OpenCollections Manual

Daniel Antal, CFA Ádám Lázár Andor Kornél Barát
Anna Márta Mester

2024-09-13

Table of contents

Introduction

Reprex's new `OpenCollections` system wants to help small and large enterprises work with big data without huge investments into data infrastructure. `OpenCollections` is an element of our collaborative toolkits that enables owners of small, local databases to remain competitive in training AI in the age of big data. It helps to fill your databases with up-to-date information, find and correct errors, and connect your database entries to new information as you need them without further IT and data investments.

The `OpenCollections` component of our solutions aims to interconnect inventories, collections, and repertoires. We want to enable private entities, like music distributors, rights management agencies, and film producers, to synchronise their IT systems with public GLAM memory institutions: archives, libraries, museums, and statistical agencies. We want to enable the enrichment of your inventory or repertoire management from interconnected databases to improve automated sales processes and the training or sales, inventory management or other AI algorithms.

Like many applications in the European open data field, `OpenCollections` is built around Wikibase. This open-source software system has built one of the world's most extensive knowledge graphs and knowledge bases, Wikidata, which synchronises the knowledge base of the 329 versions of Wikipedia with global databases, libraries, statistical authorities, company houses and other digital infrastructure.

This manual is not aimed at IT professionals or engineers. Wikibase has many thousands users with a simple and intuitive user interface. With this manual we are aiming for data stewards, data curators, librarians, archivists, inventory managers, who are responsible for documenting, updating repertoires, intellectual property assets, rights claims, webshop inventories, inventory management, and want to automate their processes, or train AI algorithms to do a better job for them.

Chapter 1 will need to be rewritten; it is currently taken from our observatory handbook, which deals with data collection programs, not broader collecting programs.

Chapter 2 is a very brief introduction to tidy data and text. It is a very brief introduction to keeping information tidy for automated computer use and easy database import.

Chapter 3 offers a typology of collections and the most prevalent problems with collections: ambiguous names, hard-to-translate descriptions, mismatched names and titles. Such problems appear in all large-scale applications and can negatively affect business, sales, legal or research integrity. We give some tips on how to work with our systems to prevent such problems

or to resolve existing collection management problems with automated data improvement, enrichment or updating.

Chapter 4 introduces Wikidata and other Open Knowledge Graphs. Using Wikidata, Wikipedia's document database, as an example, we show how to organise knowledge into a graph database and connect it with other knowledge graphs on the Internet of Data.

Chapter 5 introduces the adaptability of Wikibase and enterprise knowledge graphs that are tailored to your needs, and can handle highly confidential data.

Chapter 6 shows how to get familiar with the system in our Sandbox.

The creation of `OpenCollections` accounts is explained step-by-step in Section 6.1.

Wikibase has been open source for a long time, but it is in its infancy as a supported open-source product. `ReprexBase`, our distribution, is enhanced with know-how, and our software libraries help you manage this knowledge system to be tailored to your needs. Wikibase has been successfully used in many EU projects, including the creation of the EU Knowledge Graph (see: Section 5.5, (Diefenbach, Wilde, and Alipio 2021)). It also has training material on the EU Academy. While Wikibase is fully open-source and accessible, it is a complicated system that requires many extensions and adoptions to support a data-sharing space or a public-private knowledge base like ours. Reprex's extensions aim to make data importing and enrichment easier and less costly and make data export more reusable.

Using Wikibase allows coordination with Wikidata, which evolved into a central hub on the web of data and it is one of the largest existing knowledge graphs, and perhaps the best known open one. It is synchronised with knowledge from respected public institutions like Eurostat, the German National Library or BBC, and it is one of the backbones of many web services like Google Search. Wikibase *is scalable* to very big graphs.

1 Inspiration

Data curators, as professionals, are responsible for managing, maintaining, and enhancing the quality of an organisation's data. Their work is instrumental in making data easily accessible, accurate, and relevant to the organisation's needs. Large organisations collaborate closely with data and knowledge engineers, analysts, scientists, and other stakeholders to establish a robust data ecosystem.

Because of the dominance of micro- and small enterprise (institution) sizes in the music sector and many other creative industries and service sectors, very few competent data curators and specialised data or knowledge engineers are present. Our approach to solving this problem is to do the curatorial and engineering work collectively.

- ☒ We pool those music experts within the stakeholder network who have data curatorial skills (for example, music librarians) or, due to their job, have background skills or an affinity to data curation.
- ☒ We provide these data curators with robust tools that only require a little learning.
- ☒ We centralise all the knowledge and data engineering work in the centre of the data-sharing space, i.e., at the Open Music Observatory.

To become a data curator, you do not need to be a data scientist, statistician, librarian, or data engineer. We are looking for professionals, researchers, or citizen scientists who have deep subject-domain knowledge about the data we want to improve: they know a lot about organs in churches, about species of wild bees, music publishing, or any other domain on which we collect data. Our ideal curators share a passion for data-driven evidence or visualisations, can learn tools that Wikipedia editors use, and have a robust and subjective idea about the data that would inform them in their work.

1.1 We need data

1.1.1 No Data is Available: This Scientist Stung Himself With Dozens Of Insects Because No One Else Would

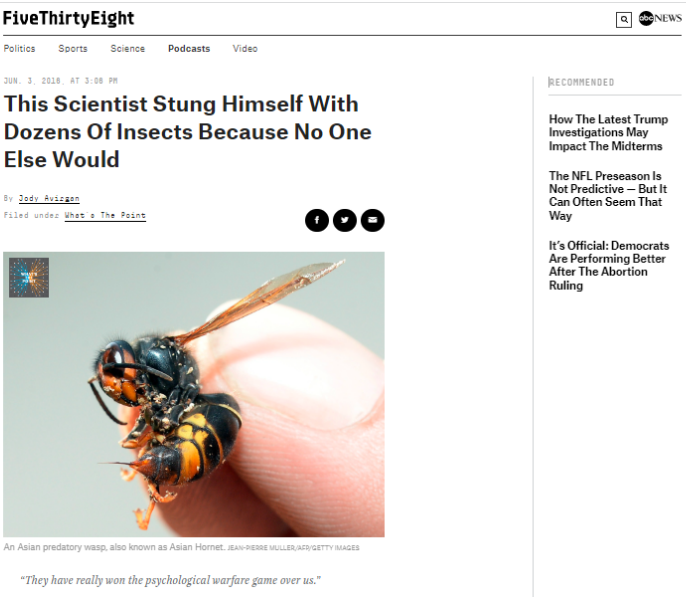


Figure 1.1: Good data curators are people who share a passion for measuring, recording and categorising the knowledge about their field, be it insects, music, or informal economy.

