
Python Client for Pilosa Documentation

Release 0.3.30

Pilosa Engineering

Apr 30, 2019

Contents:

1	pilosa package	3
1.1	Submodules	3
1.2	pilosa.client module	3
1.3	pilosa.exceptions module	3
1.4	pilosa.orm module	3
1.5	pilosa.response module	3
1.6	pilosa.validator module	3
1.7	pilosa.version module	3
1.8	Module contents	3
2	Requirements	5
3	Install	7
4	Quick overview	9
5	Indices and tables	11

Python client for [Pilosa](#) high performance distributed row index.

CHAPTER 1

pilosa package

1.1 Submodules

1.2 pilosa.client module

1.3 pilosa.exceptions module

1.4 pilosa.orm module

1.5 pilosa.response module

1.6 pilosa.validator module

1.7 pilosa.version module

1.8 Module contents

CHAPTER 2

Requirements

- Python 2.7 and higher or Python 3.4 and higher.

CHAPTER 3

Install

Pilosa client is on [PyPI](#). You can install the library using `pip`:

```
pip install pilosa
```


CHAPTER 4

Quick overview

Assuming Pilosa server is running at localhost:10101 (the default):

```
import pilosa

# Create the default client
client = pilosa.Client()

# Retrieve the schema
schema = client.schema()

# Create an Index object
myindex = schema.index("myindex")

# Create a Field object
myfield = myindex.field("myfield")

# make sure the index and field exists on the server
client.sync_schema(schema)

# Send a SetBit query. PilosaError is thrown if execution of the query fails.
client.query(myfield.set(5, 42))

# Send a Bitmap query. PilosaError is thrown if execution of the query fails.
response = client.query(myfield.row(5))

# Get the result
result = response.result

# Act on the result
if result:
    columns = result.row.columns
    print("Got columns: ", columns)

# You can batch queries to improve throughput
```

(continues on next page)

(continued from previous page)

```
response = client.query(  
    myindex.batch_query(  
        myfield.row(5),  
        myfield.row(10),  
    )  
)  
  
for result in response.results:  
    # Act on the result  
    print(result)
```

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`