



# Wikidata and OpenStreetMap

## Making our applications smarter

Akademy 2020

Volker Krause



[vkrause@kde.org](mailto:vkrause@kde.org)  
[@VolkerKrause](https://twitter.com/VolkerKrause)



# What do we need for applications?

- Code
- Content
- Infrastructure
- Data



## Examples in KDE

- Marble
- Musicbrainz use in media players
- Coordinate to timezone mapping
- Koko: Coordinate to location mapping
- KDE Itinerary (power plug compatibility checks, airport/station coordinates, public transport logos, etc)
- ...



## Data Sources

- Standalone databases
  - Often disconnected from each other, no unified identifiers, etc
  - Varying data formats and licenses
  - Often read-only
- Open Data communities
  - Wikidata
  - OpenStreetMap



## Wikidata

- “machine readable Wikipedia”
- 8 billion statements about 100 million objects
- 60 million media assets (Wikimedia Commons)
- CC0 licensed (excl. Commons)
- Very broad scope



## Wikidata – Data Model

- Qualified subject/predicate/object triples
- Subject: items (Qxxxxxx)
- Predicate: one out of ~9k properties (Pxxxx)
- Objects: primitive types, media asset or items
- Statements can be qualified (e.g. by when they were valid)
- Example: [Q1431](#) (KDE) P31 (instance of) [Q2989352](#) (free software community)



# OpenStreetMap

- 60GB of annotated geo-spatial data
  - 6 billion points, 700 million lines/polygons
- ODbL licensed
  - share-alike, attribution
- Only “now” is modeled (+ version control)



## OpenStreetMap – Data Model

- Three types of elements:
  - Nodes (points with coordinates, 100 nano-degree resolution)
  - Ways (ordered sequence of nodes)
  - Relations (sets of elements)
- Each element can be annotated with a large set of key/value pairs
- Cross-referencing with Wikidata items



How can we use this?



## Bundling Data

- Works for limited amounts of static information
- Prepare data locally and ship with application
- Allows for very efficient indexing/packing
- Sources:
  - Online query APIs
  - Derivative databases
  - Full data dumps



## Online Access

- Generic APIs of Wikidata/OpenStreetMap
  - Simple single-item access
  - Complex query services (SPARQL, OverpassQL)  

```
SELECT ?p WHERE { ?p wdt:P463 wd:Q1431. }
```
- [maps.kde.org](https://maps.kde.org): spatial-indexed raw data access
- Specialized services on own infrastructure or by 3<sup>rd</sup> parties



## Online Access - Considerations

- Privacy
  - High resolution coordinates, specific interests/activities, etc
- Complex query services are slow and cause a high server load, infeasible for most application use
- Follow Wikidata/OSM API access etiquette



## Existing Wikidata/OSM Code

- KItinerary:
  - SPARQL access to Wikidata for offline data preparation
  - QGIS generated z-order curve spatial indices
- KPublicTransport:
  - OSM access via full data dumps, OverpassQL, [maps.kde.org](https://maps.kde.org)
  - Wikidata/OSM cross-referencing
  - Wikimedia Commons use and license checks



## Conclusion

- There's lots of data we can use to make our applications smarter
- Interest in turning building blocks into libraries?
- Interest in moving data-based features like coordinate to timezone/country/region mapping to a Framework?



Questions?



# References

- Wikidata
  - Basic API: <https://www.wikidata.org/w/api.php>
  - SPARQL: <https://query.wikidata.org/>
  - Etiquette: <https://www.mediawiki.org/wiki/API:Etiquette>
  - Full database dumps: [https://www.wikidata.org/wiki/Wikidata:Database\\_download](https://www.wikidata.org/wiki/Wikidata:Database_download)
- OpenStreetMap API
  - Basic API: [https://wiki.openstreetmap.org/wiki/API\\_v0.6](https://wiki.openstreetmap.org/wiki/API_v0.6)
  - Overpass: [https://wiki.openstreetmap.org/wiki/Overpass\\_API](https://wiki.openstreetmap.org/wiki/Overpass_API)
  - Overpass Turbo: <https://overpass-turbo.eu/>
  - Full data dumps: <https://wiki.openstreetmap.org/wiki/Planet.osm>