### 記事

Minoru Horita · 2020年7月28日 2m read

Open Exchange

# Python Gateway パート V: Execute関数

この連載記事では、InterSystemsデータプラットフォーム用の<u>Python Gateway</u>について説明します。 また、InterSystems IRISからPythonコードなどを実行します。 このプロジェクトは、InterSystems IRIS環境にPythonの力を与えます。

- 任意のPythonコードを実行する
- InterSystems IRISからPythonへのシームレスなデータ転送
- Python相互運用アダプタでインテリジェントな相互運用ビジネスプロセスを構築する
- InterSystems IRISからのPythonコンテキストの保存、調査、変更、復元

## その他の記事

現時点での連載計画です(変更される可能性があります)。

- <u>パート I: 概要、展望、紹介</u>
- パート ||: インストールとトラブルシューティング
- パートⅢ:基本機能
- <u>パート IV:相互運用アダプタ</u>
- パート V: Execute関数 <-- 現在、この記事を参照しています
- パート VI:動的ゲートウェイ
- パート VII: プロキシゲートウェイ
- パート VIII:使用事例とML Toolkit

## はじめに

この記事と次の記事では、一般に(.Net/Java)ゲートウェイとして理解されているものに関する機能を説明します。 今回は、実行関数(Pythonコードを構造化されたプログラムで実行する機能)について説明します。この関数は、InterSystems IRISアプリケーションのPythonコードを活用する場合に役立ちます。

## Execute関数

名前を指定して関数を実行します。 このAPIは2つのメソッドで構成されています。

- ExecuteFunction
- ExecuteFunctionArgs

これらはシグネチャが異なっています。 ExecuteFunction は位置引数とキーワード引数に分けられた %List, %Collection.AbstractArray および動的オブジェクトを受け付けます。 ExecuteFunctionArgs は args... を受け付け、それを解析して位置引数とキーワード引数に分けます。 その後、ExecuteFunctionArgs は ExecuteFunction を呼び出します。

引数の値をエスケープするのは呼び出し側の仕事です。 isc.py.util.Converter クラスを使用して以下をエスケープします。

- 文字列
- ブール値
- 日付

- 時間
- タイムスタンプ

### **ExecuteFunction**

```
ExecuteFunction は isc.py.Main クラスのメソッドです。 シグネチャ:
- function - 呼び出す関数の名前です。 入れ子にできます (例:random.randint)。
- variable - 結果を書き込むPython変数の名前です。
- positionalArguments - Python関数の位置引数です。 次のいずれかになります。
 + $lb(val1, val2, ..., valN)
 + %Collection.AbstractIterator オブジェクト
 + JSON配列
- keywordArguments - Python関数のキーワード引数です。 次のいずれかになります。
 + $lb($lb(name1, val1), $lb(name2, val2), ..., $lb(nameN, valN))
 + %Collection.AbstractArray オブジェクト
 + フラットなJSONオブジェクト
- serialization - 結果のシリアライズ方法です。
- result - 結果をこの変数に書き込みます。
function 以外のすべての引数は省略可能です。
以下に動作例を示します。
set sc = ##class(isc.py.Main).ImportModule("random", ,.random)
set posList = $lb(1, 100)
set posCollection = ##class(%ListOfDataTypes).%New()
do posCollection.Insert(1)
do posCollection.Insert(100)
set posDynamic = [1, 100]
for positionalArguments = posList,posCollection,posDynamic {
    set sc = ##class(isc.py.Main).ExecuteFunction(random _ ".randint", positionalArgu
ments,,,.result)
    write result,!
}
set kwList = $lb($lb("a", 1), $lb("b", 100))
set kwCollection = ##class(%ArrayOfDataTypes).%New()
do kwCollection.SetAt(1, "a")
do kwCollection.SetAt(100, "b")
set kwDynamic = { "a": 1, "b": 100}
for kwArguments = kwList,kwCollection,kwDynamic {
    set sc = ##class(isc.py.Main).ExecuteFunction(random _ ".randint", ,kwArguments,,
.result)
    write result,!
}
set posList = $lb(1)
set kwDynamic = {"b": 100}
set sc = ##class(isc.py.Main).ExecuteFunction(random _ ".randint", posList, kwDynamic
,,.result)
write result,!
set posList = ##class(isc.py.util.Converter).EscapeStringList($lb("Positional: {0} {1
}! Keyword: {name}, {name2}", "Hello", "World"))
```

