IoT: Client Devices

A Libcurl Example - Code

Code!!!

Workstation first

- Test our code
- Easier environment
- Faster test/code cycles
- More stable networking

Why?

- A simple example
- Sends an HTTP GET

./requestor.c

```c
#include <stdio.h>
#include <curl/curl.h>

#define OK        0
#define INIT_ERR  1
#define REQ_ERR   2

#define URL       "http://localhost:8000"

int main(void) {
  CURL      *curl;
  CURLcode  res;

  curl = curl_easy_init();
  if (curl) {
    curl_easy_setopt(curl, CURLOPT_URL, URL);
    curl_easy_setopt(curl, CURLOPT_FOLLOWLOCATION, 1L);
    res = curl_easy_perform(curl);
    if(res != CURLE_OK) {
      return REQ_ERR;
    }
    curl_easy_cleanup(curl);
  } else {
    return INIT_ERR;
  }
  return OK;
}
```

NORMAL master ./requestor.c M unix

```
[cclamb@ubuntu:~ $ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

SimpleHTTPServer

$ python -m SimpleHTTPServer
(run this in another window)

Work — ssh -X 192.168.120.141 — 89×31

```
cclamb@ubuntu:~/Work/iot-client $ make
gcc    -c requestor.c -o requestor.o
gcc    -o test requestor.o -L/usr/lib/x86_64-linux-gnu -lcurl -lpthread
cclamb@ubuntu:~/Work/iot-client $ ls
makefile  printer.c  requestor.c  requestor.o  test
cclamb@ubuntu:~/Work/iot-client $ ./test > test.out
cclamb@ubuntu:~/Work/iot-client $
```

```
cclamb@ubuntu:~/Work/iot-client $ python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
127.0.0.1 - - [12/Jan/2017 16:55:17] "GET / HTTP/1.1" 200 -
```

```
2  0* bash   1h 51m 58s  0.0 0.0 0.1  2017-01-12  16:55  ubuntu
```