

This is the content of this Wikidata page:
https://www.wikidata.org/wiki/Wikidata:Data_access
retrieved on 26 Nov 2024

Wikidata:Data access

Wikidata currently contains over 110 million Items and over 1,3 million Lexemes, and these numbers will keep on growing. There are many methods available to access all that data—this document lays them out and helps prospective users choose the best method to suit their needs.

It's crucial to choose an access method that gives you the data you need in the quickest, most efficient way while not putting unnecessary load on Wikidata; this page is here to help you do just that.

Contents

- [1 Before we begin](#)
 - [1.1 Using Wikidata's data](#)
 - [1.2 Wikimedia projects](#)
 - [1.3 Data best practices](#)
 - [1.4 Access best practices](#)
- [2 Search](#)
 - [2.1 What is it?](#)
 - [2.2 When to use it?](#)
 - [2.3 Details](#)
- [3 Linked Data Interface \(URI\)](#)
 - [3.1 What is it?](#)
 - [3.2 When to use it?](#)
 - [3.3 Details](#)
 - [3.4 Less verbose RDF output](#)
 - [3.5 Revisions and caching](#)
- [4 Wikidata Query Service](#)
 - [4.1 What is it?](#)
 - [4.2 When to use it?](#)
 - [4.3 Details](#)
- [5 Linked Data Fragments endpoint](#)
 - [5.1 What is it?](#)
 - [5.2 When to use it?](#)
 - [5.3 Details](#)
- [6 Wikibase REST API](#)
 - [6.1 What is it?](#)
 - [6.2 When to use it?](#)
 - [6.3 Details](#)
- [7 MediaWiki Action API](#)
 - [7.1 What is it?](#)
 - [7.2 When to use it?](#)
 - [7.3 Details](#)
 - [7.4 Bots](#)
- [8 Recent Changes stream](#)
 - [8.1 What is it?](#)
 - [8.2 When to use it?](#)
 - [8.3 Details](#)

- [9 Dumps](#)
 - [9.1 What are they?](#)
 - [9.2 When to use them?](#)
 - [9.3 Details](#)
 - [9.4 Tools](#)
 - [9.5 Local query service](#)

Before we begin

Using Wikidata's data



Our logo

Wikidata offers a wide range of general data about everything under the sun. All that data is licensed [CC0](#), "No rights reserved", for the public domain.

Changes to APIs and other methods of accessing Wikidata are subject to the [Stable Interface Policy](#). Data sources on this page are **not guaranteed** to be stable interfaces.

Wikimedia projects

This document is about accessing data from outside Wikimedia projects. If you need to present data from Wikidata in another Wikimedia project, where you can employ parser functions, Lua and/or other internal-only methods, refer to [How to use data on Wikimedia projects](#).

Data best practices

We offer the data in Wikidata freely and with no requirement for attribution under [CC-0](#). In return, we would greatly appreciate it if, in your project, you mention Wikidata as the origin of your data. In so doing you help ensure that Wikidata will stay around for a long time to provide up-to-date and high-quality data. We also promote the best projects that use Wikidata's data.

Some examples for attributing Wikidata: "Powered by Wikidata", "Powered by Wikidata data", "Powered by the magic of Wikidata", "Using Wikidata data", "With data from Wikidata", "Data from Wikidata", "Source: Wikidata", "Including data from Wikidata" and so forth. You can also use one of our [ready-made files](#).



You may use the Wikidata logo shown above, but in so doing you should not in any way imply endorsement by Wikidata or by the Wikimedia Foundation.

Please offer your users a way to report issues in the data, and find a way to feed this back to Wikidata's editor community, for example through the [Mismatch Finder](#). Please share the location where you collect these issues on the [Project chat](#).

Access best practices

When accessing Wikidata's data, observe the following best practices:

- Follow the User-Agent policy – send a good User-Agent header.
- Follow the robot policy: send `Accept-Encoding: gzip, deflate` and don't make too many requests at once.
- If you get a 429 Too Many Requests response, stop sending further requests for a while (see the [Retry-After](#) response header)
- When available (such as with the Wikidata Query Service), set the lowest timeout that makes sense for your data.
- When using the MediaWiki Action API, make liberal use of the `maxlag` parameter and consult the rest of the guidelines laid out in [API:Etiquette](#).

Search

What is it?

Wikidata offers an [Elasticsearch](#) index for traditional searches through its data: [Special:Search](#)

When to use it?

Use search when you need to look for a text string, or when you know the names of the entities you're looking for but not the exact entities themselves. It's also suitable for cases in which you can specify your search based on some very simple relations in the data.

Don't use search when the relations in your data are better described as complex.

Details

You can make your search more powerful with these additional keywords specific to Wikidata: `haswbstatement`, `inlabel`, `wbstatementquantity`, `hasdescription`, `haslabel`. This search functionality is documented [on the CirrusSearch extension page](#). It also has its own [API action](#).

Linked Data Interface (URI)

What is it?

