Welcome one and all to this Wiki about the API

Now I know what you are thinking: I don’t need this wiki on how to use this API!

Well good luck then….

Na just kidding welcome to this tutorial.

I have built this tutorial in the following manner:

1.Where do I start?

2.How is the API set up?

3.what can I ask the API?

3.1 Login.

3.2 Logout

3.3 Update configuration id of the Device.

3.4 API Call for NodeMCU: Get Sensor/Actuator ID of the second device (partner in crime)

(This is where the fun begins…)

3.5 API Call for NodeMCU: Insert sensor data or Get sensor data depending on the device function

3.6 Set sensor device on the website

3.7 Set actuator device on the website

(Last but not least:)

3.8 Update the threshold for the actuator.

O and remember, if for any reason you still have questions about this don’t hesitate to send me an Email on: [Michael.van.de.Reep@hva.nl](mailto:Michael.van.de.Reep@hva.nl)

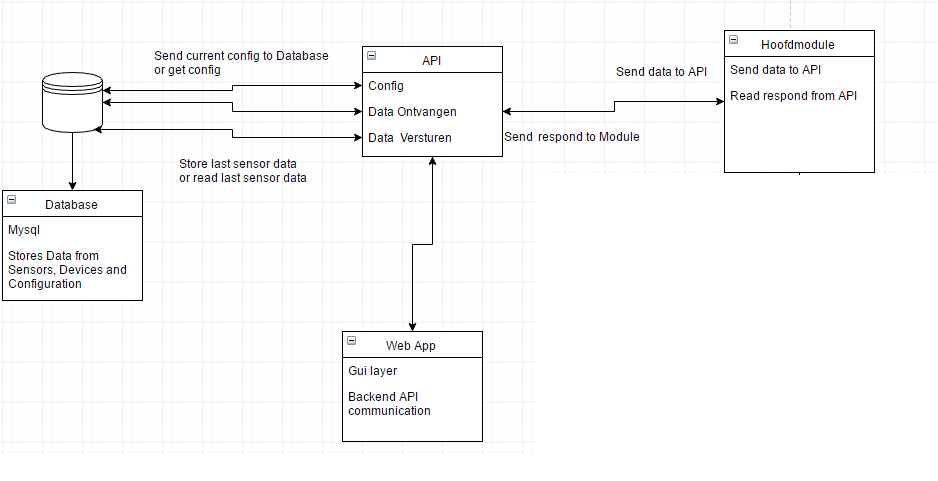
one more thing some parts of this API use Cookies at the client side.

asdasd

1.Where do I start?

So before we begin you need to keep 3 things in mind.

First. the API is built for the website and the NodeMCU cominucation.



Second. for the database layout you can look at the Database dump file:

Database\DataDump\MIOT\_Database\_Dump.zip

Or have a Look at the ERD:

Database\ERD\aqiqinl\_MIOT\_project.mwb

third. For the dev of this project we had 2 dev. places.

1. just a local apache and Mysql Database.
2. iotworkshop.aqiqi.nl witch by the time you are reading this is probably taken down.

2.How is the API set up?

first of all you can call the API file by using the url like 127.0.0.1 (Local) .api.php?[varible1]=[value1]&[Varible2]=[Value2]

When you open the API file you see that first we pick up 2 files called: config.php and database.php. I have done this for security reasons so no one can see the databse and config setting directly in the API file.

second the API file is set up in if statmends.

this is so that if you call the API file you will be automatically transferred to the correct section.

for instance if you sent a rest GET call with the variables Device 1 and Device 2 you will run the login code of the api.

The down side of this is that is you send a variable you also use in a second part you will execute that as wel.

for instance if you have a function that requires variable 1 and 2 and a function with 1,2 AND 3 it will execute both if you have variable 1 and 2 in the url

a way to prevent this from happening is to use the !isset function.

so in this instance if we send the same variables only the first function will be executed.

one last thing all variables and functions are very sensitive to upper and lowercase

3.what can I ask the API?

Let’s just say that the API is as smart as you make it.

If for instance you want it to make a sandwich you need to teach it to do so. (I would be impressed if you did)

3.1 Login.

so the first function is to login with the sensor device and the actuator device.

the sensor device here would be Test1

the actuator device here would be Test2

the url would be: .api.php? Device1=Test1&Device2=Test2

the api will now do the following checks and query’s:

set all values to uppercase.[19,20]

check if device 1 and 2 are in the database.[32,54]

if both devices are in the database we can continue[56.193]:

(welcome to logic hell…)

what happens here is that the api gets all possible combinations of device 1 and device 2 in the device link table.

the device link is used in another function but we need to make sure that if they are in there we can continue. if device 1 is in there it’s already linked to a actuator or sensor device, same case for device 2 so we check for that as well.

the logic is as followed:

1.if there is a device 1 and 2 in the device link that matches device 1 and 2 in that order then go to the dashboard and set the cookies for sensor and actuator device if they are already set in the database (for instance if someone else is logged in and is using the same sensor and actuator. this can be a partner of a 2 pair).

