

# Python KeySQL\_client for KeySQL<sup>®</sup> Server

## Creating catalog, k-objects, and store, and inserting instances

### Style used:

KeySql code:

```
CREATE CATALOG birds;  
CREATE STORE backyard_birds for catalog birds;
```

Output of KeySql queries:

```
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'summer',ACT_TYPE:'bath'}}},  
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'summer',ACT_TYPE:'eat'}}},
```

Python code :

Python code

Python output:

Python output

## String style coding using python keysql server

KeySQL Example:

```
CREATE CATALOG birds;  
CREATE STORE backyard_birds for catalog birds;
```

```
CREATE KEYOBJECT season CHAR in catalog birds;  
CREATE KEYOBJECT ACT_TYPE CHAR in catalog birds;  
CREATE KEYOBJECT activity {ACT_TYPE, season} in catalog birds;  
CREATE KEYOBJECT name CHAR in catalog birds;  
CREATE KEYOBJECT bird {name, activity} in catalog birds;
```

```
INSERT INTO backyard_birds INSTANCES  
{bird: {name: 'house sparrow', activity: {act_type: 'bath', season: 'summer'}}},  
{bird: {name: 'house sparrow', activity: {act_type: 'eat', season: 'summer'}}},  
{bird: {name: 'house sparrow', activity: {act_type: 'sing', season: 'summer'}}},
```

*Copyright (C) 2017-2020 by KEYARK, INC. - All Rights Reserved.*

*All information contained herein is proprietary and confidential to KEYARK, INC.*

*Any use, reproduction, or disclosure without the written permission of KEYARK, INC. is prohibited.*



```
{bird: {name: 'house sparrow', activity: { act_type: 'bath', season: 'fall'}}},
{bird: {name: 'house sparrow', activity: { act_type: 'eat', season: 'fall'}}},
{bird: {name: 'house sparrow', activity: { act_type: 'sing', season: 'fall'}}},
{bird: {name: 'white-crowned sparrow', activity: {act_type: 'bath', season: 'fall'}}},
{bird: {name: 'white-crowned sparrow', activity: {act_type: 'eat', season: 'fall'}}},
{bird: {name: 'white-crowned sparrow', activity: {act_type: 'sing', season: 'fall'}}};
```

```
SELECT * from backyard_birds;
```

The result of this query is:

```
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'summer',ACT_TYPE:'bath'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'summer',ACT_TYPE:'eat'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'summer',ACT_TYPE:'sing'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'bath'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'eat'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'sing'}}},
{BIRD:{NAME:'white-crowned
sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'bath'}}},
{BIRD:{NAME:'white-crowned
sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'eat'}}},
{BIRD:{NAME:'white-crowned
sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'sing'}}}
```

## Using KeySql Python driver for the same example

**Step 1** – CREATE CONNECTION, using host, port, user\_name, password

Host and port are the host and port of the KeySQL Server.

User\_name and password are given to the user.

```
test_connection = KeySqlConnection (host, port, user_name, user_password,
schema=user_schema)
```

Example of parameters:

host = "1.0.0.123" (as type string)

port = 123 (as type int)

user\_name = "user\_name\_example" (as type string)

user\_password = "user\_password\_example" (as type string)

user\_schema = "user\_schema" (as type string)

If schema is not specified, the default schema will be selected

*Copyright (C) 2017-2020 by KEYARK, INC. - All Rights Reserved.*

*All information contained herein is proprietary and confidential to KEYARK, INC.*

*Any use, reproduction, or disclosure without the written permission of KEYARK, INC. is prohibited.*



## Step 2 – Create catalog, store, and keyobjects

1. Create catalog (string can be both low and upper cases ):

```
test_connection.execute('CREATE catalog birds')
```

2. Create keyobjects:

```
test_connection.execute('CREATE KEYOBJECT season char in catalog birds')
test_connection.execute('CREATE KEYOBJECT type char in catalog birds')
test_connection.execute('CREATE KEYOBJECT name char in catalog birds')
test_connection.execute('CREATE KEYOBJECT activity {type, season} in catalog birds')
```

