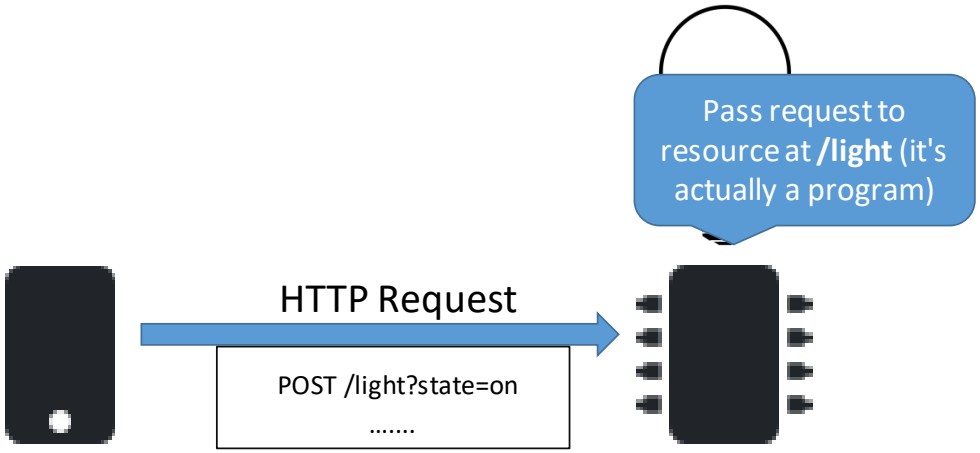
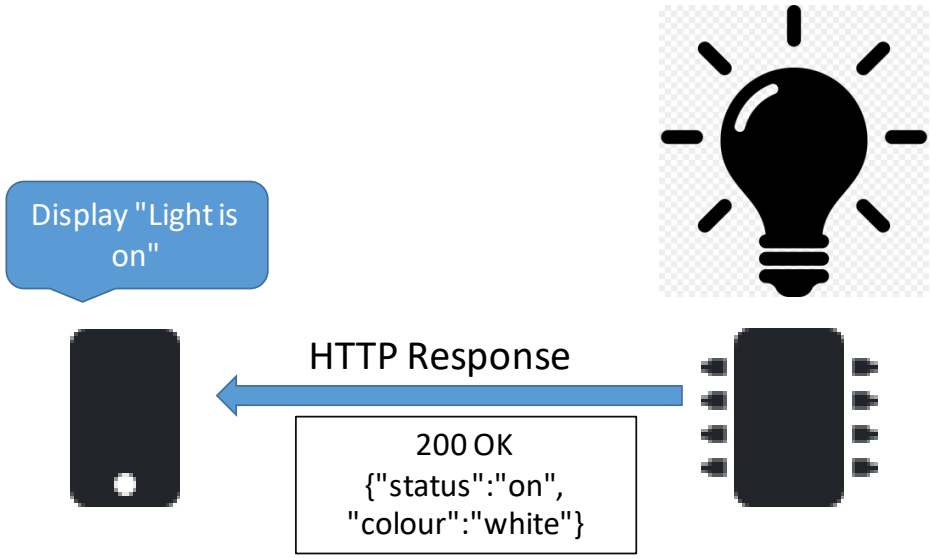
HTTP Post

HTTP Post


Pass request to resource at **/light** (it's actually a program)
HTTP Request
POST /light?state=on
.....

