

文章

[Lilian Huang](#) · 七月 13, 2022 阅读大约需 5 分钟

用 Python 的 Native API 调用类方法

InterSystems [Native SDK for Python](#) 是 InterSystems IRIS APIs 的轻量级接口，曾经只能通过 ObjectScript 使用。

准确地说，我对调用 ObjectScript 方法、类方法的能力特别感兴趣。

它可以工作，而且效果很好，但默认情况下，调用只支持标量参数：字符串、布尔值、整数和浮点数。

但如果你想：

- 传递或返回结构，例如字典或列表
- 传递或返回流

您需要编写一些粘合代码或使用这个[project](#)（使用 pip install edpy 安装）。edpy 包会给你一个简单的签名：

```
call(iris, classname, methodname, args)
```

它允许您调用任何 ObjectScript 方法并返回结果。

像这样导入它：

```
from edpy import iris
```

call 接受 4 个必需参数：

- iris - a reference to an established [IRIS object](#)
- classname - IRIS class to call
- methodname - IRIS method to call
- args - list of 0 or more arguments

参数

每个参数可以是以下其中之一：

- string 字符串（任意长度，如果大于 \$\$\$MaxStringLength 或 3641144 个符号，则会自动转换为流）
- boolean 布尔值
- integer 整数
- float 浮动
- dict 字典（转换为动态对象）
- list or tuple 列表或元组（转换为动态数组）

字典、列表和元组参数可以递归地包含其他字典、列表和元组（持续性内存）。

返回值

在返回值，我们期望是动态对象/数组或 JSON 字符串/流。在这种情况下，edpy 会首先将其转换为 Python 字符串，并在可能的情况下将其解释为 Python 字典或列表。否则，结果将按原样返回给调用者。

就是这样，但让我给你一些 ObjectScript 方法的例子，以及如何使用这个 Python 函数调用它们。

示例1: Pong

```
ClassMethod Test(arg1, arg2, arg3) As %DynamicArray
{
    return [(arg1), (arg2), (arg3)]
}
```

调为：

```
>>> iris.call(irisnative, "User.Py", "Test", [1, 1.2, "ABC"])
[1, 1.2, 'ABC']
```

这里没有惊喜。参数被打包回一个数组并返回给调用者。

示例 2: 属性

```
ClassMethod Test2(arg As %DynamicObject) As %String
{
    return arg.Prop
}
```

像这样调用：

```
>>> iris.call(irisnative, "User.Py", "Test2", [{"Prop":123}])
123
```

现在进行更嵌入式的调用：

```
>>> iris.call(irisnative, "User.Py", "Test2", [{"Prop":{"Prop2":123}}])
{'Prop2': 123}
```

如果属性太长也没关系 - 流将用于将其发送到 IRIS 和/或返回：

```
ret = iris.call(irisnative, "User.Py", "Test2", [{"Prop":"A" * 10000000}])
>>> len(ret)
10000000
```

如果您需要保证InterSystems IRIS 端的流，您可以使用[%Get](#):

```
set stream = arg.%Get("Prop", "stream")
```

如果流是 base64 编码的，您可以使用以下命令自动对其进行解码：

```
set stream = arg.%Get("Prop", "stream<base64")
```

示例 3: 字符串或流

```
ClassMethod Test3(arg As %Stream.Object) As %String
```

```
{
    set file = ##class(%Stream.FileCharacter).%New()
    set file.TranslateTable = "UTF8"
    set filename = ##class(%File).ManagerDirectory() _ "test.txt"
    do file.LinkToFile(filename)
```

```

if $isObject(arg) {
    set sc = file.CopyFromAndSave(arg)
} else {
    do file.Write(arg)
    set sc = file.%Save()
}
if $$ISERR(sc) {
    set jsonret = {"status":0, "payload":($system.Status.GetErrorText(sc))}
} else {
    set jsonret = {"status":1}
}
quit jsonret.%ToJson()
}

```

这里我们将字符串或流写入 <mgr>test.txt。

```
>>> iris.call(irisnative, "User.Py", "Test3", ["\u1f60a;"])
```

```
{'status': 1}
```

注意：在所有代码示例中输入为 “ \u1f642; ”

如果我打开文件，我会看到一个 \u1f642; 而不是两个 ?? - 所以我们保留编码。

```
>>> iris.call(irisnative, "User.Py", "Test3", ["\u1f642;" * 10000000])
```

```
{'status': 1}
```

为简便起见，我将省略文件输出，但它就在那里。

最后，通过在内部传递一个动态

对象或数组，您可以完全避免字符串/流二分法，即使您不知道该属性是比字符串限制短还是长。在这种情况下，您始终可以像流一样获取可疑属性。

示例4: 返回流

```
ClassMethod Test4(arg As %DynamicArray) As %String
```

```
{
```

```
    return arg.%Get(0)
```

```
}
```

看起来是这样：

```
>>> ret = iris.call(irisnative, "User.Py", "Test4", [["\u1f60a;" * 10000000]])
```

```
>>> len(ret)
10000000
>>> ret[:5]
'&#x1f60a;&#x1f60a;&#x1f60a;&#x1f60a;&#x1f60a;'
```

还有一件事

还有一个 `getIris(ip="localhost", port=1972, namespace="USER", username="SYSTEM", password="SYS")` 函数可以让您获得一个工作的 IRIS 对象。

所以这里有一个完整的例子，如果你想自己尝试一下：

首先加载 [User.Py class](#) 并安装 edpy python 库：

```
pip install edpy
```

然后在python调用中：

```
from edpy import iris

irisnative = iris.getIris()

iris.call(irisnative, "User.Py", "Test", [1, 1.2, "ABC"])

iris.call(irisnative, "User.Py", "Test2", [{"Prop":123}])

iris.call(irisnative, "User.Py", "Test2", [{"Prop":{"Prop2":123}}])

ret2 = iris.call(irisnative, "User.Py", "Test2", [{"Prop":{"A": * 10000000}}])

iris.call(irisnative, "User.Py", "Test3", ["&#x1f60a;"])

iris.call(irisnative, "User.Py", "Test3", ["&#x1f60a;" * 10000000])

ret4 = iris.call(irisnative, "User.Py", "Test4", [{"&#x1f60a;" * 10000000}])
```

结论

适用于 Python 的 Native SDK 是一个强大的工具，为您提供对 InterSystems IRIS 的完整且不受限制的访问。我希望这个项目可以为您节省一些时间来编组 InterSystems IRIS 调用。是否有一些它不支持的方法参数组合？如果是这样，请在评论中分享您是如何调用的。

链接：

- [Native SDK Docs](#)
- [User.Py class](#)
- [Repo](#)

原文请参考链接：<https://community.intersystems.com/node/521336>

[##嵌入式 Python](#) [#Python](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

源

URL:<https://cn.community.intersystems.com/post/%E7%94%A8-python-%E7%9A%84-native-api-%E8%B0%83%E7%94%A8%E7%B1%BB%E6%96%B9%E6%B3%95>