The Blog of Nick

Nick's Projects and Ramblings

Goal Zero Yeti Local WiFi REST API

Update: Goal Zero generously provided me with a Postman file to show all the routes supported by the Yeti. I've added info on that to the bottom.

I have acquired a new Goal Zero Yeti 3000 and was curious how to use its WiFi functionality for evil. The app is acceptable, but I've discovered you can query the local REST API using some glorious wireshark monitor mode.

Basically, hit http://GOAL_ZERO_IP/state. GET returns:

```
{
"thingName": "yetiXXXXXXXX",
"v12PortStatus": 1,
"usbPortStatus": 1,
"acPortStatus": 1,
"backlight": 1,
"app_online": 1,
"wattsIn": 245.3,
"ampsIn": 21.9,
"wattsOut": 161.3,
"ampsOut": 14.4,
"whOut": 3682,
"whStored": 2070,
"volts": 11.2,
"socPercent": 69,
"isCharging": 1,
"timeToEmptyFull": 681,
"temperature": 30,
"wifiStrength": -37,
"timestamp": 317529,
"firmwareVersion": "0.7.2",
"version": 2
}
```

and POST can be used with this JSON:

```
POST /state HTTP/1.1
Host: 10.1.1.1
Content-Type: application/json
User-Agent: YetiApp/1340 CFNetwork/1125.2 Darwin/19.4.0
Connection: keep-alive
Accept: application/json
Accept-Language: en-us
Content-Length: 19
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
```

and the payload to turn off USB:

```
{"usbPortStatus":0}
```

Or to set the 12V port:

```
{"v12PortStatus":0}
```

Or you can set all 3 states at the same time:

```
{"v12PortStatus":0,
"usbPortStatus":0,
"acPortStatus": 1}
```

- In direct connect mode, it creates a WiFi network and the phone queries its HTTP server at /state. GET returns status and POST can be used to change settings.
- In remote mode (called “anywhere connect”), it joins your network and phones home to AWS _but_ it can still be queried over the local HTTP server.

Also some other routes:

```
GET /sysinfo HTTP/1.1
Host: 10.1.1.1
Accept-Encoding: gzip, deflate
Accept: application/json
```

```
User-Agent: YetiApp/1340 CFNetwork/1125.2 Darwin/19.4.0
Accept-Language: en-us
Cache-Control: no-cache
Connection: keep-alive
```

```
{ "name": "yetiXXXXXXXXX", "model": "Yeti 3000", "firmwareVersion": "0.7.2",
"macAddress": "XXXXXXXX", "platform": "esp32" }
```

```
POST /join-direct HTTP/1.1
Host: 10.1.1.1
Content-Type: application/json
Content-Length: 0
Connection: keep-alive
Accept: application/json
User-Agent: YetiApp/1340 CFNetwork/1125.2 Darwin/19.4.0
Accept-Language: en-us
Cache-Control: no-cache
Accept-Encoding: gzip, deflate
```

When queried on my laptop, returned:

```
{ "name": "yetiXXXXXX", "model": "Yeti 3000", "firmwareVersion": "0.7.2",
"macAddress": "XXXXXXXX", "platform": "esp32", "state": {
  "thingName": "yetiXXXXXXXX",
  "v12PortStatus": 1,
  "usbPortStatus": 1,
  "acPortStatus": 1,
  "backlight": 1,
  "app_online": 1,
  "wattsIn": 229.6,
  "ampsIn": 20.5,
  "wattsOut": 142.2,
  "ampsOut": 12.7,
  "whOut": 3743,
  "whStored": 2100,
  "volts": 11.2,
  "socPercent": 70,
  "isCharging": 1,
  "timeToEmptyFull": 552,
  "temperature": 30,
  "wifiStrength": -39,
  "timestamp": 319152,
  "firmwareVersion": "0.7.2",
  "version": 2
} }
```

Other Routes from Goal Zero

Quick summary

- GET /sysinfo – gives system info from the ESP32
- POST /factory-reset with “{}” body does a factory reset
- GET /rpc/Sys.Reboot will trigger a reboot

GET/POST to /loglevel

```
{
  "app": 3,
  "ota": 3,
  "ota_cleanup": 3,
  "mqtt": 3,
  "mqtt_shadow": 3,
  "mqtt_jobs": 3,
  "yeti": 4,
  "wifi": 3,
  "http_server": 3,
  "http_client": 3,
  "main": 3,
  "uart": 3,
  "pic_btldr": 3,
  "chain": 3,
  "sys": 3
}
```

POST to /join to join the wifi

```
{
"wifi": {
"name": "NetworkName",
"pass": "Password"
},
"iot": {
"env": "prod",
"hostname": "a1xyddj5i8k7t5-ats.iot.us-east-1.amazonaws.com",
"endpoint": "https://yeti-prod.goalzeroapp.com/v1/thing"
}
}
```

GET to /wifi to show the available wifi networks the GZ can see

```
{
    "NickPixel": {
        "db": -38
    },
    "networky-saved": {
        "db": -43,
        "saved": true
    },
    "something-else": {
        "db": -67
    }
}
```

POST to /password-set to change the Yeti password

```
{
"new_password": "ABCDEFGH"
}
```

POST to /join-direct to join direct (not clear exactly what this does when in cloud mode)

POST to /start-pair again, not clear exactly what this does

Note: when I POST to /state the yeti began sending its network again but is still joined to the main WiFi network. Not sure if there's a good configuration for local network only without IoT phone home feature.

April 9, 2020 nicholas Uncategorized