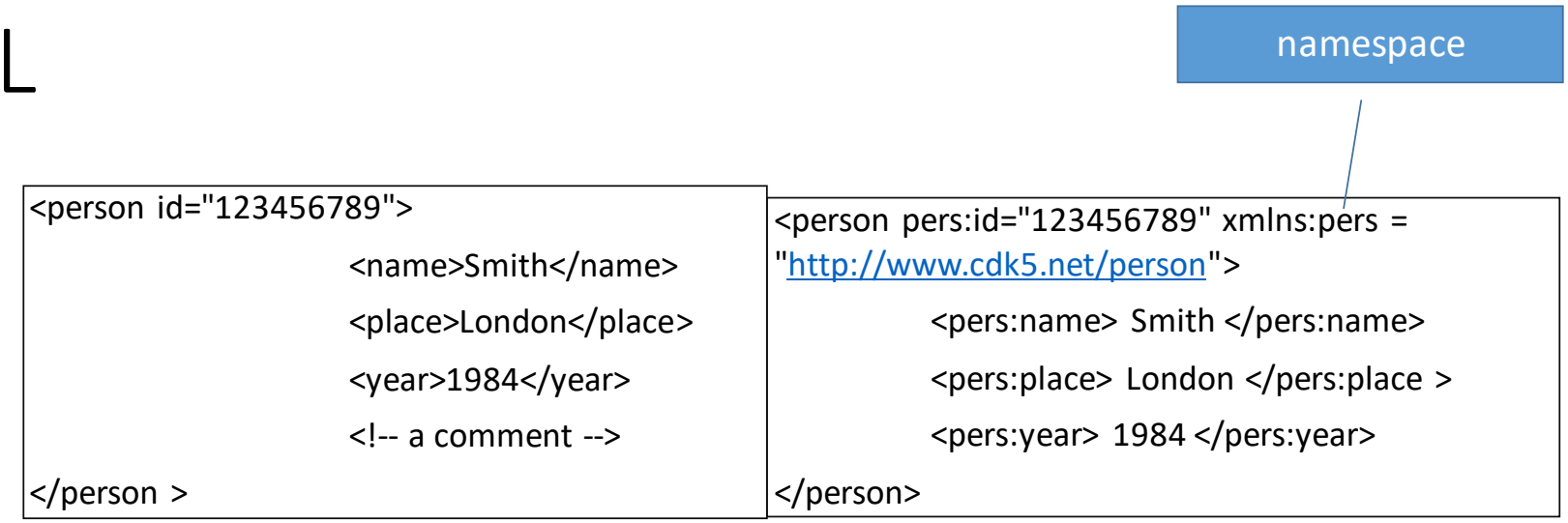
HTTP Post


Display "Light is on"
HTTP Response
200 OK
{"status":"on",
"colour":"white"}

External Data Representations
XML

- eXtensible Markup Language(XML)
- Same heritage as HTML(but XML is NOT HTML)
- XML data items are tagged with ‘markup’ strings
  - used to describe the logical structure of the data
- XML has many uses. For now we confine ourselves to external data representations
- Has many cool features including
  - Extensible
  - Textual
  - Kind of human readable and machine readable...

XML



- Above shows XML definitions of the Person structure.
  - As with xHTML, tags enclose character data.
  - Tags : <name>, <place>,<year> data:”Smith”, “London”…
- Namespaces provide a means for scoping names

JSON

- JavaScript Object Notation
- Lightweight text-based open standard designed for human readable data interchange.
- Can represent simple data structures and associative arrays.
- Good for serializing and transmitting structured data across a network


```
{
    "id":"example",
    "current_value":"500",
    "at":"2013-05-06T00:30:45.694188Z",
    "max_value":"500.0",
    "min_value":"333.0",
    "version":"1.0.0"
}
```

JSON

- JSON is a data interchange format technique
- A collection of name/value pairs.
- Application programming interfaces(APIs) exist for most programming languages

```
{
  "person": {
    "id": "123456789",
    "name": "Smith",
    "place": "London",
    "year": "1984"
  }
}
```

```
{ "temperature"  : 72.55 }
```

More on HTTP

- For an extensive overview, checkout:

http://www.ntu.edu.sg/home/ehchua/programming/webprogramming/http_basics.html

REST

- Short for Representational State Transfer
- Set of Principles for how web should be used
- Coined by Roy Fielding
  - One of the HTTP creator
- A set of principles that define how Web standards(HTTP and URIs) can be used.

Key REST Principles

1. Every “thing” has an identity
   - URL
2. Link things together
   - Hypermedia/Hyperlinks
3. Use standard set of methods
   - HTTP GET/POST/PUT/DELETE
   - Manipulate resources through their representations
4. Resources can have multiple representations
   - JSON/XML/png/...
5. Communicate stateless
   - Should **not** depend on server state.

