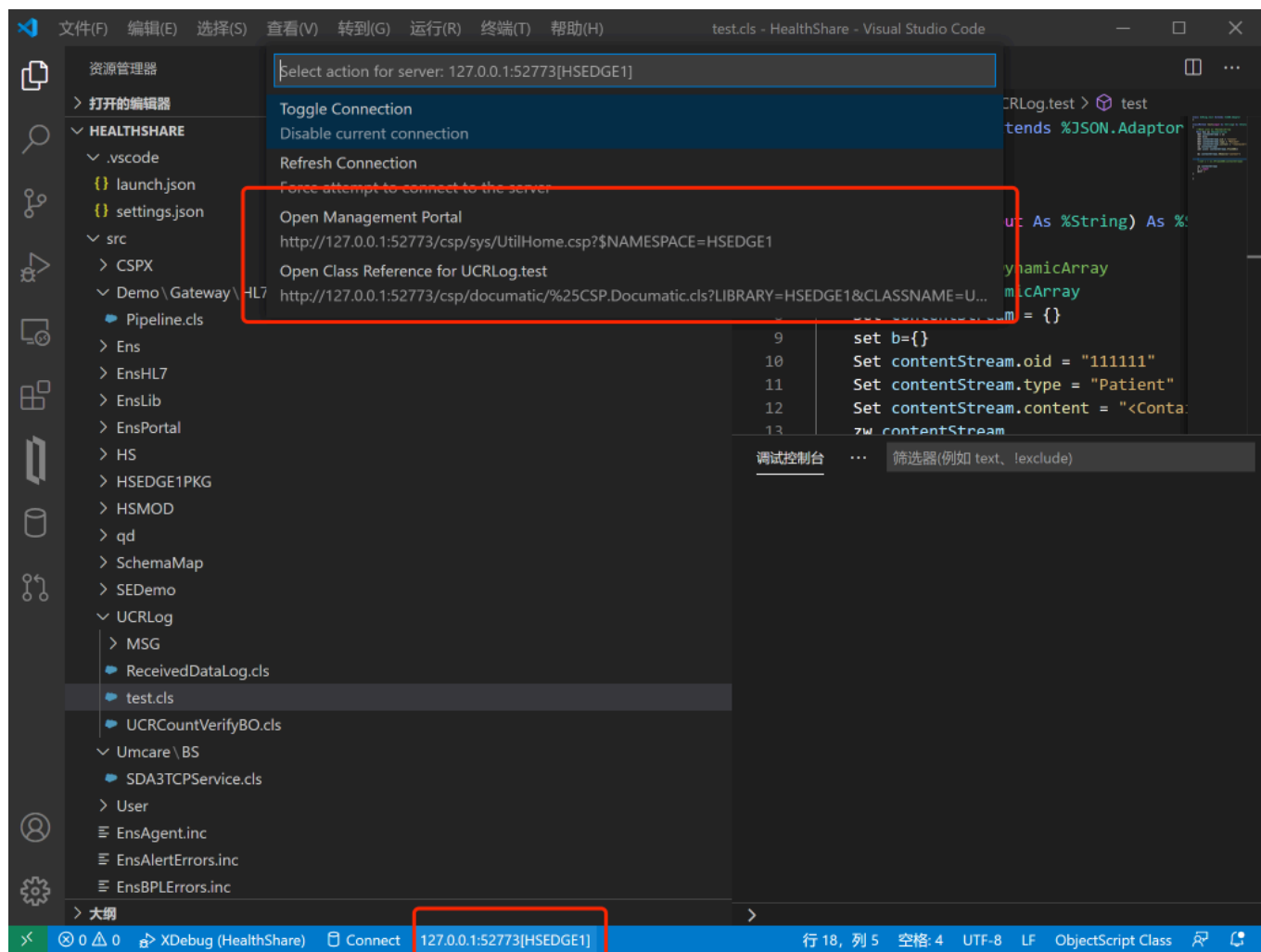
```
.vscode > {} launch.json > Launch Targets > {} XDebug
1  {
2      // 使用 IntelliSense 了解相关属性。
3      // 悬停以查看现有属性的描述。
4      // 欲了解更多信息，请访问: https://go.microsoft.com/fwlink/?linkid=829397
5      "version": "0.2.0",
6      "configurations": [
7          {
8              "type": "objectscript",
9              "request": "launch",
10             "name": "XDebug",
11             "program": "15128"
12         }
13     ]
14 }
```

