

# SWIMO – API LOCALE V2

22/08/2018

Règles :

Param :  
indique un paramètre obligatoire GET

Option :  
indique un paramètre optionnel GET

Chemin  
[IP SWIMO/cgi-bin](#)

Methode :

Get :  
param et option

Linux approach :

GET  
api =<apikey>  
serial =<n°de série>

## Table des matières

1.	GET ALL.....	- 3 -
2.	UPDATE SYST .....	- 6 -
3.	GET DEVICES .....	- 7 -
4.	UPDATE DEVICE .....	- 11 -
5.	GET ANALYSE .....	- 12 -
6.	UPDATE SENSOR.....	- 14 -
7.	ADD SENSOR.....	- 15 -
8.	DEL SENSOR .....	- 15 -
9.	ADD DEVICE.....	- 16 -
10.	DEL DEVICE .....	- 16 -
11.	ADD PLAGE.....	- 17 -
12.	DEL PLAGE .....	- 17 -
13.	CALIBRATION LINUX .....	- 19 -
14.	SETUP WIFI LINUX .....	- 21 -

# 1. GET ALL

**<ipMachine>/cgi-bin/getAll**

ip : ip local ou ip public

Get All renvoie globalement, le compte, les capteurs et équipements connectés.

## Param :

serial= <n° de série>

api= <apikey>

"user":[

{

"swimo":"v.0.11.22", VERSION DU CONTROLEUR

"version":"v.0.1.0", VERSION DE LA BASE USER

"ipMachine":"192.168.0.31",

"lang":"FR",

"volume":"40", // volume du bassin

"typeBassin":"piscine", piscine ou spa / 0 ou 1

"typeAbris": "\_INT", // \_INT ou \_EXT

"unit": "metric", metric ou imperial

```
"accueil_analyse":[
{
  "sensorType": "1", ///voir table sensor type
  "nmSensor": "4", // nmSensor correspond au n° sur circuit
  "currentDate": "2018-08-08 13:17:17",
  "etatSensor": "1", /// si =1 capteur en erreur
  "nameSensor": "temperature", //champs modifiable
  "minSensor": "26.69", //valeur minimum enregistrée depuis Minuit
  "maxSensor": "28.39", //valeur max ..
  "alarmMin": "3",      alarme de déclenchement notification
  "alarmMax": "10",    alarme de déclenchement notification
  "liveSensor": [ 8 derniers relevés espacés de 1 minute
    • [
      "26.73",
      "26.73",
      "26.73",
      "26.71",
      "26.71",
      "26.69",
      "26.78",
      "26.75"
    ]
  ],

```

"unitConsigne": "°C", // unitConsigne

"offsetSensor": "0", //ajustement du capteur + ou -

"typeCalibration": "0", //0 à 3 points de calibration

"dateCalib": "0000-00-00 00:00:00" //date de la dernière calibration



```
"accueil_appareil":[
{
  "nameAction": "pompe Filtration", //champs modifiable

