記事
Megumi Kakechi · 2022年5月31日 4m read

Embedded Python で IRIS グローバル($LB) を Pandas Dataframe に変換する方法

InterSystems IRIS 2021.2 のバージョンより、Embedded Python を使用できるようになりました。

Embedded Python で Excel のデータを IRIS グローバルに格納する方法 では pandas.DataFrame のデータを InterSystems IRIS グローバルに保存する方法をご紹介しました。
こちらの記事では、その逆の「InterSystems IRIS グローバル($LB) を pandas.DataFrame に変換する」方法をご紹介します。

以下のようなグローバルを、Embedded Python を使用して DataFrame に変換します。

```
USER>zwrite ^ISJ
^ISJ(1)=$lb("Name","Age","Address")
^ISJ(2)=$lb("??","50","??")
^ISJ(3)=$lb("??","40","??")
^ISJ(4)=$lb("??","30","??")
```

%Library.GlobalクラスのGetクエリ を使用して取得し、iris.sql.execを使用して DataFrame に格納する方法があります。
ただし、こちらの方法はリスト形式($LB)のまま DataFrame に変換します。

```
USER>do $system.Python.Shell()

Python 3.9.5 (default, Apr 15 2022, 01:28:04) [MSC v.1927 64 bit (AMD64)] on win32
Type quit() or Ctrl-D to exit this shell.
>>> mysql = "select name,value from %library.global_get('user','^ISJ',,2,2)"
>>> resultset = iris.sql.exec(mysql)
>>> dataframe = resultset.dataframe()
>>> print (dataframe)
   name            value
0   ^ISJ               4
1 ^ISJ(1) $lb("Name","Age","Address")
2 ^ISJ(2)     $lb("??","50","??")
3 ^ISJ(3)     $lb("??","40","??")
4 ^ISJ(4)     $lb("??","30","??")
>>>
```

こちらの結果の value を Name, Age, Address に分けて変換したい場合、既存の %Global.cls のクエリで行うことはできないため、

1. **IRIS側で、あらかじめリスト形式($LB)を分解してから処理する** か、
2. **Python側で、IRISのリスト形式($LB)のまま格納されたデータを文字列置換などして DataFrame を作り直す**

のどちらかを行う必要があります。

上記1の「**IRIS側で処理する**」場合、カスタムクラスクエリ
を使用してグローバル内のリストの各データを返すクエリを作成し、それをSQL経由でアクセスする方法が考えられます。
カスタムクラスクエリを使用する方法は、
SQL文ではなくユーザコードでクラスクエリを記述する方法はありますか？ の記事でご紹介しています。

上の記事で紹介しているサンプルクラスを使用して、^ISJ
のリストを要素別に抽出するサンプルを作成してみました。

^ISJの value の結果列が$LB形式で3要素なので、上の記事で使用しているサンプルを以下のように変更します。

★GFetchクラスメソッド

```
//Set Row=$LB($na(@glvn@(x)),@glvn@(x))
 Set Row=$LB($na(@glvn@(x)),$LIST(@glvn@(x),1),$LIST(@glvn@(x),2),$LIST(@glvn@(x),3))
```

★Gクエリ

```
// Query G(glvn As %String) As %Query(CONTAINID = 0, ROWSPEC = "Node:%String, Value:%
String") [ SqlProc ]
Query G(glvn As %String) As %Query(CONTAINID = 0, ROWSPEC =
"Node:%String, Value1:%String, Value2:%String, Value3:%String") [ SqlProc ]
```

実行例は以下のようになります。

```
USER>do $system.Python.Shell()

Python 3.9.5 (default, Apr 15 2022, 01:28:04) [MSC v.1927 64 bit (AMD64)] on win32
Type quit() or Ctrl-D to exit this shell.
>>> mysql="select * from SQLUSER.TestStoredProc1_G('^ISJ')"
>>> resultset = iris.sql.exec(mysql)
>>> dataframe = resultset.dataframe()
>>> print (dataframe)
     node value1 value2  value3
0 ^ISJ(1)  Name   Age Address
1 ^ISJ(2)    ??    50      ??
2 ^ISJ(3)    ??    40      ??
3 ^ISJ(4)    ??    30      ??
>>>
```

今回使用したサンプルはこちらにあります

https://github.com/Intersystems-jp/GlobalToPandasDataframe

~

(2022/06/01現在)
日本語対応はしていませんが、以下のような方法もあります。こちらは将来のバージョンで日本語対応予定です。
%SYS.Pythonクラスの ToList()メソッド**を使用して、IRISリストをPythonリストに変換する方法**です。

１．以下のようなクラスを作成します。

```
ClassMethod toPythonList(gname As %String, i As %Integer) As %SYS.Python
{
    quit ##class("%SYS.Python").ToList(@gname@(i))
}

ClassMethod getDataFrame(gname As %String) [ Language = python ]
{
    import iris
    import pandas as pd
    g = iris.gref(gname)
    cnt=g[None]
    newlist=[]
    for i in range(1,cnt+1):
     datalist=iris.cls(__name__).toPythonList(gname, i)
     newlist.append(datalist)
    newdf=pd.DataFrame(newlist[1:cnt],columns=[newlist[0][0],newlist[0][1
],newlist[0][2]])
    print(newdf)
}
```

２．次のように実行します。使用するのは ^ISJ2
のような英数字のみのデータが含まれるリスト形式のグローバルです。

```
USER>zw ^ISJ2
^ISJ2=4
^ISJ2(1)=$lb("Name","Age","Address")
^ISJ2(2)=$lb("Sato","50","Tokyo")
^ISJ2(3)=$lb("Kato","40","Osaka")
^ISJ2(4)=$lb("Ito","30","Kyoto")

USER>do ##class(%SYS.Python).Shell()

Python 3.9.5 (default, Apr 15 2022, 01:28:04) [MSC v.1927 64 bit (AMD64)] on win32
Type quit() or Ctrl-D to exit this shell.
>>> iris.cls('User.PythonTest').getDataFrame('^ISJ2')
   Name  Age  Address
0  Sato   50    Tokyo
1  Kato   40    Osaka
2   Ito   30    Kyoto
```

#Embedded Python #Python #InterSystems IRIS #InterSystems IRIS for Health

ソースURL:https://jp.community.intersystems.com/post/embedded-python-%E3%81%A7-iris-%E3%82%B0%E3%83%AD%E3%83%BC%E3%83%90%E3%83%ABlb-%E3%82%92-pandas-dataframe-%E3%81%AB%E5%A4%89%E6%8F%9B%E3%81%99%E3%82%8B%E6%96%B9%E6%B3%95