1.2. if there is a device 1 and 2 in the device link that matches device 1 and 2 in that order and the sensor and actuator cookies are set go to dashboard. you don’t need to refresh them.

1.3 if there is a device 1 and 2 in the device link that matches device 1 and 2 but it’s not the same device you entered it will redirect back to login with a message saying device 1 or 2 is already set in the database.

2. if there is no result from the query’s about if the device is in the device link table it means it’s not in there or there is but its linked to another device already handle it with the correct error messages.

3. if you get nothing from all the above steps that means the devices are not in any order in the database so create the device link, set the cookies for device 1 and 2 and redirect to dashboard.

3.2 Logout

What if you want to logout?

No problem we can do that its simpel we let the website post a hidden value called logout (with the value 1) to the API.

the url would be: .api.php?Logout=1

the API will check to see if you have cookies like sensor\_id and Actuator\_ID[195,237]

wat we need to do is set the actuator en sensor back to active 0 to make them inactive again.

(SQL, get set… fire!)

do the same for the other values you want to change at logout.

o one more thing if there are any more cookies we can simply delete them by the small loop at [229,233]

afther that simply redirect to the homepage.

3.3 Update configuration id of a Device.

if we want to set a device to a specific configuration we need to send a config ID to the device on startup.

We can do that by selecting a actuator or sensor on the website.

it will send the folloeing varibles:

DeviceId in this case: test1

a configuratien ID: 13

usage api.php?deviceid=test1&configuratie=13

it will update the config in the database where the device id matches the device id given in the url.[238,254]

3.4 API Call for NodeMCU: Get Sensor/Actuator ID of the second device (partner in crime)

if we want to tell the actuator or sensor device what sensor or actuator id is joint to it we can simply send the following varibles:

DEviceId: test1

deviceFunctie: [sensor/actuator]

usage: api.php?deviceid=test1&deviceFunctie=[sensor/actuator]

if the defive function is sensor it will send a sql query to the database to search for the sensor\_ID

if the device function is actuator it will send a sql query to the database to search for the actuator\_ID

[255,295]

Remember what I told you about the downside of using variables to fire of code?

*“for instance if you have a function that requires variable 1 and 2 and a function with 1,2 AND 3 it will execute both if you have variable 1 and 2 in the url”*

the next function that I will describe (3.5) also uses deviceId and DeviceFunction…..

you can solve this by adding a (!isset($\_GET['value']))) to the if statement. this way you will isolate the if statement from the next one since this function will not be called with a value variable.

a more lazy way to solve this is to put this if statement after the section of 3.5.

3.5 API Call for NodeMCU: Insert sensor data or Get sensor data depending on the device function

now this is where it gets interesting.

this function is a complicated one.

we use this one to either update the sensor log with a value from the senor or get the value for the actuator device.

usage:

deviceId=TEST2

deviceFunctie=[actuator/sensor]

sensorId=001

value=1 if deviceFuntion is sensor de value is a must but if its an actuator it will not be used but you’ll need to specify it non the less.

api.php?deviceId=TEST2&deviceFunctie=[actuator/sensor]&sensorId=001&value=1

(Once more in to the breach…)

make sure the device\_ID is in the database.

you’ll notice that I have put the execute of the sql query in the if statement. this will make sure it can execute the query. if not it will give an error message and go to the else.

well then if the device id checks out we can get going.

1. if the deviceFunctie = sensor[326,386]

api.php?deviceId=TEST1&deviceFunctie=sensor&sensorId=001&value=100

it will check to see if the sensor and the Sensor module are linked in the database.

now here is the fun part. we wanted to make sure there are only ~20 entry’s in the database per sensor.

what the API will do if ask the database how many there are in there and if the row count is less than 20 it will insert the sensor data. if it is more than 20 it will search the database for the first/ oldest entry and it will update that one.

you can override this by manualy adding more sensor log data or raising the 20 entrys limit by changing the value at rule 341

2. if the deviceFunctie = actuator[386,467]

api.php?deviceId=TEST2&deviceFunctie=actuator&sensorId=001&value=1

here we enter the senor id from which we need the sensor value from and we add value=1 just for funciezz (that or the function wont start :) ).

now here we will need the device link table for. (remember the logic bom at 3.1?)

next get the sensor last sensor value from the database in a descending orientation and limit the output to 1 entry.

store the value for later… first we need to get the configuration id form the database and the threshold

pfff… we are done here… o echo the results in the following manner: configuration, sensor value, threshold.