```

```

"idActionType": "1",

"nmAction": "1", // n° de port sur circuit

"textDevice": "AUTO", // renvoie le code du programme en cours

"consigne": "20", // renvoi -1 ou ensemble vide si non utilisé

"unitConsigne": "mg/l", // renvoi -1 ou ensemble vide si non utilisé

"soldeNiveau": "-1", // renvoi -1 ou ensemble vide si non utilisé

"consoJour": "14:53:24.000", // TEMPS DE FONCTIONNEMENT FORMATÉ H :M :S

"isOff": "1", //etat de l'équipement : 0 =ETEIND,1=ALLUMÉ

"securite": "1", // 0 1 (accepte plages) 2 (bouton poussoir, non visible sur serveur) 3 (non visible sur serveur)

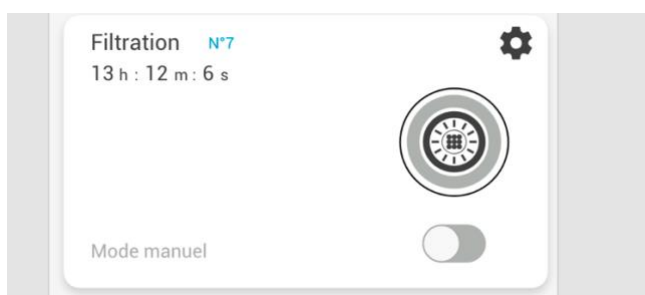
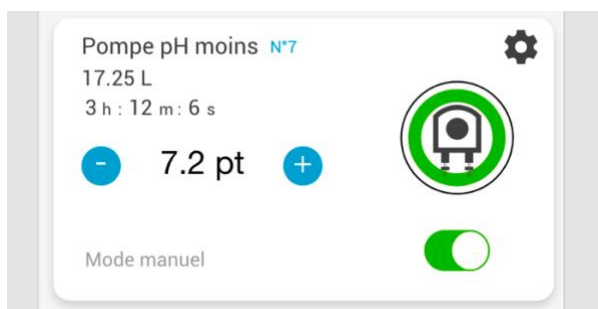
"selectedIndex": "2", //index de stateAct ou code mode (0-1-2)

"stateAct": ["1V2V3"],

"codeMode": ["ONVOFFVAUTO"],

"currentDate": "2018-08-08 13:18:48"

```



## 2. UPDATE SYST

**<ipMachine>/cgi-bin/updateSyst?**

### Param :

serial= <n° de série>

api= <apikey>

### Option :

unit metric, imperial (°c, °f)

lang fr, en, es

type 0=piscine, 1=spa

vol volume en m3 (step=0.1)

abris \_INT / \_EXT

### answer :

true

possibilité de valider chaque param indépendamment les uns des autres,

### 3. GET DEVICES

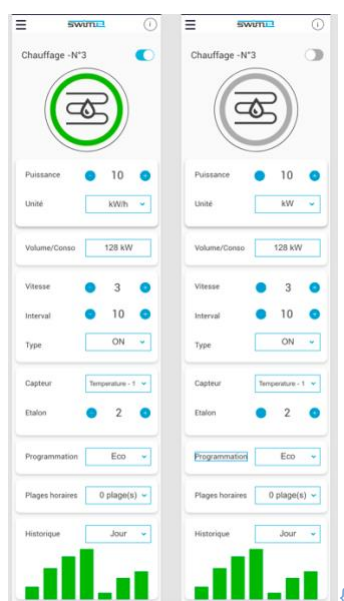
`<ipMachine>/cgi-bin/getDevices?nmAction=5`

Param :

serial= <n° de série>

api= <apikey>

nmAction



"devices":[

{

"nameAction":"pompe Chlore",

"idActionType":"4",

"number":"2", renvoie number et nmAction v1 et V2

"nmAction":"2",

"textDevice":"ECO",//renvoie le codeProg

"isOff":"1",

"securite":"1",

"selectedIndex":"2",

```

"stateAct":["1/2/3"],

"codeMode":["ON/OFF/AUTO"],

"power":"20.000000", puissance/debit : champs val

"unitPower":""," champs unitPower default

"consigne":"1.400000", con

"unitConsigne":"mg/l",

(dev en cours)
"niveau":"100", s i#-1 ou « « afficher L (litres) champs bid

"vitesse":"0", vit

"typeVitesse":"ON", type

"interval":"1", interval

"etalon":"0", etalon de déclenchement, 0.3 pour ph moins par exemple

"nmSensor":"13", sensor (depuis liste sensor user)

```

## PROGRAMME

```

"selectedSeq":"0", codeSeq

"statProg":["1/2/3"],

"codeProg":["ECO/MAX/CHOC "],

"currentDate":"2018-08-09 18:54:31"

```

```

}

```

```

],

```

```

"plages":[

  {

    "action":"5", V1

    "nmAction":"5", V2

    "start":"22:00",

    "end":"22:15",

    "days":"1234567"

  }
]

