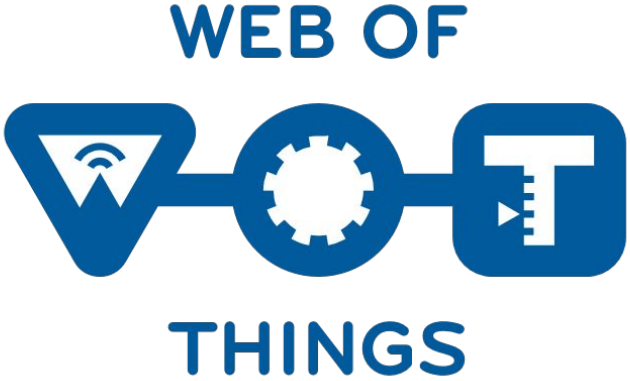


WoT Scripting API

W3C WoT vF2F, October 2021

Table of Contents

Consume options - reduce TD size & complexity

Problem: TDs can easily become huge with various protocols, security schemes and multiple IP addresses

Consumer could:

- Ask for given binding and/or security only
- Provide options initially

Proposed API change

- From `Promise<ConsumedThing> consume(ThingDescription td);`
- To `Promise<ConsumedThing> consume(ThingDescription td, optional ConsumeOptions options);`

- At the moment each interaction can specify *formIndex*
- Consume options may include hints for desired binding, security scheme, ...

Expose options

- Separate API entry point
  - Different security for client API (discovery + consume) and expose API
- Expose options could include passing preferences to runtimes
  - Supported features: protocol binding preferences, security, ...
  - Security schemes
    - Client could pass list of supported/desired security schemes, host selects first one supported, or …
    - Could implementations encapsulate this negotiation?
- Proposed API change

  From:

```
Promise<ExposedThing> produce(ExposedThingInit init);
```

  To:

```
Promise<ExposedThing> produce(ExposedThingInit init,
                              optional ExposeOptions options);
```

Action state and control

- Missing API for cancelling and/or query an action. 2 approaches:
  - WoT Profile Action Model (separate *op*'s, i.e. cancelaction, queryaction)
    - cancelaction ... (more complex: stopping by other consumers possible?)
    - queryaction
  - Model by Cristiano (control object)
- Proposed change in API (control object)
  - From:

    Promise<InteractionOutput> invokeAction(DOMString *actionName*,
    optional InteractionInput *params* = null,
    optional InteractionOptions *options* = null);

    To:

    Promise<ActionControl> invokeAction(DOMString *actionName*,...);

  - *ActionControl* is an object that includes action state, query/cancel methods and a resolver for *InteractionOutput*
  - Multiple clients can invoke an Action and will get back an *ActionControl* object with updated state.
- Open points
  - action queues (implementation detail?)
  - one action at a time vs multiple actions at the same time
  - who is allowed to stop a launched action, security (Management layer)

Discovery

- Discovery spec not yet fully ready
- Scripting discovery API scope maps to 2nd stage discovery
  - 1st stage is related to introduction/provisioning and Scripting API may not be available yet in that run-level (e.g. in OCF)
  - OCF has different run-levels for 1st and 2nd stage discovery (certification!)
  - Could we have separate API entry points for 1st and 2nd stage discovery?
    - different security requirements
    - 1st (provisioning) phase is protocol and certification dependent
- What's next
  - Other ways of discovery? (currently: *directory*, *direct*)
    - *multicast* - Things in the visible network
    - *realm* - Things in the visible network that are part of the same security realm
    - *nearby* - Things around me (Bluetooth, NFC, etc.)
    - *query* (e.g. SPARQL) https://github.com/w3c/wot-scripting-api/issues/321
  - Security-privacy concerns
  - Issues: https://github.com/w3c/wot-scripting-api/labels/discovery

