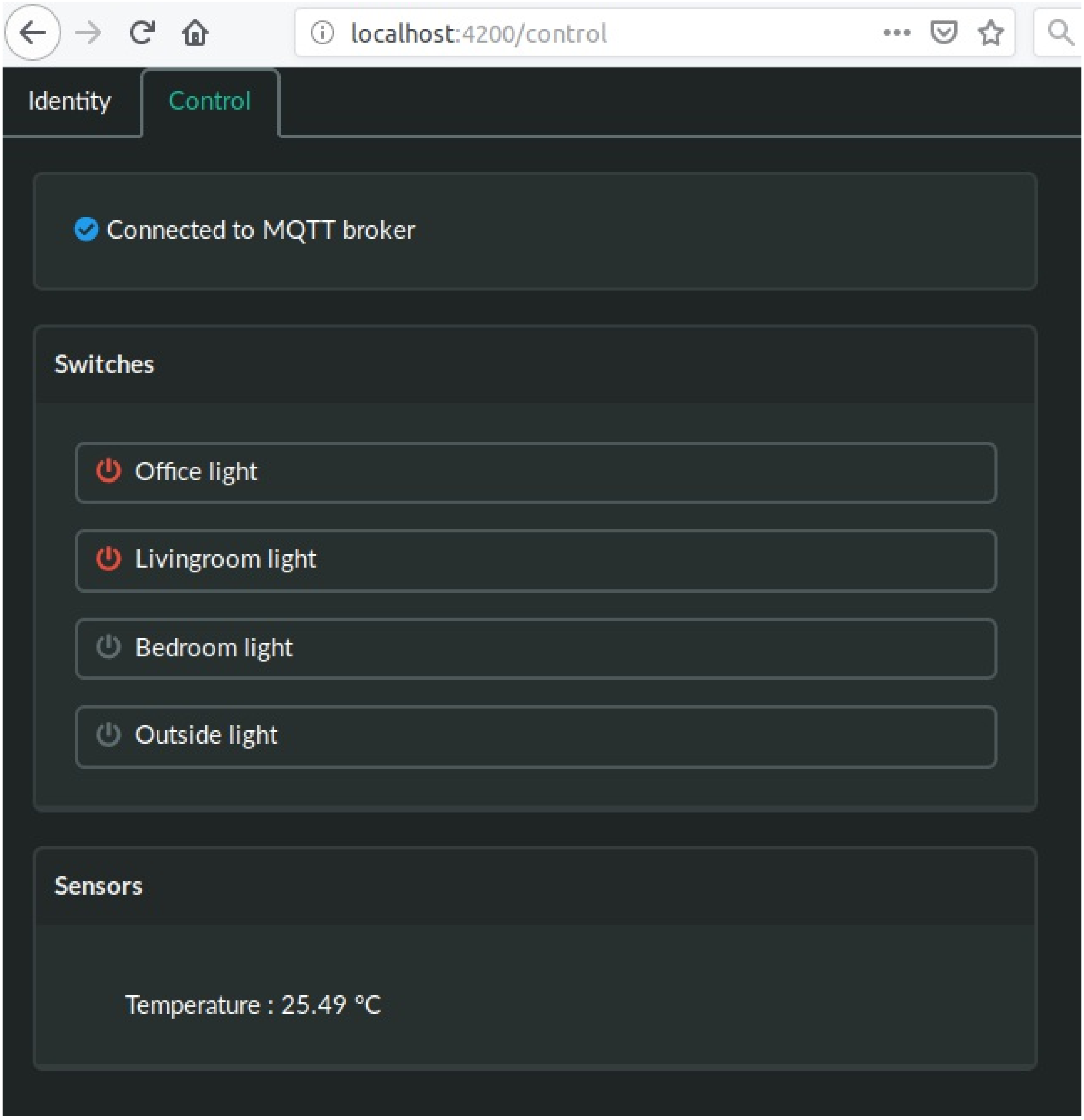
Otto

Otto is a simple web application to control switches and view sensor output using MQTT. Connection settings are configured through the web interface, and stored in the browser's local storage.



✓ Connected to MQTT broker

Temperature : 25.49 °C

Requirements

- MQTT server with websockets enabled
- Some sort of service that controls the switches (see switch topics below)

- Retained topics for device discovery. See home assistant documentation for format of auto discovery messages.

Running Otto

Just navigate to https://otto.zenly.xyz/

All user data is kept in the browser, so there's no need to install the app yourself, but if you *do* want to:

```
docker run --rm -p 8080:80 dougg/otto
```

Device Configuration and Messages

The interface reads discovery messages from the MQTT broker and users these to configure what devices to display and control.

For example, suppose we configure otto to use a "discover prefix" of "homeassistant", and we publish a retained message like this:

- topic: `homeassistant/switch/sprinkler/config`
- payload: `{"name": "Sprinkler system", "state_topic": "homeassistant/switch/sprinkler/state", "command_topic": "homeassistant/switch/sprinkler/set"}`

Clicking on the web interface button to turn the sprinkler on will send a message on topic `homeassistant/switch/sprinkler/set` with a payload of `ON`

Once the sprinkler has been turned on, whatever is controlling it is expected to set `homeassistant/switch/sprinkler/state` to `ON` and set the retain flag on that messag

Likewise for turning our example sprinkler off.

Use of "set" and "state" messages is compatible with what Home Assistant expects.

Development Environment

Run `make up logs` to bring otto up and point a browser to http://localhost:4200/