The `Schmidt Pain Index`, as its informally known, runs from 1-4. The common honey bee serves as its anchor point, a solid 2. At the top end of the scale lie the bullet ant and the tarantula hawk (which is neither a tarantula nor a hawk; it's a wasp). Watch the video with Dr. Schmidt, and listen to the whole interview here. This Scientist Stung Himself With Dozens Of Insects Because No One Else Would.

1.1.2 Nobody Counted Them Before: Big Data Is Saving This Little Bird

"We need to improve conservation by improving wildlife monitoring. Counting plants and animals is really tricky business." Big Data Is Saving This Little Bird

1.2 From Datasets and Files to Living Web Resoures

1.2.1 Web 3.0

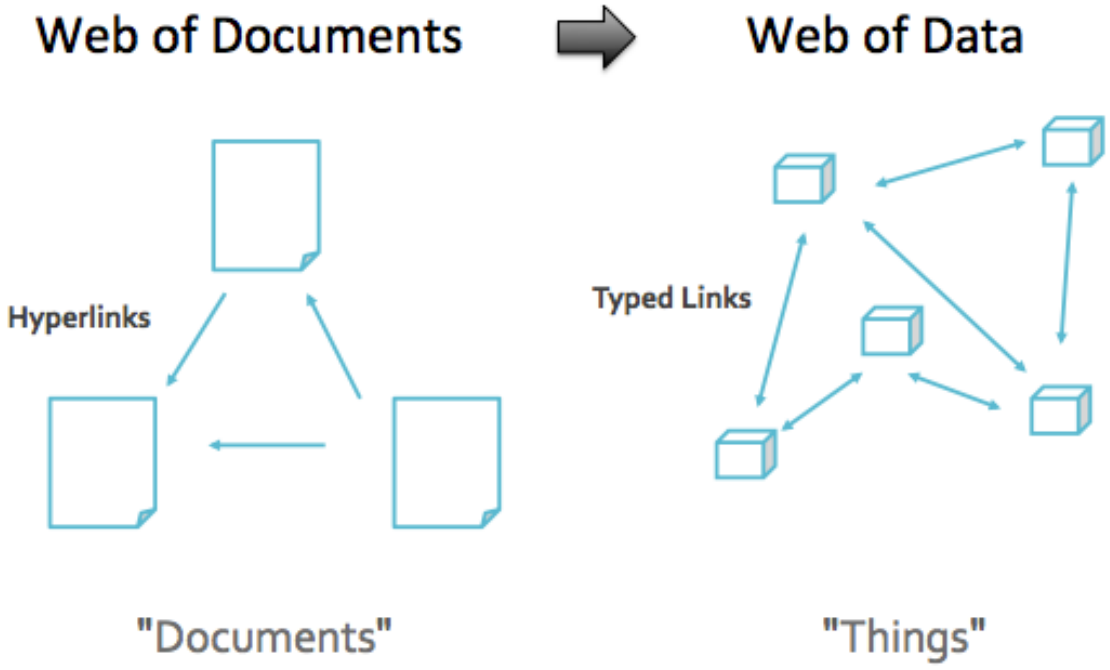


Figure 1.2: We are structuring your knowledge in a way that it results in datasets that can be connected similarly to the World Wide Web pages.

1.3 Remain Critical: Ethical Data, Trustworthy AI

Sometimes we put our hands on data that looks like a unique starting point to create a new indicator. But our indicator will be flawed if the original dataset is flawed. And it can be flawed in many ways, most likely that some important aspect of the information was omitted, or the data is autoselected, for example, under-sampling women, people of colour, or observations from small or less developed countries.

1.3.1 Machine Learning from Bad Data: Weapons of Math Destruction, Algorithms of Oppression

Cathy O'Neil: Weapons of math destruction, which O'Neil are mathematical models or algorithms that claim to quantify important traits: teacher quality, recidivism risk,

creditworthiness but have harmful outcomes and often reinforce inequality, keeping the poor poor and the rich rich. They have three things in common: opacity, scale, and damage. https://blogs.scientificamerican.com/roots-of-unity/review-weapons-of-math-destruction/](https://blogs.scientificamerican.com/roots-of-unity/review-weapons-of-math-destruction/)

In Algorithms of Oppression, Safiya Umoja Noble challenges the idea that search engines like Google offer an equal playing field for all forms of ideas, identities, and activities. Data discrimination is a real social problem; Noble argues that the combination of private interests in promoting certain sites, along with the monopoly status of a relatively small number of Internet search engines, leads to a biased set of search algorithms that privilege whiteness and discriminate against people of colour, especially women of colour.

1.3.2 Big Data Creates Inequalities: Data Feminism

Catherine D'Ignazio and Lauren F. Klein: Data Feminism. This is a much-celebrated book and with a good reason. It views AI and data problems from a feminist point of view, but the examples and the toolbox can be easily imagined for small-country biases, racial, ethnic, or small enterprise problems. A very good introduction to the injustice of big data and the fight for a fairer use of data, and how bad data collection practices through garbage in-garbage out lead to misleading information or even misinformation.

1.3.3 Bad Data collection Used for Modeling: Why The Bronx Burned

Why The Bronx Burned. Between 1970 and 1980, seven census tracts in the Bronx lost more than 97 percent of their buildings to fire and abandonment. In his book The Fires, Joe Flood blames the misguided "best and brightest" effort by New York City to increase government efficiency. With the help of the Rand Corp., the city tried to measure fire response times, identify redundancies in service, and close or re-allocate fire stations accordingly. What resulted, though, was a perfect storm of bad data: The methodology was flawed, the analysis was rife with biases, and the results were interpreted in a way that stacked the deck against poorer neighbourhoods. The slower response times allowed smaller fires to rage uncontrolled in the city's most vulnerable communities. Listen to the podcast here.

