Services / Parsoid Quarterly Review

Wikimedia Foundation - Q2 of FY14-15
Agenda

Team intro - 1-2 minutes
Parsoid - 10-12 mins
Services - 10-12 mins
Discussion - 5-10 mins
Team and staffing numbers slide

Subbu
Marc
C. Scott
Arlo
James, since Nov 17
Marko, since Dec 15

Services

FY 2013-2014
FY 2014-2015
FY 2015-2016
Q4
Q1
Q2
Q3
Q4
Q1
Parsoid
What we said + What we did
Background: High level objectives

- Work towards read views served by Parsoid HTML
  - Long term goal: replace PHP parser
- Continue improving editing support for VE (and other clients)
- Code cleanup, technical debt, testing infrastructure
- Research new applications
<table>
<thead>
<tr>
<th>Objective</th>
<th>Measure of success</th>
<th>Dependency</th>
<th>ETA</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrate with RESTbase</td>
<td>---</td>
<td>Services</td>
<td>Done</td>
<td>V2 API ready and tested with RESTbase</td>
</tr>
<tr>
<td>CSS based customization of Cite</td>
<td>Site-specific customizations replicated in CSS in modern browsers w/ fallback for</td>
<td>Community, Core, VE</td>
<td>End of Q3</td>
<td>CSS and HTML changes ready with customizations for 3 wikis; Final pieces being worked on for deploy</td>
</tr>
<tr>
<td>extension</td>
<td>older browsers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Support a subset of complex</td>
<td>Clean visual diff on previously dirty pages</td>
<td>---</td>
<td>Done</td>
<td>Finished by end of Dec; Deployed Jan 28, 2014</td>
</tr>
<tr>
<td>templates</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add stable id support</td>
<td>VE can switch between HTML and WT editing</td>
<td>---</td>
<td>Q4</td>
<td>Initial work done; Postponed to Q3/Q4</td>
</tr>
<tr>
<td>Objective</td>
<td>Measure of success</td>
<td>Dependency</td>
<td>ETA</td>
<td>Status</td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------</td>
<td>------------</td>
<td>------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Improvements to <code>&lt;nowiki&gt;</code> insertion <em>(Unplanned)</em></td>
<td>Fewer nowikis in tests, Fewer complaints from frwiki and other wikis</td>
<td>---</td>
<td>Done</td>
<td>Quote and link handling vastly improved</td>
</tr>
<tr>
<td>Use Promises API</td>
<td>---</td>
<td>---</td>
<td>Ongoing</td>
<td>First pass: API is promises-based</td>
</tr>
<tr>
<td>Improve tests</td>
<td>More tests; Fewer failures; Better coverage</td>
<td>--</td>
<td>Ongoing</td>
<td>More parser tests; more mocha tests; added code coverage monitoring</td>
</tr>
<tr>
<td>Fixes to core PHP parser <em>(Unplanned)</em></td>
<td>Improved compatibility between Parsoid and PHP parser</td>
<td>--</td>
<td>Ongoing</td>
<td>Improved link parsing and serialization</td>
</tr>
<tr>
<td>Objective</td>
<td>Measure of success</td>
<td>Dependency</td>
<td>ETA</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------------------------</td>
<td>------------</td>
<td>-----</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>Language variant support</td>
<td>Identical rendering as core parser; passing tests</td>
<td>---</td>
<td>Q3</td>
<td>Initial work done; Postponed</td>
</tr>
<tr>
<td>Maintain RT testing infrastructure</td>
<td>---</td>
<td>---</td>
<td>Q4</td>
<td>Some fixes to keep it operational; More work needed</td>
</tr>
<tr>
<td>Get wikilint production-ready</td>
<td>---</td>
<td>---</td>
<td>??</td>
<td>Postponed (not enough resources)</td>
</tr>
<tr>
<td>Research new applications</td>
<td>---</td>
<td>---</td>
<td>??</td>
<td>Postponed (not enough resources)</td>
</tr>
</tbody>
</table>
What we learned
What we learned

- Important to prioritize work based on Parsoid client requirements -- easy to get sucked into fixing edge cases..
- `<nowiki>` insertion is surprisingly hard to get right (to be robust, correct, and insert as few as necessary).
- Among mediawiki devs, CSS-based customization for Cite was not as controversial as we feared.
- We are spread a little too thin for all that needs to get done.
- Documentation & blog posts are important but always get de-prioritized.
Metrics & other key accomplishments
Key accomplishments

