

文章

[Jingwei Wang](#) · 二月 14 阅读大约需分钟

[Open Exchange](#)

使用嵌入式Python实现InterSystems IRIS 互操作性

1. 互操作性嵌入式Python(interoperability-embedded-python)

这个概念旨在展示 iris互操作性 如何与嵌入的python一起使用。

1.2. 示例代码

```
import grongier.pex
import iris
import MyResponse

class MyBusinessOperation(grongier.pex.BusinessOperation):

    def OnInit(self):
        print("[Python] ...MyBusinessOperation:OnInit() is called")
        self.LOGINFO("Operation OnInit")
        return

    def OnTeardown(self):
        print("[Python] ...MyBusinessOperation:OnTeardown() is called")
        return

    def OnMessage(self, messageInput):
        if hasattr(messageInput, "_IsA"):
            if messageInput._IsA("Ens.StringRequest"):
                self.LOGINFO(f"[Python] ...This iris class is a Ens.StringRequest with this message {messageInput.StringValue}")
                self.LOGINFO("Operation OnMessage")
                response = MyResponse.MyResponse("...MyBusinessOperation:OnMessage() echos")
                return response
```

1.3. 注册一个组件

不是ObjectScript 代码

通过Grongier.PEX.Utils.RegisterComponent()方法。

启动一个嵌入式Python shell:

```
/usr/irissys/bin/irispython
```

然后使用这个类方法将一个python文件添加到组件列表中,以实现互操作性

```
iris.cls("Grongier.PEX.Utils").RegisterComponent(<ModuleName>,<ClassName>,<PathToPyFile>,<OverWrite>,<NameOfTheComponent>)
```

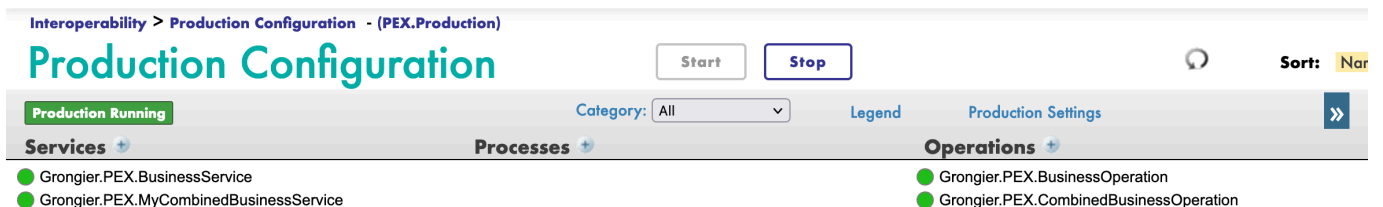
例如：

```
iris.cls("Grongier.PEX.Utils").RegisterComponent("MyCombinedBusinessOperation","MyCombinedBusinessOperation","/irisdev/app/src/python/demo/",1,"PEX.MyCombinedBusinessOperation")
```

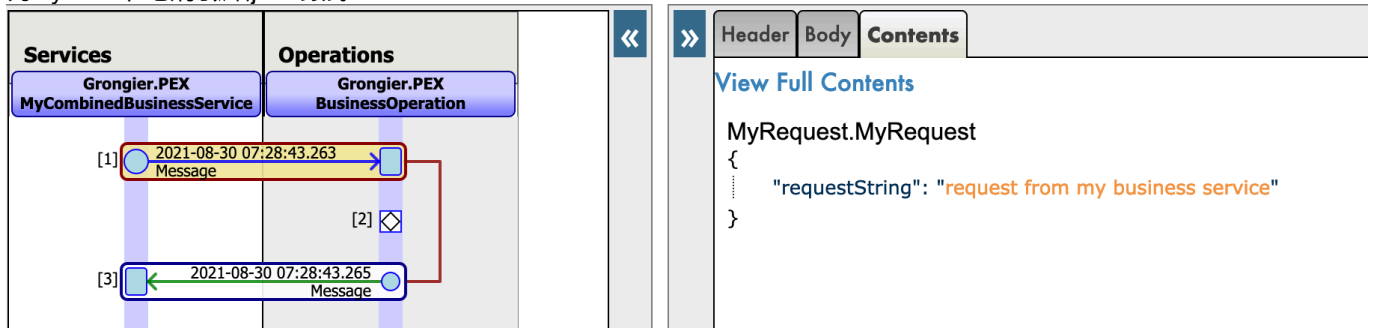
2. 演示

Production有四个Python组件：

- 两个业务服务组件：
 - 用Grongier.PEX.MyCombinedBusinessService持续给业务操作组件发送同步消息
 - 这些消息是JSON格式的Python对象,存储于Grongier.PEX.Message类
 - Thoses messages are python objects casted in JSON and stored in Grongier.PEX.Message.
 - Python代码：src/python/demo/MyCombinedBusinessService.py
 - Grongier.PEX.MyBusinessService是一个用于写消息日志的原始业务服务组件,无其他作用。
 - Python 代码：src/python/demo/MyBusinessService.py
- 两个业务操作组件：
 - Grongier.PEX.BusinessOperation是用于接收业务服务组件Grongier.PEX.MyCombinedBusinessService的消息
 - Python 代码：src/python/demo/MyBusinessOperation.py
 - Grongier.PEX.CombinedBusinessOperation可以接收Ens.StringRequest消息,并返回Ens.StringResponse消息
 - Python 代码：src/python/demo/MyCombinedBusinessOperation.py