1.3.4 Bad Incentives Are Blocking Better Science

Bad Incentives Are Blocking Better Science "There's a difference between an answer and a result. But all the incentives are pointing toward telling you that as soon as you get a result, you stop." After the deluge of retractions, the stories of fraudsters, the false positives, and the high-profile failures to replicate landmark studies, some people have begun to ask: " Is science

broken?". Listen to the pdocast [Science is Hard]tttps://podcasts.apple.com/us/podcast/10-science-is-hard/id1011406983?i=1000391467935)

1.4 Reality Check

1.4.1 Looking Behind Data: Moving to America's Worst Place to Live

Christopher Ingraham wrote a quick blog post for The Washington Post about an obscure USDA data set called the `natural amenities index`, which attempts to quantify the natural beauty of different parts of the country. He described the rankings, noted the counties at the top and bottom, hit publish and did not think much of it. Almost immediately, he started to hear from the residents of northern Minnesota, who were not very happy that Chris had written, "The absolute worst place to live in America is (drumroll, please) … Red Lake County, Minn." He could not have been more wrong … a year later he moved to Red Lake County with his family.

2 Tidy work

, - . All happy families are alike; each unhappy family is unhappy in its own way.

Our `OpenCollections` systems are complex system which are intended to be used in trustworthy AI applications. They follow the Anna Karenina principle: a deficiency in any of a number of factors dooms an endeavour to fail. Consequently, a successful endeavour (subject to this principle) is one for which every possible deficiency has been avoided.

Once the data is messy, there is a semantic ambiguity (an ambiguity in the intended use or meaning of data) that will render automation impossible or will lead to logical faults when software agents or algorithms use your data. You must keep your numeric and text data tidy at all times. The best way to keep data and text tidy is to keep it simple. Very simple.

Simplicity is simple, if you start simple and keep it that way. Simplifying messy text and messy data is always challenging.

Collective work involving data and data annotations and descriptions requires a shared understanding of the syntax and file formats.

Our curators need to be familiar with two ideas.

- ☒ Tidy data means that tabular datasets are organised in a simple but particular manner. All observations are in rows, and all measured variables or characteristics are in columns, with no merged columns or rows. This is the optimal formatting for working with relational databases, and it is also a helpful start for graph databases. (See: Section 2.1.)
- ☒ Word processors like Word Work on different operational systems like Windows, MacOS, and Linux, creating very different text files and adding their formatting and other metadata to what you type. When we work together on the World Wide Web, we need something simpler than HTML but a bit more rich than a plain text file, clearly separating text editing from text formatting. The various markup notations, for example, *markdown*, are conventions for indicating that you want to make a text part **bold** or *italics* that works on all computer systems exactly the same way.(See: Section 2.2.)

2.1 Tidy data

Our data stewardship must follow the tidy data principle, which has very complex computer science and information management consequences, but for the curators of data, it boils down to an organised simplicity.

Tidy data is a standard way of mapping the meaning of a dataset to its structure. A dataset is messy or tidy depending on how rows, columns and tables are matched up with observations or collection items, and the measures and types of variables.

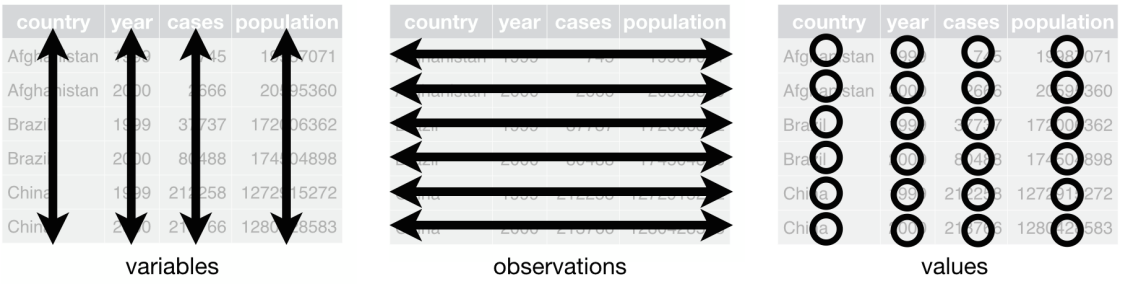


Figure 2.1: Following three rules makes a dataset tidy: variables are in columns, observations are in rows, and values are in cells. From R For Data Science - 12. Tidy Data

In tidy data:

- ☒ **1. Every table column is a variable**. We do not use colours (our machine-to-machine pipelines is colourblind). If we need comments or specifications, we add a new column.
- ☒ **2. Every row in the table represents an observation, or an individual piece of a collection.** Every variable belonging to `Bulgaria` is in the `Bulgaria` row, and there is one and only `Bulgaria row`.
- ☒ **3. Every cell is a single value**. Your blue male apron length is 97? The length column of the blue male apron row is 97.

> **Note**
>
> A `tidy dataset` is black-and-white, and each table cells contains one element of knowledge that cannot be further divided.

We repeat in negative terms these seamingly simple principles:

- ☐ **Two observed values: two columns**. We do not use colours (our machine-to-machine pipelines is colourblind). If we need comments or specifications, we add a new column. The apron has a variable length of 96-98? Use a column for min and max length, and in the cases when there is a single number, put 97 as both max length and minimum length. But never write 96-98 in on column, it must be either 96 or 98.
- ☐ **Two items or two observations: two rows**: Do you have two types of blue male aprons: write a separate line for the kitchen apron and the gardening apron. Or just call it apron 1 and apron 2. Do you have two responsdents in the apron customer satisfaction form: that will be two rows.
- ☒ **We never merge cells!** A tidy dataset has no divided or joined columns and no divided or joined rows. Never write 96-98 into a cell, because 96 goes to a separate cell than 98 because it has a different meaning: 96 means the minimum length of the apron, and 98 the maximum. Is there an apron with a length of 85 cm? That goes to a different row, because it refers to the length of a different type of apron.

Looks easy? If you start with a tidy table, it *is* very easy. If you have to tidy up a messy data table or an entire database, it often requires many years of data-wrangling experience to get it right first.

Is there science behind this? Yes, and it is more complicated than it sounds. In computer science or algebraic terms, you must organise your data to Codd's 3rd standard form. If you start from a well-organised table, it is a piece of cake to keep it that way. Reorganising messy information into a tidy format requires a lot of experience. Understanding that the ambiguity in the meaning of 96-98 should be resolved by treating them as two separate values, one meaning minimum possible length and the other maximum length, will not come naturally for everybody. But we will help in those cases.

