# 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

James, **since Nov 17**

Marko, **since Dec 15**

C. Scott

Arlo

Services

Gabriel

*FY 2013-2014*

Q4

*FY 2014-2015*

Q1 Q2 Q3 **Q4**

*FY 2015-2016*

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

| Objective | Measure of success | Dependency | ETA | Status |
|---|---|---|---|---|
| Integrate with RESTbase | --- | Services | Done | V2 API ready and tested with RESTbase |
| CSS based customization of Cite extension | Site-specific customizations replicated in CSS in modern browsers w/ fallback for older browsers | Community, Core, VE | End of Q3 | CSS and HTML changes ready with customizations for 3 wikis; Final pieces being worked on for deploy |
| Support a subset of complex templates | Clean visual diff on previously dirty pages | --- | Done | Finished by end of Dec; Deployed Jan 28, 2014 |
| Add stable id support | VE can switch between HTML and WT editing | --- | Q4 | Initial work done; Postponed to Q3/Q4 |

| Objective | Measure of success | Dependency | ETA | Status |
|---|---|---|---|---|
| Improvements to <nowiki> insertion **(Unplanned)** | Fewer nowikis in tests, Fewer complaints from frwiki and other wikis | --- | Done | Quote and link handling vastly improved |
| Use Promises API | --- | --- | Ongoing | First pass: API is promises-based |
| Improve tests | More tests; Fewer failures; Better coverage | -- | Ongoing | More parser tests; more mocha tests; added code coverage monitoring |
| Fixes to core PHP parser **(Unplanned)** | Improved compatibility between Parsoid and PHP parser | -- | Ongoing | Improved link parsing and serialization |

| Objective | Measure of success | Dependency | ETA | Status |
|---|---|---|---|---|
| Language variant support | Identical rendering as core parser; passing tests | --- | Q3 | Initial work done; Postponed |
| Maintain RT testing infrastructure | --- | --- | Q4 | Some fixes to keep it operational; More work needed |
| Get wikilint production-ready | --- | --- | ?? | Postponed (not enough resources) |
| Research new applications | --- | --- | ?? | Postponed (not enough resources) |

# 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

| Objective | Measure of success | Dependency | ETA | Status |
|---|---|---|---|---|
| Support for upright images | | VE | Q3 | Various WIPs |
| Experiment improving efficiency of data-mw, data-parsoid encoding | Reduces raw size of HTML in common and pathological cases | VE and other clients | Q3, Ongoing | Several ideas to try |
| Implement stable id support | VE can switch between HTML and WT editing | VE, Services | Q4 | WIP in place |
| Any additional API support required for data-parsoid stripping | | | Q3 | |

| Objective | Measure of success | Dependency | ETA | Status |
|---|---|---|---|---|
| Language variant rendering | Identical rendering as core parser; passing tests | | Q3 | Several patches up for review |
| CSS based customization of Cite extension | Site-specific customizations replicated in CSS in modern browsers w/ fallback for older browsers | Community, Core, VE | End of Q3 | CSS and HTML changes ready with customizations for 3 wikis; Final pieces being worked on for deploy |
| Initiate work for template scoping | | ECT, VE, Core | Q4 | Several ideas and outlines in place; Initial discussions happened last 2 weeks |

# 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

| Objective | Measure of success | Dependency | ETA | Status |
|---|---|---|---|---|
| RESTBase v1 deployment | Test, stabilize and deploy RESTBase with basic Parsoid HTML & metadata storage & API | Parsoid, Ops | Q2 | Delayed<br>Deploy ETA: Mid-February |
| Performance | < 15ms 95th for small (<10k) resources, <250ms 95th for 2mb resources | | Before deploy | Achieved<br>93ms 95th percentile across 3½ day random read run over all of enwiki |
| Build the team | Hire at least two awesome engineers & onboard them | | end Q2 | Done<br>One more req open |

| Objective | Measure of success | Dependency | ETA | Status |
| --- | --- | --- | --- | --- |
| Mathoid roll-out | Mathoid SVG + MathML rendering mode in production | | Q2 | Achieved<br>Released Math 2.0 Oct 23, worked with community |
| Work with Chris & core team on requirements for auth service | Develop minimal API, clarify architectural background | Core | | Done. Core plans still unclear, but MVP (PHP API authz & CSRF end point) as fall-back. |
| Tooling and Service infrastructure | Work with ops & release engineering on better deploy pipeline, guidelines for services | ops, release engineering | Q3+ | Ongoing<br>Some momentum on tooling at dev summit; exchange of ideas & code between services, but no formal guidelines yet |

