Article
Heloisa Paiva · Mar 9, 2023 2m read

Open Exchange

SQLAlchemy - the easiest way to use Python and SQL with IRIS's databases

Introduction

In some of the last few articles I've talked about types between IRIS and Python, and it is clear that it's not that easy to access objects from one side at another.

Fortunately, work has already been done to create SQLAlchemy-iris (follow the link to see it on Open Exchange), which makes everything much easier for Python to access IRIS' objects, and I'm going to show the starters for that.

Thank you @Dmitry Maslennikov !

Installing

To install, simply open your terminal with administrators' access and type

```
pip install sqlalchemy-iris
```

and that will also install the pre-requisites for you, if needed.

Usage

Now, on a python file, you can import the module, connect to the database and play with sqlalchemy in any way you want. If you feel comfortable, you may follow the steps:

- Import "createengine" from sqlalchemy and create the engine with the string "iris://username:password@IP:port/namespace". Of course you can import the whole module, but "createengine" creates an instance of Engine (sqlalchemy.engine, for more info click here) has all the subclasses necessary for what I'm presenting here.

```
from sqlalchemy import create_engine

engine = create_engine("iris://_SYSTEM:SYS@localhost:1972/SAMPLE")
```

- Create the connection (sqlalchemy.engine.connection, for more info click here) so you can work with transactions, simple execute, etc.

```
conn = engine.connect()
```

Great! Now you have configured access to your database.

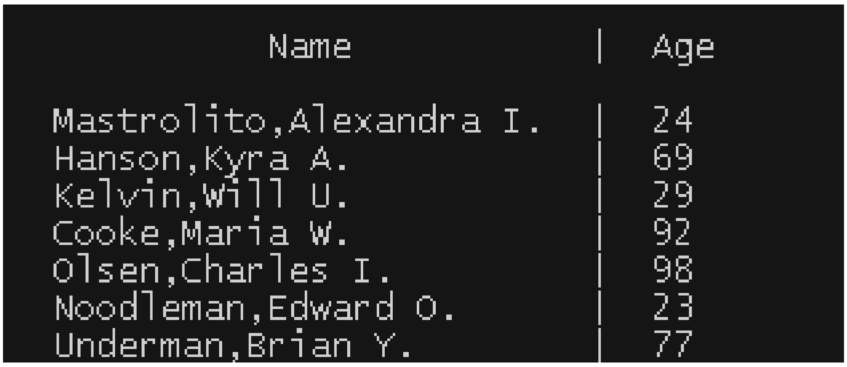
For a simple SELECT query, and to iterate through the result set, do the following (as usual, using as example the table created in this article):

```
query = 'SELECT Name, Age  from Sample.PersistentData WHERE Age >=21'
result = conn.exec_driver_sql(query)
```

Now, *result* is a CursorResult (sqlalchemy.engine.CursorResult). You can check everything you can do with a CursorResult here, in the official documentation. This is all you have to do for a simple iteration:

```
print("Name, Age")
for row in result:
    print(row[0], ", ", row[1])
```

and with a little formatting you may have an output like this:



```
              Name               |  Age

 Mastrolito,Alexandra I.         |  24
 Hanson,Kyra A.                  |  69
 Kelvin,Will U.                  |  29
 Cooke,Maria W.                  |  92
 Olsen,Charles I.                |  98
 Noodleman,Edward O.             |  23
 Underman,Brian Y.               |  77
```

You can also check the official documentation for more details and possibilities, now that you know where to start.

PS.: you can also import "text" from sqlalchemy and execute the query as

```
result = conn.execute(text(query))
```

which will produce the exact same results.

Conclusion

You can also perform DDL and any other DML statements, and there's even bigger support to work with ORM (*Object Relational Mapping),* but since this is just a "kick off" I'm not going further.

Would you like more tutorials with SQLAlchemy? Maybe next time using ORM?
Feel free to contact me for any doubts!

#Innovatium #Python #SQL #Tutorial #InterSystems IRIS #Open Exchange
Check the related application on InterSystems Open Exchange

Source URL:https://community.intersystems.com/post/sqlalchemy-easiest-way-use-python-and-sql-iriss-databases