A Wiki Front End in 40 lines of code, or Doing Cool Things with Wiki Content

Aka: *Parsoid Power!*

http://etherpad.wikimedia.org/p/parsoidpower

C. Scott Ananian <cscott@cscott.net>
Wikimedia Foundation

(And thanks to Gabriel Wicke for many slides!)

T105175
Wikimania 2015
July 15, 2015

Outline

- Introduction to Parsoid HTML5
- Data extraction with Parsoid HTML5
- Introduction to RESTBase API
- Let’s build something real!

Time permitting:

- Using compressed dumps

Before we dive in...

Who uses this stuff?

- **VisualEditor**: modify HTML, serialize to WT
- **Flow**: store HTML, edit WT
- **Kiwix**: massage for offline usage
- **OCG**: generate PDFs via HTML->LaTeX
- **cxserver**: adjust links (later: templates), machine translate**editProtectedHelper gadget**: edit template params
- **Google Knowledge Graph**: extract semantic info from transclusions, tables, lists, categories...
- **Batea**! A bot for checking citations for WikiProject:Medicine
- **You!** "Printable page"? New wiki front ends or experiments?

HTML5 with RDFa

Wikitext: `[[Main Page]]`

HTML: `<a rel="mw:WikiLink" href="./Main_Page">Main Page</a>`

Match: `document.querySelectorAll('a[rel~="mw:WikiLink"]')`

Wikitext: `[http://example.com Link content]`

HTML: `<a rel="mw:ExtLink" href="http://example.com">Link content</a>`

Match: `document.querySelectorAll('a[rel~="mw:ExtLink"]')`

Page properties

Wikitext: `[[Category:Foo]]`

HTML: `<link rel="mw:PageProp/Category" href="./Category:Foo">`

Match: `document.querySelectorAll('[rel~="mw:PageProp/Category"]');`

Wikitext: `__NOTOC__`

HTML: `<meta property="mw:PageProp/notoc">`

Match: `document.querySelectorAll('[rel~="mw:PageProp/notoc"]');`

Similar: `forcetoc, newsectionlink, nonewsectionlink, nogallery, hiddencat, nocontentconvert, notitleconvert, noeditsection, noindex, index, staticredirect`

Images

Wikitext: `[[Image:Foo.jpg|left|caption]]`

HTML:

```
<figure typeof="mw:Image" class="mw-default-size">
  <a href="./File:Foo.jpg">
    <img resource="./File:Foo.jpg"
      src="//upload.wikimedia.org/something/3/3a/Foo.jpg"
      width="200" height="220"></a>
  <figcaption>caption</figcaption>
</figure>
```

Match:
```
document.querySelectorAll(
     'figure, [typeof~="mw:Image"]'
);
```

Transclusions

Wikitext: {{foo|a}}

HTML:

```
<span typeof="mw:Transclusion"
 about="#mw-t1" id="mw-t1"
 data-mw='{...}'>some content here</span>
<span about="#mw-t1">more content here</span>
```

