

Xidel: An XQuery interpreter for HTML

We wanted to automate webpages, querying HTML without a browser

How could you do that?

XPath? Most HTML XPath engines support only XPath 1.
Does not even have regular expressions

CSS? Can find simple element, but cannot calculate anything with it

JavaScript? DOM syntax really sucks for queries
(getElementById, ...), did not work without a browser

jQuery, node.js? did not exist back then (summer 2006)

So we wrote (an HTML parser and) a custom query engine partly compatible with XPath

Xidel: An XQuery interpreter for HTML

We needed more features and in the course of time made it a standard compatible

- XPath 2 interpreter
- XQuery 1 interpreter
- JSONiq interpreter

Xidel: An XQuery interpreter for HTML

A few extensions to normal XQuery:

- Additional functions

css Find something with CSS: `css("#foo .class")/(. * 100)`

form Convert a `<form>` element to a HTTP request (encoded as JSONiq object)

inner-html, outer-html Serialize a node to html (encoded as string)

... ..

- Pattern matching
Really great way to query HTML. See on Wednesday
- Variable assignments:
`$var := 123`

Xidel: An XQuery interpreter for HTML

A few changes to normal XQuery:

- Case insensitive element matching
`/foo/BAR` or `/FOO/bar` are the same query
- Case insensitive, "natural" string comparison
`"a" eq "A"` is true, `"12" < "100"` too
(actually just a different default collation)
- Namespaces are resolved dynamically during node testing
`/foo:bar` returns `<foo:bar xmlns:foo=".."/>` even if `foo` is not defined in the query
- Dynamic typing
`"39" + 3` is 42

Xidel: An XQuery interpreter for HTML

A few changes to normal JSONiq:

- Properties readable with node tests
`{"a": 123}/a` returns 123
- Properties readable with dot notation
`{"a": 123}.a` returns 123
- Modifiable objects and arrays
`$object := {"a": 123}, $object("a") := 456`

Xidel: An XQuery interpreter for HTML

Going beyond a single document

- `xidel http://site -e "select content 1" -f "//a (:select links:)" -e "select content 2"`
This will extract the "content 1" from site, and "content 2" from all sites the first site has links to.
- `xidel http://site -f "//a" -e "select content 1" -e "select content 2"`
This will extract "content 1" and "content 2" from all sites the first site links to (and nothing from from site1)
- `xidel http://site -f "form((//form)[1], {'username': 'name', 'password': '...'})" -f 'css("a#link-id")' -e //query`
This will login on site, follow a link with id "link-id"

Available as open source projects:

Xidel Standalone command line XQuery interpreter
(<http://xidel.sourceforge.net>)

Internet Tools Library for FreePascal (http://benibela.de/sources_en.html#internettools or
<https://github.com/benibela/internettools>)

VideLibri (German) desktop program and Android app to
access library catalogues/OPACs
(<http://videlibri.sourceforge.net>)