2.1.1 Wide & Long Formats

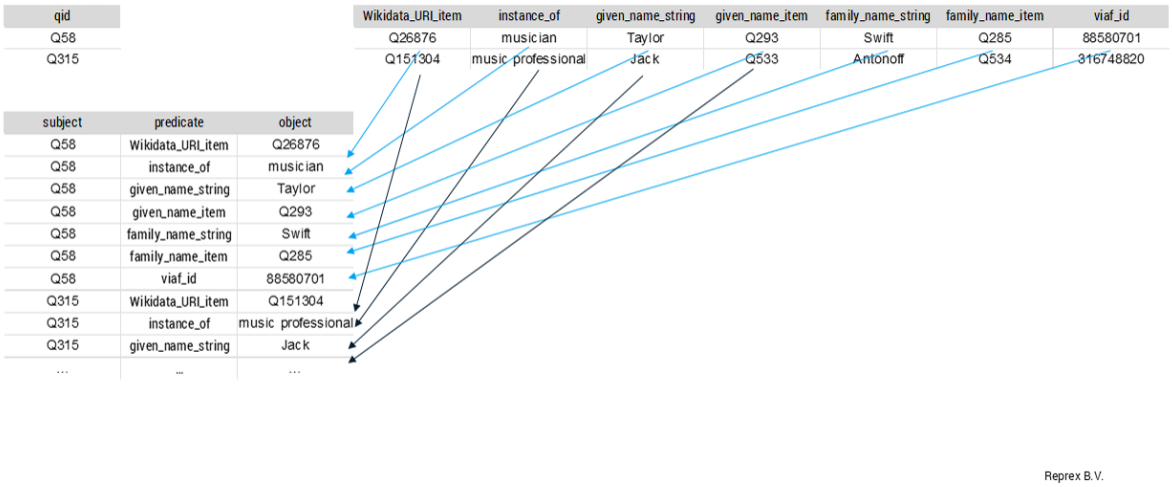


Figure 2.2: Tidy data tables can be pivoted: in this example a tidy wide-format data table is pivoted to a long-form table which has exactly three columns, a subject, a predicate and object, i.e., the semantic triples of knowledge management.

The tidy format is unambiguous: we always know that a number or string (value) belongs to its observational subject (in the rows) and the measured property variable (in the columns). Because the meaning is unambiguous, it can be transposed to different formats without loss of knowledge or misunderstandings.

Our knowledge base applications and Wikibase requires the three-column semantic triple format, because it can be organised into a graph; relational database managers usually prefer the wide format, because in this case every observed property of a data subject is in one record.

> **Note**
>
> A `tidy dataset` is black-and-white, and each table cells contains one element of knowledge that cannot be further divided.

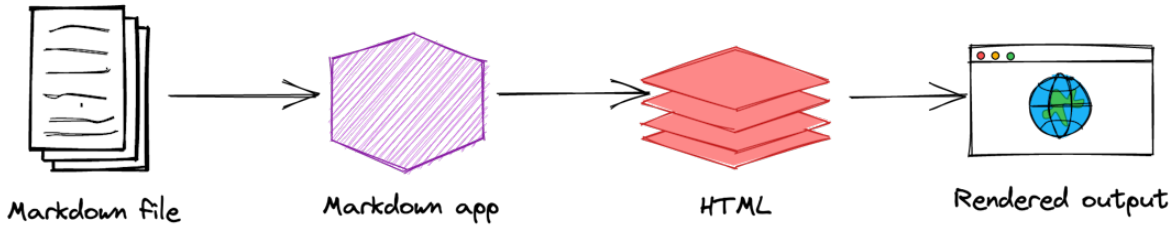
If your table is tidy, it will be easy to reuse in relational or graph database, or it can import easily into a spreadsheet or statistical program. Any further formatting with colours, divided columns, merged rows will stop the data portability, because only you will know why columns or rows are merged, divided, or coloured.

2.2 Markup text

We create interconnected, interoperable (web) resources. We want to ensure that our research results are findable, accessible, and reusable. It must work in Word and Works, Notebook and VIM, Windows, MacOS, and Linux, with Latvian, English, Greek, and Thai character sets.

The World Wide Web has been a source of high interoperability and findability in the last 30 years, with the introduction of the HTTP protocol and the standardization of the HTML text markup language. We use a much-simplified version of HTML called Markdown.

Markdown text opens on MacOS, Windows, or Linux. It is very easy to translate into HTML, Word, Libre Office, Google Docs, LaTeX, or PDF. Markdown is a simplified HTML text notation that works well with word processors.



If you want Word output, Word is rendered instead of HTML. You can also create a PDF or EPUB and even a PPTX output.

2.2.1 Markdown editors

There are countless Markdown editors. Because Markdown is so simple, you can, if you want to, edit markdown files in Notepad, WordPad (Windows) or VIM (Linux).

Most word processors support markdown. For example, Google Docs has a free extension that converts and document from Docs to markdown.

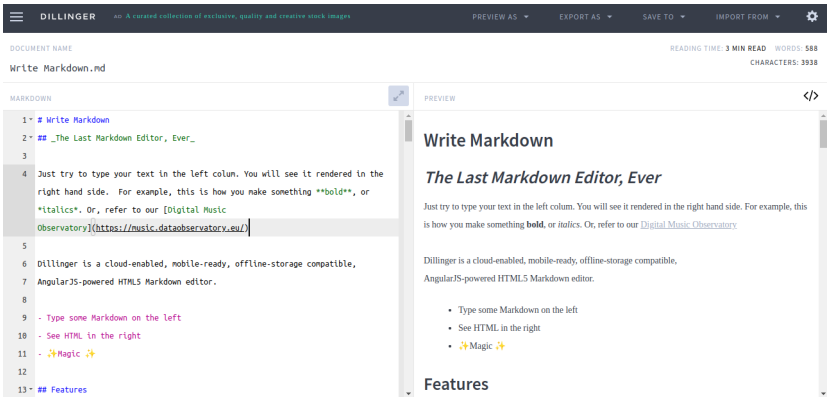


Figure 2.3: Dilinger is one of the best editors, and it is particularly suitable foor first-time markup users, as you immediately get visual feedback on how you mark up your text.

There are several online Markdown editors that you can use to try writing in Markdown. Dillinger is one of the best online Markdown editors. Just open the site and start typing in the left pane. A preview of the rendered document appears in the right pane.

- Basic Syntax
- Extended Syntax

> **Using Word or Works**
>
> You can work on Word, your iWorks suite, or any preferred word processor. However, you will lose margin settings, font typefaces and sizes, background colors, and other finishing touches.
> We discourage the use of word processors for footnotes and bibliographic references due to their varying treatment of such metadata. Our systems rely on standard BibLatex bibliographic references and a simple notation for footnotes, ensuring consistency and reliability.
> Our recommended markdown editor is Quarto. You can copy and paste text from Word or other word processors into Quarto, and it will retain **bold**, *italics*, and headings.
> Remember, we want to create text that machines and people can read, too, to avoid fancy aesthetics. Keep the text fancy (but of course, you can dress it up in Word or Adobe Illustrator later).