```

80 Itérations possibles (pagination ou slide)

```

"consos":[

```



```
{
  "temps": "206",
  "currentDate": "2018-08-09 18:54:31"
},
{
  "temps": "109",
  "currentDate": "2018-08-09 17:54:17"
},
{
  "temps": "301",
  "currentDate": "2018-08-09 17:49:15"
},
{
  "temps": "301",
  "currentDate": "2018-08-09 17:44:14"
},
{
  "temps": "301",
  "currentDate": "2018-08-09 17:39:13"
},
{
  "temps": "302",
  "currentDate": "2018-08-09 17:34:12"
},
{
  "temps": "301",
  "currentDate": "2018-08-09 17:29:11"
},
{
```

```
"temps": "301",  
"currentDate": "2018-08-09 17:24:10"  
},  
{  
"temps": "301",  
"currentDate": "2018-08-09 17:19:09"
```

## 4. UPDATE DEVICE

**cgi-bin/updateDevice?nmAction=2**

### Param :

serial= <n° de série>

api= <apikey>

nmAction ou number

### Option :

val	valeur debit ou puissance
con	point de consigne
seq	sequence V1
bid	niveau bidon
unitPower	unité puissance (liste unité)
vit	code vitesse (entier )
interval	interval (entier)
type	type devitesse (ON ou WEEK)
sensor	liste sensor du user
etalon	etalon point de consigne
codeSeq	le selected Index du programme
index	le selected index du statut on/off/auto
name	le nom du device

### Answer :

true

## 5. GET ANALYSE

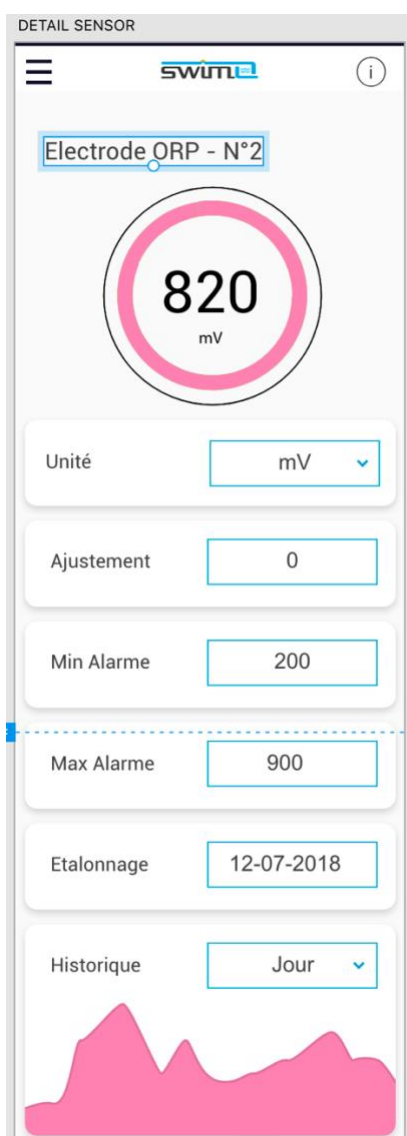
`cgi-bin/getAnalyse?nmSensor=2`

Param :

serial= <n° de série>

api= <apikey>

nmSensor ou number



"analyse":{

```

{
  "sensorType": "5",
  "number": "2", number v1 ou nmSensor v2
  "nmSensor": "2",
  "currentDate": "2018-08-09 19:14:32",
  "etatSensor": "1",
  "nameSensor": "Electrode ORP", champs modifiable // nameSens
  "alarmMin": "100", Alarme min < une notification push ou email (via serveur)
  "alarmMax": "1000", Alarme max > une notification push ou email (via serveur)
  "liveSensor": "809",
  "offsetSensor": "0", setsens
  "unitConsigne": "mV", unitConsigne
  "typeCalibration": "2",
  "dateCalib": "0000-00-00 00:00:00"
}

```

#### DEFAULT HISTORIQUE 24H -144 itérations

```

"historique": [
  {
    "releve": "809",
    "insertion": "2018-08-09 19:14:32"
  },
  {
    "releve": "813",
    "insertion": "2018-08-09 19:09:31"
  },
  {
    "releve": "800",
    "insertion": "2018-08-09 19:04:30"
  }
]