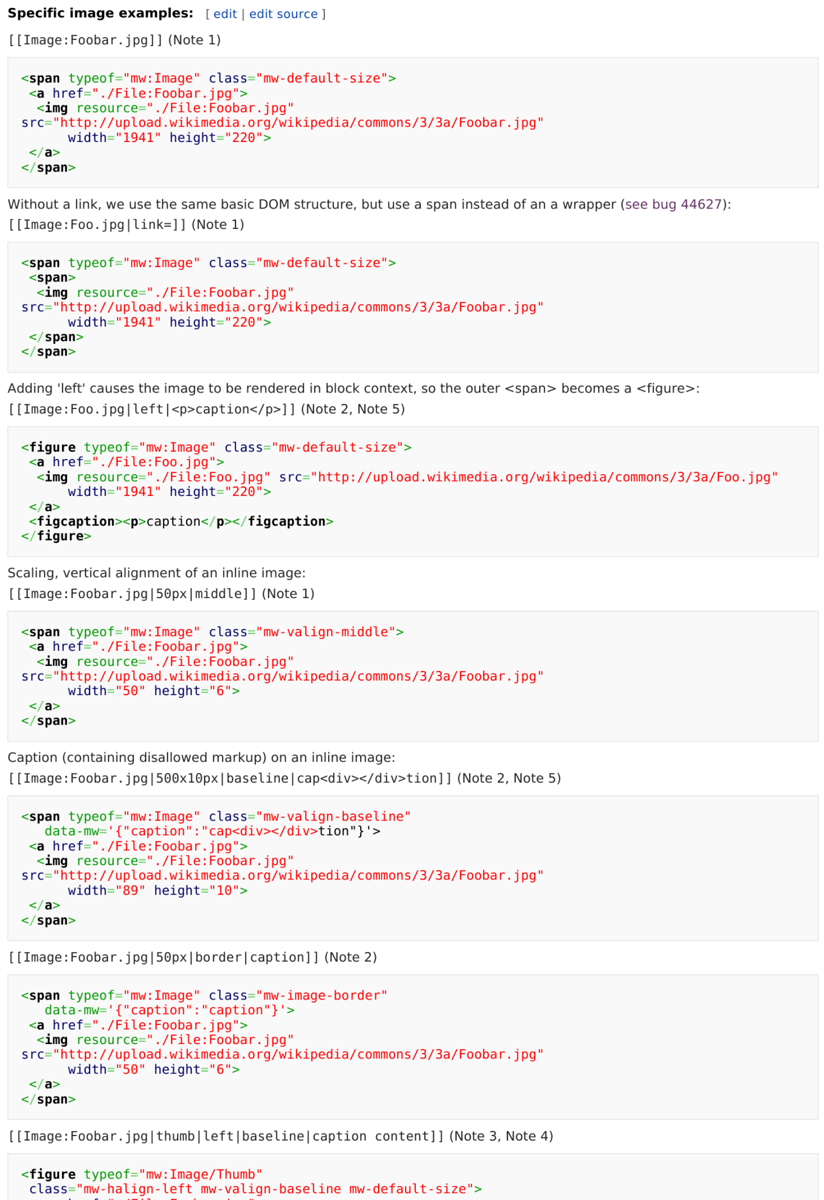
data-mw:

```
{"parts": [
  {
    "target": {
      "wt": "foo",
      "href": "./Template:Foo"
    },
    "params": {
      "1": {
        "wt": "a"
      }
    }
  }
]}
```

Match: `document.querySelectorAll('[typeof~="mw:Transclusion"]')`

DOM spec

mediawiki.org/wiki/Parsoid/MediaWiki_DOM_spec



Specific image examples: [edit | edit source]

[[Image:Foobar.jpg]] (Note 1)

```
<span typeof="mw:Image" class="mw-default-size">
 <a href="./File:Foobar.jpg">
  <img resource="./File:Foobar.jpg"
src="http://upload.wikimedia.org/wikipedia/commons/3/3a/Foobar.jpg"
      width="1941" height="220">
 </a>
</span>
```

Without a link, we use the same basic DOM structure, but use a span instead of an a wrapper (see bug 44627):

[[Image:Foo.jpg|link=]] (Note 1)

```
<span typeof="mw:Image" class="mw-default-size">
 <span>
  <img resource="./File:Foobar.jpg"
src="http://upload.wikimedia.org/wikipedia/commons/3/3a/Foobar.jpg"
      width="1941" height="220">
 </span>
</span>
```

Adding 'left' causes the image to be rendered in block context, so the outer <span> becomes a <figure>:

[[Image:Foo.jpg|left|<p>caption</p>]] (Note 2, Note 5)

```
<figure typeof="mw:Image" class="mw-default-size">
 <a href="./File:Foo.jpg">
  <img resource="./File:Foo.jpg" src="http://upload.wikimedia.org/wikipedia/commons/3/3a/Foo.jpg"
      width="1941" height="220">
 </a>
 <figcaption><p>caption</p></figcaption>
</figure>
```

Scaling, vertical alignment of an inline image:

