

WoT Testing Framework

Overview of Current Tools and Future

Ege Korkan, Siemens AG
[@egekorkan](#)

Thing Description Playground

TD Playground

Notes from the group:
- Can be also called TD Validator

- Current version being only a validation tool, we need a *real* playground! Like JSON-LD Playground

We are including the following new features and are looking for new ideas :)

Github: <https://github.com/thingweb/thingweb-playground>

Online: <http://plugfest.thingweb.io/playground/>

New Features

Notes from the group:
- TDs should be in the
gists of the Thingweb
account

- Live validation
- Saving TD as a gist online:
 - Could be saved in the user's GitHub or W3C or Thingweb for documentation/statistics purposes
→ Opinions?
- Filling the editor with an example TD
 - What kind of TDs are annoying to understand and would be helpful for beginners?

New Features

Notes from the group:
- Locating the error is very difficult at the current version

- Optional Assertion Testing on the front-end
 - Implies some “heavy” data exchange
- Importing TD from a URL or computer
- Better error messages

Demo

Future Work

Notes from the group:
- Both are needed

Script based validation as an npm package?

A server to send TDs and get validation results?

Any pain points at the moment?

Behavior Testing: Exposed Things

Exposed Thing Testing

A tool to fuzz test all the interactions of the Thing

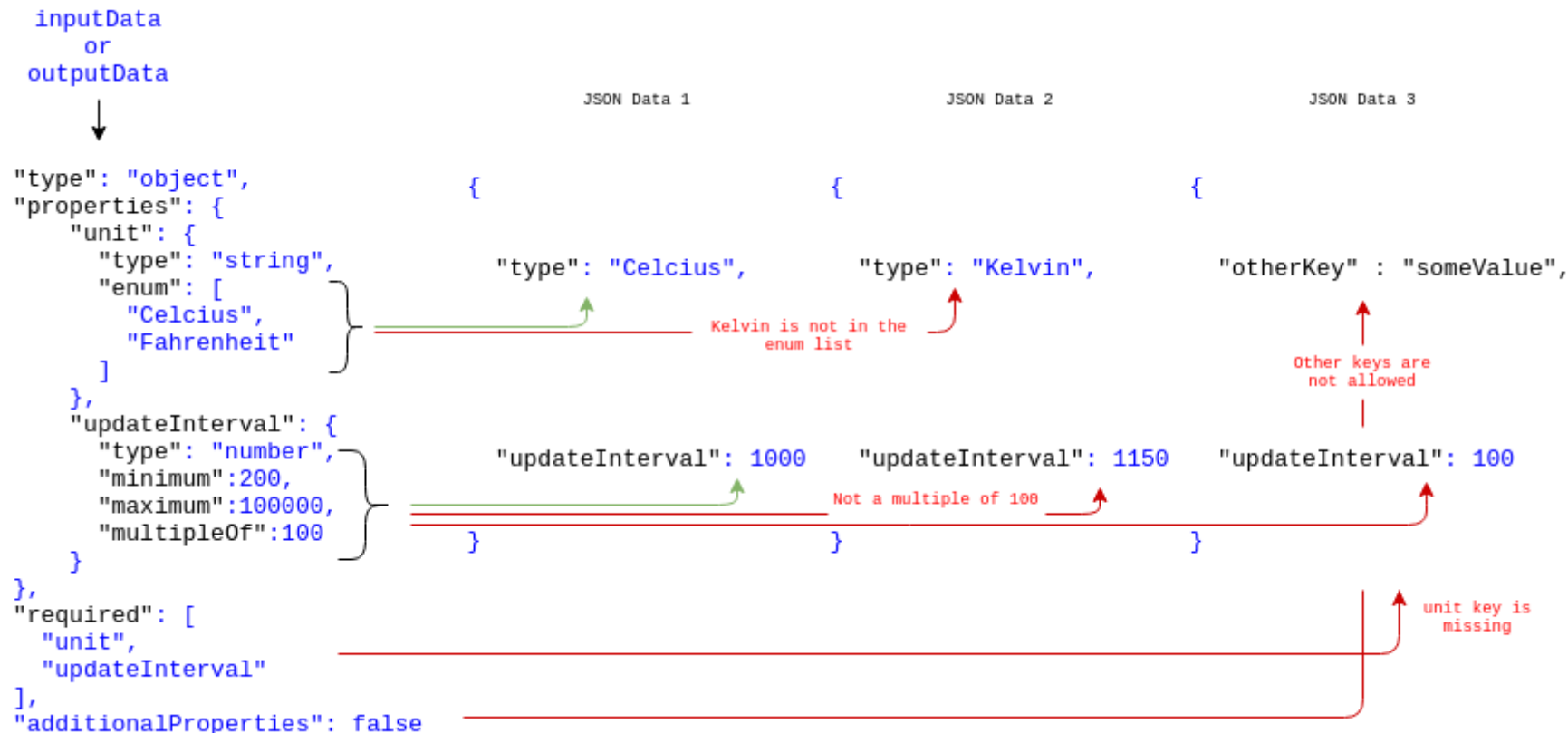
Available [here](#)