```

## 6. UPDATE SENSOR

`cgi-bin/updateSensor?nmSensor=2`

### Param :

serial= <n° de série>

api= <apikey>

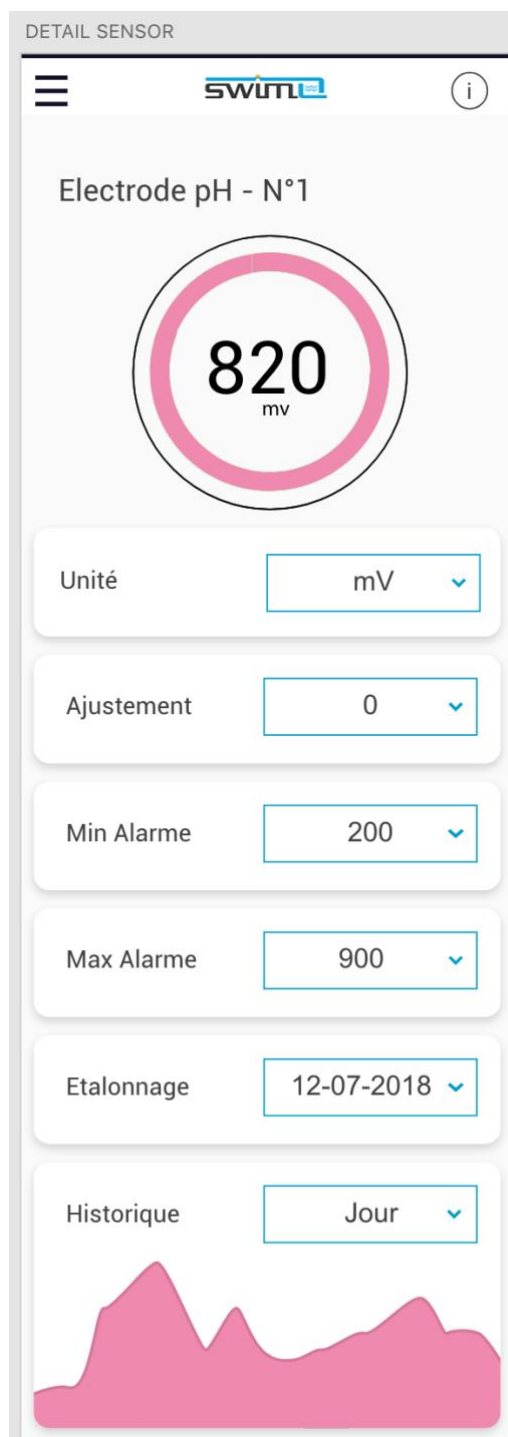
nmSensor ou number

### Option :

Setsens	offsetSensor
Min	adjustment
Max	define minAlarm
nameSens	define MaxAlarm
unitConsigne	rename the sensor
	unité de la liste

answer :

true



## 7. ADD SENSOR

cgi-bin/sensorAdd?sensor=2&nmSensor=2

### Param :

serial= <n° de série>

api= <apikey>

nmSensor ou number (n°signal)

sensor (sensorType)

@ Ajoute un nouveau capteur

## 8. DEL SENSOR

cgi-bin/sensorDel?nmSensor=2

### Param :

serial= <n° de série>

api= <apikey>

nmSensor ou number (n°signal)

@ supprime un capteur

## 9. ADD DEVICE

cgi-bin/relayAdd?action=2&nmAction=2

### Param :

serial= <n° de série>

api= <apikey>

nmAction ou number (n° relay)

action (action Type)

@ Ajoute un nouvel appareil

## 10. DEL DEVICE

cgi-bin/relayDel?nmSensor=2

### Param :

serial= <n° de série>

api= <apikey>

nmAction ou number (n° relay)

@ supprime un appareil



## 11. ADD PLAGE

cgi-bin/plageAdd?nmAction=2&start=08:00&end=12:00&days=123

### Param :

serial= <n° de série>

api= <apikey>

nmAction ou number (n° relay)

start (debut de plage formatée 08:00)

end(fin de plage formatée 16:00)

days (recurrence jour au choix 1234567)

retour :liste des plages

```
"plages":[  
  {  
    "action":"5", V1  
    "nmAction":"5", V2  
    "start":"22:00",  
    "end":"22:15",  
    "days":"1234567"
```

