The Technology Behind Wikidata

Daniel Kinzler

Wikimania 2013

Wikibase

- ContentHandler
- API-based Editing
- Sitelink Integration
- Statement transclusion
- Linked Data Interface
- Change Propagation
- Usage Tracking
- Queries

Wikibase

- Data Items
- Linked to pages on Wikipedia and Wikivoyage
- Have labels and descriptions
- Have statements about properties
- Data repository provides data to client wikis

ContentHandler

- Page content doesn't have to be wikitext
- We can still use page history, protection, watchlist, etc.

- ContentHandler provides handling for other kinds of data:
- Display
- Diffs
- Edit?

API Based Editing

- No direct editing of JSON
- Editing the whole page does not make sense
- Specialized tools for editing individual bits:
- JavaScript widgets using...
- Specialized API modules.

Sitelink Integration

- Automatic Wikipedia language links
- Local and automatic links are combined

- Currently:
- Wikibase ”pretends” the links come from the Wikitext
- When the links change, the page is re-rendered

- Future:
- Wikibase hooks in when language links are rendered
- No re-rendering, automatic links not in the database

Statement Transclusion

- Using values from wikidata in wikitext
- Parser function for the simple text
- Lua binding for more complex tasks
- No nice Lua library for high level access things yet

Linked Data Interface

- Linked data uses resolvable URIs
- Different URIs for things and descriptions:
- http://www.wikidata.org/wiki/Q42 Vs
  http://www.wikidata.org/entity/Q42
- Content Negotiation is used to serve the content in a format that the client can understand:
- http://www.wikidata.org/wiki/Q42 Vs
  http://www.wikidata.org/wiki/Special:EntityData/Q42.json Vs
  http://www.wikidata.org/wiki/Special:EntityData/Q42.xml Vs
  http://www.wikidata.org/wiki/Special:EntityData/Q42.rdf Vs
  http://www.wikidata.org/wiki/Special:EntityData/Q42.n3
- RDF mapping still incomplete (no statements)

Change Propagation

- Client wikis need to be notified about changes
- Changes are recorded in the database (like recentchanges)
- Dispatcher script pushes changes to client wikis
- The client wiki re-renders the appropriate page(s)

- Currently, pages are linked to items by a 1:1 relationship
- Only the items ”linked” page on the wiki is purged
- Labels of connected items are missed

Usage Tracking

- Access to any item from any client page
- We need to update the right pages when an item changes
- Tracking table on the client wiki
- Still need to check 800 or so wikis when dispatching
- A central tracking table would be huge

- Solution: track pages on the clinet wiki,
- track wikis on the repository

Queries

- Automatically generate lists
- Initially, for a fixed value of a single property
- No qualifiers, no intersection, no chains
- Initial implementation in MySQL
- Future: use (probably) elastic search for more complex queries
- Also support spacial queries for geo-coordinates (perhaps using PostGIS)

WIKIDATA