Conditional Query Parameters for CoAP Observe

draft-core-conditional-attributes-11

Status of the draft

- In WGLC, changes from -10 to -11 based on shepherd review
  - 8 issues in total (Github #50 to #57)
- Changelog of -11:
  - Title of the document changed, and conditional attributes are now called conditional query parameters for accuracy.
  - Clarifying decimal support for conditional control parameters.
  - Text for error handling of type mismatches added
  - Editorial fixes
- 3 issues still open for WG discussion

Open issue 1: Handling complex data structures (Issue #52)

- *"The document states that these query parameters are meant for scalar or boolean values, not for strings or complex data structures. For example, how does this work with resources serialized in SenML for which the value field is decimal but there are other fields like units, timestamp, etc?"*

- Select one (more?) of these options (I prefer option 2 ☺):
  - Have default scope, eg conditional parameters (c.gt, c.lt, etc.) only apply to the v (value) field in SenML or c.pmin apply to the t (time) field
  - Specifying those explicitly in the query eg GET /sensor/temp?c.gt=v:20 (this will allow querying other SenML fields apart from v)
  - Finding some way to handle resource states that have multiple records (eg should only that one record that changes be included in the response, or all records belonging to that set)
  - Don't support complex data structures

Open issue 2: Mixing with non-registered query parameters (Issue #53)

- Should all the query parameters provided in the observe request be registered or do we allow mixing with non-registered query parameters

- Proposal
  - Allow mixing with non-registered query parameters, eg

```
GET
coap://coap.me:5683/sensor?precision=0.1&c.band&c.gt=3
0&c.con&|c.pmax=10
```

Open issue 3: Explaining Resource State Projection (Issue #51)

- “The idea of Resource State Projection is mentioned briefly in the Overview but isn't fully explained. If it's an important concept for understanding the document, it should be clearly defined and explained”

- Proposal (One of these options, TBD in ML?):
  - Expand the current explanation (see proposed text in github issues)
  - Provide an idea of Resource State Projection in another document and refer to it as an informative source
  - Reword to be tighter aligned with bullet point 3 in RFC 7641 Section 1.4:

```
<coap://server/temperature/critical?above=42>, which changes its
state based on the client-specified parameter value either every
few seconds to the current temperature reading if the
temperature exceeds the threshold or to "OK" when the reading
drops below;
```