Research Report

Easier Access for Programmers to Wikidata

Wikimedia Deutschland e.V.

Easier Access for Programmers to Wikidata—Intro

Executive Summary

Aims: We want to improve the data access for programmers and tool builders for the open community project Wikidata.

Methods: We studied the motivations, activities and problems of...
– 4 subject matter experts in open data
– 13 Programmers, 11 of them with experience in working with Wikidata.
Additionally, 89 community members participated in a survey to collect an overview of self-assessed skills in various data access technologies.

Findings:

– People need to know why they should use Wikidata in the first place. What Wikidata offers is different from platforms offering thematically curated data in tabular form, which was how most subject matter experts usually accessed data.

– Navigating the documentation for data access was a hurdle for both newcomers and even very experienced members of the Wikidata community. The main problem is not one of flawed content but of navigation and finding action-oriented, easy-to-try information.

– For people who want to become more involved and create more complex combinations of tools, it is most likely that learning from and with a community of like-minded people is beneficial. The difficulties to get into contact with others poses a relevant barrier of entry.

Content

Research Interests

We want to improve the data access for programmers and tool builders for the open community project Wikidata. Wikidata is a database storing general statements about things, e.g. "[Michael Müller] is [head of government] of [Berlin]". Everyone can edit these statements and improve them. Using stored data about other cities, you can get "All people who are heads of government of cities in Germany".

It thus is similar to Wikipedia, but instead of an article about something, there is a list of statements about a concept or thing, which makes Wikidata's data easier to use in computer programs: It is hard to parse an article text but it is relatively easy to process Wikidata's data and people already do this, e.g. Histropedia for showing timelines like this one: http://histropedia.com/timeline/466wbb666m/Discoveries-in-mathematics – based on Wikidata's data.

The research participants will be people who build digital tools which use semantic data. They will have different skill levels in different means to access the data (e.g. API use, SQL, Blazegraph...).

The research could be used to guide future work on Wikidata's APIs and other means of data access, the documentation and support for the community in self organization- and self-education, with particular focus on newcomers.

Research Methods

Our research was explorative and aimed to be able to describe and explain motivations, activities and problems of our participants when using Wikidata for their work.

Our data was collected in conversations with our participants. Since all participants were remote, we used tools like Google Meet or Skype for our conversations.

The interviews were semi-structured. We used an interview guide which listed the topics we wanted to explore. The interviews were 30min to 1h:30min long. The length of the interview depended on the mutually available time frame and the questions explored. In almost all interviews, two researchers were present: one person leading the interview, the other person taking notes.

After the interview, the data was pseudonymized as an additional safeguard (The notes were already taken pseudonymously, but given that they were written live, it can happen that e.g. names are written down instead of placeholders).

The data collection was shared between Jan Dittrich (Wikimedia Deutschland e.V.) and the design research agency futur2. While Jan Dittrich focused on research with developers, futur2 did collect the data with the subject-matter-experts.

This report is based on the initial analysis by futur2 and Jan Dittrich as well as on a re-reading and analysis of the original notes from the conversations with both programmers and subject matter experts.

Participants

We recruited participants from different groups.

Subject Matter Experts (SME): Futur2 recruited four Subject Matter Experts. They all were active in open data and activism using open data. Some of them could program, some not, though all had skills of working with data. They were picked to include people who do not use Wikidata yet but could be a potential target group and for giving a broader view of the open data ecosystem.
The Subject matter expert id-codes start with "SME-"

Developers: We recruited 13 developers to give insights in their current use of APIs. A large part (10 people) of the developers were recruited from the Wikidata community and thus had experiences with our existing APIs. All developers recruited from the community were male, despite attempts to get a more gender diverse representation (The people who answered in our survey were mainly male, too). Two of the developers did have no experience with Wikidata and its APIs at all. An additional developer had heard about Wikidata and used the API in the past, yet did not work extensively with it or would consider themselves part of the core community of the project. Most developers came from Europe. Two came from Africa, one from South America (See the corresponding section in the survey).

Findings: Data needs of potential data reusers

Participants look for data useful for the cause they have in mind

The subject matter experts had very specific causes they used data for like predicting food shortages, monitoring infection rates or government transparency. They evaluate data sources and tools for how much they can help them to support their causes.