2.2.2 Wikipedia & MediaWiki

The documentation of our knowledge base and terminological agreements is documented in MediaWiki, the software that makes Wikipedia editable, too. It uses a form of markdown for

an interoperable and simple editing of interlinked documents, images, and data documents.

3 Collections

The term "collection" is elusive. Museums, libraries, and archives have never reached a consensus on what it means because there are so many ways and motivations to collect things. Physical collections often spoke for themselves. The idea of a digital collection made the process of collecting more abstract and fuzzy, as we can generate very large and heterogeneous collections.

Our manual aims to remain practical and seeks no definition of collection or curation. For our purposes, a digital collection is an organised set of digital artefacts organised by a curator or successive curators in institutions, following a more or less well-defined policy. The collection aims to preserve digital artefacts that users can use or consult, and the users need to find these artefacts: books, articles, photographs, historical dresses, sound recordings, stamps, and biographies. To retrieve individual items from the collection, we employ names (name titles) and identifiers, and to help with searching, browsing, and learning, we use categorisations in the collection.

3.1 Curator

Curators of physical collections have been recognised as professionals who search, acquire, preserve, research and communicate the individual items of collections to be preserved for further generation in musea: "…the notions of curation and curator to denote the person in charge of all tasks directly related to objects in a museum collection (i.e. their preservation, research, and communication) become firmly established in the English-speaking world only as late as the nineteenth century, their generalized use coinciding with the rise of museum professionalism." (Dallas 2016)

In the digital era, without the limitations of transport costs, storage space costs, temperature or lightning requirements, we can create much larger collections; on the Internet they can attract a global user base. Digital curation requires a reflection on the physical curatorial policies.

> "In a pragmatic approach, actors of digital curation include not just information professionals but also those involved in all aspects of the creation and reuse of a broad range of information objects. The latter comprise not just digital research data, static digital resources, and databases, but also derivations and performances

of such objects, and representations of domain knowledge, including indigenous and community based." (Dallas 2016)

3.2 Collection types

3.2.1 Playlists, repertoires, libraries

The archetype of libraries contains books organized by title, author and topic, or music libraries by title, author and genre.

Library-type collections use the Dublin Core metadata set, organized around titles, authors, and short descriptions.

> Libraries, playlist
>
> Your collection will likely depend on the Dublin Core, DataCite or Europena mandatory metadata fields. For example, to place an item of your collection into Europeana you must identify each item with a `title` and/or a `description`; in a library system you will use titles, often with subtitles or alternative (translated) titles.
>
> - ☒ You will use the names of author(s), like *Mark Twain*, or *John Lennon* and *Paul McCartney*.
> - ☒ You will use titles, like *The Adventures of Huckleberry Finn* and *Hey Jude* and *Symphony No. 2.*. Literary works and classical music works often have translated titles like *2. szimfónia*.
> - ☒ You will use publication or public release or copyright registration dates (or at least years.)
>
> Because there may be several identically named authors or titles (think about the Symphony No. 2.), you will need unique identifiers for your items.

Libraries suffer from name ambiguities and often name entity disambiguation. For example, many songs are called "Machinist," and many authors are called "James Campbell." Sometimes, names and titles need to be matched, which causes search errors, royalty payment errors, etc. See further details in the subsequent Section 3.3.1 part of this chapter.

3.2.2 Webshops, galleries, museums

Galleries and other exhibition places often show only (on the front page) a selection of diverse items available in your inventory. You do not only keep books or sound recordings but also

keep various items (merchandise, tote bags, etc.); the items need to be better described with author-title relationships; after all, who is the 'author' of a tote bag?

Gallery-type collections use a CIDOC-like information model for metadata and usually rely more heavily on thesauri to describe many different entities or things with a consistent language that is well understood by machines and people alike.

Webshops, galleries, museums

Your collection will likely depend on a broader conceptual model like CIDOC, and well-established controlled vocabularies like AAT.

- ☒ You will use titles, like *Mona Lisa* and *Ohne Titel*. Titles are often translated (*Without title*) or not useful for identification (like *Ohne Titel*).
- ☒ Because the title may not be a good identifier, you will use short descriptions, like Tour *T-Shirt Female Medium*, *Tour T-Shirt Male XL*, *Blue kitchen apron from the 19th century*. In such cases, the title may be a shorter version of the description *This wonderful Tour T-Shirt is available in blue, yellow, and green for women.*
- ☒ Various further information points on provenance may be recorded ("Designed in California", "Found in the Friesland region of the Netherlands") etc.

Good descriptions are essential because your users may look for very different items in your collections. Good descriptions can be easily translated from English to Dutch or Latvian, and machines can read them or translate them without error. You will focus on using keywords, keyword chains, or descriptions that come from a controlled vocabulary, a classification, or a thesaurus.
Unless your enterprise or organisation has its ontology, we will use CIDOC as a basis. CIDOC is a complex, event-based information model and you do not have to learn it. We need to ensure that the most important metadata about your collection is imported or entered into Wikibase so that we can export it, for example, into a CIDOC-compliant RDF.

The challenge with galleries is that they have to describe many things consistently and independently from natural languages. For example, a dress historian may use the color blue to describe a cooking aprons. How do we make sure that `blue`, `blauw`, `kék`, , or , labels are understood the same way, so that we can compare English, Dutch, Russian or Georgian collections? (See Section 3.6 later in this chapter.)

3.2.3 Documents, question banks, archives

Archives and document databases often contain millions of various documents or other records. Compared to libraries and galleries, individual collection items usually have a lower value and

a much lower level of documentation. An archive may contain millions of documents, but only a few may be interesting for our age or use case. Titles are often non-existent because the `document #3217454` is not very helpful for the user.

Archives emphasize the provenance of their collections. We may have thousands of emails, which must be those of a late novelist or a former CEO. If they are boxed, the origin of the box, when it was boxed, and other aspects of their recording history are the most important guides for the person who wants to find *that* email sent to the editor about the final changes in a novel, or the *final aproval* of an investment project.

Archives use the RiC conceptual model, ontology, or a metadata system on prior international archival standards.

3.2.4 Registers

Registers are collections that aim for completeness. They register every limited liability company in a jurisdiction, every copyright-protected musical work in a country, every living person, and every living musician in a city.

