Web Scraping with Mechanize

Presented by Dean Stautberg

Mechanize

Documentation: http://mechanize.rubyforge.org

Code: https://github.com/tenderlove/mechanize/tree/master

Description: "The Mechanize library is used for automating interaction with websites. Mechanize automatically stores and sends cookies, follows redirects, can follow links, and submit forms. Form fields can be populated and submitted. Mechanize also keeps track of the sites that you have visited as a history."

Web Scraping

Retrieving data by parsing HTML from a web site or application that doesn't provide a proper API.

Handling cookies and redirects (like Mechanize does) is an important part, because some apps will deny you access if your code does not act like a normal web browser.

Can be fragile. Scraping code can break if the layout of the HTML changes.

Retrieving a Page

```
require 'rubygems'
require 'mechanize'
# Load and parse the page
agent = Mechanize.new
page = agent.get('http://google.com/')
# Access elements of the page
page.body  # the raw html
page.links
page.forms
page.forms.first.fields
```

Finding Content on the Page

Finding links by text:

```
agent.page.links.find { |l| l.text == 'News' }
agent.page.link_with(:text => 'News')
agent.page.links_with(:text => 'News')
```

Using XPath (http://www.w3.org/TR/xpath)

```
# Find all anchor tags within the div with id 'ires'
page.search("//div[@id='ires']//a")
```

Using CSS selectors (http://www.w3.org/TR/CSS2/selector.html)

```
page.search("div#ires a")
```

Submitting a Form

This retrieves the form "f" by name, and gives a value to field "q", and then submits the form.

```
google_form = page.form('f')
google_form.q = 'ruby'
page = agent.submit(google_form)
```

Other Things You Can Do

- Change user agent to mimic certain browsers
- Set http authentication headers
- Use an http proxy
- Swap in a different html parser instead of nokogiri
- Use browser "transactions" to help revert back to certain points in your browsing history

Scraping vs. API's

Be sure to check for an API before resorting to web scraping.

- Twitter API: https://dev.twitter.com/
- Google Search API: http://code.google.com/apis/customsearch/v1/overview.html
- Netflix API: http://developer.netflix.com/

API's will usually release new versions without breaking backward compatibility. Scraped HTML will usually break without warning.

If you have to resort to scraping, at least Mechanize makes it as easy as possible.