The data they need can vary widely: Participants e.g. used Covid-19 case statistics, weather data, market prices of crops, street quality data and geodata for their purposes.

SME-B: "my work over the last few years, has specifically been in democratic alternatives such as citizens assemblies, and participatory budgeting"

The programmers who talked about why they use Wikidata partly also used Wikidata for specific needs related to a cause like getting primary data for a research project (P3), supplementing metadata for a research project (P9), getting a list of words with a certain end syllable for using them in a language-focused app (P1) or analysing Wikipedia contributions (P7).

Subject Matter Experts use many different search strategies and data sources

That people come with specific causes and ideas in mind also means that they are focused in their search for data. They, however, are not bound to a specific platform. They used different tools to find the data they need, e.g.

- search on github to find tools that use similar data to check their data sources
- use google to search for datasets
- check out open data government initiatives.

Some people settle with portals that provide data for their concern and continue to use them, e.g https://ourworldindata.org/ .

SME-P:" …government open data portal. You go and look at what the government offers if you are into government data. Then you have the, you know, the Open Knowledge Society, they have links."

It is not completely clear how people decide whether they find the data is trustworthy or not; the data indicated that several factors play a role like the provider of the platform and if it is seen to be open source and or non-profit and if the data is used or created by other organizations which participants know as trustworthy like the UN..

*SME-*M: *"I will get five links for CO_2 per capita, but which one is relevant which one is more testable so in order to see that, I will go through five of them, and then I will decide that which I will use"*

Table data is well known, graph data not so much

The subject matter experts we talked to used mainly tabular data and tools that work well on such data. This means they tried to find csv or Excel files, downloaded them and used them in their tools. After downloading the tabular data, they would use a tool like Excel.

SME-B.: "I mean, I've been an Excel hacker for over a decade"

SME-M: "I will download a CSV file and I will play with the data set or if I'm presenting I'll try to create interactive visualization or something like that."

While some of the subject matter experts could program, programming was only used some times and there were other tools that also were used in concert with the data like the database-like Airtable and graphic applications to build great-looking diagrams.

Semantic and graph-like data was not well known to participants. Some knew that it exists, but they could do without it.

*SME-*P: *"I didn't find it [Wikidata] relevant for my work. So I didn't really get into that I found more other sources that are more relevant for what I need to do"*

It is not clear how many of the developers from the Wikidata community benefited from the semantically networked structure of Wikidata or if they could have used table data equally well.

Findings: Accessing the Data from code

People do not find the documentation they need

People need to learn how to use Wikidata's API before they can successfully interact with it. However, even very tech savvy people – like the two programmers without Wikidata experience and even participants with Wiki-Experience – had trouble finding the needed information and got lost in linked pages between text on not-item-pages of Wikidata.org, pages on Mediawiki.org and the corresponding API documentation.

At least two participants described the API as overwhelming (P1, P5).

P1: *"intimidatingly many options, very off-putting"*

One person reported that while they were working with Wikidata for some time, they did only find out that there is an API and documentation for it far later (they used a software called *PyWikiBot* instead). Another participant told that they did learn about a very useful API call after several years active and sustained involvement by being told by another community member.

Participants with a longer experience in navigating the Wiki* documentation seemed not to get lost easily. The experienced people pointed out that they missed best practices or examples that would show established ways to solve common problems in regards to querying for data. Participants found examples helpful. This sentiment was shared by long term members as well as people without Wikidata experience (P1,P2,P4,P5, P11, P13). Particularly, the SPARQL examples got praise (P9, P13, P11).

The documentation content itself was not a problem and was sometimes described as "good" (P11, P2). While the *content* might not be a problem, the *finding* and *navigating* the documentation seems to be a big problem for beginners and experts alike.

Participants use many different tools and access different parts of the infrastructure

Participants combined a lot of different technologies and tools. They also preferred to use the tools they had at hand. There was rarely a "standard" way of doing something.

Python seems to be the most used language among our participants, but R and PHP and even Smalltalk were used, too. Data was accessed via SPARQL, API, Dump or PyWikibot[1]. There were some participants praising SPARQL, while others disliked it or found it complicated.