Consequences of *recent* changes

ExposedThing no longer extends ConsumedThing

- How to use ConsumedThing as ExposedThing (e.g., properly unit-test)
  a. Consume ExposedThing
  b. Own Constructor
  c. Discover using discovery function

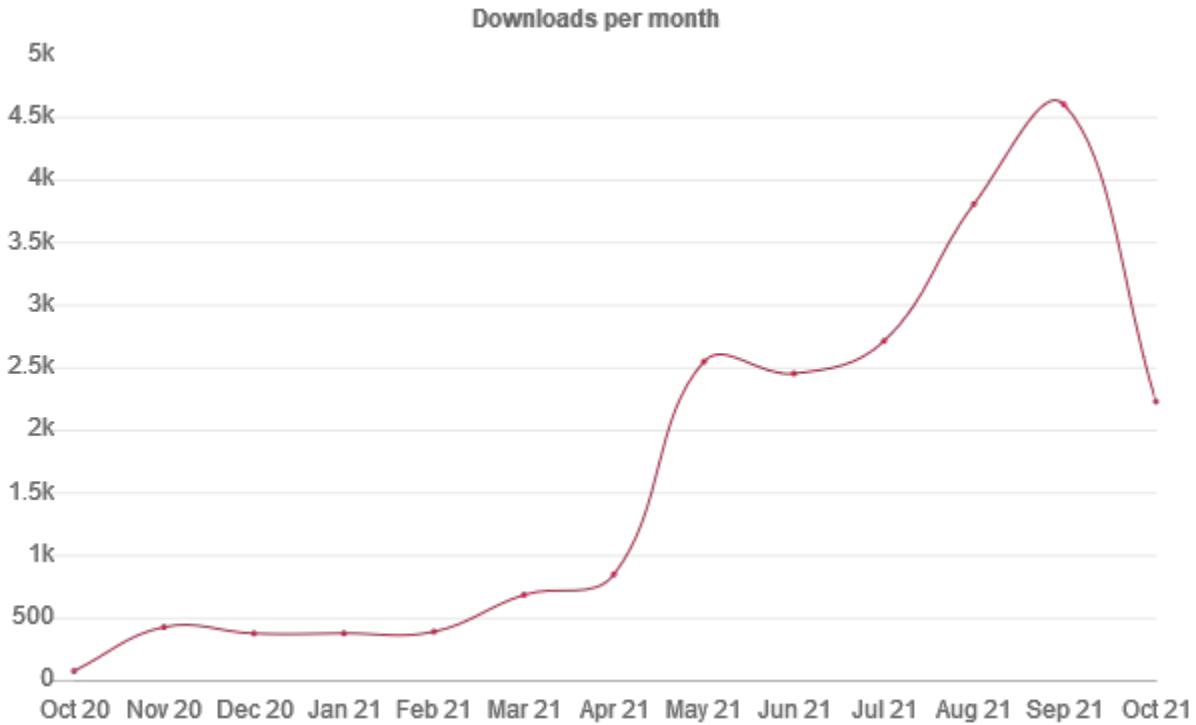
Eclipse node-wot - Status/Update

- Readable Stream support
- New release v0.8.0 planned for the next weeks
- There are pending PRs to align missing parts
  - Strategy: First implement and then update spec? Time-consuming.

- Where node-wot is used (feedback is welcome):
  - Siemens
  - Netzo
  - UNIBO
  - WebOfThings WoT Adapter
  - ECHONET
  - NHK web demo
  - …

Eclipse node-wot - Statistics

https://npmstats.org/@node-wot:td-tools



Scripting API - Charter Thoughts

What do be done in this charter?

- Support ***all*** TD features?
  - TD Canonicalization
    e.g., getThingDescription() reporting canonical TD ?
- Discovery features support
- Binding Templates
- WoT Profile?
  - Within Scripting API scope?

What to do in upcoming Charter?

- Keep up with *new* features
- TD signing ?
- Security Management, Scripting Management, ...