



WoT Thing Directory

by LinkSmart

Farshid Tavakolizadeh (Fraunhofer)

2020-06-22

WoT Online F2F

Outline

- Current Status
- Naming
- JSONPath / XPath Query Languages
- Directory Meta Info
- Pagination
- DNS-SD Service Type
- TDs without ID
- Directory TD

Current Status

- Service Discovery
 - DNS-SD registration
- RESTful API
 - HTTP API
 - OpenAPI specs
 - Thing Description (TD) CRUD, catalog, and validation
 - XPath 3.0 and JSONPath query languages
 - TD validation with JSON Schema
 - Request authentication and authorization
 - JSON-LD response format (except for JSON fragments)
- Persistent Storage
 - LevelDB
- CI/CD (Github Actions)
 - Automated testing
 - Automated builds and releases (Docker images, binaries)

Naming

- Thing Directory¹ (TD)
 - Nice, but shares the same acronym with Thing Description (TD)
 - Adds ambiguity in deployments, API, code, documentation
 - Examples:
 - "/td": Thing Directory or Thing Description endpoint?
 - "The context of TD has been changed": both have context
 - Hypermedia proposal²: Thing Directory (TDir)
- Change the name?
 - Thing Description Directory?
 - Directory Service?
 - ...
- Introduce a short form?
 - tdir
 - tds

¹ <https://www.w3.org/TR/wot-architecture/#dfn-thing-directory>

² <https://github.com/w3c/wot-thing-description/tree/master/proposals/hypermedia-control>

JSONPath / XPath

- Filtering

- /td?jsonpath=\$[?(@.title=='Temperature Sensor')]
- /td?xpath=*[title='Temperature Sensor']

- JSON-LD response

- Array of TD with original context and schema

```
{
  "@context": "https://linksmart.eu/thing-directory/context.jsonld",
  "@type": "Catalog",
  "items": [
    {
      "@context": "https://www.w3.org/2019/wot/td/v1",
      "id": "urn:example:1234",
      ...
    }
  ],
  "page": 1,
  "perPage": 100,
  "total": 1
}
```

JSONPath / XPath

- Value/object selection:
 - Select: `/td?xpath=**//properties/status//href`
 - Filter and select:
 - `/td?xpath=*[title='Terrace Temperature Sensor']/properties/status//href`
- JSON fragment responses
 - Array of “href”s

```
{
  "@context": "https://linksmart.eu/thing-directory/context.jsonld",
  "@type": "Catalog",
  "items": [
    "https://example.com/status"
  ],
  "page": 1,
  "perPage": 100,
  "total": 1
}
```

JSONPath / XPath

- Pros
 - Short and expressive
 - Passed as URL query parameters
 - Value/object selection saves both client and server resources
- Cons
 - Fragmented responses (selection queries) are not JSON-LD compliant
 - Schema can get messy (e.g. when querying multiple values with different types)
 - Most implementations perform queries on the serialized data only
 - Injection software attack?
 - JSONPath: no formal specification

Related Issues:

Consider Role "Part of" TDs in Query Results: <https://github.com/w3c/wot-discovery/issues/21>

Compare Query and Filter approaches: <https://github.com/w3c/wot-discovery/issues/17>

Directory Meta Attributes

- TTL
 - The validity of TD
 - The maximum duration for client caching
 - Similar to TTL of DNS records
 - The lifetime of TD inside the directory after registration/update/observation
- Update time (inside directory)
 - Registration or update time of TD
 - Required internally to calculate expiry (=update+ttl)
 - May be useful for other applications (e.g. list all TDs updated today)

Directory Meta Attributes

Approach #1

- Add "ttl" attribute to TD specs
- Or extend the context for directory only
- Read-only "meta" in responses, only when requested (nice to have)
 - Digital proof inside TD based on LD-Proofs*
draft

* <https://w3c-ccg.github.io/ld-proofs>

```
/td/urn:example:1234?include_directory=true
{
  "@context": [
    "https://www.w3.org/2019/wot/td/v1",
    {"dir": "https://www.w3.org/2019/wot/xxx/v1"},
  ],
  "id": "urn:example:1234",
  "title": "ExampleSensor",
  "created": "2020-06-12T08:35:44.897112128Z",
  "modified": "2020-06-12T14:38:57.411137834Z",
  ...
  "ttl": 300,
  "dir:meta": { // present because of the query parameter
    "created": "2020-06-12T08:35:44.897112128Z",
    "modified": "2020-06-12T08:35:44.897112128Z",
    "expiry": "2020-06-12T08:40:44.897112128Z", // modified+ttl
    "lastSeen/lastActive": "2020-06-23T14:38:57.411137834Z", // health check / poke
  }
}
```

