記事
Toshihiko Minamoto · 2023年1月16日 7m read

Open Exchange

Embedded Python テンプレート

開発者の皆さん、こんにちは！

InterSystems IRIS で embedded python を使用する一般的なプロジェクトの出発点として推奨できる、最小限の embedded python テンプレート をご紹介しましょう。

特徴:

- Embedded python対応
- Embedded Pythonの3つの開発方法の例
- VSCode開発対応
- Dockerが利用可能
- オンラインデモが可能
- ZPM First開発対応。

以下、その特徴について説明しましょう。

まず、Embedded Pythonについて説明します。この機能は InterSystems IRIS 2021.2 に搭載されており、InterSystems IRIS と python を使用したソリューションを開発することができます。 IRIS 2021.2以降では、InterSystems IRISとの共有メモリ環境でpythonスクリプトを実行でき、python開発者は、コードがデータに近いデータベースを使用する際に、独自のオプションを得ることができます。

Embedded Pythonによる3つの開発モード

ObjectScript から python のライブラリをコールする

これは、%SYS.Python クラスにより、Pythonのライブラリをインポートし、ObjectScirptを介してPythonをコールすることができるようになったからです。 ドキュメント, 例. 以下のコードを参照してください。

```
ClassMethod Today() As %Status
{
  Set sc = $$$OK
  Set dt = ##class(%SYS.Python).Import("datetime")
  write dt.date.today().isoformat()
  Return sc
}
```

Python で ObjectScript クラスメソッドを記述する

実際、開発者はメソッドのシグネチャに [Language=python] タグを付けて、pure python でコーディングできるようになった。また、ObjectScriptのクラスやグローバルを参照するためのヘルパーpythonライブラリ "iris"も用意されています。 ドキュメント、例、以下のサンプルコード

```
ClassMethod CreateRecordPython(propValue As %VarString, ByRef id As %Integer) [ Langu
age = python ]
{
    import iris
    obj=iris.cls(__name__)._New()
    obj.Test=propValue
    sc=obj._Save()
    id=obj._Id()
    return sc
}
```

InterSystems IRISソリューションのpure pythonでのコーディング

これは、開発者がIRISを扱う方法の3番目のオプションです。
ここでは、pythonスクリプトをIRISに接続する必要があり、これはENV変数とCallInサービスの "On "を介して行うことができます（詳細は以下を参照）。一度セットアップされたPythonスクリプトは、IRISとの共有メモリで実行されます。ここで、"iris" ライブラリが非常に役に立ちます。 ドキュメント, 例.

```
def create_rec(var):
    obj=iris.cls('dc.python.PersistentClass')._New()
    obj.Test=var
    obj._Save()
    id=obj._Id()
    return id

# ?????????
from datetime import datetime
now=str(datetime.now())
print("dc.python.PersistentClass ?????????????")
print(create_rec(now))

## SQL??????????????
def run_sql(query):
    rs=iris.sql.exec(query)
    for idx, row in enumerate(rs):
        print(f"[{idx}]: {row}")

query="dc_python.PersistentClass ?? * ?????"
print("SQL?????? "+query)
run_sql(query)
```

Dockerが可能

テンプレートリポジトリは、コンテナ内でIRISを実行し、Embedded Pythonの調整に必要なすべてをセットアップします。

環境変数。Embedded PythonはIRISに接続してPythonスクリプトを実行するために、特定の環境変数の設定を必要とします。以下は、それを助ける設定をdockerfileに記述しています：

```
# init Python env
ENV PYTHON_PATH=/usr/irissys/bin/irispython
ENV SRC_PATH=/irisrun/repo
ENV IRISUSERNAME "SuperUser"
ENV IRISPASSWORD "SYS"
ENV IRISNAMESPACE "USER"
```

また、Embedded Python は CallIn サービスを "ON" にする必要があり、これは docker build フェーズで iris.script で行われます。

```
; Embedded Python?callin???
    do ##class(Security.Services).Get("%Service_CallIn",.prop)
    set prop("Enabled")=1
    set prop("AutheEnabled")=48
    do ##class(Security.Services).Modify("%Service_CallIn",.prop)
```

また、あなたのソリューションには、いくつかの Python ライブラリのインストールが必要かもしれません。これは、リポジトリのルートにある requirements.txt とdockerfileにある pip3 コールによって提供されます：

```
pip3 install -r requirements.txt && \
```

VSCode開発対応

VSCodeでdockerを使って開発するのはとても便利です。Embedded pythonのVSCodeを使ったIRISソリューションをdockerで開発する場合、Devcontainer mode に切り替える必要があります。 .devcontainer フォルダに devcontainer.json ファイル を導入してください。これは、どのDockerサービスと連携する必要があるかを記述しており（私たちのケースではirisです）。コンテナ内で動作しているIRISが使用するPythonエンジンが提供するVSCode内のPythonスクリプトを実行するのに役立ちます。 devcontainer.json ファイルには、コンテナモードでどの extensions を使用する必要があるのかを記述したセクションもあります。

```
"extensions": [
        "ms-python.python",
        "ms-python.vscode-pylance",
        "intersystems-community.vscode-objectscript",
        "intersystems.language-server",
        "intersystems-community.servermanager",
        "ms-vscode.docker"
    ],
```

ZPMを使用したEmbedded Pythonソリューションのインストール

このテンプレートは、「ZPM first」開発リポジトリとして設定されています。つまり、開発したコードはすべて module.xml に記述され、Docker イメージをビルドするたびに ZPM モジュールとしてインストールされます。つまり、開発者は毎回 iris.script の 次の行 からコーディングを開始します：

```
zpm "load /home/irisowner/irisbuild/ -v":1:1
```

そして、Embedded pythonのコードもZPMモジュールに記述され、FILECOPYを介してインストールされます：

この式は、リポジトリ内の /python フォルダにあるすべての python スクリプトをパッケージ化し、ターゲット IRIS インストールの libdir 内の python/ フォルダにインストールしたいことを意味します。python スクリプトが ${libdir}python/ フォルダにコピーされると、ターゲット IRIS マシンの ObjectScirpt または Python からのインポート コールに利用できるようになります。

注意：他のPythonコードを誤って上書きしないように、Pythonスクリプトのフォルダ名称は、ユニークなものにしてください。

このテンプレートがあなたのお役に立つことを願っています。フィードバック、特にプルリクエストは大歓迎です。

#Embedded Python #開発環境 #InterSystems IRIS #VSCode
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/embedded-python-%E3%83%86%E3%83%B3%E3%83%97%E3%83%AC%E3%83%BC%E3%83%88