P1*: "I see SPARQL queries and am put off by this"*

P5*: "Wikidata without the query service [SPARQL] would not be much"*

P11*: "On Wikidata there is the "beautiful SPARQL system"*

The provided examples for SPARQL were always seen as good, though. Some participants needed to move away from SPARQL since it got too slow for their purposes, so they rewrote parts of their code to work with the API although they would otherwise have continued using SPARQL (P3, P5). One participant needed to do mass imports of huge amounts of data, for which tools like *quick statements*[2] were too slow, too.

The data that participants got through SPARQL, API or PyWikiBot was used in different ways – further processed by scripts in python or R, fed into local big data infrastructure, for web applications or dashboards to contribute back to Wikidata.

One participant (P5) also pointed out that for a successful creation of a useful service, you need knowledge of the data that is on Wikidata, its modeling and how your service will be used. If you want to build an app for showing the next hospital and you query for a list of hospitals you might provide totally useless results, since your list would also include hospitals that are defunct, historical or fictional, as P5 told.

[1] PyWikibot is a script collection for the programming language Python and uses the Mediawiki API
[2] Quick Statements is a data import tool, in which you define data in a line-based text format.

The heterogeneous tool system indicates that there are a lot of possibilities for interacting with Wikidata in the way that feels best for people. However, it also can make getting started hard, since everyone uses some different combination of tools. Particularly when querying performance becomes a bottleneck (for example with some SPARQL-queries) it is hard to know which solution or alternative tool might be helpful to use and how to make it work.

Communities and Knowledge Exchange

Exchange with other people in the community was important for our participants. Contact with others can play a large role when getting to know Wikidata (P11). Getting to know can also be supported by being part of related communities like being a Wikipedia Editor (P2, P7) or knowing DBPedia[3] (P3).

Other people are also relevant for learning about useful API functions (P5). One of the participants had direct contact with developers from the Wikidata team (P13).

The exchange seems to be rather distributed to various channels that you need to know and sometimes also be invited to. There are rather specific and local channels like special interest twitter or chat groups (P5); one participant exchanged mainly on the country's Wikimedia-Org chat group (P9). The two developers without Wikidata experience also mentioned github and stackoverflow as important sources for information. This makes sense as they could not know or quickly find out how to enter the Wikidata specific discussions, so those general platforms are a plausible starting point. None of the participants mentioned "project chat"[4] or the Wikidata mainlinglist[5].

Two developers with community experience mentioned that they are unsure about how to act on Wikidata and said they are unsure which actions are appropriate or not. P7 described this as uncertainty of how to "fit in", P9 framed it as being unsure about the right ways to edit and retrieve data as well as uncertainty about community policies.

While knowledge exchange in the community seems to be very important, there seems to be no core place where people go or a clear trajectory. Instead there is the use of many different platforms of which some are somewhat accessible via Wikidata (like the Telegram group[6]) while others are separate from it. On one hand this shows self organization and differentiation, on the other hand it probably makes getting into the community hard, since it is not obvious how to start exchanging with other community members. The pattern of a

[3] wiki.dbpedia.org is older (2007) than Wikidata (2012) and similar in what it offers. The data of DBPedia is generated from an analysis of Wikipedia articles.
[4] https://www.wikidata.org/wiki/Wikidata:Project_chat, an place for exchange based on Wikipages on Wikidata.
[5] https://lists.wikimedia.org/mailman/listinfo/wikidata
[6] Telegram is an instant message service similar to WhatsApp. Currently the group has about 500 members and a relatively high frequency of messages – I guess about 20-100 every day. One participant mentioned that this is too much to be useful for them.
https://t.me/joinchat/AZriqUj5UagVMHXYzfZFvA

bricolage- or tinkering-infrastructure is similar to the use of tools and programming languages.

Summary

To attract programmers to Wikidata, they first need to know about Wikidata and why they should use it for the ideas they have in mind. This is particularly important, as the way data is represented in Wikidata might not be familiar to potential users and it might only work for some of their use cases.

When people want to use the data on Wikidata and query for it, the largest barrier is finding the documentation they need to produce working code. People get lost in a mass of linked pages and can not find practical information like examples and best practices. The presentation and structure of the current API is not ideal, but once found, the content is seen as useful. The problem is not one of flawed texts but of navigation and finding action-oriented, easy-to-try information.