The Linked Data Interface provides access to individual entities via URI:

<http://www.wikidata.org/entity/Q???>. Such URIs are called concept URIs. Note concept URIs use HTTP, not HTTPS.

When to use it?

Use the Linked Data Interface when you need to obtain individual, complete entities that are already known to you.

Don't use it when you're not clear on which entities you need – first try searching or querying. It's also not suitable for requesting large quantities of data.

Details



Meet [Q42](#)

Each Item or Property has a persistent [URI](#) made up of the Wikidata concept namespace and the Item or Property ID (e.g., [Q42](#), [P31](#)) as well as concrete data that can be accessed by that Item's or Property's data [URL](#).

The namespace for Wikidata's data about entities is

<https://wikidata.org/wiki/Special:EntityData>.

Appending an entity's ID to this prefix (you can use `/entity/` for short) creates the abstract (format-neutral) form of the entity's data URL. When accessing a resource in the `Special:EntityData` namespace, the special page applies [content negotiation](#) to determine the output format. If you opened the resource in a browser, you'll see an HTML page containing data about the entity, because web browsers prefer HTML. However, a linked-data client would receive the entity data in a format like JSON or RDF – whatever the client specifies in its `HTTP Accept:` header.

For example, take this *concept URI* for Douglas Adams – that's a reference to the real-world person, not to Wikidata's concrete description:

<http://www.wikidata.org/entity/Q42>

As a human being with eyes and a browser, you will likely want to access data *about* Douglas Adams by using the concept URI as a [URL](#). Doing so triggers an HTTP redirect and forwards the client to the data URL that contains Wikidata's data *about* Douglas Adams: <https://www.wikidata.org/wiki/Special:EntityData/Q42>.

When you need to bypass content negotiation, say, in order to view non-HTML content in a web browser, you can specify the format of the entity data by appending the corresponding extension to the data URL; examples include `.json`, `.rdf`, `.ttl`, `.nt` or `.jsonld`. For example, <https://www.wikidata.org/wiki/Special:EntityData/Q42.json> gives you Item Q42 in JSON format.

Less verbose RDF output

By default, the RDF data that the Linked Data interface returns is meant to be complete in itself, so it includes descriptions of other entities it refers to. If you want to exclude that information, you can append the query parameter `?flavor=dump` to the URL(s) you request.

By appending `&flavor` to the URL, you can control exactly what kind of data gets returned.

- `?flavor=dump`: Excludes descriptions of entities referred to in the data.
- `?flavor=simple`: Provides only truthful statements (best-ranked statements without [qualifiers](#) or [references](#)), along with sitelinks and version information.
- `?flavor=full` (default): An argument of "full" returns all data. (You don't need to specify this because it's the default.)

If you want a deeper insight into exactly what each option entails, you can take a peek into the [source code](#).

Revisions and caching

You can request specific revisions of an entity with the `revision` query parameter: <https://www.wikidata.org/wiki/Special:EntityData/Q42.json?revision=112>.

The following URL formats are used by the user interface and by the query service updater, respectively, so if you use one of the same URL formats there's a good chance you'll get faster (cached) responses:

- <https://www.wikidata.org/wiki/Special:EntityData/Q42.json?revision=1600533266> (JSON)
- <https://www.wikidata.org/wiki/Special:EntityData/Q42.ttl?flavor=dump&revision=1600533266> (RDF, without descriptions of other entities)

Wikidata Query Service

Main page: [Wikidata:SPARQL query service/Wikidata Query Help](https://www.wikidata.org/wiki/Wikidata:SPARQL_query_service/Wikidata_Query_Help)

What is it?

The Wikidata Query Service (WDQS) is Wikidata's own SPARQL endpoint. It returns the results of queries made in the SPARQL query language: <https://query.wikidata.org>

When to use it?

Use WDQS when you know only the characteristics of your desired data.

Don't use WDQS for performing text or fuzzy search – `FILTER(Regex(...))` is an antipattern. (Use [search](#) in such cases.)

WDQS is also not suitable when your desired data is likely to be large, a substantial percentage of all Wikidata's data. (Consider using a [dump](#) in such cases.)

Details

You can query the data in Wikidata through our SPARQL endpoint, the [Wikidata Query Service](#). The service can be used both as an interactive web interface, or programmatically by submitting GET or POST requests to <https://query.wikidata.org/sparql>.

The query service is best used when your intended result set is scoped narrowly, i.e., when you have a query you're pretty sure already specifies your resulting data set accurately. If your idea of the result set is less well defined, then the kind of work you'll be doing against the query service will more resemble a search; frequently you'll first need to do this kind of search-related work to sharpen up your query. See the [Search](#) section.

Linked Data Fragments endpoint

What is it?

The Linked Data Fragments (LDF) endpoint is a more experimental method of accessing Wikidata's data by specifying patterns in triples: <https://query.wikidata.org/bigdata/ldf>. Computation occurs primarily on the client side.

When to use it?

Use the LDF endpoint when you can define the data you're looking for using triple patterns, and when your result set is likely to be fairly large. The endpoint is good to use when you have significant computational power at your disposal.

