

---

# **Asyncflux Documentation**

*Release 0.1+*

**Jorge Puente-Sarrín**

August 09, 2015



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Documentation</b>	<b>5</b>
<b>3</b>	<b>License</b>	<b>7</b>
<b>4</b>	<b>Indices and tables</b>	<b>9</b>
4.1	Package Documentation . . . . .	9
4.2	Release Notes . . . . .	12
	<b>Python Module Index</b>	<b>13</b>



Asynchronous client for [InfluxDB](#) and [Tornado](#).



---

## Installation

---

You can use `pip` to install Asyncflux:

```
$ pip install git+https://github.com/puentesarrin/asyncflux.git
```





---

## Documentation

---

`Sphinx` is needed to generate the documentation. Documentation can be generated by issuing the following commands:

```
$ cd docs
$ make html
```

Or simply:

```
$ python setup.py doc
```

Also, the current documentation can be found at [ReadTheDocs](#).



---

**License**

---

Asyncflux is available under the [Apache License, Version 2.0](#).



---

## Indices and tables

---

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

## 4.1 Package Documentation

### 4.1.1 `asyncflux` Package

`asyncflux.version = '0.1+'`  
Current version of `Asyncflux`.

### 4.1.2 `asyncflux` Modules

#### `asyncflux.client` – Connection to InfluxDB

```
class asyncflux.client.AsyncfluxClient (host=None, port=None, username=None, password=None, is_secure=False, io_loop=None, **kwargs)
```

```
client[db_name] || client.db_name
```

Get the `db_name` *Database* on *AsyncfluxClient* client.

```
HOST = 'localhost'
```

```
PORT = 8086
```

```
USERNAME = 'root'
```

```
PASSWORD = 'root'
```

```
host
```

```
port
```

```
base_url
```

```
username
```

```
password
```

```
request (*args, **kwargs)
```

```
ping (*args, **kwargs)
query (*args, **kwargs)
get_databases (*args, **kwargs)
get_database_names (*args, **kwargs)
create_database (*args, **kwargs)
drop_database (*args, **kwargs)
get_users (*args, **kwargs)
get_user_names (*args, **kwargs)
create_user (*args, **kwargs)
change_user_password (*args, **kwargs)
drop_user (*args, **kwargs)
grant_privilege (*args, **kwargs)
revoke_privilege (*args, **kwargs)
grant_admin_privileges (*args, **kwargs)
revoke_admin_privileges (*args, **kwargs)
```

#### asyncflux.database – Database level operations

```
class asyncflux.database.Database (client, name)
```

```
BATCH_SIZE = 5000
client
name
query (*args, **kwargs)
write (*args, **kwargs)
write_points (*args, **kwargs)
get_measurements (*args, **kwargs)
get_tag_keys (*args, **kwargs)
get_tag_values (*args, **kwargs)
get_series (*args, **kwargs)
drop_series (*args, **kwargs)
grant_privilege_to (*args, **kwargs)
revoke_privilege_from (*args, **kwargs)
get_retention_policies (*args, **kwargs)
get_retention_policy_names (*args, **kwargs)
create_retention_policy (*args, **kwargs)
alter_retention_policy (*args, **kwargs)
```

`drop_retention_policy (*args, **kwargs)`

`drop (*args, **kwargs)`

### asyncflux.user – Tools for InfluxDB users

`class asyncflux.user.User (client, name, admin=None)`

`ADMIN = False`

`client`

`name`

`admin`

`change_password (*args, **kwargs)`

`grant_privilege_on (*args, **kwargs)`

`revoke_privilege_on (*args, **kwargs)`

`grant_admin_privileges (*args, **kwargs)`

`revoke_admin_privileges (*args, **kwargs)`

`drop (*args, **kwargs)`

### asyncflux.retentionpolicy – Tools for databases' retention policies

`class asyncflux.retentionpolicy.RetentionPolicy (database, name, duration, replication, default=None)`

`DEFAULT = False`

`database`

`name`

`duration`

`replication`

`default`

`alter (*args, **kwargs)`

`drop (*args, **kwargs)`

### asyncflux.testing – Unit testing support for asynchronous code

#### asyncflux.util – General-purpose utilities

`asyncflux.util.asyncflux_coroutine (f)`

A coroutine that accepts an optional callback.

Given a callback, the function returns None, and the callback is run with (result, error). Without a callback the function returns a Future.

`asyncflux.util.sanitize_url (url)`

`asyncflux.util.batches (source, batch_size)`

## 4.2 Release Notes

### 4.2.1 Next Release

*Very soon*

- Initial release.
- Added Sphinx docs and [ReadTheDocs](#) configuration.



