Article
Heloisa Paiva · Mar 17, 2023 2m read

Types in Python and in ObjectScript

Why I've decided to write this

In my last article I've talked about returning values with Python. But returning them is simple, what can make it harder is what I'm going to talk about today: where the value is treated.

Python object in IRIS

Following the example of the last aricle, we have the method:

```
Class python.returnTest [ Abstract ]
{

ClassMethod returnSomething(pValue... As %String) As %Integer [ Language = python ]
{
 return pValue
}

}
```

Then, we'll have as a return a Python object, that IRIS interprets as the class %SYS.Python. So if I call the method with two values, like this:

```
Set returnValue = ##class(python.returnTest).returnSomething("Hello", "World")
```

we'll have as returnValue something like "1@%SYS.Python". Then, to access the strings, I can do the following:

```
Write returnValue."__getitem__"(0)
```

and in the terminal we'll have "Hello".

The attributes that you can use depend on the type, but you're gonna find them by running a "help". It's simple:

```
Set builtins = ##class(%SYS.Python).Builtins()
Do builtins.help(returnValue)
```

and then you'll have the kind of returnValue and the methods that you can youse - to compare, change or simply view it, or what else you might want.

If you execute the "help" without parameters, i.e. "Do builtins.help()", you'll have everything you can do with all types.

IRIS object in Python

Again with the last article's example, we'll have:

```
import iris

def main():
    connection_string = "localhost:1972/samples"
    username = "_system"
    password="sys"
    connection = iris.connect(connection_string, username, password)

    irispy = iris.createIRIS(connection)

    test = irispy.classMethodObject("python.returnTest",
"returnSomething", "Hello", " World!")

    print(test.invoke("__getitem__", 0))

    return test


print(main())
```

Here, we made the connection like the previous articles until we have the *irispy*, that we use to access everything the package Iris has. So I found the method classMethodObject() in [Native SDK for Python Quick Reference](), which is the adequate one to call the "returnSomehting" and have an object returned.

Other examples are classMethodVoid(), classMethodBoolean(), classMethodIRISList(), etc.

So the "test" os am IRIS object, since it was created with an IRIS method. To access the methods from an IRIS object in Python, I can use .inovke("oject name", "arguments") or .invokeVoid(). To access the properties or configure them, I can use .set() and .get().

#Embedded Python #Innovatium #Python #InterSystems IRIS

Source URL:https://community.intersystems.com/post/types-python-and-objectscript