

# Best practices for reusing Wikidata's data in the Wikimedia Projects

**User:Mike Peel** 

30 May 2025

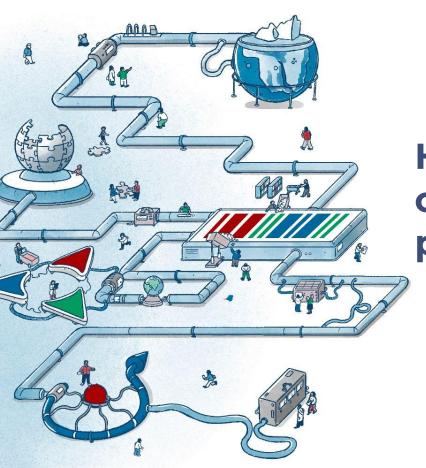




### Best practices for reusing Wikidata's data in the Wikimedia Projects

- How to transclude Wikidata content into Wikimedia projects
- Examples of current use
- The tricky parts...





## How to transclude Wikidata content into Wikimedia projects

## Getting content out of Wikidata

- Lua Scribunto interface.
  - https://www.mediawiki.org/wiki/Extension:Wikibase\_Client/Lua
- Can access directly in Lua codes, or in templates
  - {{#statements:P1476|from=Q191380}}will display "Notre-Dame de Paris"
- Module:WikidataIB flexible Wikidata module

#### Overview [edit source]

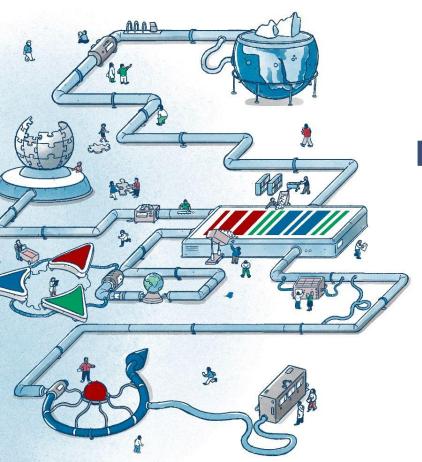
The module provides these calls specifically for use in infoboxes at present:

1. getValue - main call, used to get the value(s) of a given property

10. getCoords gets coordinates and passes them through {{Coord}}

- 2. getQualifierValue given: (1) a property; (2) its value; (3) a qualifier's propertyID, returns values which match
- 3. getValueByQual gets the value of a property which has a qualifier with a given entity value
- 4. getValueByLang gets the value of a property which has a qualifier P407("language of work or name") whose value has
- the given language code

  5. getValueByRefSource gets the value of a property which has a reference "stated in" (P248) whose value has the given entity-ID
- 6. getProp0fProp if the value(s) of prop1 are of type "wikibase-item" then it returns the value(s) of prop2 of each of those wikibase-items
- 7. getAwardCat if the item has values of P166 (award received), then it examines each of those awards for P2517 (category for recipients of this award) and it returns the corresponding category, with the item's P734 (family name) as sort key, or no sort key if there is no family name
- 8. getIntersectCat for each value of the prop1 it fetches the value's main category and then each value of prop2, then it returns all of the categories representing the intersection of those properties
- 9. getSum0fParts scans the property 'has part' (P527) for values matching a list. If the matched values have a qualifier 'quantity' (P1114), those quantities are summed and returned (but zero returns nil)



#### **Examples of current use**



## Wikidata Infobox on Wikimedia Commons

#### Wikidata and Sister Projects

[Collapse]



- Started in 2018, bot-deployed in 5.4 million Commons categories!
- Small infobox to add context for category contents, without hiding category contents
- Using Wikidata information only (no local definitions) - completely multilingual code: {{Wikidata Infobox}}
- Images on Wikidata shown alongside Commons category - are best images selected? (+ night image, map, ...?)
- Authority control at the bottom
- Continuous development with community input and suggestions















(when no Wikidata is sitelinked)

#### Wikipedia Infoboxes



- More complicated on Wikipedias
- Many different infoboxes covering different topics, different layouts etc.
- Consensus varies between Wikipedias (wide deployment on Catalan, tricky on English, others mixed).
- Some Wikipedias require referenced Wikidata information only (please add your references!)
- Automatically update when content is added to Wikidata. Add a picture on Wikidata -> it immediately appears in infoboxes across many languages.
- Improves cross-Wikimedia integration.