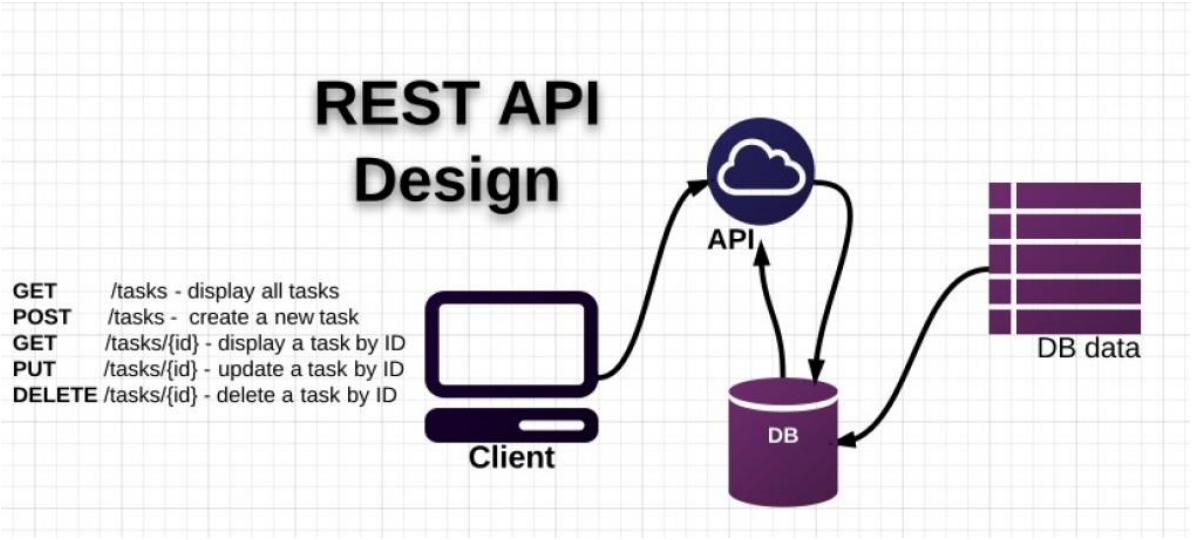
“API First” approach

- Collaboratively design, mockup, implement and document an API **before** the application or other channels that will use it even exist.
- Uses “clean-room” approach.
  - the API is designed with little consideration for the existing technology estate.
  - the API is designed as though there are no constraints.

source: http://www.programmableweb.com/news/introduction-to-api-first-design/analysis/2016/10/31

Traditional API Design

- API design happens after the release of some a data-rich application
  - Existing application “wrapped” in API
- Created as an afterthought.
  - Tightly bound application needs data/function exposed as API.
  - Shoe-horned in as a separate entity.



Advantages of Web APIs

- Suits multi-device environment of today.
- An API layer can serve multiple channels/devices.
  - Mobile/tablet/IoT device
- Scalable, modular, cohesive and composeable
  - If designed properly!(e.g. microservice/Rest architecture)
- Concentrate on function first rather than data

APIs in the Internet of Things

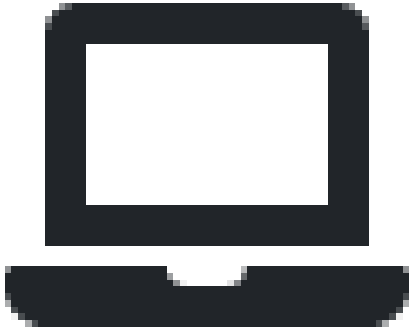
- Many new IoT devices emerging.
- Devices are limited on their own
  - Accompanying APIs invite innovation and generate value
- "Build a better mousetrap, and the world will beat a path to your door" - Ralph Waldo Emerson
  - e.g. Rentokil believe they have using APIs
  - Rentokil increased operational efficiency and compliance through the automatic notifications of a caught animal and its size.
  - Core to this are web APIs.

https://www.computerworlduk.com/it-business/rentokil-on-iot-rat-traps-cash-for-apps-incentives-apis-3612866/

HTTP Web API on an IoT devices

- Easy to set up a Web server on a Raspberry Pi(or smaller device):
  - Connects sensors/actuators to web
  - Access and Control your devices via the Web:
    - Web application program interface(Web API)

Demo



Client

GET /temperature HTTP/1.1

HTTP/1.1 200 OK
.....
{"temperature": 22.45}

Server