Building Mozilla Webthings

Owning home automation via HTTP

Material: https://chrpaul.de/fosdem2020
Learn from other adapters
Email Sender
Simple email sending actions. Needs to be configured with email and app password.

by Mozilla IoT (license)
## Using existing protocols

<table>
<thead>
<tr>
<th>Transport</th>
<th>Best practices and data formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP stack</td>
<td>Semantic web</td>
</tr>
<tr>
<td>HTTP</td>
<td>OAuth</td>
</tr>
<tr>
<td>HTTPS</td>
<td>JSON (with JSON Schemas)</td>
</tr>
<tr>
<td>HTTP/2</td>
<td>REST API</td>
</tr>
<tr>
<td>Websocket</td>
<td>Websocket API</td>
</tr>
<tr>
<td></td>
<td>DNS Service Discovery</td>
</tr>
</tbody>
</table>
Web of Things is about interoperability!
What does the Thing do?

GET http://192.168.0.31/things
Authorization: Bearer eyJhbGciOiJFUzI1NiR5... 
Accept: application/json
WebThings manifest

You MUST declare:

● properties,
● actions and
● events
Properties

- MUST have a JSON Schema type ("integer", "number", "string", etc.)
- May have a semantic @type (e.g. "InstantaneousPowerProperty")
- may be readOnly: true
WebThings manifest

You MAY declare: @type (array of strings)

- e.g. Light, ColorControl, DoorSensor, TemperatureSensor
- Source: https://iot.mozilla.org/schemas/
WebThings manifest

- Actions
  - e.g. fade
- Events
  - e.g. overheated
WebThings manifest
{
    "@context": "https://iot.mozilla.org/schemas/",
    "@type": ["Light", "ColorControl"],
    "id": "https://mywebthingsserver.com/things/lamp",
    "Name": "Lamp",
    "description": "A web connected lamp",
    "properties": {
        "color": {
            "@type": "ColorProperty",
            "type": "string",
            "title": "Color",
            "description": "The color of light",
            ...}
Lights in the morning

If the time of day is 08:30 and DateTime is not Weekend, set Bedroom On/Off to true
If the time of day is 08:30 and DateTime is not Weekend, set Bedroom On/Off to true.
Benefits of a gateway

- Automation with rules
- Privacy / access control
- Bridging protocols / interfaces
- Remote access
- Logging
- Configuration hub
Email Sender

Simple email sending actions. Needs to be configured with email and app password.

by Mozilla IoT (license)

Configure  Remove  Enable
CO₂ Monitor
CO₂ Monitor Adapter
by Christian Paul (license)
Six lessons learned
Automate pushing code to your Raspberry Pi

Also restart the gateway service at the same time
Use setCachedValueAndNotify for debugging

Use setCachedValue to avoid disk writes
Access the logs via SSH

The access tokens for logs in the front-end get reset on every restart
A .git folder disables the checksum test

For each production release you’ll need to generate SHA256SUMS
Prepare for separate releases, if you have architecture-dependent code
The common license of addons is the MPL-2.0

It helps contributing to your project because there’s already a lot of similar code under that license
Further questions/ suggestions:
christian@chrpaul.de

Also, I’m looking for an awesome open-source job!
Get Involved

Build a Web Thing
Build your own IoT device which uses the Web Thing API

Create an Adapter
Create an adapter add-on to bridge an existing IoT device to the web

Hack on Project Things
Help us develop our Web of Things implementation
Bonus slides
For answering questions or other curious minds
WebThings

An open platform for monitoring and controlling devices over the web.

Learn More

WebThings Gateway

A software distribution for smart home gateways focused on privacy, security and interoperability

WebThings Framework

A collection of re-usable software components to help developers build their own web things
Mozilla’s Decentralized WoT Approach Improves Privacy, Security, Interoperability

Mozilla gateway is in your home

No data

Your data

https tunnel

just web

Vendor gateways send data to cloud

Your data stored in many IoT vendor data centers

“Your” data

apps galore

Enabling IoT devices to be discoverable on the web ≠ Connecting IoT device data to the cloud
Tips for developing an addon

● Adding a .git folder disables the check of checksums.
● Develop on your computer, push to your Raspberry Pi.
  ○ Add scripts to push the code and restart the gateway to increase your productivity
● Access the logs via SSH
● Use setCachedValueAndNotify for debugging of values.
  ○ Use setCachedValue in a release to avoid disk writes.
● Prepare for separate releases, if you have architecture-dependent code.
● The common license of addons is the MPL-2.0.
Pre-defined things (Mozilla’s schema)

OnOffSwitch, MultiLevelSwitch, BinarySensor, MultiLevelSensor, ColorControl, EnergyMonitor, SmartPlug, Light, MotionSensor, DoorSensor, TemperatureSensor, LeakSensor, PushButton, Camera, VideoCamera, Alarm

Source: https://iot.mozilla.org/schemas/
The issues of IoT

- Varying network protocols
- Varying application protocols
- Varying apps per vendor
- Most apps are only available for Android and iOS
- Most apps are only available on the main app stores
- The vendor’s cloud stores all events
- Price is high; Resale value is low
Turn the Kitchen Light on

OK, turning the kitchen light on.

Turn the Bedroom green.

OK, turning the bedroom green.

How can I help?
Thanks

Christian Paul

- Open source enthusiast
- Web developer
- Care about data security and accessibility
- Not a Mozilla employee