
superdatabase3000

Release 0.2.0

JeanMax

Mar 31, 2020

CONTENTS:

1	superdatabase3000 package	1
1.1	DbServer	1
1.2	DbClient	1
Index		3

SUPERDATABASE3000 PACKAGE

1.1 DbServer

class superdatabase3000.**DbServer** (*sock_filename=None, hdf_filename=None*)

A server running your favorite database.

Parameters

- **sock_filename** (*str*) – path to the DbServer socket (default: “/tmp/superdatabase3000.sock”)
- **hdf_filename** (*str*) – path to the DbServer hdf store (default: “~/superdatabase3000.hdf”)

Example

```
>>> server = DbServer(  
...     sock_filename="/tmp/db.sock",  
...     hdf_filename="/tmp/db.h5"  
... )  
>>> server.read_loop()
```

read_loop()

Poll events in an infinite loop. Check for events on the server.

This will handle:

- accepting/removing clients
- sending messages (the ones added to the queue with `send_to`)
- reading messages

1.2 DbClient

class superdatabase3000.**DbClient** (*sock_filename=None*)

A client to interact with your favorite database.

Parameters **sock_filename** (*str*) – path to the DbServer socket (default: “/tmp/superdatabase3000.sock”)

Example

Assuming a server is already launched

```
>>> client = DbClient("/tmp/db.sock")
>>> df = client.select("/toto")
```

delete (*table*, *where*)

Drop the rows matching the ‘where’ close on the given ‘table’.

Parameters

- **table** (*str*) – the name of the table to query
- **where** (*str*) – for the ‘where’ syntax, see: https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html#querying-a-table

Returns *bool* – False if the table doesn’t exist.

drop (*table*)

Drop the given ‘table’.

Parameters **table** (*str*) – the name of the table to query

Returns *bool* – False if the table doesn’t exist.

insert (*table*, *df*)

Insert the DataFrame ‘df’ to the given ‘table’. Might update values.

The table must always be sorted, so you should prefer inserting to the end (with growing indexes) if you need better performances.

Parameters

- **table** (*str*) – the name of the table to query
- **df** (*DataFrame*) – the DataFrame to insert in the database

Returns *bool* – False if the table doesn’t exist.

select (*table*, *where=None*, *columns=None*, *start=None*, *stop=None*)

Select rows (as a DataFrame) from the given ‘table’.

Parameters

- **table** (*str*) – the name of the table to query
- **where** (*str*) – for the ‘where’ syntax, see: https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html#querying-a-table
- **columns** (*list*) – a list of columns that will limit the returned columns
- **start** (*int*) – row number to start selection (negative index allowed)
- **stop** (*int*) – row number to stop selection (negative index allowed)

Returns *The selected DataFrame, otherwise None if something funky happens.*

INDEX

D

`DbClient` (*class in superdatabase3000*), 1
`DbServer` (*class in superdatabase3000*), 1
`delete()` (*superdatabase3000.DbClient method*), 2
`drop()` (*superdatabase3000.DbClient method*), 2

I

`insert()` (*superdatabase3000.DbClient method*), 2

R

`read_loop()` (*superdatabase3000.DbServer method*),
1

S

`select()` (*superdatabase3000.DbClient method*), 2