Directory Meta Attributes

Approach #2

- Add "ttl" attribute to TD
- Define a second resource type for meta info

```
/td/urn:example:1234
```

```
{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "id": "urn:example:1234",
  "title": "ExampleSensor",
  "created": "2020-06-12T08:35:44.897112128Z",
  "modified": "2020-06-12T14:38:57.411137834Z",
  ...
  "ttl": 300
}
```

```
/xxx/urn:example:1234
```

```
{
  "@context": "https://www.w3.org/2019/wot/xxx/v1",
  "id": "urn:example:1234",
  "created": "2020-06-12T08:35:44.897112128Z",
  "modified": "2020-06-12T08:35:44.897112128Z",
  "expiry": "2020-06-12T08:40:44.897112128Z", // modified+ttl
  "lastSeen": "2020-06-23T14:38:57.411137834Z", // health check / poke
  "link": "/td/urn:example:1234" // link to TD
}
```

Directory Meta Attributes

Approach #3

- New resource type
 - Out-of-band information
- API will accept/respond the new resource type, not just td

```
/not-just-td/urn:example:1234
{
  "@context": "https://www.w3.org/2019/wot/xxx/v1",
  "id": "urn:example:1234",
  "td": {
    "@context": "https://www.w3.org/2019/wot/td/v1",
    "id": "urn:example:1234",
    "title": "ExampleSensor",
    "created": "2020-06-12T08:35:44.897112128Z",
    "modified": "2020-06-12T14:38:57.411137834Z",
    ...
  },
  "ttl": 300,
  "created": "2020-06-12T08:35:44.897112128Z",
  "modified": "2020-06-12T08:35:44.897112128Z",
  "expiry": "2020-06-12T08:40:44.897112128Z", // modified+ttl
  "lastSeen": "2020-06-23T14:38:57.411137834Z", // health check / poke
}
```

Directory Meta Attributes

LinkSmart's choice:

Approach #1

Single resource type

- "ttl" inside the TD
- If no "ttl", directory may
 - Keep forever
 - Reject
- Proof and Proof Chain* based on LD-Proofs draft

Related Issues:

Consider TTL User cases: <https://github.com/w3c/wot-discovery/issues/18>

Consider how to sign TDs in a directory service: <https://github.com/w3c/wot-discovery/issues/24>

Add integrity protection to TDs: <https://github.com/w3c/wot-security/issues/166>

```
/td/urn:example:1234
{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "id": "urn:example:1234",
  "title": "ExampleSensor",
  "created": "2020-06-12T08:35:44.897112128Z",
  "modified": "2020-06-12T14:38:57.411137834Z",
  ...
  "ttl": 300,
  "proofChain": [ // https://w3c-ccg.github.io/ld-proofs/#proof-chains
    {}, // original TD proof
    {} // proof added by directory
  ]
}
```

* <https://w3c-ccg.github.io/ld-proofs/#proof-chains>

Pagination

/td?page=2&per_page=10

```
{
  "@context": "https://linksmart.eu/thing-directory/context.jsonld",
  "@type": "Catalog",
  "items": [
    // 10 TDs
  ],
  "page": 2,
  "perPage": 10,
  "total": 25,
  "nextLink": "/td?page=3&per_page=10" // replace page/perPage with next link?
}
```

- Offset and limit?

/td?offset=10&limit=10

DNS-SD Type

- LinkSmart uses:
 - Type: `_wot._tcp`
 - Subtype: `_directory`
 - TXT Record: `"version=<api-version>" "td=/td"`

- Browse for WoT directory services

```
$ avahi-browse _directory._sub._wot._tcp --resolve
+ vpn_tun0 IPv4 Plugfest Local Thing Directory      _wot._tcp      local
= vpn_tun0 IPv4 Plugfest Local Thing Directory      _wot._tcp      local
  hostname = [iot-linksmart-white.local]
  address = [192.168.30.132]
  port = [8081]
  txt = ["version=v1.0.0-beta.15" "td=/td"]
```

- Browse for all WoT services

- How to distinguish results?