为Python本地消息新增json跟踪：



3. 前置需

Git和Docker必须已安装

4. 使用Docker安装

从git拉取repo到本地

```
git clone https://github.com/grongierisc/interpeorability-embedded-python
```

在安装路径打开terminal, 并运行:

```
docker-compose build
```

使用IRIS容器运行你的项目:

```
docker-compose up -d
```

5. 无Docker安装

在iris 实例上安装 grongier_pex-1.0.0-py3-none-any.whl :

```
/usr/irissys/bin/irispython -m pip install grongier_pex-1.0.0-py3-none-any.whl
```

然后, 加载ObjectScript类:

```
do $System.OBJ.LoadDir("/opt/irisapp/src", "cubk", "*.cls", 1)
```

6. 运行How to Run the Sample

打开production, 并运行。

示例代码会开始运行。

7. repo中包含内容What's inside the repository

7.1. Dockerfile

Dockerfile包含在容器中安装一些python的依赖项, 例如pip, venv和sudo
然后创建dev目录将git仓库复制进入目录

Dockerfile启动IRIS并导入Titanics csv文件, 然后为Python Shell激活 %Service_CallIn。
可以使用使用相关的docker-compose.yml来轻松地设置额外的参数, 如端口号和映射键和主机文件夹位置。

这个dockerfile以安装python模块的**霰** 结束。

最后一部分是关于安装jupyter notebook和它的内核。

使用.env文件来调整在docker-compose中使用的dockerfile。

7.2. .vscode/settings.json

VSCode配置文件 [VSCode ObjectScript plugin](#)

7.3. .vscode/launch.json

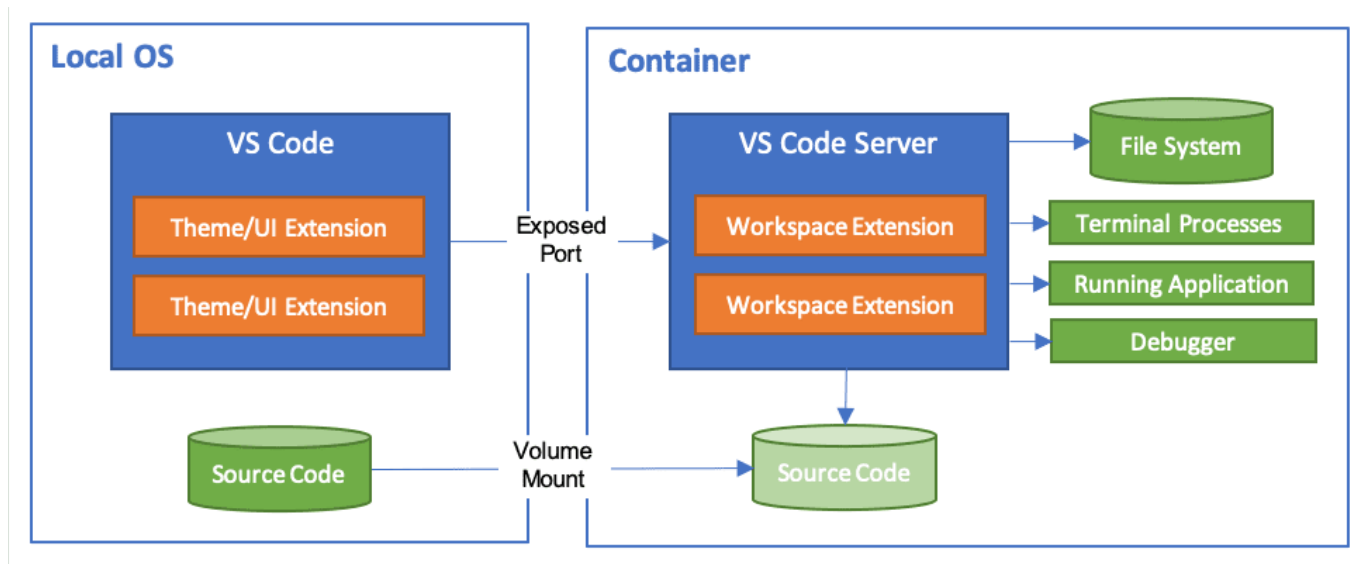
VSCode ObjectScript 调试配置文件

[获取更相关信息](#)

7.4. .vscode/extensions.json

如果你想在容器中用VSCode运行,可以添加扩展。

[获取更相关信息](#)



