

Artículo

[Luis Angel Pére...](#) · 24 ene, 2023 Lectura de 3 min

## Cómo usar Python Embebido para generar un flujo continuo de datos

¡Hola desarrolladores!

Python tiene un ecosistema muy extenso y potente que contiene miles de librerías y paquetes disponibles, especialmente en Data Science.

Por ello, quería hacer un primer intento de usar la funcionalidad Python Embebido para, de forma sencilla, importar una librería de Python llamada `datetime`, generar datos con un componente timestamp y persistirlo en InterSystems IRIS for Health. También funcionará en InterSystems IRIS.

He dividido este pequeño proyecto en 2 partes principales:

- Configuración del entorno de Python Embebido siguiendo la documentación de InterSystems IRIS for Health
- Creación del conjunto de datos timestamp como ejemplo usando Python Embebido y persistencia de los mismos en IRIS usando el paquete `datetime`

### 1. Configuración Python Embebido en InterSystems IRIS for Health 2022.1

- En Windows, abre el terminal CMD:

```
C:\InterSystems\IRISHealth\bin>irispip install --target C:\InterSystems\IRISHealth\mgr python numpy
```

Nota: Este es el paso para comprobar el entorno de Python Embebido, porque cuando usé IRIS 2021.0 descubrí que el `irispip` no funciona

- Probar la funcionalidad Python importando el paquete `math` de Python

Abre el terminal de IRIS y ejecuta lo siguiente en el namespace USER: `set pymath = ##class(%SYS.Python).Import("math")`

Después, ejecutar `write pymath.pi`, Podrás ver que has llamado correctamente al paquete Python, con el siguiente resultado:

```
USER>write pymath.pi
```

```
3.141592653589793116
```

- También podemos empezar el shell de Python abriéndolo desde el terminal de IRIS:

```
do ##class(%SYS.Python).Shell()
```

```
Class User.PythonFirstTry Extends %RegisteredObject
{

ClassMethod pyHello() As %Status
{
set pythonBuiltins = ##class(%SYS.Python).Builtins()
do pythonBuiltins.print("Hello World!")
}

ClassMethod pyForLoop() [ Language = python ]
{
for i in range(5):
print("Python")
}

}
```

He escrito dos métodos de clase que puedes poner en un archivo .cls, compilarlos y ver el resultado.

Ejecuta los métodos así:

```
USER>do ##class(User.PythonFirstTry).pyHello()
```

Hello World!

```
USER>do ##class(User. PythonFirstTry).pyForLoop()
```

Python

Python

Python

Python

Python

## 2. Usar la librería de Python para generar un flujo de datos y persistir en InterSystems IRIS

Librería de Python necesaria para la generación del flujo de datos:

- Import datetime package (no requiere instalación extra usando CLI)

Además del paquete datetime, intenté usar otras librerías y paquetes. Sin embargo, algunas de ellas no están presentes de forma nativa con Python Embebido. Para instalarlas, abre el terminal CMD (para Windows), ejecuta lo siguiente (XX se reemplaza con el nombre del paquete):

```
C:\InterSystems\IRISHealth\bin>irispip install XX
```

Estos son los pasos seguidos para generar datos y persistir en IRIS.

1. Crea la clase %Persistent
2. Crea la Property
3. Define un método de clase (por ejemplo, GetData) y especificar [Language = python]
4. Genera datos llamando a la librería datetime de Python, usando este fragmento de código:

```
Class PYTHONLEARNINGPKG.toIris Extends %Persistent
{

Property dayTime As %String;

Debug this method
ClassMethod GetData() As %Status [ Language = python ]
{
    import iris
    import datetime
    timeNow = datetime.datetime.now()
    print(timeNow)
    setTime = iris.cls("PYTHONLEARNINGPKG.toIris")._New()
    setTime.dayTime = str(timeNow)
    setTime._Save()
}
```

Nota: timeNow necesita ser convertido en un tipo de datos String

5. Ejecuta el método GetData() a través del Terminal de IRIS, ejecuta la consulta SQL

Nota: Podrías usar un bucle for simple para controlar el volumen de datos que es generado mediante este método.

« **Wizards** » **Actions** » Open Table **Tools** » **Documentation** »

Catalog Details **Execute Query** Browse SQL Statements

Execute Show Plan Show History Query Builder Display Mode ▼ Max 1000 more

```
SELECT  
ID, dayTime  
FROM PYTHONLEARNINGPKG.toIris
```

Row count: 1000 Performance: 0.029 seconds 1323 global references 52389 commands executed 0 disk read latency (ms) Cached Query: [%sqlcq.PYTH](#)

17	2022-06-28 14:43:06.497285
18	2022-06-28 14:43:06.497285
19	2022-06-28 14:43:06.497285
20	2022-06-28 14:43:06.497285
21	2022-06-28 14:43:06.497285
22	2022-06-28 14:43:06.497285
23	2022-06-28 14:43:06.497285
24	2022-06-28 14:43:06.497285
25	2022-06-28 14:43:06.497285
26	2022-06-28 14:43:06.497285
27	2022-06-28 14:43:06.497285
28	2022-06-28 14:43:06.503832
29	2022-06-28 14:43:06.503832
30	2022-06-28 14:43:06.503832
31	2022-06-28 14:43:06.503832
32	2022-06-28 14:43:06.503832
33	2022-06-28 14:43:06.503832

[#Embedded Python](#) [#InterSystems IRIS](#) [#InterSystems IRIS for Health](#)

URL de  
fuente: <https://es.community.intersystems.com/post/c%C3%B3mo-usar-python-embecido-para-generar-un-flujo-continuo-de-datos>