Since it's experimental, don't use the LDF endpoint if you need an absolutely stable endpoint or a rigorously complete result set. And as mentioned before, only use it if you have sufficient computational power, as the LDF endpoint offloads computation to the client side.

Details

If you have partial information about what you're looking for, such as when you have two out of three components of your triple(s), you may find what you're looking for by using the [Linked Data Fragments](#) interface at <https://query.wikidata.org/bigdata/ldf>. See the [user manual](#) and [community pages](#) for more information.

Wikibase REST API

Main page: [Wikidata:REST API](#)

What is it?

The Wikibase REST API is an OpenAPI-based interface that allows users to interact with, retrieve and edit items and statements on Wikibase instances – including of course Wikidata: [Wikidata REST API](#)

When to use it?

The Wikibase REST API is still under development, but for Wikidata it's intended to functionally replace the [Action API](#) as it's a dedicated interface made just for Wikibase/Wikidata.

The use cases for the Action API apply to the Wikibase REST API as well. Use it when your work involves:

- Editing Wikidata
- Getting direct data about entities themselves

Don't use the Wikibase REST API when your result set is likely to be large. (Consider using a [dump](#) in such cases.)

It's better not to use the Wikibase REST API when you'll need to further narrow the result of your API request. In such cases it's better to frame your work as a [search](#) (for Elasticsearch) or a [query](#) (for WDQS).

Details

The Wikibase REST API has [OpenAPI documentation](#) using [Swagger](#). You can also review the [developer documentation](#).

MediaWiki Action API

What is it?

The Wikidata API is MediaWiki's own Action API, extended to include some Wikibase-specific actions: <https://wikidata.org/w/api.php>

When to use it?

Use the API when your work involves:

- Editing Wikidata
- Getting data about entities themselves such as their revision history
- Getting all of the data of an entity in JSON format, in small groups of entities (up to 50 entities per request).

Don't use the API when your result set is likely to be large. (Consider using a [dump](#) in such cases.)

The API is also poorly suited to situations in which you want to request the current state of entities in JSON. (For such cases consider using the [Linked Data Interface](#), which is likelier to provide faster responses.)

Finally, it's probably a bad idea to use the API when you'll need to further narrow the result of your API request. In such cases it's better to frame your work as a [search](#) (for Elasticsearch) or a [query](#) (for WDQS).

Details

The MediaWiki Action API used for Wikidata is meticulously documented [on Wikidata's API page](#). You can explore and experiment with it using the [API Sandbox](#).

There are multiple Wikibase specific endpoints. Here are some example requests:

- [wbsearchentities search for New York, NY with limit=1](#) gives: [New York City \(Q60\)](#). This endpoint is very valuable because it is fast and can be used as a source of inference in tools and applications. For most well described items in Wikidata this endpoint works surprisingly well.

Bots



We welcome well-behaved bots

You can also access the API by using a bot. For more on bots, see [Wikidata:Bots](#).

Recent Changes stream

What is it?

The Recent Changes stream provides a continuous stream of changes of all Wikimedia wikis, including Wikidata: <https://stream.wikimedia.org>

When to use it?

Use the Recent Changes stream when your project requires you to react to changes in real time or when you need all the latest changes coming from Wikidata – for example, when running your own query service.

Details

The Recent Changes stream contains all updates from all wikis using the [server-sent events](#) protocol. You'll need to filter Wikidata's updates out on the client side.

You can find the web interface [at stream.wikimedia.org](https://stream.wikimedia.org) and read all about it on the [EventStreams](#) page.

Dumps

Main page: [Wikidata:Database download](#)

What are they?

Wikidata dumps are complete exports of all the Entities in Wikidata: <https://dumps.wikimedia.org>

When to use them?

Use a dump when your result set is likely to be very large. You'll also find a dump important when setting up your own query service.

Don't use a dump if you need current data: the dumps take a very long time to export and even longer to sync to your own query service. Dumps are also unsuitable when you have significant limits on your available bandwidth, storage space and/or computing power.

Details

If the records you need to traverse are many, or if your result set is likely to be very large, it's time to consider working with a database dump: ([link to the latest complete dump](#)).

You'll find detailed documentation about all Wikimedia dumps [on the "Data dumps" page on Meta](#) and about Wikidata dumps in particular on the [database download](#) page. See also [Flavored dumps](#) above.

Tools

- [JsonDumpReader](#) is a PHP library for reading dumps.
- At [\[1\]](#) you'll find a Go library for processing Wikipedia and Wikidata dumps.
- You can use [wdumper](#) to get partial custom RDF dumps.

Local query service

It's no small task to procure a Wikidata dump and implement the above tools for working with it, but you can take a further step. If you have the capacity and resources to do so, you can host your own instance of the Wikidata Query Service and query it as much as you like, out of contention with any others.

To set up your own query service, follow [these instructions](#) from the query service team, which include procuring your own local copy of the data. You may also find useful information in Adam Shorland's [blog post](#) on the topic.

This page was last edited on 30 September 2024, at 19:22.