server

index
/Users/aminaiqbal/GitHub/FogPrototype/server.py

Modules

zmq

Functions

process_sensor_data(received_data)

```
Process received sensor data from client.
:param received_data: received sensor data
:return: processed sensor data
```

Data

context = <zmq.Context(1 socket)>
data_queue = deque([])
failed_data_queue = deque([])
socket = <zmq.Socket(zmq.REP)>

client

index
/Users/aminaiqbal/GitHub/FogPrototype/client.py

Modules

config time zmq

Classes

builtins.object

Client

class **Client**(builtins.object)

```
A class to represent a local component.
```

Methods defined here:

__init__(self)

```
Construct all necessary attributes for the Client class.
```

init_sensors(self)

```
Initialise the sensors by connecting a socket of the client to the configured ports of the sensors
```

process_server_response(self, server_response=None)

```
Process the server's response
Args:
    server_response: response that client received from sever

Returns: server_response, if True
```

receive_data_from_sens(self)

```
Receive data from sensors and store it in a queue
```

receive_data_from_server(self)

```
Receive data from the server and store it in a queue, needs to be called in a timed thread to make sure server
is responding
Returns: True, if data was received
```

send_data_to_server(self)

```
Send data to the server starting with the data that failed to send to the server
if the server is not available, store the data in an extra queue for priority processing when the server is
available again

Returns: True, if data was sent
```

start(self)

```
Start the client processes for receiving and sending data asynchronously
```

Data descriptors defined here:

__dict__

```
dictionary for instance variables (if defined)
```

__weakref__

```
list of weak references to the object (if defined)
```

Functions

Lock = allocate_lock(...)

```
allocate_lock() -> lock object
(allocate() is an obsolete synonym)

Create a new lock object. See help(type(threading.Lock())) for
information about locks.
```

virtual_sensor

index
/Users/aminaiqbal/GitHub/FogPrototype/virtual_sensor.py

Modules

config
random
sys
time
zmq

Classes

builtins.object

VirtualSensor

class **VirtualSensor**(builtins.object)

```
VirtualSensor(port, name, coordinates)

A class to represent a virtual sensor.
```

Methods defined here:

__init__(self, port, name, coordinates)

```
Construct all necessary attributes to set up the zmq socket.
:param port: port to bind to socket
:param name: name of sensor
:param coordinates: coordinates of sensor
:param socket_type: zmq socket type
```

get_coordinates(self)

```
Get the coordinates of the virtual sensor.
:return: coordinates of virtual sensor
```

get_name(self)

```
Get the name of the virtual sensor.
:return: name of virtual sensor
```

get_port(self)

```
Get the port of the virtual sensor.
:return: port of virtual sensor
```

start_sensor(self, rand_interval, send_interval)

```
Start the virtual sensor.
:param rand_interval: random values for the virtual sensor
:param send_interval: time interval between sending sensor data
```

Data descriptors defined here:

__dict__

```
dictionary for instance variables (if defined)
```

__weakref__

```
list of weak references to the object (if defined)
```