

---

# TS3Py Documentation

*Release 0.4.2*

**Thor77**

February 08, 2017



<b>1</b>	<b>API</b>	<b>3</b>
1.1	Low-level . . . . .	3
1.2	High-level . . . . .	3



TS3Py is a object-orientated interface to Teamspeak3's query-interface.

Let's take a look:

```
>>> s = Server('ts.example.com')
>>> s.login('serveradmin', 'password')
>>> s.virtual_servers
[Server1, Server2]
>>> s.virtual_servers[0].clients
[serveradmin from 1.1.1.1:39004, thor77]
```



## 1.1 Low-level

```
class ts3py.Query(ip, port=10011)
```

```
    command(cmd, params={}, options=[])
```

Send a command to the Teamspeak3-server and return the response.

**Parameters**

- **cmd** (*string*) – command
- **params** (*dict*) – parameters appended to the command
- **options** (*list*) – options appended to the command

**Returns** response of the command (if any)

**Return type** list

```
    connect(ip, port=10011)
```

Connect to a Teamspeak3-Server.

**Parameters**

- **ip** (*string*) – ip of the Teamspeak3-server
- **port** (*int*) – port of the Teamspeak3-server's query-interface

```
    disconnect()
```

Disconnect from the Teamspeak3-server.

## 1.2 High-level

```
class ts3py.objects.Server(host, port=10011)
```

```
    gm(message)
```

Send a text message to all clients on all virtual servers

**Parameters** **message** (*str*) – message to be sent

```
    login(username, password)
```

Login to an user-account on the server.

**Parameters**

- **username** (*str*) – name of the user
- **password** (*str*) – password of the user

**logout()**

Logout from the current session

**virtual\_servers**

Return a list of virtual-servers on that server.

**Returns** list of virtual-servers

**Return type** list

**class** ts3py.objects.**VirtualServer** (*query, params={}*)

**channel**

Return a list of channel on that virtual-server.

**Returns** list of channel

**Return type** list

**clients**

Return a list of clients on that virtual-server.

**Returns** list of clients

**Return type** list

**command** (*cmd, params={}, options=[]*)

Run a command in the context of the virtual-server.

**Parameters**

- **cmd** (*string*) – command
- **params** (*dict*) – parameters appended to the command
- **options** (*list*) – options appended to the command

**Returns** response of the command (if any)

**Return type** list

**class** ts3py.objects.**Channel** (*server, params={}*)

**clients**

Return a list of clients in that channel.

**Returns** list of clients

**Return type** list

**class** ts3py.objects.**Client** (*server, params={}*)



## C

Channel (class in ts3py.objects), 4  
channel (ts3py.objects.VirtualServer attribute), 4  
Client (class in ts3py.objects), 4  
clients (ts3py.objects.Channel attribute), 4  
clients (ts3py.objects.VirtualServer attribute), 4  
command() (ts3py.objects.VirtualServer method), 4  
command() (ts3py.Query method), 3  
connect() (ts3py.Query method), 3

## D

disconnect() (ts3py.Query method), 3

## G

gm() (ts3py.objects.Server method), 3

## L

login() (ts3py.objects.Server method), 3  
logout() (ts3py.objects.Server method), 4

## Q

Query (class in ts3py), 3

## S

Server (class in ts3py.objects), 3

## V

virtual\_servers (ts3py.objects.Server attribute), 4  
VirtualServer (class in ts3py.objects), 4