Published on InterSystems Developer Community (https://community.intersystems.com)

### **ExecuteFunctionArgs**

ExecuteFunctionArgs は isc.py.Main クラスのメソッドです。 シグネチャ:

- function 呼び出す関数の名前です。 入れ子にできます (例:random.randint)。
- variable 結果を書き込むPython変数の名前です。
- serialization 結果のシリアライズ方法です。
- result 結果をこの変数に書き込みます。
- args... 関数の引数です。

#### ExecuteFunctionArgs

は、関数の署名から正しい位置引数とキーワード引数を決定しようとします(可能な場合)。

ExecuteFunctionArgs

が正しい引数の仕様を解釈できない場合(そして問題が発生する場合)は、ExecuteFunction を直接呼び出すことをお勧めします。 例:

```
set sc = ##class(isc.py.Main).ImportModule("random", ,.random)
set sc = ##class(isc.py.Main).ExecuteFunctionArgs(random _ ".randint", , ,.result, 1,
 100)
write result.!
set string = ##class(isc.py.util.Converter).EscapeString("Positional: {0}, {1}, {2},
{3}")
set arg1 = ##class(isc.py.util.Converter).EscapeString("Hello")
set arg2 = ##class(isc.py.util.Converter).EscapeString("World")
set arg3 = ##class(isc.py.util.Converter).EscapeString("Alice")
set arg4 = ##class(isc.py.util.Converter).EscapeString("Bob")
set sc = ##class(isc.py.Main).ExecuteFunctionArgs("str.format",,,.result, string, arg
1, arg2, arg3, arg4)
write result,!
set string = ##class(isc.py.util.Converter).EscapeString("Positional: {0} {1}! Keywor
d: {name}, {name2}")
set arg1 = ##class(isc.py.util.Converter).EscapeString("Hello")
set arg2 = ##class(isc.py.util.Converter).EscapeString("World")
set kwargs = "**" _ {"name":"Alice","name2":"Bob"}.%ToJSON()
set sc = ##class(isc.py.Main).ExecuteFunctionArgs("str.format",,, .result, string, ar
g1, arg2, kwargs)
write result,!
```

## 要約

Python Gatewayを使用すると、InterSytems IRISとPythonをシームレスに統合できます。これを使用し、Pythonの機能をInterSystems IRISアプリケーションに追加してください。

## リンク

- Python Gateway
- Python Gatewayのサンプル

- Python 3.6.7 (64ビット版) のインストール
- Pvthonのドキュメント/チュートリアル

## イラスト付きガイド

ML Toolkitユーザーグループには、イラスト付きのガイドもあります。 ML Toolkitユーザーグループは、InterSyste ms社のGitHub組織の一部として設定されている非公開GitHubリポジトリです。 このリポジトリは、Python Gatewayを含むML

Toolkitコンポーネントをインストール、学習、またはすでに使用している外部ユーザーを対象としています。 ML Toolkitユーザーグループに参加するには、以下の内容を含む簡単なメールを MLToolkit@intersystems.com 宛に送信してください (グループメンバーが議論中にあなたを認識して特定するために必要です)。

- GitHubのユーザー名
- 氏名(英文字表記の名前、姓の順)
- 組織(勤務先、通学先、または在宅勤務)
- 役職(組織での実際の役職、「学生」、または「無所属」)
- 国(本拠地としている国)

#Python #InterSystems IRIS

InterSystems Open Exchangeで関連アプリケーションを確認してください

Y-XURL: https://jp.community.intersystems.com/post/python-gateway-%E3%83%91%E3%83%BC%E3%83%88-v%EF%BC%9Aexecute%E9%96%A2%E6%95%B0