For people who want to become more involved and build more complex toolchains it is most likely that learning from and with a community of like-minded people is hard to achieve from the outside. How to interact with Wikidata in more complex setups as well as the channels for communication are hard to understand as both seem to have grown organically. People new to the community will most likely struggle with questions of best practices, finding good examples and places for exchange.

Appendix

Glossary of terms

- → **Items**: Wikidata consists of items. Items stand for a concept, so there is an item for "Mount Everest", an item for "Kilogram" etc.
- → **API**: In this document an API is a way to get data from a computer in a standardized way.
- → **SPARQL**: A special purpose programming language for querying data. It is particularly suited for databases structured like Wikidata.
- → **SQL**: A special purpose programming language for querying for data. It is particularly suited for databases that work with linked tables.
- → **REST**: A way to structure an → *API*
- → **Open Data**: "the idea that some data should be freely available to everyone to use and republish as they wish…The goals of the open-source data movement are similar to those of other "open(-source)" movements such as open-source software…"[7]
- → **Open Knowledge**: "Open knowledge (or free knowledge) is knowledge that one is free to use, reuse, and redistribute…The concept is related to open source"[8]
- → **Python, R, Smalltalk, Javascript**: Programming languages
- → **Dump**: A big file with data (from Wikidata) that can be downloaded.
- → **CSV**: A widespread file format for tabular data.
- → **Hadoop**: A software for working with large datasets. It can distribute the calculations to several computers.
- → **Spark**: An extension for →*Hadoop*, mainly for machine learning.

Survey

The survey was done independently from the qualitative research to assess who the current users of our APIs are and in which technologies they have experience.

- → Participants: 89
- → Data collection via google forms
- → We sent out calls for participation in the survey via the Wikidata mailinglist and the Wikidata weekly newsletter
- → Description of the survey:
  "The Wikidata team at Wikimedia Deutschland is interested in making it easier for people who write code to use Wikidata's data in their own applications and tools.

[7] https://en.wikipedia.org/w/index.php?title=Open_data&oldid=972455402
[8] https://en.wikipedia.org/w/index.php?title=Open_knowledge&oldid=970891242

Thus, we would like to learn more about the people who have accessed Wikidata's data from the code of their own applications and tools. We will use what we learn to improve the ease of access and usefulness of the ways in which we provide the data.

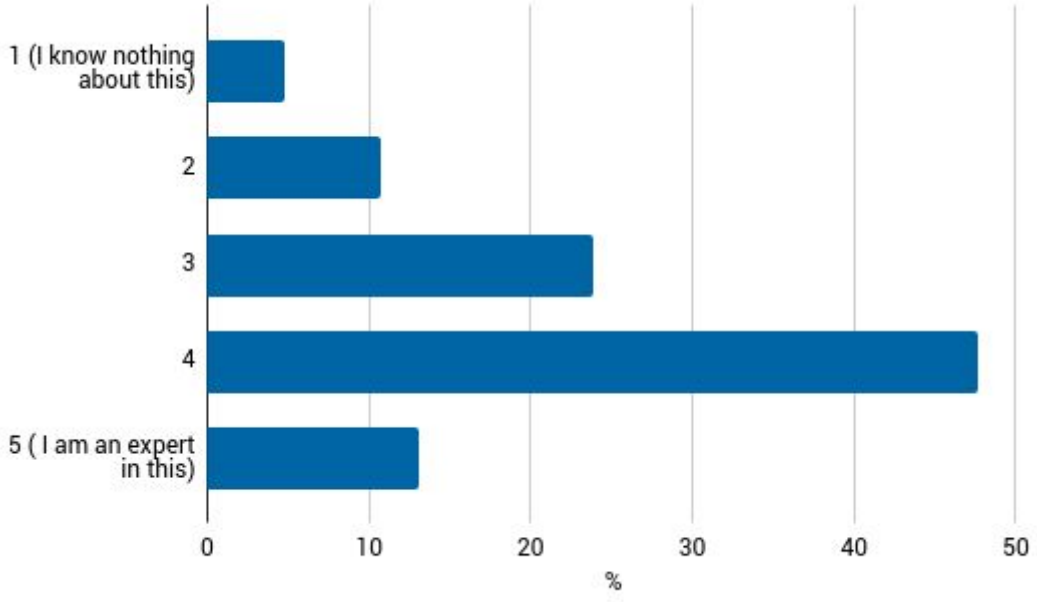
The participation is anonymous and there are no required answers. The data will only be shared in an aggregated form.