WoT Test Bench is being updated to generate the [server side assertion results \(.csv\) on behavior](#)

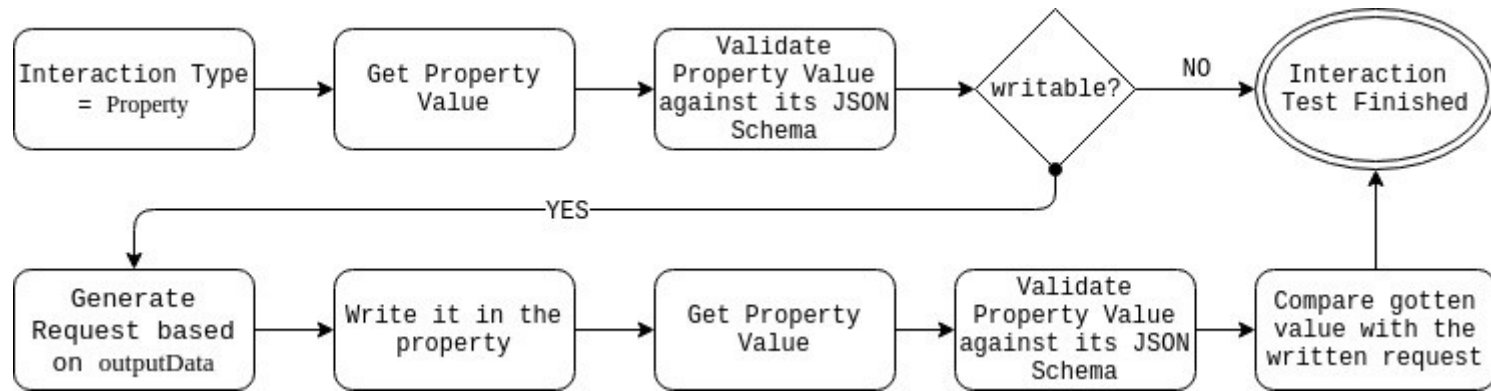
Assertion IDs:

server-data-schema, server-data-schema-extras, server-uri-template

Exposed Thing Testing



Exposed Thing Testing



Similar for Actions

Notes from the group:
- Note that testing whether your write was succesful is not an assertion

Behavior Testing: Consumers

Deployment – Opinions!

How would you be able / like to use it?

- As an online server → Access to only online Exposed Things
- Installing on your own machine → Installation overhead
- Testing from the browser → Restriction to HTTP

Notes from the group:

- The 1st and 3rd can be combined so that when connection to the device is not possible, we revert to HTTP only or ask to download and install the tool

Consumer Capability Testing

What about testing the consumer side?

- Client-data-schema
 - A Thing acting as a Consumer when interacting with another target Thing described in a WoT Thing Description MUST generate data organized according to the data schemas given in the corresponding interactions.
- client-data-schema-accept-extras
 - A Thing acting as a Consumer when interacting with another Thing MUST accept without error any additional data not described in the data schemas given in the Thing Description of the target Thing.
- client-data-schema-no-extras
 - A Thing acting as a Consumer when interacting with another Thing MUST NOT generate data not described in the data schemas given in the Thing Description of that Thing.
- client-uri-template
 - A Thing acting as a Consumer when interacting with another Thing MUST generate URIs according to the URI Templates, base URIs, and form href parameters given in the Thing Description of the target Thing.

Virtual Thing

- [Virtual Thing](#) (on npm) to generate possible Things,
 - **Can** be used to generate a report.
 - **Can** be hosted online on Thingweb.
- We need to identify important TDs that cover many features, as our test set that a client should pass.

Behavior Testing: Break Them!

- Safety testing
 - Sending invalid inputs to Exposed Thing
 - Returning invalid data to Consumers
- Security testing
 - Penetration testing based on security information on the TD
 - Invalid security information on Exposed Things