Registers can be library-like (for example, for copyright-protected literary or musical works), or more archival, for registering every birth and death certificate to create a population register. Like in the case of archives, data provenance is important. As opposed to archives, registers add new items and delete or make them obsolete; when people move away, companies are liquidated, or the copyright term of musical work expires.

> From business records to archives
>
> Your main challenge is that you have many very similar items in your collections, which are usually not very interesting and therefore researchers or curators do not spend time to individual describe and title them.
>
> - ☒ It is important to retain information about the record's structure: the letter has 3 pages, and the individual page is the 2nd of 3 pages.
> - ☒ Provenance is recorded with utmost care: the letters from the private drawer of the CEO, the private journal of the author, and the company's the counts in the year 1832.
> - ☒ Like libraries, our role is to connect people to the collection item, broadening the understanding of its significance. This connection is not limited to an author or editor role but extends to various roles such as project sponsor, judge, correspondence partner, sibling, etc.
>
> The international archival standards were modernised into RiC (Records in Context) for linking on the internet in 2023. We use the RIC ontology and conceptual model to work with archival documents. Our curators do not have to work with RIC directly in all cases,

but they must use OpenCollections in a way that records they record the key metadata of RIC. We will set up a Wikibase for you in a way that can be translated to RIC (and earlier archival standards.)

Registers can be formed around libraries, galleries, and archives, but they always have a time dimension, showing valid from and valid till date of every item.

3.3 Identifying, Naming, and Describing Collection Items

3.3.1 Naming people and indvidual things

When interacting with the world of persons, things, and relations, we use human language and name the persons and things. When naming people, for example, we use a first name or a full name. Names can be unambiguous or have a certain level of ambiguity that can be resolved in a context. In the United States alone, more than 38,000 men were named James Smith, and more than 32,000 women were named Maria Garcia in 2013 (**hartman_john_smith_et_al?**); identification by full name is an error-prone process.

`Taylor` is a unisex English name, and `Swift` is a family name that is not uncommon in English-speaking countries. The full name `Taylor Swift` name can refer to the American female superstar Taylor (Alison) Swift, the American male photographer Taylor Swift, or the event manager of Grand Hyatt New York, a woman who grew up in Missouri and used to sing in groups. (Newsweek: What It's Like to Be Named Taylor Swift in 2014)

Taylor M. Swift, woman from New York:

> Taylor Swift, New York: Facebook shut off my profile because they thought I was impersonating her. She must have been 15, so I was 18 or 19. She started to get popular and Facebook contacted me saying, "We are so sorry, but any impersonation of any kind is forbidden." I sing, too, and in college I was in a singing group and they thought I was literally impersonating her because people would write on my wall [about performances]. I had to send in three forms of ID. I think it took three-and-a-half weeks to get it back. Now my [Facebook] name is Taylor [middle name] because I can't have my first and last name on there… On my business cards, I have Taylor M. Swift.

Another Taylor Swift, a man from Seattle:

> Taylor Swift, Seattle: I get probably two or three emails [meant for Swift] a day. I've incorporated my middle name into my primary email, but I've held onto that one because why not?

The management of large collections and their databases requires unambiguous identification. It is avoidable that Taylor Swift, the photographer in Seattle, receives the royalties of the `Gold Rush` song; it is equally unacceptable that he cannot sell his photographs because his name is confused with the famous musician's namesake.

The names are replaced with a unique string in a database or an application that works with databases, like a museum inventory book, a copyright register, or a library catalogue. This string is often a string of numeric digits.

- `Uniqueness`: a given identifier must specify ("point to") one and only one person in the name space; in a personal record collection, there may not be identically named artist, however, in a global collection like the complete catalogue of Spotify, YouTube or Apple Music, there are many namesakes. With the ability to connect, link, join digital collections, names are less and less likely to be unique.
- `Persistence`: people's names are not permanent, and do not enable unambiguous specification of entities for an indefinite period. In many cultures, people change names when married (or divorced), particularly women; but there are many other reasons for a change of a person's name. In music and other arts, artist often use pseudonyms from a given time period.

Tips for people's names

- ☒ Try to record all name variants.
- ☒ Be aware of the differences of the Eastern and Western name order.
- ☒ Thrive to use global, unique, persistent identifiers.
- ☒ When there is no truly global identifier, create one in OpenCollections.

The `Eiffel Tower`, `Tour Eiffel`, `Eiffel-torony`, `Eiffeltoren` names refer to the same building in English, French, Hungarian and Dutch. While the building is individual, it has many names. Using a street address or the geocoordinates would be tempting; but street addresses keep changing. The geocoordinates do not show elevation (in case you would need the storey number), and there was *something* in another time, before the Eiffel Tower was built on the location of 48° 51' 29.1348'' North and 2° 17' 40.8984'' East. A popular location identifier, `geonames` identifies this famous building with 6254976; Wikidata uses the Q243 identifier.

The `Symphony No. 2` suffers from the same problem (it is `2. szimfónia` in Hungarian and `Symfonie nr. 2` in Dutch), but also from the fact that it is given to many musical works: it may refer to Opus 36 of Ludwig van Beethoven (Symphony No. 2 in D Major, Op. 36), or Symphony No. 2 in C Minor by Gustav Mahler, or Opus 73, Symphony No. 2 in D Major, by Johannes Brahms.

In collections, "information for display should be in a format and with syntax that is easily read and understood by users. This may be accomplished through data in the form of free

text or concatenated displays, allowing for the expression of the nuances of language necessary to relay the uncertainty and ambiguity that are common in art information." (Harpring and Baca 2016, p429) Most collection management system use a `title` and a `description` field to achieve this affect; titles and descriptions are used in library, archive and museum-type memory institutions. Software codes and information systems also need good names, and coming up with good names is often considered as the one of the most difficult task in computer science. (Allamanis et al. 2015)

Tips for individual names of things

- ☒ Choose a preferred name that is easy to read, and may be understood for most (or a plurality) of your users.
- ☐ It may not be possible to record all name variants; use the ones that may be relevant for your users.
- ☒ Thrive to use global, unique, persistent identifiers.
- ☒ When there is no truly global identifier, create one in OpenCollections.

