

Using XPath 3.1 (and its friends) to work with JSON

Ash Clark (e/em/eir)
Code4Lib 2021

[https://github.com/amclark42/
code4lib_xpath-to-work-with-json](https://github.com/amclark42/code4lib_xpath-to-work-with-json)

Overlap between XML and JSON

- one “root”,
- hierarchy,
- meaningful groupings, and
- common formats for data interchange.

Maps and arrays in XPath 3.1

```
map {  
  "greeting": "Hello",  
  "who": [  
    "World",  
    <orgName>Code4Lib 2021</orgName>,  
    $attendees-from-Northeastern  
  ]  
}
```

Maps and arrays in XPath 3.1

```
map:get($myMap, "greeting")  
  eq $myMap("greeting")  
(: and these `eq $myMap?greeting` :)
```

```
array:get($myArray, 1)  
  eq $myArray(1)  
(: and these `eq $myArray?1` :)
```

Maps and arrays in XPath 3.1

```
$myMap[?greeting eq "Hello"]
```

```
$myArray[?* [. eq "World"]]
```

Takeaways from my use of maps and arrays

- Key-value pairs are *great* for mapping parameter names onto human-readable labels!
- The nesting of arrays means I can do complex ordering of values.
- I can cache the XML representation of JSON in my XML database, which gives me indexing and quick query results.
- I can create RESTXQ APIs which return XML or JSON on request (or HTML, after applying an XSL transformation).
- I can pass maps around instead of creating XQuery functions with oodles of required parameters.