- Two major areas of rendering diffs fixed / close to being fixed.
- RESTBase integration completed.
- Regular deploys (1-2 times a week) without incident.
- Testing (Oct ’13 → today):
  - 84% code coverage (higher if tracing & debugging is also tested)
  - More parser tests (2% more tests; 23K+ total across 6 modes)
  - More passing tests (87.5% → 88.2%; lots of false failures)
  - RT testing improved
    - zero char diffs went from 85.2% → 85.28%
    - zero semantic diffs still at 99.84%
What’s next
Background: High level objectives

- Primary Focus: Supporting VE goals this quarter
- Push forward towards Parsoid-HTML read views
- Initiate work to properly scope output of templates
  - Easier to reason about for editors
  - Removes WYSIWYG surprises from VE
  - Enables incremental parsing
    - faster parsing
    - reduced load on API cluster
<table>
<thead>
<tr>
<th>Objective</th>
<th>Measure of success</th>
<th>Dependency</th>
<th>ETA</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support for upright images</td>
<td></td>
<td>VE</td>
<td>Q3</td>
<td>Various WIPs</td>
</tr>
<tr>
<td>Experiment improving efficiency of data-mw, data-parsoid encoding</td>
<td>Reduces raw size of HTML in common and pathological cases</td>
<td>VE and other clients</td>
<td>Q3, Ongoing</td>
<td>Several ideas to try</td>
</tr>
<tr>
<td>Implement stable id support</td>
<td>VE can switch between HTML and WT editing</td>
<td>VE, Services</td>
<td>Q4</td>
<td>WIP in place</td>
</tr>
<tr>
<td>Any additional API support required for data-parsoid stripping</td>
<td></td>
<td></td>
<td>Q3</td>
<td></td>
</tr>
<tr>
<td>Objective</td>
<td>Measure of success</td>
<td>Dependency</td>
<td>ETA</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>------------------</td>
<td>-------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Language variant rendering</td>
<td>Identical rendering as core parser; passing tests</td>
<td></td>
<td>Q3</td>
<td>Several patches up for review</td>
</tr>
<tr>
<td>CSS based customization of Cite extension</td>
<td>Site-specific customizations replicated in CSS in modern browsers w/ fallback for older browsers</td>
<td>Community, Core, VE</td>
<td>End of Q3</td>
<td>CSS and HTML changes ready with customizations for 3 wikis; Final pieces being worked on for deploy</td>
</tr>
<tr>
<td>Initiate work for template scoping</td>
<td></td>
<td>ECT, VE, Core</td>
<td>Q4</td>
<td>Several ideas and outlines in place; Initial discussions happened last 2 weeks</td>
</tr>
</tbody>
</table>
Asks
Asks

- Additional engineer(s)
- Community liaison:
  - Cite CSS styling
  - Wikitext linting / template fixes
- CSS help:
  - Reduce rendering diffs to allow image-based testing
  - Validate mobile rendering of HTML page views
  - Prepare for Parsoid-based “Printable page”
Services
What we said & what we did
<table>
<thead>
<tr>
<th>Objective</th>
<th>Measure of success</th>
<th>Dependency</th>
<th>ETA</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTBase v1 deployment</td>
<td>Test, stabilize and deploy RESTBase with basic Parsoid HTML &amp; metadata storage &amp; API</td>
<td>Parsoid, Ops</td>
<td>Q2</td>
<td>Delayed Deploy ETA: Mid-February</td>
</tr>
<tr>
<td>Performance</td>
<td>&lt; 15ms 95th for small (&lt;10k) resources, &lt;250ms 95th for 2mb resources</td>
<td>Before deploy</td>
<td></td>
<td>Achieved 93ms 95th percentile across 3½ day random read run over all of enwiki</td>
</tr>
<tr>
<td>Build the team</td>
<td>Hire at least two awesome engineers &amp; onboard them</td>
<td>end Q2</td>
<td></td>
<td>Done One more req open</td>
</tr>
<tr>
<td>Objective</td>
<td>Measure of success</td>
<td>Dependency</td>
<td>ETA</td>
<td>Status</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------------</td>
<td>------------</td>
<td>-----</td>
<td>-------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Mathoid roll-out</td>
<td>Mathoid SVG + MathML rendering mode in production</td>
<td></td>
<td>Q2</td>
<td>Achieved Released Math 2.0 Oct 23, worked with community</td>
</tr>
<tr>
<td>Work with Chris &amp; core team on requirements for auth service</td>
<td>Develop minimal API, clarify architectural background</td>
<td>Core</td>
<td></td>
<td>Done. Core plans still unclear, but MVP (PHP API authz &amp; CSRF end point) as fall-back.</td>
</tr>
<tr>
<td>Tooling and Service infrastructure</td>
<td>Work with ops &amp; release engineering on better deploy pipeline, guidelines for services</td>
<td>ops, release engineering</td>
<td>Q3+</td>
<td>Ongoing Some momentum on tooling at dev summit; exchange of ideas &amp; code between services, but no formal guidelines yet</td>
</tr>
</tbody>
</table>
Things we didn't originally plan

