RI testing that enumerated values are correct

Written by: David Herron - Evoke Systems - david@davidherron.com or david.herron@evokesystems.com

Date: February 3, 2025

Introduction

In *Recommended Implementation for utilizing enumeration schemas in the OpenADR RI* we discussed how the RI, or any VEN or VTN, can use the schema files to implement data validation.

In Definition.md, we have several enumeration tables describing certain values that can be sent via the OpenADR protocol. Some are sent by the VTN to VENs, and others are sent by VENs to the VTN.

This paper focuses on how the RI and the test suite should validate conformance with the OpenADR protocol.

We start from assuming that the RI VTN and VEN implementations will incorporate the schema files, and utilize auto-generated validation functions in a manner similar to what is described.

By having implemented data validation for the values corresponding to the enumeration tables the RI will exhibit this beavior:

- RI VEN will detect invalid values supplied by the VTN, and throw errors
- RI VEN will ensure it does not generate invalid values
- RI VTN will detect invalid values supplied by a VEN, and throw errors
- RI VTN will ensure it does not generate invalid values

Compliance Testing

Certain software products - such as communications protocol, programming platforms where there are multiple implementations - need to validate that an implementation is compliant with the specification.

There is a specification - such as the Java APIs or the OpenADR 3.x protocol - and there are multiple implementations of that specification created by multiple

organizations. Compliance testing aims to ensure that the implementation is correctly conforming to the specification.

Therefore the compliance tests for OpenADR must exercise every API endpoint as well as ensure that data is correctly structured. The correctly structured data includes the objects whose content is defined in the enumeration tables.

Testing a VEN or VTN for compliant behavior on enumerated values

At the high level testing whether a VEN or VTN is correctly handling the enumerated values requires correct data validation.


Figure 1. Testing VTN compliance with valid or invalid enumerated data
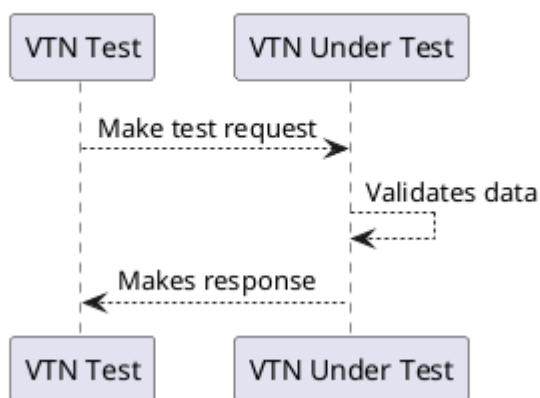
This diagram shows the VTN Test suite validating a VTN being tested for compliance. A similar diagram can be drawn for the VEN Test suite and a VEN being tested for compliance.

In both cases, a compliant VEN or VTN will reject objects that have incorrect values in enumerated fields, and accept objects that have correct values in those fields.

The test suite must validate on every API endpoint.

Figure 2. For each API endpoint, test valid enumerated values
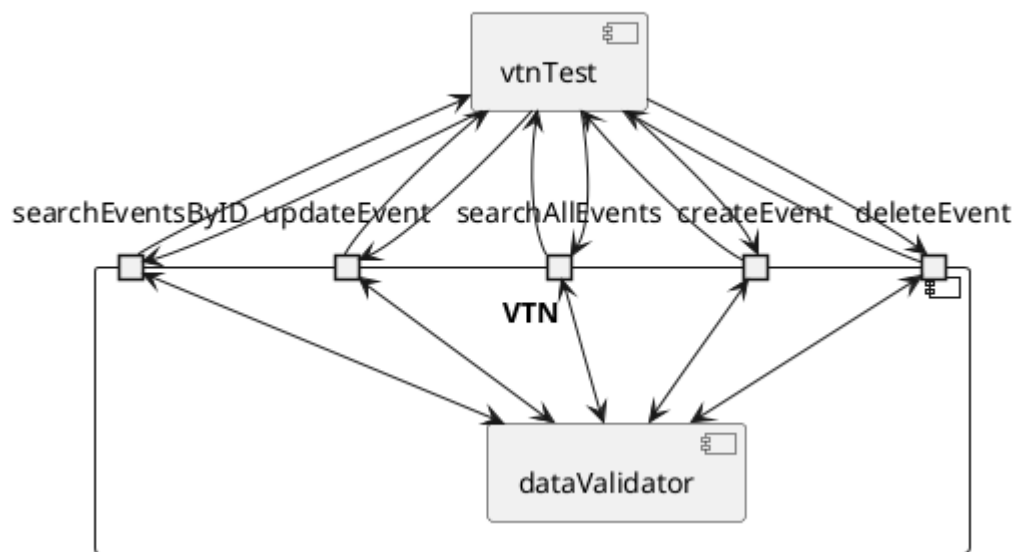
This only lists the endpoints for the `/events` endpoints. The same is true for all other groups of endpoints.

Put another way -

- For each API endpoint
  - Identify the object & enumerated fields in that object
    - For each enumeration *type*
      - Send valid data
      - Send valid data but too many values
      - Send data that matches the correct type but is incorrect range
      - Send data with incorrect type

There are four clusters of data to test for each enumeration. For example consider the SIMPLE payload:

```
valid:
    - type: SIMPLE
      values: [ 0 ]
    - type: SIMPLE
      values: [ 1 ]
    - type: SIMPLE
      values: [ 2 ]
    - type: SIMPLE
      values: [ 3 ]
tooMany:
    - type: SIMPLE
      values: [ 0, 1, 2, 3 ]
```

```yaml
outOfRange:
    - type: SIMPLE
      values: [ 999 ]
    - type: SIMPLE
      values: [ 4 ]
    - type: SIMPLE
      values: [ 1.4 ]
    - type: SIMPLE
      values: [ 2.5 ]
    - type: SIMPLE
      values: [ -1 ]
    - type: SIMPLE
      values: [ -2 ]
    - type: SIMPLE
      values: [ -3 ]
incorrectType:
    - type: SIMPLE
      values: [ '0' ]
    - type: SIMPLE
      values: [ '1' ]
    - type: SIMPLE
      values: [ '2' ]
    - type: SIMPLE
      values: [ '3' ]
    - type: SIMPLE
      values: [ 'three' ]
    - type: SIMPLE
      values: [ 'five' ]
```

A valid SIMPLE payload is a `values` array with a single integer with values between 0 and 3 inclusive.

The values shown here are representative incorrect and correct values.