**a**

`asyncflux.client`, 9  
`asyncflux.database`, 10  
`asyncflux.retentionpolicy`, 11  
`asyncflux.testing`, 11  
`asyncflux.user`, 11  
`asyncflux.util`, 11



**A**

ADMIN (asyncflux.user.User attribute), 11  
 admin (asyncflux.user.User attribute), 11  
 alter() (asyncflux.retentionpolicy.RetentionPolicy method), 11  
 alter\_retention\_policy() (asyncflux.database.Database method), 10  
 asyncflux.client (module), 9  
 asyncflux.database (module), 10  
 asyncflux.retentionpolicy (module), 11  
 asyncflux.testing (module), 11  
 asyncflux.user (module), 11  
 asyncflux.util (module), 11  
 asyncflux\_coroutine() (in module asyncflux.util), 11  
 AsyncfluxClient (class in asyncflux.client), 9

**B**

base\_url (asyncflux.client.AsyncfluxClient attribute), 9  
 BATCH\_SIZE (asyncflux.database.Database attribute), 10  
 batches() (in module asyncflux.util), 11

**C**

change\_password() (asyncflux.user.User method), 11  
 change\_user\_password() (asyncflux.client.AsyncfluxClient method), 10  
 client (asyncflux.database.Database attribute), 10  
 client (asyncflux.user.User attribute), 11  
 create\_database() (asyncflux.client.AsyncfluxClient method), 10  
 create\_retention\_policy() (asyncflux.database.Database method), 10  
 create\_user() (asyncflux.client.AsyncfluxClient method), 10

**D**

database (asyncflux.retentionpolicy.RetentionPolicy attribute), 11  
 Database (class in asyncflux.database), 10

DEFAULT (asyncflux.retentionpolicy.RetentionPolicy attribute), 11  
 default (asyncflux.retentionpolicy.RetentionPolicy attribute), 11  
 drop() (asyncflux.database.Database method), 11  
 drop() (asyncflux.retentionpolicy.RetentionPolicy method), 11  
 drop() (asyncflux.user.User method), 11  
 drop\_database() (asyncflux.client.AsyncfluxClient method), 10  
 drop\_retention\_policy() (asyncflux.database.Database method), 10  
 drop\_series() (asyncflux.database.Database method), 10  
 drop\_user() (asyncflux.client.AsyncfluxClient method), 10  
 duration (asyncflux.retentionpolicy.RetentionPolicy attribute), 11

**G**

get\_database\_names() (asyncflux.client.AsyncfluxClient method), 10  
 get\_databases() (asyncflux.client.AsyncfluxClient method), 10  
 get\_measurements() (asyncflux.database.Database method), 10  
 get\_retention\_policies() (asyncflux.database.Database method), 10  
 get\_retention\_policy\_names() (asyncflux.database.Database method), 10  
 get\_series() (asyncflux.database.Database method), 10  
 get\_tag\_keys() (asyncflux.database.Database method), 10  
 get\_tag\_values() (asyncflux.database.Database method), 10  
 get\_user\_names() (asyncflux.client.AsyncfluxClient method), 10  
 get\_users() (asyncflux.client.AsyncfluxClient method), 10  
 grant\_admin\_privileges() (asyncflux.client.AsyncfluxClient method), 10

grant\_admin\_privileges() (asyncflux.user.User method), 11

grant\_privilege() (asyncflux.client.AsyncfluxClient method), 10

grant\_privilege\_on() (asyncflux.user.User method), 11

grant\_privilege\_to() (asyncflux.database.Database method), 10

## H

HOST (asyncflux.client.AsyncfluxClient attribute), 9

host (asyncflux.client.AsyncfluxClient attribute), 9

## N

name (asyncflux.database.Database attribute), 10

name (asyncflux.retentionpolicy.RetentionPolicy attribute), 11

name (asyncflux.user.User attribute), 11

## P

PASSWORD (asyncflux.client.AsyncfluxClient attribute), 9

password (asyncflux.client.AsyncfluxClient attribute), 9

ping() (asyncflux.client.AsyncfluxClient method), 10

PORT (asyncflux.client.AsyncfluxClient attribute), 9

port (asyncflux.client.AsyncfluxClient attribute), 9

## Q

query() (asyncflux.client.AsyncfluxClient method), 10

query() (asyncflux.database.Database method), 10

## R

replication (asyncflux.retentionpolicy.RetentionPolicy attribute), 11

request() (asyncflux.client.AsyncfluxClient method), 9

RetentionPolicy (class in asyncflux.retentionpolicy), 11

revoke\_admin\_privileges() (asyncflux.client.AsyncfluxClient method), 10

revoke\_admin\_privileges() (asyncflux.user.User method), 11

revoke\_privilege() (asyncflux.client.AsyncfluxClient method), 10

revoke\_privilege\_from() (asyncflux.database.Database method), 10

revoke\_privilege\_on() (asyncflux.user.User method), 11

## S

sanitize\_url() (in module asyncflux.util), 11

## U

User (class in asyncflux.user), 11

USERNAME (asyncflux.client.AsyncfluxClient attribute), 9

username (asyncflux.client.AsyncfluxClient attribute), 9

## V

version (in module asyncflux), 9

## W

write() (asyncflux.database.Database method), 10

write\_points() (asyncflux.database.Database method), 10