- 进入调试界面，点击xDebug 旁边的绿色启动按钮开始调试，其他同Class Method 调试方式

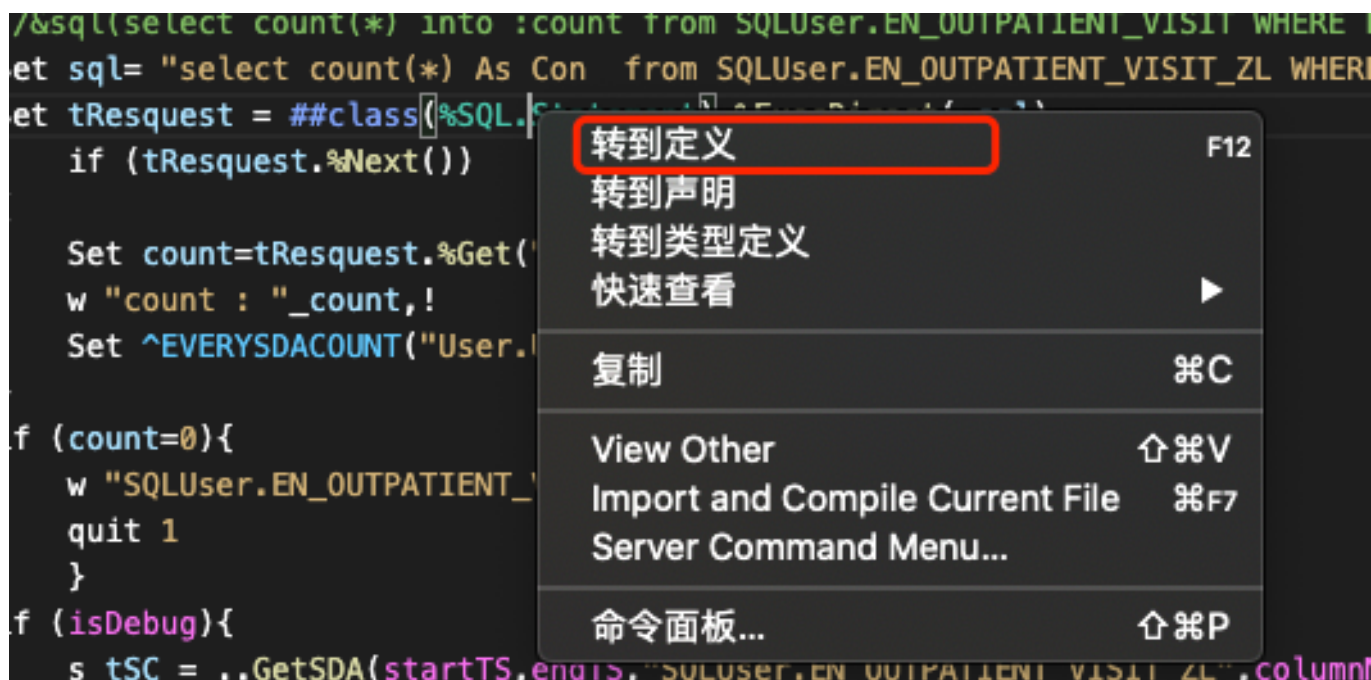


6. 关于VSCode几个编辑器小技巧：

在底部栏点击当前连接的服务器会出现提示菜单，从这里可以便捷进入实例管理界面和class reference



在任意的类或者方法上点击右键可以转到定义。如果使用了#dim预定义变量，则可以在变量上使用转到变量声明，而且可以使用转到类型定义类型的定义。



7. 尾声

- 当前版本不支持xml 代码文件导入！VSCode community中已经有人提到过这个问题，开发者回复之后的版本很有可能会加入
- 本项目的Github的issue地址，可以在这里提出问题：<https://github.com/intersystems-community/vscode-objectscript/issues>
- 支持的产品：Caché/Ensemble 2016.2 以及更新, 以及所有的基于IRIS 的产品版本.

[#InterSystems IRIS](#)

源

URL:

<https://cn.community.intersystems.com/post/%E7%B2%BE%E5%8D%8E%E6%96%87%E7%AB%A0%E7%BD%AE%E9%A1%B6-%E4%BD%BF%E7%94%A8vscode-%E8%BF%9B%E8%A1%8Ciris-%E5%BC%80%E5%8F%91>