

/temp

GET <http://192.168.1.100/temp>

geeft de temperatuur in JSON format weer.

Request:

GET <http://192.168.1.100/temp> HTTP/1.1

Response:

HTTP/1.1 200 OK

Content-Type: application/json

Connection: clode

{ "Weerstation": 1, "Temperatuur": 25 }

Opmerkingen:

- geeft 404 NOT FOUND als er geen meet data is.
- Als het een fout request is geeft die 400 BAD REQUEST

/lux

GET <http://192.168.1.100/lux>

geeft de lichtintensiteit in JSON format weer.

Request:

GET <http://192.168.1.100/lux> HTTP/1.1

Response:

HTTP/1.1 200 OK

Content-Type: application/json

Connection: close

{ "Weerstation": 1, "Lichtintensiteit": 245 }

Opmerkingen:

- geeft 404 NOT FOUND als er geen meet data is.
- Als het een fout request is geeft die 400 BAD REQUEST

/

GET <http://192.168.1.100/>

geeft alle meetwaarden in html weer.

Request:

GET <http://192.168.1.100/> HTTP/1.1

Response:

HTTP/1.1 200 OK

Content-Type: application/json

Connection: close

<!DOCTYPE HTML>

<html>

"Temperatuur: 20 (min: 12, max: 30)"

</br>"

"Lichtintensiteit: 260"

</html>

Opmerkingen:

– geeft 404 NOT FOUND als er geen meet data is.

– Als het een fout request is geeft die 400 BAD REQUEST

/conf

GET <http://192.168.1.100/conf>

geeft alle meetwaarden en configuratiewaarden in JSON format

Request:

GET <http://192.168.1.100/conf> HTTP/1.1

Response:

HTTP/1.1 200 OK

Content-Type: application/json

Connection: close

```
{ "Weerstation": 1, "maximale grenswaarde": 30, "minimale  
grenswaarde": 12, "Temperatuur": 25, "Lichtintensiteit": 424 }
```

/conf

PUT <http://192.168.1.11/conf>

Request:

PUT <http://192.168.1.11/conf> HTTP/1.1

Content-Type: application/json

Content-length: 22

{"max": 35, "min": 10}

Response:

HTTP/1.1 200 OK

Content-Type: application/json

Connection: close

{"max": "<http://192.168.1.100/conf/max/35>", "min": "<http://192.168.1.100/conf/min/10>"}