```
#######################################################
#
# Yuktix REST API document
#
# version 1.0
# @author rajeev jha
#
# #####################################################


# information needed to write the client

To write a client for Yuktix REST API, you will need

A) API endpoint - The server domain/IP and port, application path and method name
e.g. http://api1.yuktix.com:8080/sensordb/v1/echo
is the echo method of sensordb application version v1, deployed on api1.yuktix.com, port 8080

B) The  HTTP Method(verb) to use. POST | GET | PUT etc.
+ The content-type for method arguments (application/json, text/plain etc. A wrong content-type usually
results in HTTP error 415.

C) Right headers
we need to populate the public API key inside Authorization header.
you may need to set more headers depending on the method.

The Authorization header should be like
Authorization: Signature=3884191b-3951-4576-8544-f34a1e5615e9

where Signature  field is your public API key available from your account.

D) Return codes
The API will return HTTP code as
$response = API_call
$code = $response["code"]

code 200 is for success.
on error - you get non-200 code
please see a good HTTP status code document for HTTP error codes.
we try to follow the HTTP codes as closely as possible.

200 - processed Ok.
401 - error with authorization
400 - bad arguments
500 - unknown internal server error


# ################################
# sample curL session
# ################################

* Hostname was NOT found in DNS cache
*   Trying 127.0.0.1...
* Connected to localhost (127.0.0.1) port 9090 (#0)
> POST /sensordb/v1/module/devices HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:22.0) Gecko/20130405 Firefox/22.0
Host: localhost:9090
Accept: */*
Authorization: Signature=b18b5fffefa240417c62c03b6a273cd9
Content-Type: application/json; charset=UTF-8
Content-Length: 24

* upload completely sent off: 24 out of 24 bytes
< HTTP/1.1 200 OK
< Access-Control-Allow-Origin: *
< Access-Control-Allow-Headers: origin, content-type, accept, authorization, x-requested-with
< Access-Control-Allow-Credentials: true
< Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS, HEAD
< Access-Control-Max-Age: 1728000
< Content-Type: application/json
< Date: Sun, 07 Jun 2015 07:54:01 GMT
< Content-Length: 4330
<

#########################
# Yuktix Public API
#########################

To use yuktix public API, you would need an API key. To Register for API key
- open www.yuktix.com/app/login.php
- click on Sign in using Google+
```

```
- create an account using Google OAuth
- After sign in, click on My Account in Toolbar and note down your API key


1) module devices
To access all devices in a public module, e.g. AWS

endpoint: http://api1.yuktix.com:8080/sensordb/v1

POST /module/devices
Headers

Authorization: Signature=<your-public-api-key>
Content-Type: application/json
Body (raw, application/json)
params: {"map":  {"name" : "AWS"}}

The public API key should be supplied as Authorization header

A sample curL session

POST /sensordb/v1/module/devices HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:22.0) Gecko/20130405 Firefox/22.0
Host: api1.yuktix.com:8080
Accept: */*
Authorization: Signature=3884191b-3951-4576-8544-f34a1e5615e9
Content-Type: application/json; charset=UTF-8

2) Module device detail

POST /module/device

Authorization: Signature=<your-public-api-key>
Content-Type: application/json
Body (raw, application/json)
params: {"map":  {"name" : "AWS", "serialNumber" : "test001"}}

3) Module device archive


POST /module/device/archive

Authorization: Signature=<your-public-api-key>
Content-Type: application/json
Body (raw, application/json)

{"map":  {"module" : "AWS", "serialNumber" : "GSN002"}}
{"map":  {"module" : "AWS", "serialNumber" : "GSN002", "interval" : "900"}}
{"map":  {"module" : "AWS", "serialNumber" : "GSN002", "interval" : "900", "end" : "1435472869000"}}
{"map":  {"module" : "AWS", "serialNumber" : "GSN002", "start" : "1435472869000", "end" :
"1435472869000"}}


The API will fetch all datapoints between start and end timestamps.
start and end parameters are unix timestamps in millis (13 digits)
The interval is in seconds (900 means 15 minutes)

Rules to determine start and end timestamps
-----------------------------------------------
a) if end is not specified it defaults to now()
b) if start is not specified it defaults to end - interval
c) if interval is not specified it defaults to 86400
d) interval is ignored for explicit start and end

The results are sorted on unix_timestamp in ascedning order.  The API will return maximum 1000 datapoints.
The responsibility of paginating over the range lies with the client.
suppose there are more than 1000 datapoints between  your start and end timestamps.
clients should track max(unix_timestamp) returned from API and issue fresh call
with new start and end timestamps.

4) Module latest device archive

POST /module/device/archive/latest
Authorization: Signature=<your-public-api-key>
Content-Type: application/json
Body (raw, application/json)

{"map" : {"module": "AWS" , "serialNumber" : "test001", "limit" : "1" }}
{"map" : {"module": "AWS" , "serialNumber" : "test001", "interval" : "900" }}

This method complements (3) and is used when we are interested in latest points to have arrived in an
```

```
This method complements (3) and is used when we are interested in latest points to have arrived in an
interval.
return points between start and end timestamp.

start and end calculations
---------------------------
interval is in seconds
end timestamp is now()
start timestamp is  now() - interval
interval defaults to 900
if interval cannot be parsed to a number - we use 900


limit is number of rows to return. max. number of returned rows is 1000.
if interval is not supplied then API will return last <limit> points. (no matter when they arrived)
The results are sorted on unix_timestamp in Descending order.

5) Module device computations
POST /module/device/computation

Authorization: Signature=<your-public-api-key>
Content-Type: application/json
Body (raw, application/json)

{"map" : {
    "channel": "RH",
    "frequency": "__DAILY__",
    "module": "AWS",
    "package" : "AWS",
    "serialNumber": "gkvk001",
    "start": "1420945200000",
    "end": "1421031600000"
    }
}

allowed frequencies are:

__DAILY__
__HOUR__
__MINUTE__

start : start time in unix_timestamp millis (13 digit)
end : end time in unix_timestamp millis (13 digit)
module : the public module for grouping together the devices
package: computation package to generate reports on module devices


6) Module device timeseries data

POST /module/tsdb
Authorization: Signature=<your-public-api-key>
Content-Type: application/json
Body (raw, application/json)

{ "channel" : "RH", "serialNumber" : "GSN002" }
return last 30 time series points for this serialNumber + channel.


{ "channel" : "RH", "serialNumber" : "GSN002", "duration" : "7day" }
duration is supplied in human time
30second
15minute
4hour
7day
2week
1month

start and end timestamps
-------------------------
@imp: start and end unix timestamps are in seconds (10 digit)
@imp: duration is supplied in human time units

if duration is supplied then end is supplied value or defaults to now()
start = end - duration
explicit start and end timestamps will be used when supplied
{ "channel" : "RH", "serialNumber" : "GSN002", "end" : "1435476594", "duration" : "7day" }
```

