記事
Toshihiko Minamoto · 2022年7月26日 5m read

Open Exchange

Jupyter Notebooks に ObjectScript を追加する方法

Jupyter Notebook は、多数の異なるマークアップ言語とプログラミング言語でコードを実行できるセルで構成された対話型環境です。

Jupyter はこれを実現するために適切なカーネルに接続しなければなりませんが、 ObjectScript カーネルがなかったため、それを作成することにしました。

こちらから試すことができます。

結果を少し覗いてみましょう。

Jupyter カーネルの基礎

Jupyter カーネルはいくつかの方法で作成できます。 ここでは、Python ラッパーカーネルを作成することにしましょう。

ipykernel.kernelbase.Kernel のサブクラスを作成して、特定の言語で実行されるコードを受け取る doexecute メソッドを実装する必要があります。

つまり、ある ObjectScript コードを取得して、何らかの方法で実行し、ノートブックにその結果を返すという概念です。

でも、実際にはどうすればよいのでしょうか。 では、その方法をさらに噛み砕いて説明しましょう。

ObjectScript コードを IRIS に送る

まず初めに、コードを IRIS に送る必要があります。 ここで使用するのが、Python 用の IRIS Native API です。

ここでは、irisnative パッケージをインポートして、接続を確立するだけです。

```
def get_iris_object():
  # InterSystems IRIS ????????
  connection = irisnative.createConnection('iris', 51773, 'IRISAPP', '_SYSTEM', 'SYS'
)

  # iris ??????????
  return irisnative.createIris(connection)
```

その後で、この接続を使用して、IRIS データベースに格納されているクラスを呼び出すことができます。

```
def execute_code(self, code):
        class_name = "JupyterKernel.CodeExecutor"
        return self.iris.classMethodValue(class_name, "CodeResult", code)
```

CodeExecutor クラスと CodeResult メソッドは何に使用されているのでしょうか。

ではそれを見てみましょう。

ObjectScript コードを実行する

このクラスの目的は、1 行の ObjectScript コードを実行して、実行の結果を含む JSON オブジェクトを返すことです。 コードを CodeResult の vstrCommand 変数に渡します。

まず、IO を現在のルーチンにリダイレクトします。その後、渡されたコードを XECUTE コマンドで実行し、IO を元の場所にリダイレクトして、結果を返します。

```
Include %sySystem

Class JupyterKernel.CodeExecutor
{

ClassMethod CodeResult(vstrCommand As %String) As %String [ ProcedureBlock = 0 ]
{
        set tOldIORedirected = ##class(%Device).ReDirectIO()
        set tOldMnemonic = ##class(%Device).GetMnemonicRoutine()
        set tOldIO = $io
        try {
            set str=""
            set status = 1
            //IO ??????????????????????????????????
            use $io::("^"_$ZNAME)

            //???????????
            do ##class(%Device).ReDirectIO(1)

            XECUTE (vstrCommand)

        } catch ex {
            set str = ex.DisplayString()
            set status = 0
        }

        //????????/??????????????????
        if (tOldMnemonic '= "") {
            use tOldIO::("^"_tOldMnemonic)
        } else {
            use tOldIO
        }
        do ##class(%Device).ReDirectIO(tOldIORedirected)

        quit {"status":(status), "out":(str)}.%ToJSON()

rchr(c)
    quit
rstr(sz,to)
    quit
wchr(s)
    do output($char(s))
    quit
wff()
    do output($char(12))
```

```
    quit
wnl()
    do output($char(13,10))
    quit
wstr(s)
    do output(s)
    quit
wtab(s)
    do output($char(9))
    quit
output(s)
    set str = str _ s
    quit
}

}
```

結果を表示する

ObjectScript コードを実行しましたが、次はどうすればよいでしょうか。 その結果を表示する必要があります。

例外がなければ、行単位で結果を表示するだけで済みます。

が、渡されたコードで例外が発生したのであれば、実行を停止し、失敗した行番号、行そのもの、そして発生した例外を表示しなければなりません。

アプリを起動する

このカーネルを自分で試してみましょう。以下のようにして試すことができます。

前提条件

git と Docker がインストール済みであることを確認してください。

リポジトリを、以下のようにローカルディレクトリに Clone/git pull します。

```
$ git clone https://github.com/Vekkby/objectsriptkernel.git
```

このディレクトリでターミナルを開き、以下を実行します。

```
$ docker-compose up -d --build
```

操作方法

ブラウザから以下を使用してノートブックサーバーにアクセスできます。

```
localhost:8888
```

'work' ディレクトリ内に 'hello.ipynb' というサンプルノートブックがあります。

投票をお願いします

このアプリは、IRIS Native API コンテストに参加しています。 こちら からこのアプリに投票してください。

#API #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

ソースURL:https://jp.community.intersystems.com/post/jupyter-notebooks-%E3%81%AB-objectscript-%E3%82%92%E8%BF%BD%E5%8A%A0%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95