# WoT Directory API

Registration, Notification, Error Responses

Farshid Tavakolizadeh (Fraunhofer)

2020-10-20

WoT Online F2F

# Outline

- API Overview

- Registration API

- Notification API

- Error Responses

# Directory API Overview

- Registration
    - Management (CRUD) of Thing Descriptions
    - CRUD spec in First Public Working Draft (FPWD)

- Notification
    - Registration events
    - Spec in FPWD

- Search
    - Syntactic and semantic search
    - Spec in FPWD
    - More in the next presentation

- Management
    - Administration (server, security config)
    - To be specified. Should this even be in the spec?

API is defined in a Thing Description and supporting text.

# Registration API

RESTful[1] HTTP[2] Create, retrieve, update, and delete (CRUD) operations:

**/td**

- POST a TD[3]
  - Create an Anonymous TD -> Response header provides a system-generated identifier (i.e. UUID v4)

**/td/{id}**

- GET a TD
  - Retrieve a TD with the given identifier

- PUT a TD[3]
  - Create/Update a TD with the given identifier –> Response status is 201 Created/204 No Content

- PATCH a TD[3]
  - Update a Partial TD or whole TD with the given identifier

- DELETE a TD
  - Delete a TD with the given identifier

[1] RESTful Design for Internet of Things Systems: https://tools.ietf.org/html/draft-irtf-t2trg-rest-iot-06
[2] Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content: https://tools.ietf.org/html/rfc7231
[3] [Recommended] TD validation against TD JSONSchema

# Registration API (cont'd)

TD Link example:
```
{
    "@context":"…",
    "@type":"Link",
    "title":"Title of linked TD",
    "security":"…",
    "securityDefinitions":"…",
    "links":[
        {
            "rel":"furtherExploration",
            "href":"/td/1234",
            "type":"application/td+json"
        }
    ]
}
```

Beyond FPWD

RESTful Listing (CRUD**L**):

**/td**

- Get TDs
    - Get a list of all TDs
    - Requires pagination/streaming to reduce server and client load
    - **Pagination**: limit total TDs per page
        - How to deal with a single, large TD?
            - Replace large TDs with TD links? Inconsistent results, but easy to process if links have a specific @type.
            - List only links to TDs? Too much overhead to query every TD separately.
            - Erase the problem: prevent creation of TDs that are larger than a configurable value?
            - Give control to client to choose (Full TDs|TD Links). Server can reject the request if it doesn't want to return a list that is too big.
    - Streaming: next slide

Related Issues:
Handle huge set of Thing Descriptions: https://github.com/w3c/wot-discovery/issues/16
Add links section to directory information model: https://github.com/w3c/wot-discovery/issues/34

# Registration API (cont'd)

<u>Beyond FPWD</u>

RESTful Listing (CRUD**L**):

**/td**

- Get TDs
  - Get a list of all TDs
  - Requires pagination/streaming to reduce server and client load
  - **Streaming**
    - Revisit this in the follow-up discovery versions?
    - gRPC/HTTP2 (not just good for TD stream fetch, but also request/response and notifications)
      - Defining the protobuf message for TD may be challenging

Related Issues:
Handle huge set of Thing Descriptions: https://github.com/w3c/wot-discovery/issues/16
Add links section to directory information model: https://github.com/w3c/wot-discovery/issues/34

# Notification API

TD change events. Based on Server-Sent Events (SSE)*

Features:

- Re-connecting clients can continue from the last received event (part of SSE)

- Client-side event filtering (part of SSE)

- Server-side filtering

- TD change diffs

Related issues:
Decide how to securely do notifications from a Directory: https://github.com/w3c/wot-discovery/issues/42
Directories should notify subscribers when a TD is updated: https://github.com/w3c/wot-discovery/issues/28

* Server-Sent Events (SSE): https://www.w3.org/TR/eventsource/

# Notification API (cont'd)

Event Types:

- created_td, updated_td, deleted_td

Event Filtering:

- /events -> all events

- /events?**type**=created_td&**type**=deleted_td -> all create and delete events

- /events?**type**=updated_td&**td_id**=urn:example:1234 -> updates on TD with urn:example:1234 ID

- [Optional] /events?**jsonpath**=$[?(@.title=~/.*Temperature Sensor/) -> updates on TDs with title ending with Temperature Sensor

Related issues:
Decide how to securely do notifications from a Directory: https://github.com/w3c/wot-discovery/issues/42
Directories should notify subscribers when a TD is updated: https://github.com/w3c/wot-discovery/issues/28

# Notification API (cont'd)

Event Data:

- JSON Object

- td_id set to TD identifier

- [Optional] When /events?**include_changes**=true
    - created_td event: created_td set to the new TD
    - updated_td event: td_updates set to TD diffs

Related issues:
Decide how to securely do notifications from a Directory: https://github.com/w3c/wot-discovery/issues/42
Directories should notify subscribers when a TD is updated: https://github.com/w3c/wot-discovery/issues/28

# Error Responses

Based on Problem Details*


For now, type is omitted (defaults to about:blank)


Suggestion:

- Globally defined, protocol-agnostic WoT error types

```
Error response example:
{
   "type":"https://w3c.org/wot/invalid_td",
   "title":"Invalid Registration",
   "status":400,
   "detail":"Validation against JSONSchema detected one or more errors",
   "instance":"/error/123e4567-e89b-12d3-a456-426655440000",
   "validation-errors":[
      "(root): security is required",
      "properties.lightInformation.properties.state.properties.lastupdated.type:
properties.lightInformation.properties.state.properties.lastupdated.type must
be one of the following: \"boolean\", \"integer\", \"number\", \"string\",
\"object\", \"array\", \"null\""
   ]
}
```

Related Issues:
Define error responses: https://github.com/w3c/wot-discovery/issues/44
Error conditions for form requests: https://github.com/w3c/wot-thing-description/issues/303
Error Handling: https://github.com/w3c/wot-scripting-api/issues/200

* Problem Details: https://tools.ietf.org/html/rfc7807

# Other Open Issues

- Identifier of TD in directory

- Using OAuth scopes for fine-grained access control

  - E.g. allow updates but prevent creation (same HTTP path and method)

- Equivalent OpenAPI spec: https://github.com/w3c/wot-discovery/issues/82

- Anonymous TD Deduplication: https://github.com/w3c/wot-discovery/issues/48

- Define directory vocab: https://github.com/w3c/wot-discovery/issues/54

- Out-of-band information: https://github.com/w3c/wot-discovery/issues/30

- TTL Use cases: https://github.com/w3c/wot-discovery/issues/18

And many more https://github.com/w3c/wot-discovery/issues

# Thank you for listening

## WoT Discovery Draft

[https://w3c.github.io/wot-discovery](https://w3c.github.io/wot-discovery)

## Fraunhofer FIT

[https://www.fit.fraunhofer.de/en.html](https://www.fit.fraunhofer.de/en.html)

Contact: Farshid Tavakolizadeh <firstname.lastname@fit.fraunhofer.de>