#### Wikidatapowered lists on Wikipedia (and other projects)

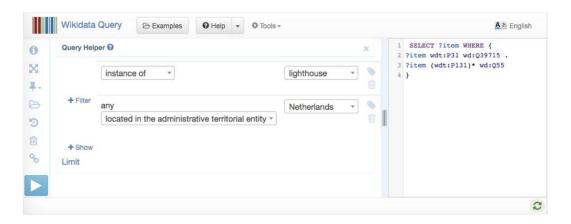
- Can quickly create lists using Wikidata information, based on Wikidata queries, that are transposed on (some) mainspace Wikipedias
- Whatever you can fetch using a Wikidata query, you can turn into a list (Listeriabot)
- As Wikidata updates, so does the list
- Works across different languages using the same dataset



### Wikidata queries

- SPARQL querying is powerful, but writing them in the right format can be tricky
  - Wikidata:Request\_a\_query can help!
- Only Wikidata content: can't edit the lists locally, and local changes will be lost

#### query.wikidata.org



## Cite Q for automatically generated citations

- Currently millions of Wikidata items for journal articles - part of Wikicite
- Can use this information to build structured references using Wikidata QID
- Can be used in any language!
- Automatically link to DOIs, author names, etc.
- Can auto-update all,
   e.g., preprint -> publication,
   or if a paper is withdrawn/replaced

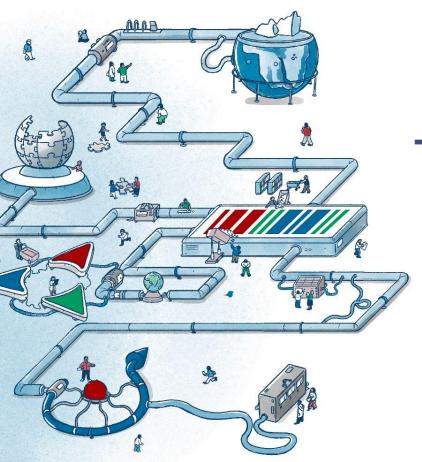
<ref>{(Cite journal|author = J. E. Carlstrom|title = The 10 Meter South Pole Telescope |journal = Publications of the Astronomical Society of the Pacific |volume = 123 |issue = 903 |pages = 568-581 |arxiv = 0907.4445 |display-authors=etal|year = 2011 |doi = 10.1086/659879 |bibcode=2011PASP.123..568C}}

#### <ref>{{Cite Q|Q56603073}}</ref>

^ a b J. E. Carlstrom; P. A. R. Ade; K. A. Aird; et al. (May 2011). "The 10 Meter South Pole

Telescope". *Publications of the Astronomical Society of the Pacific*. **123** (903): 568–581.

arXiv:0907.4445 ∂. Bibcode:2011PASP..123..568C ₺. doi:10.1086/659879 ₺. ISSN 0004-6280 ₺. Wikidata Q56603073.



#### The tricky parts...



#### The tricky parts... Technical

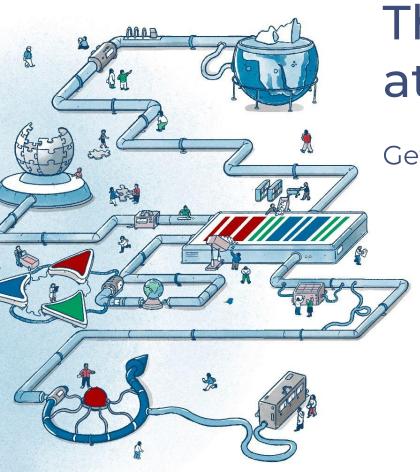
- Mostly solved, but complicated
- Built-in Lua functions get you part of the way
- WikidataIB gets you most of the rest
- Getting formatting right is often the most difficult thing
- #ifexist is finally fixed!
- Listeria is definitely a hack would be nice to include queries directly in articles...
- Wikidata Bridge to edit in-place would be nice



#### The tricky parts... Social

- The key part!
- Community has to want and support Wikidata integration - without this it's an uphill struggle.
- "Not made on this wiki" is a thing!
- Have to iterate with the community to make sure integration is working exactly as they want it
- E.g., Wikidata Infobox on Commons is collaboratively developed with the wider community and frequently iterated on!
- What works on one wiki might not work on another





### Thanks for your attention!

Get in touch:

**User talk:Mike Peel** 

Questions?