7) API /user/auth/google

Content-Type: application/json
Body (raw, application/json)

```
{
"id" : "1",
"given_name" : "rajeev",
"family_name" : "jha",
"email" : "jha.rajeev@gmail.com",
"link" : "http://www.google.com/1",
"picture": "http://www.google.com/pic/xyz.png"
}
```

RESPONSE

```
{
  "code" : 200,
  "result" : {
    "loginId" : "11",
    "sessionKey" : "683900c3-c7d4-4fc0-830e-595a617624f9",
    "firstName" : "rajeev",
    "lastName" : "jha",
    "handle" : "1",
    "email" : "jha.rajeev2@gmail.com",
    "module" : "__NULL__",
    "accountName" : "__google_1",
    "publicAPIKey" : "85cbd4e0-c50c-42ba-b256-776ad290a0fb"
  }
}
```

8) API /module/snapshot
Content-Type: application/json
Body (raw, application/json)

```
{"module" : "AWS" }
```

RESPONSE

```
Array
(
    [code] => 200
    [response] => {
  "code" : 200,
  "result" : [ {
    "id" : "41",
    "T" : "31.5",
    "RH" : "24.2",
    "address" : "HSR Layout, Bangalore",
    "tsUnix" : "1437578705883",
    "P" : "97577.0",
    "serialNumber" : "GSN002",
    "LWS" : "153.0",
    "longitude" : "76.6333",
    "latitude" : "12.283333",
    "Rain" : "4"
  }, {
    "id" : "42",
    "T" : "33.6",
    "RH" : "27.5",
    "address" : "Kanpur",
    "tsUnix" : "1437587915717",
    "P" : "99115.0",
    "serialNumber" : "GSN007",
    "LWS" : "147.0",
    "longitude" : "80.3334",
    "latitude" : "26.4607",
    "Rain" : "3"
  } ]
}
)
```