[[Image:Foobar.jpg|50px|middle]] (Note 1)

```
<span typeof="mw:Image" class="mw-valign-middle">
 <a href="./File:Foobar.jpg">
  <img resource="./File:Foobar.jpg"
src="http://upload.wikimedia.org/wikipedia/commons/3/3a/Foobar.jpg"
      width="50" height="6">
 </a>
</span>
```

Caption (containing disallowed markup) on an inline image:

[[Image:Foobar.jpg|500x10px|baseline|cap<div></div>tion]] (Note 2, Note 5)

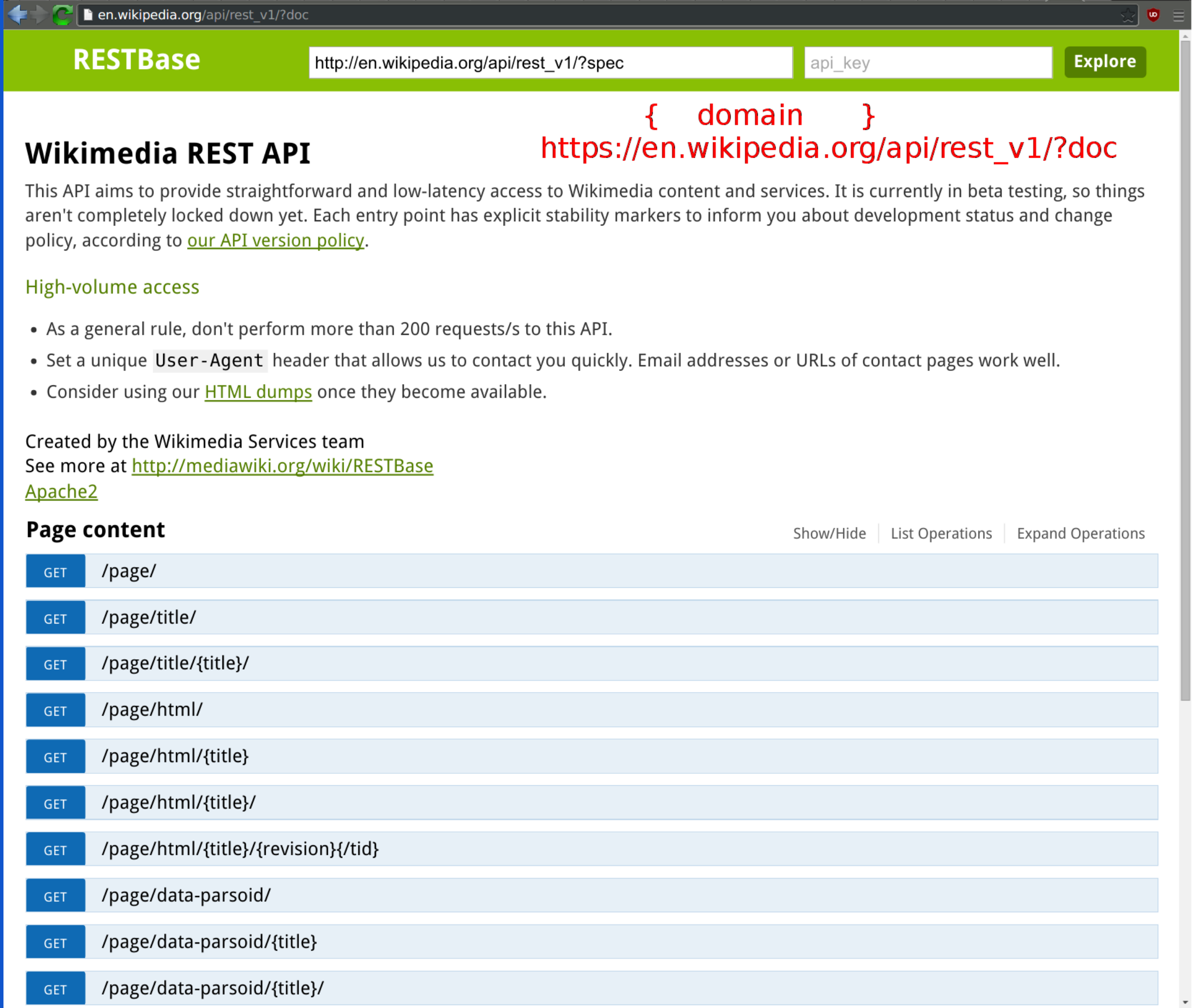
```
<span typeof="mw:Image" class="mw-valign-baseline"
   data-mw='{"caption":"cap<div></div>tion"}'>
 <a href="./File:Foobar.jpg">
  <img resource="./File:Foobar.jpg"
src="http://upload.wikimedia.org/wikipedia/commons/3/3a/Foobar.jpg"
      width="89" height="10">
 </a>
</span>
```