3.3.2 Naming categories, groups of individual entities, and non-individual items

> When discussing art vocabulary for categorizing works of art, we are really talking about the controlled terminology used to *index* art works. For our purposes, *indexing* refers to a conscious activity performed by knowledgeable cataloguers who consider the retrieval implications of the indexing terms that they apply to information objects; we are not referring to an automated process that simply parses every word in a text into indexes, as search engines like Google do on the open Web. Controlled vocabulary for art refers to standardized words and phrases used to refer to ideas, physical characteristics, people, places, events, subject matter, and many other concepts related to art, architecture, and other cultural heritage. The most important functions of a controlled vocabulary are to gather together variant terms and synonyms referring to concepts, and to link concepts in a logical order or into categories. Are a *rose window* and a *Catherine wheel* the same thing? How is *pot-metal glass* related to the more general term *stained glass*? The links and relationships in a controlled vocabulary ensure that these relationships are defined and maintained, for both cataloguing and retrieval.(Harpring and Baca 2016, p426)

Information for display should be in a format and with syntax that is easily read and understood by users. This may be accomplished through data in the form of free text or concatenated displays, allowing for the expression of the nuances of language necessary to relay the uncertainty and ambiguity that are common in art information. In addition, certain key elements of information must be formatted to allow for retrieval, using controlled vocabularies where appropriate.

Tips for naming things

- ☒ Whenever possible, use an open, public, trusted controlled vocabulary or thesaurus to create generic names (“male shirt”)
- ☐ It is a good practice to use several thesauri, even though for usability a preferred (main) thesaurus may be preferred.
- ☒ Use the same controlled vocabularies to identify categories, subgroups, keywords.
- ☒ Thrive to use global, unique, persistent identifiers of the definitions of your controlled vocabulary.
- ☒ When there is no truly global definition, create one in OpenCollections.

3.4 Identifiers

“An identifier is an unambiguous label which specifies an entity. In computer science terms, an identifier is a name; the entities named occupy a specific domain of application,the namespace, and identify points in that namespace.” (N. Paskin 1999)

- `Uniqueness`: a given identifier must specify (“point to”) one and only one person or thing in the name space. If we work on the internet, then the identifier must be a globally unique string, because the name space can perpetually grow.
- `Persistence`: is permanence of naming, enabling unambiguous specification of entities for an indefinite period.

  > A numbering scheme is a formal standard, an industry convention, or an arbitrary internal system such as an incremented production serial number etc., to arrive at a consistent syntax for denoting and distinguishing separate members of a class of entities. […] The important point here is that the resulting number is simply a label string (a “noun”). It does not, of itself, create a string that is actionable in a digital or physical environment (a “verb”) without further steps being taken. It may be used (and probably will be used) in databases, or it may be incorporated into another mechanism later. (Norman Paskin 2003, 30–31).

Because modern IT systems can contain information about billions and billions of things, it is less and less desirable to only use the 0…9 numeric characters for this purpose, and often, a random string of alphanumeric characters is used. Many so-called hash applications ensure that even if you record billions of entities or transactions, they are given a unique string. Following Norman Paskin, it is a good distinction to consider these identifiers as a simple label string or a “noun”. `0000 0004 6613 4394` is simply a computer-language equivalent of Taylor (Alison) Swift; it is the International Standard Name Identifier for the said artist. In the universe of the Spotify music platform, the string 06HL4z0CvFAxyc27GXpf02 identifies the same famous artist.

☒ A library catalogue contains information about books. Books are usually identified by title, author name, publisher, and publishing data because often the same library has many James Campbells or similar-titled books, etc. A unique global identifier is the International Standard Book Number.

☒ A music playlist contains sound recordings. The recordings are often referred to by the name of the performer(s) and the title of the music work that they perform; however, in global systems, we may have dozens of same-name performers and even hundreds of same-title works (just think about Symphony No.2!). Instead, we can identify the performers with the ISNI International Standard Name Identifier and the recordings with the Spotify Track ID or the ISRC International Standard Recording Code.

☒ A dress history database may identify specimens of shirts and aprons; as there may be many similar aprons, they usually do not have a specific name. Instead, they are either identified with a generic name, like `Male apron from the 19th century`, or by an inventory number.

i Note

The most common standard numbering schemes of interest in digital rights management and digital asset management include

- ISBN: International Standard Book Numbering (ISBN)
- ISSN: International Standard Serial Number (ISSN)
- ISRC: International Standard Recording Code (ISRC)
- ISRN: International Standard Technical Report Number (ISRN)
- ISMN: ISO 10957:1993 International Standard Music Number (ISMN)
- ISWC: ISO 15707:2001 International Standard Musical Work Code (ISWC)
- ISAN: Draft ISO 15706: International Standard Audiovisual Number (ISAN)
- ISTC: Draft ISO 21047: International Standard Text Code (ISTC)

3.4.1 Actionable identifiers

Paskin calls identifiers that can initiate an action in a digital or physical environment actionable identifiers, similar to verbs.

If in your home database, `artist-0001` refers to Taylor Swift, it is just a "noun", a replacement of Taylor Swift. However, 0000 0004 6613 4394 and 06HL4z0CvFAxyc27GXpf02 are actionable. Clicking https://isni.org/isni/0000000078519858 informs you via your browser or your library system by sending a package of standard metadata that this woman is not Taylor M. Swift from New York or the Taylor Swift, the photographer from Seattle. Similarly, https://open.spotify.com/artist/06HL4z0CvFAxyc27GXpf02 allows you to check out and even listen to all the released songs of the most famous Taylor Swift.

3.4.2 Local and global identifiers

`T Σ ,` both stand for "Taylor Swift" with different character sets and `Teilora Svifta` is a Latvian version of the same name. We can say that they are suitable in a Greek, Georgian or Latvian database. Similarly, database management systems provide (local) unique identifiers for every CD or music sheet of the author.

If in your home database, `artist-0001` may refer to the same artist. The problem with connecting databases and exchanging information about the the artist known as "Taylor Swift" is to ensure that `artist-0001`, `Teilora Svifta` is exchanged with data about 0000 0004 6613 4394, or 06HL4z0CvFAxyc27GXpf02, or , and not the photographer Taylor Swift or any other person.

`Taylor Swift` is a name, not an identifier. In most contexts, it correctly identifies Taylor M. Swift, Taylor Swift, and Taylor Alison Swift, but there are mistakes.

- 06HL4z0CvFAxyc27GXpf02 is a local but public identifier. It works only in the Spotify universe, but you can check that any music connected to `06HL4z0CvFAxyc27GXpf02` is performed by Taylor Swift.
- 0000000078519858 is a global identifier because the ISNI consortium ensures that nobody will ever get the same identifier again; furthermore, the identifier follows an international standard and remains forever open.

Global identifiers aim to work across databases; they are not specific to your computer system or a specific library catalogue. The use of global identifiers is essential to making various databases, data carriers, or their systems interoperable.