Time needed: 1-5min".

Question: How experienced are you with accessing Wikidata's data from code?

Accessing data means using e.g. the API, SQL, SPARQL or dumps to access Wikidata's data

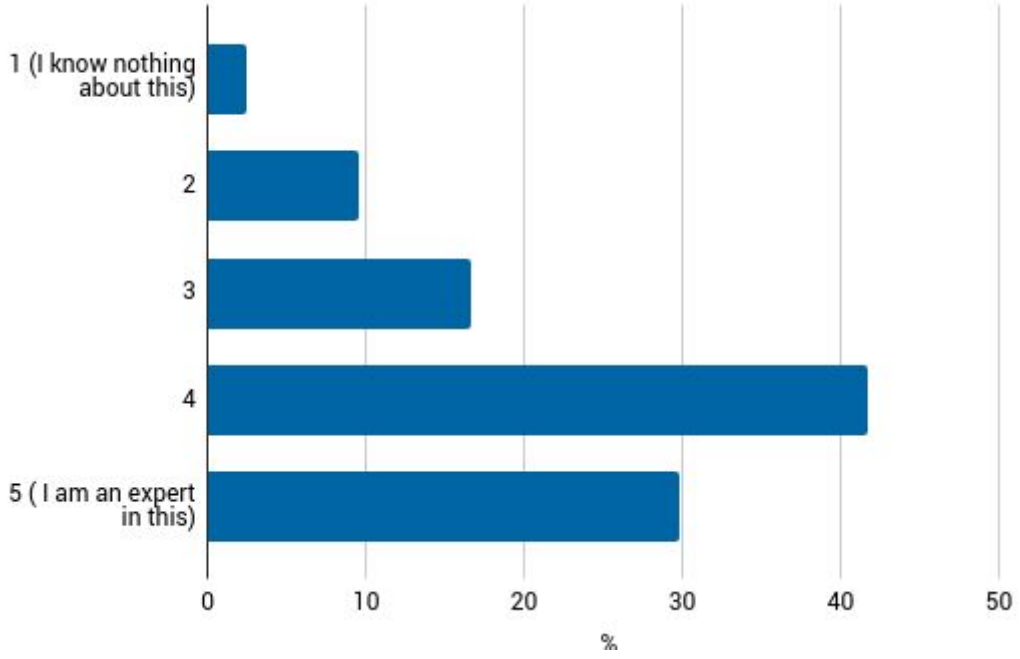
Results: (Horizontal axis is percentages)



Question: How experienced are you with accessing data from code in general?

"in general" means that this is about experiences on working with data of any project or product, not just Wikimedia ones. "Accessing data" means using e.g. APIs, SQL or web scraping

Results: (Horizontal axis is percentages)



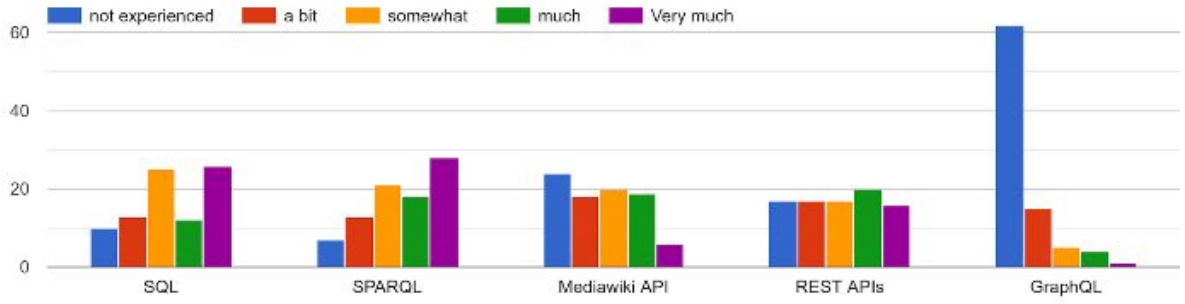
Question: How much experience do you have in using these technologies for accessing data in general?

