Modeling JSON Text with YANG

draft-lhotka-yang-json-01

Ladislav Lhotka
\langle lhotka@nic.cz \rangle

31 July 2012
Purpose

The primary aim is to enable validation of JSON text against YANG data models.

Writing JSON mapping rules for YANG directly would be difficult: XPath is not defined for JSON, ...

Instead, a 1-1 translation procedure between JSON and YANG-compatible XML is defined.

*JSON text is valid iff the corresponding XML document is valid.*

Side effect: the translation procedure is bidirectional (invertible), driven by a YANG data model and achieves better results than generic XML→JSON translators.
Namespaces

JSON names live in namespaces, but explicit namespace identifiers are rarely needed.

Format of qualified names in JSON:

<module name>:<local name>

*No prefixes, no URIs.*

Rules:

1. Namespace identifier must be used if the local name alone is not unique among its sibling nodes.
2. Otherwise, the namespace identifier is optional.
## Translation to JSON

<table>
<thead>
<tr>
<th>Node type</th>
<th>JSON</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>leaf</td>
<td>name/scalar</td>
<td>&quot;mtu&quot;: 1500</td>
</tr>
<tr>
<td>container</td>
<td>name/object</td>
<td>&quot;ipv4&quot;:{&quot;enabled&quot;:true,...}</td>
</tr>
<tr>
<td>leaf-list</td>
<td>name/array of scalars</td>
<td>&quot;class&quot;:[&quot;info&quot;,...]</td>
</tr>
<tr>
<td>list</td>
<td>name/list of objects</td>
<td>&quot;interface&quot;:[{...},{...}]</td>
</tr>
</tbody>
</table>

When translating scalar values, YANG datatype is taken into account – numeric datatypes map to JSON numbers, etc.

Values of type “empty” are mapped to `[null]`. 
Example

```text
module ex-json {
    namespace "http://example.com/ex-json";
    prefix ej;
    import ietf-inet-types { prefix inet; }
    container top {
        list address { key "seqno";
            leaf seqno { type uint8; }
            leaf ip {
                type inet:ip-address;
                mandatory true;
            }
        }
        container phases {
            typedef angle {
                type decimal64 {
                    fraction-digits 2;
                }
                units "radians";
            }
            leaf max-phase {
                default "6.28";
                type angle;
            }
            leaf-list phase {
                type angle;
                must ". <= ../max-phase";
                min-elements 1;
            }
        }
    }
}
```

```json
{
    "ex-json:top": {
        "address": [
            {
                "seqno": 1,
                "ip": "192.0.2.1"
            },
            {
                "seqno": 2,
                "ip": "2001:db8:0:1::1"
            }
        ],
        "phases": {
            "phase": [0.79, 1.04, 3.14]
        }
    }
}
```
Applications

This work is used in draft-bierman-netconf-yang-api-00.

With caution, YANG can be used for modeling non-NETCONF JSON data.