3. Create store:

```
test_connection.execute('CREATE store backyard_birds for catalog birds')
```

To check if all items are created:

```
reply = test_connection.execute("SHOW catalog birds")
```

The reply will contain all created k-objects:

```
['ACTIVITY {SEASON,TYPE}', 'BIRD {ACTIVITY,NAME}', 'NAME CHAR', 'SEASON CHAR', 'TYPE CHAR']
```

## Step 3 – Insert instances into the store

Instances can be inserted one-by-one or as a list

Insert one instance:

```
test_connection.execute("insert into backyard_birds instances {bird: {name: 'house sparrow',  
activity: {type: 'bath', season: 'summer'}}}")
```

Insert list of instances:

```
test_connection.execute("insert into backyard_birds instances \  
{bird: {name: 'house sparrow', activity: {type: 'eat', season: 'summer'}}},\  
{bird: {name: 'house sparrow', activity: {type: 'sing', season: 'summer'}}},\  
{bird: {name: 'house sparrow', activity: {type: 'bath', season: 'fall'}}},\  
{bird: {name: 'house sparrow', activity: {type: 'eat', season: 'fall'}}},\  
{bird: {name: 'house sparrow', activity: {type: 'sing', season: 'fall'}}},\  
{bird: {name: 'white-crowned sparrow', activity: {type: 'bath', season: 'fall'}}},\  
{bird: {name: 'white-crowned sparrow', activity: {type: 'eat', season: 'fall'}}},\  
{bird: {name: 'white-crowned sparrow', activity: {type: 'sing', season: 'fall'}}}")
```

**Step 4 - Check inserted instances:**

```
reply = test_connection.execute("select * from backyard_birds")  
print(reply)
```

The result is a list of python classes:

```
[<class 'keysql_client_package.keysql_mapper.BIRD'>,  
<class 'keysql_client_package.keysql_mapper.BIRD'>,  
<class 'keysql_client_package.keysql_mapper.BIRD'>,  
<class 'keysql_client_package.keysql_mapper.BIRD'>,  
<class 'keysql_client_package.keysql_mapper.BIRD'>,  
<class 'keysql_client_package.keysql_mapper.BIRD'>,  
<class 'keysql_client_package.keysql_mapper.BIRD'>,  
<class 'keysql_client_package.keysql_mapper.BIRD'>,  
<class 'keysql_client_package.keysql_mapper.BIRD'>]
```

To get results as keysql strings, query should be executed with flag `keep_str_res=True`  
(keep string results)

*Copyright (C) 2017-2020 by KEYARK, INC. - All Rights Reserved.  
All information contained herein is proprietary and confidential to KEYARK, INC.  
Any use, reproduction, or disclosure without the written permission of KEYARK, INC. is prohibited.*



```
reply = test_connection.execute("select * from backyard_birds", keep_str_res=True)
print(test_connection.get_last_str_result())
```

The output will be standard keysql output:

```
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'summer',ACT_TYPE:'bath'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'summer',ACT_TYPE:'eat'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'summer',ACT_TYPE:'sing'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'bath'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'eat'}}},
{BIRD:{NAME:'house sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'sing'}}},
{BIRD:{NAME:'white-crowned sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'bath'}}},
{BIRD:{NAME:'white-crowned sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'eat'}}},
{BIRD:{NAME:'white-crowned sparrow',ACTIVITY:{SEASON:'fall',ACT_TYPE:'sing'}}}
```

Another query:

```
test_connection.execute('create keyobject cnt number in catalog birds')
reply = test_connection.execute("select name, season, count(season) as cnt from backyard_birds
group by season", keep_str_res=True)
print(test_connection.get_last_str_result())
```

Result:

```
{RESULT:{#NAME:{NAME:'white-crowned sparrow',NAME:'house sparrow',NAME:'house
sparrow',NAME:'white-crowned sparrow',NAME:'house sparrow',NAME:'white-crowned
sparrow'},SEASON:'fall',CNT:6}},
{RESULT:{#NAME:{NAME:'house sparrow',NAME:'house sparrow',NAME:'house
sparrow'},SEASON:'summer',CNT:3}}
```

**Last Step** Disconnect from server

```
test_connection.disconnect()
```