

Article

[Athanasios Hatzis](#) · Jul 23, 2018 5m read

---

## InterSystems Cache Python Object-Relational Mapper

Hi,

this is a public announcement for the first release of InterSystems Cache Object-Relational Mapper in Python 3. Project's main repository is located at Github ([healiseu/InterSystemsCacheORM](https://github.com/healiseu/InterSystemsCacheORM)).

### About the project

CacheORM module is an enhanced OOP porting of InterSystems [Cache-Python binding](#). There are three classes implemented:

- CacheClient This is the super class of CachePython module. It wraps two functions from intersys.pythonbind module [pythonbind3.connection\(\)](#) and [pythonbind3.database\(\)](#).
- CacheQuery A subclass of CacheClient that wraps methods and adds extra functionality in [intersys.pythonbind.database](#) and [intersys.pythonbind.query](#) classes
- CacheClass A subclass of CacheClient, that wraps methods and adds extra functionality in [intersys.pythonbind.database](#) and [intersys.pythonbind.object](#) classes

The [intersys.pythonbind package](#) is a Python C extension that provides Python application with transparent connectivity to the objects stored in the Caché database.

### Source Code

The project's code that is released to the public was originally written and used as a module of [TRIADB](#) project.

### Tests and Demos

There are two folders in this release:

- [testCacheORM](#) contains python jupyter notebook files that demonstrate CacheQuery and CacheClass
- [testCacheBinding](#) are tests written for InterSystems Cache python binding

One can simply compare tests with demos to appreciate the work in this project to leverage intersystems cache python binding. For example

```
# InterSystems Cache Python binding for queries
```

```
import intersys.pythonbind3

# Create a connection
user="_SYSTEM";
password="123";
host = "localhost";
port = "1972";
url = host+"["+port+"]:SAMPLES"
conn = intersys.pythonbind3.connection()

# Connect Now to SAMPLES namespace
conn.connect_now(url, user, password, None)

# Create a database object
samplesDB = intersys.pythonbind3.database(conn)

# create a query object
cq = intersys.pythonbind3.query(samplesDB)

# prepare and execute query
sql = "SELECT ID, Name, DOB, SSN FROM Sample.Person"
cq.prepare(sql)
cq.execute()

# Fetch rows
for x in range(0,10):
    print(cq.fetch([None]))
```

Same code in only 4 lines using CacheORM python module

```
from CacheORM import CacheQuery
```

```
samplesquery = CacheQuery(namespace='SAMPLES', username='SYSTEM', password='SYS', dl=99)
```

```
samplesquery.executesql('SELECT ID, Name, DOB, SSN FROM Sample.Person')
```

```
samplesquery.printrecords(10)
```

You can view the output from this python Jupyter Notebook at my [Microsoft Azure CacheORM library](#)

## Another example, this time with Cache-Python Objects

```
# Demo of InterSystems Cache Python binding with Samples namespace and Sample.Person class
```

```
import intersys.pythonbind3
```

```
conn = intersys.pythonbind3.connection( )
```

```
conn.connect_now('localhost[1972]:SAMPLES', '_SYSTEM', '123', None)
```

```
samplesDB = intersys.pythonbind3.database(conn)
```

```
##% Create a new instance of Sample.Person to be husband
```

```
husband = samplesDB.create_new("Sample.Person", None)
```

```
ssn1 = samplesDB.run_class_method("%Library.PopulateUtils", "SSN", [])
```

```
dob1 = samplesDB.run_class_method("%Library.PopulateUtils", "Date", [])
```

```
husband.set("Name","Hatzis, Athanassios I")
husband.set("SSN",ssn1)
husband.set("DOB",dob1)

# Save husband
husband.run_obj_method("%Save",[])
print ("Saved id: "+str(husband.run_obj_method("%Id",[])))

### Create a new instance of Sample.Person to be wife
wife = samplesDB.create_new("Sample.Person", None);
ssn2 = samplesDB.run_class_method("%Library.PopulateUtils","SSN",[])
dob2 = samplesDB.run_class_method("%Library.PopulateUtils","Date",[])
wife.set("Name","Kalamari, Panajota");
wife.set("SSN",ssn2)
wife.set("DOB",dob2)

# Save wife
wife.set("Spouse",husband);
wife.run_obj_method("%Save",[]);
print ("Saved id: " + str(wife.run_obj_method("%Id",[])))

### Relate them
husband.set("Spouse",wife);
husband.run_obj_method("%Save",[]);

wife.set("Spouse",husband);
wife.run_obj_method("%Save",[]);

# Open an instance of the Sample.Person object
athanID=217
athanPerson = samplesDB.openid("Sample.Person",str(athanID),-1,-1)

# Open another instance
otherID=3
otherPerson = samplesDB.openid("Sample.Person",str(otherID),-1,-1)

# Fetch some properties
print ("ID: " + otherPerson.run_obj_method("%Id",[]))
print ("Name: " + otherPerson.get("Name"))
print ("SSN: " + otherPerson.get("SSN"))
print ("DOB: " + str(otherPerson.get("DOB")))
print ("Age: " + str(otherPerson.get("Age")))
```

Same code using CacheORM python module, i.e. object-relational mapping

```
from CacheORM import CacheClass

# Create an instance of PopulateUtils to call built-in CACHE class method
populateUtils = CacheClass(namespace='%SYS', cachepackage='%Library', cacheclass='PopulateUtils', username='_SYSTEM', password='SYS')

# Create CacheClass Instance
husband = CacheClass(username='_SYSTEM', password='SYS', dl=99)

# Create and populate a new CacheClass Object
```

```
husband.new()
husband.set_value("SSN",populateUtils.class_method("SSN"))
husband.set_value("Name", "Hatzis, Athanassios I")
husband.set_value("DOB", populateUtils.class_method("Date"))

# Save husband
husband.save()

# Create another CacheClass object
wife = CacheClass(username='_SYSTEM', password='SYS')
wife.new()
wife.set_value("SSN",populateUtils.class_method("SSN"))
wife.set_value("Name", "Kalamari, Panajota")
wife.set_value("DOB", populateUtils.class_method("Date"))
wife.save()

# Relate them
wife.set_refobj("Spouse", person._cache_id)
wife.save()

husband.set_refobj("Spouse", female._cache_id)
husband.save()

# Get Object References
person.get("Spouse").get("Name")
female.get("Spouse").get("Name")

# Open an existing object with id=3 and read cache properties
person = CacheClass(username='_SYSTEM', password='SYS', objectID='3')
print(f"ID:{person.id}\nSSN: {person.get('SSN')}\nName: {person.get('Name')}\nDateOfBirth: {person.get('DOB')}")
```

You can view the output from this python Jupyter Notebook at my [Microsoft Azure CacheORM library](#)

[#Object Data Model](#) [#Python](#) [#Caché](#)

---

Source URL:<https://community.intersystems.com/post/intersystems-cache-python-object-relational-mapper>