[[Image:Foobar.jpg|50px|border|caption]] (Note 2)

```
<span typeof="mw:Image" class="mw-image-border"
   data-mw='{"caption":"caption"}'>
 <a href="./File:Foobar.jpg">
  <img resource="./File:Foobar.jpg"
src="http://upload.wikimedia.org/wikipedia/commons/3/3a/Foobar.jpg"
      width="50" height="6">
 </a>
</span>
```

[[Image:Foobar.jpg|thumb|left|baseline|caption content]] (Note 3, Note 4)

```
<figure typeof="mw:Image/Thumb"
 class="mw-halign-left mw-valign-baseline mw-default-size">
```


en.wikipedia.org/api/rest_v1/?doc
RESTBase
http://en.wikipedia.org/api/rest_v1/?spec
api_key
Explore
{ domain }
https://en.wikipedia.org/api/rest_v1/?doc

Wikimedia REST API

This API aims to provide straightforward and low-latency access to Wikimedia content and services. It is currently in beta testing, so things aren't completely locked down yet. Each entry point has explicit stability markers to inform you about development status and change policy, according to our API version policy.

High-volume access

- As a general rule, don't perform more than 200 requests/s to this API.
- Set a unique `User-Agent` header that allows us to contact you quickly. Email addresses or URLs of contact pages work well.
- Consider using our HTML dumps once they become available.

Created by the Wikimedia Services team
See more at http://mediawiki.org/wiki/RESTBase
Apache2

Page content

GET /page/
GET /page/title/
GET /page/title/{title}/
GET /page/html/
GET /page/html/{title}
GET /page/html/{title}/
GET /page/html/{title}/{revision}{/tid}
GET /page/data-parsoid/
GET /page/data-parsoid/{title}
GET /page/data-parsoid/{title}/

https://en.wikipedia.org/api/rest_v1/?doc

Documentation

API:

https://{project}/api/rest_v1/?doc

DOM spec:

https://www.mediawiki.org/wiki/Parsoid/MediaWiki_DOM_spec

Let's try it out!

Building a wiki front end:

https://jsfiddle.net/cscottnet/ceshcL3u/

Going further...

Let’s make it an offline wiki!

https://jsfiddle.net/cscottnet/eqysbbq1/

Uses https://mozilla.github.io/localForage/ for local storage.

Where next?

- Download feedback, compression, controllable spider depth
- Sidebar and special fun features
- Editing support (embed VisualEditor)
- Offline editing
- Better UX
- Better article layout
- ...you tell me!

Parsoid is bidirectional

What about converting HTML to Wikitext?

https://jsfiddle.net/cscottnet/hfdmhp8L/

Batch: (alpha) HTML dumps

- http://dumps.wikimedia.org/htmldumps/dumps/
- XZ-compressed SQLite databases
- https://phabricator.wikimedia.org/T93396

```
CREATE TABLE data(
    title TEXT,
    revision INTEGER,
    tid TEXT,
    body TEXT,
    page_id INTEGER,
    namespace INTEGER,
    timestamp TEXT,
    comment TEXT,
    user_name TEXT,
    user_id INTEGER,
    PRIMARY KEY(title ASC, revision DESC)
);
```