"in general" means that this is about experiences on working with data of any project or product, not just Wikimedia ones. With "Mediawiki API" we mean the so-called "Action API":

https://www.mediawiki.org/wiki/API:Main_page

	not experienced	a bit	somewhat	much	Very much
SQL	○	○	○	○	○
SPARQL	○	○	○	○	○
Mediawiki API	○	○	○	○	○
REST APIs	○	○	○	○	○
GraphQL	○	○	○	○	○

Results: Vertical axis is absolute counts (not percentages!)



Question: What were your experiences in accessing Wikidata's data?

Tell us about what worked and what did not work well for you. Do you remember any specific obstacles that were hard to overcome?

Langantwort-Text

Results:

In their experiences, people described experiences and obstacles as free-form text.

Problems with response performance showing as lags (11 times mentioned) and timeouts (9 times) were mostly mentioned together with SPARQL. This is a topic that came up in the qualitative research interviews, see section "Participants use many different tools and access different parts of the infrastructure")

6 answers mentioned documentation, three of these six specifically pointing out the needed documentation is hard to find or navigate. This is also an experience participants in the

qualitative research interviews talked about, see section "People do not find the documentation they need".

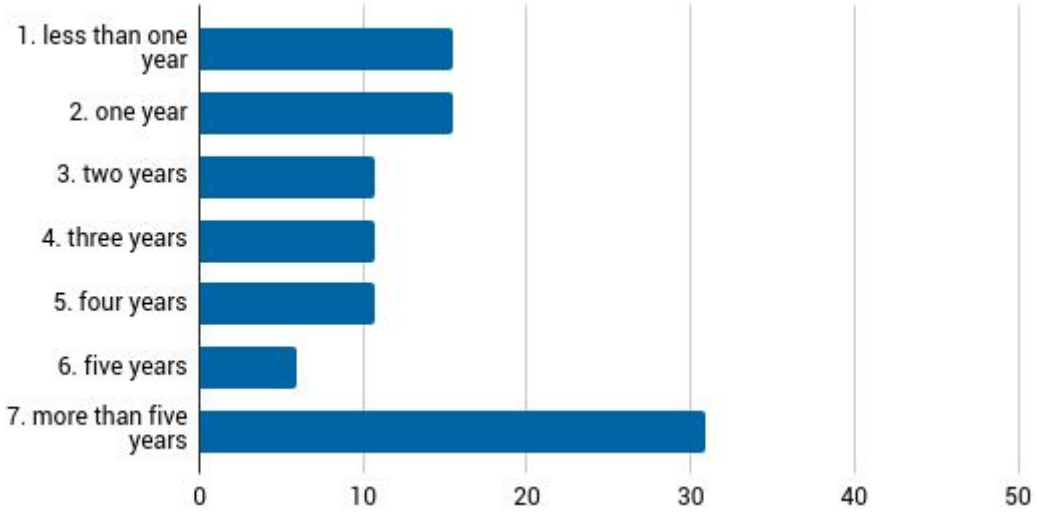
SPARQL is mentioned very often (24 times). However, this does not automatically mean that SPARQL is inherently causing many problems, as it seems to be a very frequently used method to deal with data among the participants – see the survey answers to "How much experience do you have in using these technologies for accessing data in general?" .

[On new page – About you]
Question: How long have you been active in the Wikidata Community?

- ○ less than one year
- ○ one year
- ○ two years
- ○ three years
- ○ four years
- ○ five years
- ○ more than five years

Results:

Horizontal axis is percentages



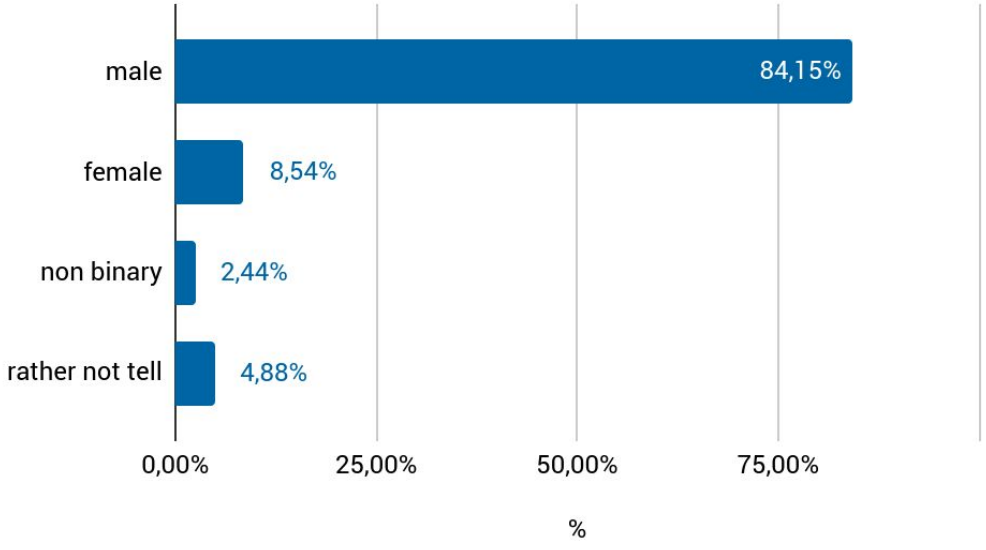
Question: Gender

If you prefer to self-describe, use the text input field ("other")

- female
- male
- non-binary
- prefer not to disclose
- Weitere...

[Note: Ideally, “weitere” or “others” would rather be named “prefer to self describe” as suggested in Spiel et.al.: How to do better with gender on surveys: a guide for HCI researchers, acm interactions, Volume 26, Number 4 (2019), Pages 62-65, however, in google forms the text is predefined.]

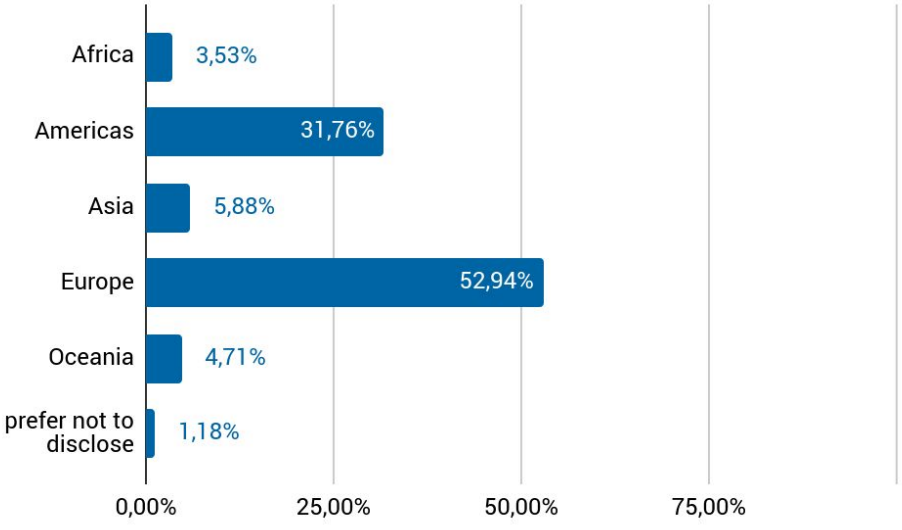
Results: (Horizontal axis are percentages)



Question: I currently live in…

- ○ Europe
- ○ Asia
- ○ Africa
- ○ Americas
- ○ Oceania
- ○ prefer not to disclose

Results:
(Horizontal axis are percentages.)



Authors and License

Research and Report was conducted and written by Jan Dittrich, Wikimedia Deutschland e.V., the futur2 critical design studio and the UX Team at Wikimedia Deutschland e.V.

This research would not have been possible without the support of our participants from the Wikidata community and beyond.

October 2020

Wikimedia Deutschland
Gesellschaft zur Förderung Freien Wissens e.V.

Postfach 61 03 49 / 10925 Berlin
Tempelhofer Ufer 23-24 / 10963 Berlin

Telefon: +49 (0)30 219 158 26-0
Telefax: +49 (0)30 219 158 26-9

E-Mail: info@wikimedia.de
Website: http://wikimedia.de
Blog: http://blog.wikimedia.de
Twitter: http://twitter.com/WikimediaDE
Facebook: http://facebook.com/WMDEeV