- TXT records?
- Dedicated service type for directory?

```
$ avahi-browse _wot._tcp --resolve
+ vpn_tun0 IPv4 Plugfest Local Thing Directory      _wot._tcp      local
+ vpn_tun0 IPv4 myRasPiLED                          _wot._tcp      local
= vpn_tun0 IPv4 Plugfest Local Thing Directory      _wot._tcp      local
  hostname = [iot-linksmart-white.local]
  address = [192.168.30.132]
  port = [8081]
  txt = ["version=v1.0.0-beta.15" "td=/td"]
= vpn_tun0 IPv4 myRasPiLED                          _wot._tcp      local
  hostname = [ktrapi.local]
  address = [192.168.30.135]
  port = [1880]
  txt = ["retrieve=/.well-known/wot-td"]
```

TDs without ID

- **TDs with ID**

```
PUT /td/urn:example:1234
```

```
{
  "id": "urn:example:1234",
  "title": "My TD",
  ...
}
```

- **TDs without ID**

```
POST /td
```

```
{
  "title": "My TD",
  ...
}
```



```
GET /td/urn:uuid:0ff82c68-a9dd-4342-91a9-a326b8ee2d50
```

```
{
  "id": "urn:uuid:0ff82c68-a9dd-4342-91a9-a326b8ee2d50"
  "title": "My TD",
  ...
}
```

Response Location Header: urn:uuid:0ff82c68-a9dd-4342-91a9-a326b8ee2d50

Directory TD

```
{
  "@context": "https://www.w3.org/2019/wot/td/v1",
  "title": "LinkSmart Thing Directory",
  "securityDefinitions": {
    "bearer_sc": { "scheme": "bearer" }, // currently only OpenID Connect's ID Token
    "basic_sc": { "scheme": "basic" } // internally, OAuth2 Password Grant to get an ID Token
  },
  "security": [ "bearer_sc", "basic" ],
  "actions": {
    "td": {
      "uriVariables": {
        "id": { "type": "string", "format": "iri-reference" }
      },
      "forms": [
        {
          "href": "/td",
          "htv:methodName": "POST",
          "contentType": "application/ld+json"
        },
        {
          "href": "/td/{id}",
          "htv:methodName": "PUT",
          "contentType": "application/ld+json"
        },
        {
          "href": "/td/{id}",
          "htv:methodName": "PATCH",
          "contentType": "application/ld+json"
        },
        {
          "href": "/td/{id}",
          "htv:methodName": "DELETE"
        }
      ]
    }
  }
}
```

```
"properties": {
  "td": {
    "uriVariables": {
      "id": { "type": "string", "format": "iri-reference" },
      "page": { "type": "number", "minimum": 1 },
      "per_page": { "type": "number", "minimum": 1, "maximum": 100 },
      "oneOf": {
        "xpath": { "type": "string" },
        "jsonpath": { "type": "string" }
      }
    },
    "forms": [
      {
        "href": "/td/{id}",
        "htv:methodName": "GET",
        "contentType": "application/ld+json"
      },
      {
        "href": "/td{?page,per_page,xpath,jsonpath}",
        "htv:methodName": "GET",
        "contentType": "application/ld+json"
      }
    ]
  },
  "validation": {
    "forms": [
      {
        "href": "/validation",
        "htv:methodName": "GET",
        "contentType": "application/ld+json"
      }
    ]
  }
}
```

<https://github.com/linksmart/thing-directory/blob/master/apidoc/directory-td.jsonld>

Directory TD

- Is TD meant to replace the OpenAPI specs?
 - Cannot define different response types based on HTTP status codes
 - E.g. 4xx/5xx error responses
 - Few ways to define the operations:
 - HTTP Methods
 - Interaction Affordance subclasses: properties/actions
 - PropertyAffordance: readOnly/writeOnly
 - No OpenID Connect security definition
- Collaborate with the AsyncAPI Initiative?

Thank you for listening

LinkSmart Thing Directory

<https://github.com/linksmart/thing-directory>

LinkSmart

<https://linksmart.eu/>

<https://github.com/linksmart>

Fraunhofer FIT

<https://www.fit.fraunhofer.de/en.html>

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