对于使用嵌入式python非常有用。

7.5. src 文件夹

```
src
??? Grongier
?   ??? PEX // ObjectScript classes that wrap python code
?     ??? BusinessOperation.cls
?     ??? BusinessProcess.cls
?     ??? BusinessService.cls
?     ??? Common.cls
?     ??? Director.cls
?     ??? InboundAdapter.cls
?     ??? Message.cls
?     ??? OutboundAdapter.cls
?     ??? Python.cls
?     ??? Test.cls
?     ??? Utils.cls
??? PEX // Some example of wrapped classes
```

```
? ??? MyBusinessOperationWithAdapter.cls
? ??? MyBusinessOperationWithIrisAdapter.cls
? ??? MyBusinessOperationWithPythonAdapter.cls
? ??? MyBusinessService.cls
? ??? MyOutboundAdapter.cls
? ??? Production.cls
??? python
??? demo // Actual python code to run this demo
? ??? MyBusinessOperation.py
? ??? MyBusinessOperationWithAdapter.py
? ??? MyBusinessOperationWithIrisAdapter.py
? ??? MyBusinessProcess.py
? ??? MyBusinessService.py
? ??? MyCombinedBusinessOperation.py
? ??? MyCombinedBusinessProcess.py
? ??? MyCombinedBusinessService.py
? ??? MyInboundAdapter.py
? ??? MyLoggingOperation.py
? ??? MyNonPollingStarter.py
? ??? MyOutboundAdapter.py
? ??? MyRequest.py
? ??? MyResponse.py
? ??? MySyncBusinessProcess.py
? ??? SimpleObject.py
??? dist // Wheel used to implement python interoperability components
? ??? grongier_pex-1.0.0-py3-none-any.whl
??? grongier
? ??? pex // Helper classes to implement interoperability components
?     ??? _BusinessHost.py
?     ??? _BusinessOperation.py
?     ??? _BusinessProcess.py
?     ??? _BusinessService.py
?     ??? _Common.py
?     ??? _Director.py
?     ??? _InboundAdapter.py
?     ??? _Message.py
?     ??? _OutboundAdapter.py
?     ??? __init__.py
??? setup.py // setup to build the wheel
```

8. 如何增加一个新组件

8.1. 进站适配器InboundAdapter

使用python执行进站适配器,使用grongier.pex.InboundAdapter子类,覆盖OnTask()方法。

8.2. 出站适配器OutboundAdapter

使用python执行出站适配器,使用grongier.pex.OutboundAdapter子类,实现所需on方法。

8.3. 业务服务组件BusinessService

使用python执行业务服务组件,使用grongier.pex.BusinessService子类,覆盖OnProcessInput()方法。

8.4. 业务流程组件BusinessProcess

使用python执行业务流程组件,使用grongier.pex.BusinessProcess子类,覆盖 OnRequest(), OnResponse() and OnComplete()方法。

8.5. 业务操作BusinessOperation

使用python执行业务流程组件,使用grongier.pex.BusinessOperation子类,覆盖 OnMessage()方法。

8.6. 注册一个组件

启动一个嵌入式python shell:

```
/usr/irissys/bin/irispython
```

然后使用这个方法将一个新的py文件添加到组件列表中,以实现互操作性

```
iris.cls("Grongier.PEX.Utills").RegisterComponent(<ModuleName>,<ClassName>,<PathToPyFile>,<OverWrite>,<NameOfTheComponent>)
```

例如:

```
iris.cls("Grongier.PEX.Utills").RegisterComponent("MyCombinedBusinessOperation","MyCombinedBusinessOperation","/irisdev/app/src/python/demo/",1,"PEX.MyCombinedBusinessOperation")
```

8.7. 直接使用Grongier.PEX

如果你不想使用RegisterComponent,你可以添加一个Grongier.PEX.Business*组件并配置其属性

- %module :
 - 你的python代码的模块名称
- %classname :
 - 你的组件的类名
- %classpaths
 - 你的组件所在的路径。
 - 除了PYTHON_PATH之外,可以有一个或多个Classpaths(用'|'字符分隔)

例如:

The screenshot shows the 'Production Settings' sidebar on the left with 'Operations' expanded, listing several business operation configurations. The main panel is titled 'Grongier.PEX.BusinessOperation' and contains tabs for 'Settings', 'Queue', 'Log', 'Messages', 'Jobs', and 'Actions'. The 'Settings' tab is active, showing an 'Apply' button, a search field, and several expandable sections: 'Informational Settings', 'Basic Settings', and 'Python BusinessOperation'. The 'Python BusinessOperation' section is expanded, showing fields for '%classname' (MyBusiness0peration), '%module' (MyBusiness0peration), '%settings' (empty), and '%classpaths' (/irisdev/app/src/python/demo).

9. 其他工作

- 仅业务服务组件和业务操作组件可以被测试
- 在适配器上工作

10. 认证

大部分代码来自Mo Cheng和Summer Gerry的PEX for Python,

注册部分来自于尚未发布的IRIS 2021.3的功能。

[##嵌入式 Python #Python #InterSystems IRIS 在 InterSystems Open Exchange 上检查相关应用程序](#)

源 URL: <https://cn.community.intersystems.com/post/%E4%BD%BF%E7%94%A8%E5%B5%8C%E5%85%A5%E5%BC%8Fpython%E5%AE%9E%E7%8E%B0intersystems-iris-%E4%BA%92%E6%93%8D%E4%BD%9C%E6%80%A7>