- Wikidata Query Service bootstrapping: proposed graph dbs & Titan, investigating public API using MQL
- Handled production issues (including security incident) with OCG / PDF renderer
  - Service isolation great for security
  - Apparmor fails unsafe
- Good amount of prep work for SOA track at dev summit
RESTBase background

● Swagger-spec-driven API proxy backed by Cassandra storage and internal services
● Focus on content, performance, enabling fast iteration for mobile & others
● Provide shared service infrastructure: consistent API with HTTPS & (eventually) SPDY, monitoring, logging, throttling, authorization, security checks (CSRF, sanitization) & -headers
● Product goals:
  ○ Speed up content API / Parsoid clients incl. VE
  ○ Enable section editing & micro-contributions on mobile
What we learned
What we learned: Collaboration, Org

- Working closely with ops was & is critical
  - hampered by lack of resources in ops, difficult to tackle deeper ops issues from our end (access, responsibility)
- Need third-party distribution strategy for SOA acceptance
- Dev Summit consensus: we should be moving forward wrt SOA
  - Discussions show general concerns around technical details
  - We lack a common language to discuss the problems that Services is/will be solving
- Need guidelines, requirements, and improved infrastructure for services: development, deployment, orchestration, API docs
What we learned: **Technical Side**

- Can slim down HTML to current mobile size, but need to strip data-mw too
- Researched strategy for efficient section editing / micro-contributions
  - granularity should probably be section-based; current element ids have ~20% size penalty
- Apps have need for flexible HTML rewriting (currently done on client, perf hit)
- Demand for HTML dumps from Kiwix, Google & others
Metrics & other key accomplishments
RESTBase random read latency (enwiki)

98ms 95th percentile max over three-day run at 1250 req/s, random enwiki reads; mean ~15ms
Testing & code coverage

CR on GitHub, Travis testing of full stack incl. Cassandra & Parsoid
API spec driven testing

- RESTBase: 81%
- RESTBase-cassandra: 83%
- Parsoid: 84%
What’s next
Our next challenges

● Support edit performance work with fast content API exposing lean HTML
● Support micro contributions & fast VE edits with section edit API
● Accelerate API and service development via generic solutions to common problems
  ○ consistent API, monitoring, logging, caching / storage, auth, throttling, validation, security headers
● Common guidelines for service development and deployment
● Encourage move towards state / stateless separation
<table>
<thead>
<tr>
<th>Objective</th>
<th>Measure of success</th>
<th>Dependency</th>
<th>ETA</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESTBase deployment</td>
<td>production deploy of RESTBase with basic Parsoid HTML and metadata storage</td>
<td>Ops</td>
<td>2015/02/15</td>
<td></td>
</tr>
<tr>
<td>VE speed-up</td>
<td>HTML size matching current mobile output (~ ⅓ current size on large pages); separate data-mw API; direct access to HTML without PHP API proxying</td>
<td>Parsoid, VE, Ops</td>
<td>End Q3</td>
<td></td>
</tr>
<tr>
<td>“Fast” section editing and retrieval on mobile and desktop.</td>
<td>Prototype section-level edit &amp; view API, collab with mobile &amp; VE</td>
<td>Mobile, VE</td>
<td>End Q3</td>
<td></td>
</tr>
</tbody>
</table>
Asks

● DevOps capacity
  ○ Distributed storage / Cassandra skills (also for WDQ)
  ○ SOA architecture / infrastructure planning & implementation

● RelEng capacity
  ○ Need to work out a solid distribution strategy for SOA acceptance