## 12. DEL PLAGE

cgi-bin/plageDel?nmAction=2&start=08:00

### Param :

serial= <n° de série>

api= <apikey>

nmAction ou number (n° relay)

start (debut de plage formatée 08:00)

retour :liste des plages encore actives

```
"plages":[  
  {  
    "action":"5", V1  
    "nmAction":"5", V2  
    "start":"22:00",  
    "end":"22:15",  
    "days":"1234567"
```

## 13. CALIBRATION LINUX

MEMO : "typeCalibration": "2", indique 2 points de calibration

**<ipMachine>/cgi-bin/calibrateStart**

Param :

serial= <n° de série>

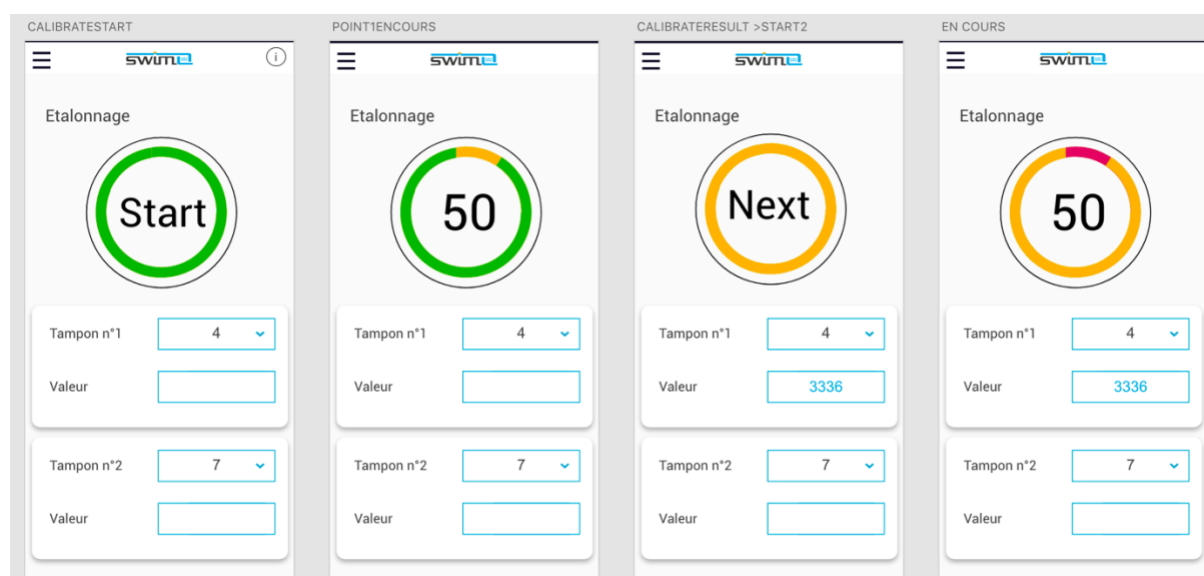
api= <apikey>

number ou nmSensor: n° de signal

point : n° du tampon

value : valeur de la solution tampon

*Démarrer avec bouton Start*



(au bout de 60 secondes, lancer)

**<ipMachine>/cgi-bin/calibrateResult**

## Param :

serial= <n° de série>

api= <apikey>

number ou nmSensor: n° de signal

point : n°de point

## Answer :

"valeur": "3424" ou True

ou

false

**Recommencer procedure selon nbre typeCalibration, en affichant**

*Bouton NEXT*

*Finaliser avec bouton END*

## 14. SETUP WIFI LINUX



<http://< ipMachine >/cgi-bin/getSSID>

renvoie la liste des réseaux wifi brute

## SET WIFI

<http://< ipMachine >/cgi-bin/setSSID?>

### Param :

SSID= <nom du reseau>

PASS= <password>

serial= <n° de série>

api= <apikey>

## RESET WIFI

<http://< ipMachine >/cgi-bin/resetWifi?>

### Param :

serial= <n° de série>

api= <apikey>