# 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

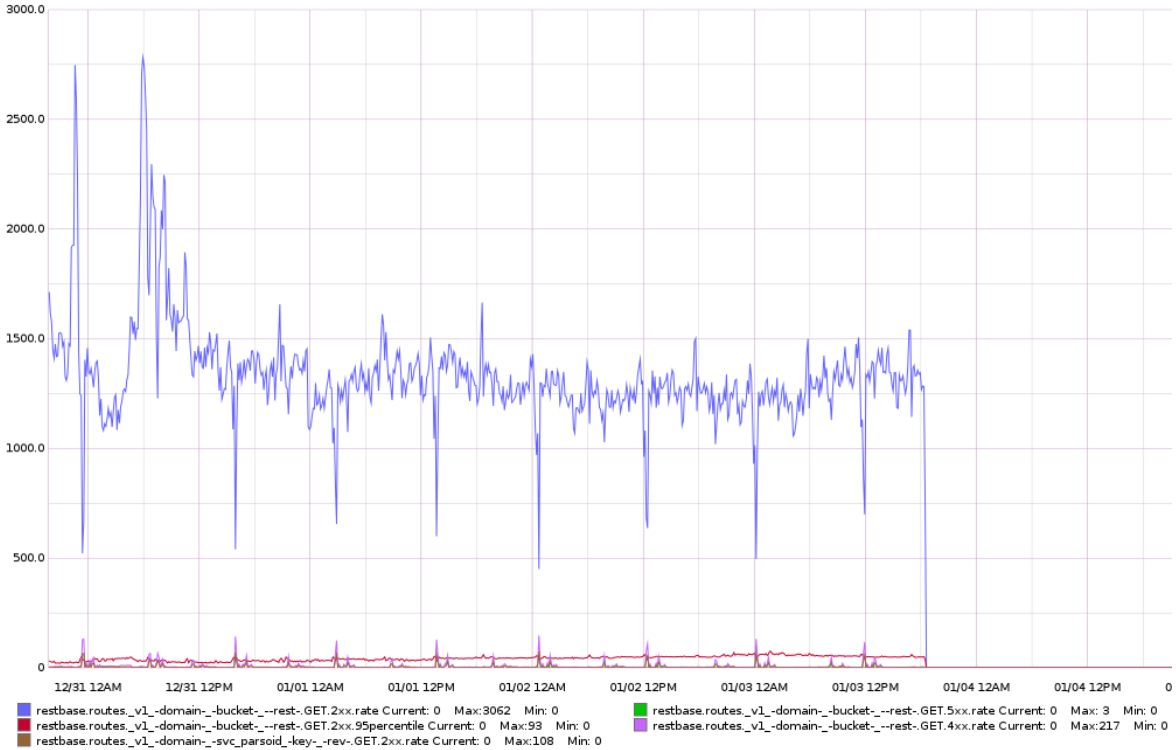
# What we learned: Technical Side

- 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

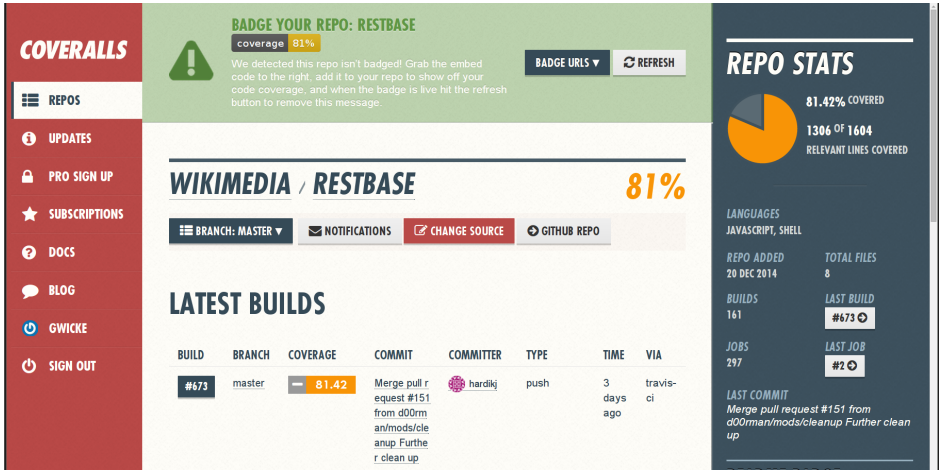


# Testing & code coverage

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

| Objective | Measure of success | Dependency | ETA | Status |
|---|---|---|---|---|
| RESTBase deployment | production deploy of RESTBase with basic Parsoid HTML and metadata storage | Ops | 2015/02/15 | |
| VE speed-up | HTML size matching current mobile output (~ ⅓ current size on large pages);<br>separate data-mw API;<br>direct access to HTML without PHP API proxying | Parsoid, VE, Ops | End Q3 | |
| “Fast” section editing and retrieval on mobile and desktop. | Prototype section-level edit & view API, collab with mobile & VE | Mobile, VE | End Q3 | |

# 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