Handling HTML Markup with Drupal's Migrate API

Benji Fisher

JANUARY 31, 2020

Introduction

About me

Benji Fisher
Hook 42
drupal.org: benjifisher
twitter: @benji17fisher

Follow along

Find a link to this presentation on my GitLab Pages:

- https://benjifisher.gitlab.io/slide-decks/index.html

Drupal 8 Migrate API

- Upgrade Drupal 6 and Drupal 7 sites
- Migrate sites from other systems to Drupal
- Imports from external systems (feeds)

A robust, flexible tool.

Migrate API: structured data

- file attachments
- related taxonomy terms
- references to authors
- references to other nodes

Migrate API: unstructured text

What about unstructured text with HTML Markup?

- Regular expressions (old)
- HTML parsing (recent)

Our approach: we wrote new Migrate process plugins in *Migrate Plus* for Pega Systems/Isovera.

Outline

- Introduction
- Parsing HTML: regexp
- Parsing HTML: DOMDocument
- Drupal 8 Migrate API
- (Possible) Future
- Conclusion

Parsing HTML: regexp

At a glance

HTML

↓

Regular expression

↓

HTML or extract strings

Simple example (?)

Extract the URL from

```
<a href="https://www.drupal.org">
Drupal home page
</a>
```

Parsing HTML: preg_match()

Extract the URL:

```
$markup = '<a
        href="https://www.drupal.org"
        home page</a>';

$regexp = '/<a href="([^"]+)">/';
preg_match($regexp, $markup,
        $matches);
$url = $matches[1];
```

Parsing HTML: not so simple

Complications:

- HTML tags: match `a` or `A`
- Other attributes: `class`, `id`, `name`, ...
- Single quotes or double quotes
- Newlines within the HTML element
- Are escaped quotes (like `\"`) allowed in a URL?

Trick question: do not reinvent the wheel!

Parsing HTML: examples

Complications:

- `<a href="https://www.drupal.org">Drupal home page</a>` (original)
- `<A href="https://www.drupal.org">Drupal home page</A>`
- `<a class="ext-link" href=...`
- `<a href='https://www.drupal.org'>Drupal home page</a>`
- `<a href="https://www.dr\"upal.org">Drupal home page</a>`

Parsing HTML: right answer, wrong question

```
$regexp = '/<\s*a\b'
      . '[^>]*\bhref'
      . '\s*=\s*'
      . '(["\'])([^"\']+)\1'
      . '/i';
preg_match($regexp, $markup,
           $matches);
$url = $matches[2];
```

Parsing HTML: innocent question

From StackOverflow:

I need to match all of these opening tags:

```
<p>
<a href="foo">
```

But not these:

```
<br />
<hr class="foo" />
```

Parsing HTML: Cthulhu (1/3)

The answer:

> *You can't parse [X]HTML with regex. Because HTML can't be parsed by regex. Regex is not a tool that can be used to correctly parse HTML.*
> ...

Parsing HTML: Cthulhu (2/3)

The answer:

> *Every time you attempt to parse HTML with regular expressions, the unholy child weeps the blood of virgins, and Russian hackers pwn your webapp. Parsing HTML with regex summons tainted souls into the realm of the living. ...*

Parsing HTML: Cthulhu (3/3)

The answer:

Have you tried using an XML parser instead?

Parsing HTML: DOM

At a glance

HTML

↓

Document Object Model (DOM)

↓

HTML or extract strings

DOMDocument basics

The DOM extension uses GNOME's `libxml` library in the background. DOM includes XML Path Language (XPath) traversing.

```
$document = new \DOMDocument();
$document->loadHTML($markup);
$xpath = new \DOMXPath($document);

foreach ($xpath->query('//a') as
         $html_node) {
  $href = $html_node-
         >getAttribute('href');
```

Using Drupal's Html Utility Class

```
use Drupal\Component\Utility\Html;
$document = Html::load($markup)
$xpath = new \DOMXPath($document);
```

XPath Examples

With `$xpath->query($selector)`, ...

`$selector`	**Matches**
`//a`	all <a> elements
`//a[class="external"]`	all <a> elements with `class="external"`
`//li[class="nav"]/a`	all <a> elements direct children of `<li class="nav">`

XPath Example (Complicated)

```
ANEDS/ANED[
  @s:id = ../ANEDOA[
    nc:OR/@s:ref = "ORG0"
  ]/CDR/@s:ref
]/NED/NEDQ[
  ../NEDRC/text() = "ExpeditedDenial"
]
```

From usdoj/foia-api on GitHub (whitespace added, tags abbreviated)

DOMDocument output

After processing, return an HTML string:

```
$processed_html
  = $document->saveHTML();
```

Drupal 8 Migrate API

ETL paradigm

In Drupal 8, the Migrate API follows the standard Extract, Transform, Load (ETL) structure:

- Extract (source plugin): read data from the source
- Transform (process plugins): change data to match the site's structure
- Load (destination plugin): save the data

The Transform/process phase is the right place to handle HTML processing.

At a glance

HTML

↓

Migrate `dom*` process plugins

↓

HTML

New process plugins for managing HTML

Four process plugins in the Migrate Plus module:

- `dom`
- `dom_str_replace`
- `dom_migration_lookup`
- `dom_apply_styles`

Goal: make it easy to process text fields with proper HTML parsing.

The dom plugin

- Create DOMDocument object from string
- Create string from DOMDocument object

```
process:
  'body/value':
    -
      plugin: dom
      method: import
      source: 'body/0/value'
    # Other plugins do their work here.
    -
      plugin: dom
      method: export
```

dom_str_replace plugin

Change the subdomain during migration:

```yaml
-
  plugin: dom_str_replace
  mode: attribute
  xpath: '//a'
  attribute_options:
    name: href
  search: 'documentation.example.com'
  replace: 'help.example.com'
```

Use `str_replace()` or `preg_replace()` on the `href` attribute.

dom_apply_styles plugin

Search for an XPath expression. Replace with styles configured in the Editor module.

```
-
  plugin: dom_apply_styles
  format: full_html
  rules:
    -
      xpath: '//b'
      style: Bold
```

dom_migration_lookup

Like core Migrate's `migration_lookup` plugin.

```yaml
-
  plugin: dom_migration_lookup
  mode: attribute
  xpath: '//a'
  attribute_options:
    name: href
  search: '@/node/(\d+)@'
  replace: '/node/[mapped-id]'
  migrations:
    - article
    - page
```

(Possible) Future

More process plugins

- Migrate Media Handler provides additional DOM-based process plugins for D7 file/image fields to D8 Media entities
- DOM manipulation on process plugins (meta issue)
  - Process non-attribute strings in `dom_str_replace`
  - Remove HTML elements
- Your next project

Different parsers than DOM

Just an FYI, my goto for HTML parsing has been querypath, it's especially good if you're dealing with old-school HTML (no </p>, etc.). - mikeryan on #2958281-7

Different parsers than DOM

- `url` source plugin data parsers
- Make process plugins data extensible: use core typed data.

HTML5

```
Masterminds\HTML5::loadHTML() -> \DOMDocument
```

Conclusion

References

- Migrate API documentation on drupal.org
- Migrate Plus module home page
- Release notes for migrate_plus 8.x-5.0-rc1
- Change record describing the new DOMDocument-based plugins
- Amusing answer on StackOverflow
- Parsing Html The Cthulhu Way
- XPath documentation on MDN

Questions