Writing Zotero Web Translators

Sebastian Karcher

PLoS Webinar, November 3rd, 2014

Types of Zotero Translators

- Web Translator (Imports data from web via URL bar icon)
- Import Translator (Imports data from a file/clipboard)
- Search Translator (imports from a database based on an identifier: DOI, ISBN, PMID)
- Export Translator (exports Zotero data)

Types of Web Translators

- Screen Scrapers
  - Based on Framework
  - From Scratch
- Using Import format
  - Get it from site header
  - Get it via GET or POST
  - MARC is a special case
- Using Search (But we rarely use this)

How a Web Translator Works

- "Do I know this page?" – Target Regex
- "Can I import from this page?" – detectWeb
- Doing the actual work – doWeb

Xpaths – Pointing to content on a webpage

- xpaths are basically “directions” used to point to a part of a webpage
- A webpage is built up from a number of nested nodes
- This is what the most simple webpage looks like

```
<html>
    <head>
        <title>A Basic Webpage</title>
    </head>
    <body>
        <div id="title">Title</div>
        <div id="content" class="text">Content</div>
    </body>
</html>
```

The most basic Xpath

- Give directions: at every corner/node, tell Zotero where to go:
- Let's say we want to go go to "The Content of the webpage"
- "Take the HTLM road, take a left at"body", then take the"div" street, or in HTML:

```
/html/body/div
```

Making Xpaths more precise

- But we're still "lost" - which of the two "div" streets do we go down?
- Option 1: Take the second

```
/html/body/div[2]
```

- Option 2: Take the that has "content" as an id

```
/html/body/div[@id="content"]
```

Making Xpaths more efficient

- In an actual webpage, an xpath can be *very* long, so we'd like to make them shorter. we can use // to start anywhere in the html tree, e.g "the with"content" as an "id" anywhere on the site:

  ```
  //div[@id="content"]
  ```

- Sometimes we don't want the precise content of an attribute like id - in those case we can use contains() as in

  ```
  //div[contains(@id, "cont")]
  ```

- We can combine conditions with "and" or "or" (*in lowercase!*)

  ```
  //div[@id="content" and @class="text"]
  ```

Zotero's built in Xpath helpers

- ZU.xpathText(doc, xpath) returns the text of *all* xpath nodes, separated by comma
- ZU.xpath(doc, xpath) returns an object of all xpath nodes
- You can also use any other javaScript function like doc.evaluate or doc.getElementsBy...

Our Tools

- Scaffold - a Firefox extension to write and test the translator
- Firefox “Inspect Element” - to help us understand the structure of a webpage (there are alternatives like “Firebug”)