The line between 06HL4z0CvFAxyc27GXpf02 and 0000000078519858 is blurred. Both can be used almost all over the world, and the basic services of 06HL4z0CvFAxyc27GXpf02 are free. Spotify offers plenty of relevant music metadata and statements for free via its web player and its open API about Taylor Swift.

3.5 Identifiers and metadata

> The most common—and perhaps least useful—definition of metadata is that it is "data about data." As catchy as this definition is, however, it is entirely ambiguous. First of all, what is data? And second, what does "about" mean? (Pomerantz 2015, p19)

We use the definition of Pomerantz about metadata. The new ISO standard on Information technology — Metadata registries (MDR) defines *metadata* as data that defines and describes other data. As Pomerantz eloquently argues, this definition is not very helpful. We use his more functional (but not contradictory) definition. "Data is only potential information, raw

and unprocessed, prior to anyone actually being informed by it. [...] Data must be understood not as an abstract concept but as objects that are potentially informative. [...] Metadata Is a Statement about a Potentially Informative Object." (Pomerantz 2015, p26)

A `statement` in this semantic meaning is a meaningful declarative sentence that is either true or false.

- Taylor Swift was born in 1989.

The World Wide Web standards for metadata exchange, which are quasi-global standards, work with so-called semantic triples. Triples are the shortest possible statements: they connect a subject and an object through a predicate.

The most popular metadata language that is both human- and machine-readable, Turtle ends every statement with a dot space separated from the third element of a triple (to avoid the third string having a dot character).

```
# The URLs for the definitions:
@prefix person: <http://example.org/persons/>
@prefix relation: <http://example.org/relations/>
@prefix book: <http://example.org/books/>
@prefix works: <http://example.org/musical_works/>

# Simple triple statements:

person:Mark_Twain   relation:author books:Huckleberry_Finn .
person:Taylor_Swift relation:author works:Gold_Rush .
```

The standard *Japanese breakfast* consists of steamed white rice, a bowl of miso soup, and Japanese-style pickles (like takuan or umeboshi). In the context of music, `Japanese Breakfast` is the stage name of the Korean-American artist Michelle Zauner.

Table 3.1: Semantic Triples

Subject	Predicate	Object
Japanese Breakfast	is a	music group
Japanese Breakfast	performs the works of	Michelle Zauner
Michelle Zauner	wrote	`Machinist`
Q44555381	identifies	Michelle Zauner
0000 0004 6613 4394	identifies	Michelle Zauner
`spotify:13FGWUlqQpGugvEenEUqsu`		Machinist

The simple' subject-predicate-object' semantic statements show how we can use "statements about potentially informative objects," i.e., these playlists contain information about the authorship, performers, or identity of various music works and their recorded and sheet notation manifestations.

It would be tempting to create an identifier like `2014USJPNBRKMACH` for Machinist, and encode, for example, the release year already in the identifier itself. This is exactly what the International Standard Recording Code does. For example, the International Standard Recording Codes (ISRC) used in the music industry should refer to the country of registration, the registrant company or entity, and the year of first registration. At the time of the creation of the ISRC code, when only a few uses could be imagined (we did not even have the internet, let alone music streaming services), this may have shown foresight. But in 2024, the ISRC codes do not represent the registration countries (because some countries ran out of their code range, and there are international registrations), for various reasons, often do not unambiguously refer to the registrant, and the practices of assigning the year code allow little semantic inference to what they mean.

In information science and digital curatorial practice, it is generally accepted that identifiers should not embed and encode metadata. Embedding metadata into an identifier usually creates an incentive to later change the identifier, which can potentially harm the uniqueness of the identifier as a string and stop its persistence. As identifiers are used in newer and newer applications or contexts, issues may arise regarding what should be embedded into the string. (Maybe not the registering label but the artist? Not the release year, but the full date instead? Or the location?)

> "The intelligence derived from an identifier system must lie with metadata rather than being embedded within intelligent identifiers if the system is to be extensible and used in many contexts [...] A given entity to which an identifier is applied may have associated with it, in the identifier system, data which provide additional information, e.g., about its content, rights, etc. These metadata are potentially an infinite set. There is no such thing as »all of the metadata« for an entity, as someone may devise a system which uses a piece of associated data not previously considered and recorded in the identifier system" (N. Paskin 1999)

We do not need to encode metadata into the identifier because we can make it *actionable*. The most common actionable identifier is a URI, which looks like an internet URL but behaves differently when a human reader clicks on it in a browser or a catalogue management application tries to read it.

The ISNI identifier 0000 0004 6613 4394 is actionable. If you click on https://isni.org/isni/0000000466134394, it displays displays the following information:

> `ISNI`: 0000 0004 6613 4394 `Name`: Breakfast, Japanese Japanese Breakfast Zauner, Michelle Zauner, Michelle Chongmi `Dates`: born 1989-03-29 `Creation role`: author composer instrumentalist performer singer `Related`

`identities`: Zauner, Michelle (real name) `Notes`: identity's home page http://japanesebreakfast.rocks/ https://www.discogs.com/artist/3602279 https://www.wikidata.org/wiki/Q28104185

URIs are usually created so that when you try to open them in a browser, they display human-intended text; if a non-browser application uses them, it allows the download of a standard, machine-readable metadata description. Modern libraries, archives, museums, or rights management applications use URIs as actionable identifiers that connect the identified entity (a musical work, a sound recording, or its author) with its metadata.

3.5.1 Universal Resource Identifiers

A quasi-global standard of global, persistent, unique identifiers is the definition of the World Wide Web Consortium on Universal Resource Identifiers (URIs). A URI is "a compact sequence of characters that identifies an abstract or physical resource," which is by design separates the identification from any actionable interaction (Berners-Lee, Fielding, and Masinter 2005). At first sight, this is confusing, because URIs usually look like URLs (Universal Resource Locators), which do point to the resource, and for example, allows for their retrieval in a web browser. For example, https://publications.europa.eu/resource/authority/country/BEL is a URI.

URIs are not URLs, because they are supposed to identify things that are not on the internet: for example, physical objects, such as buildings in physical space, or mediaeval manuscripts in libraries. They do look like URL, because they often provide some service, for example, they connect to a definition or description of the "resource" they identify. The https://publications.europa.eu/resource/authority/country/BEL identifies Belgium, as a country, which is not something that you can download to your computer. By making the URI in a format of a URL, it allows a human-reader to find a more detailed description of the thing that is identified. This is particularly useful in the case of classes that refer to many things, such as `adhesive-coated paper` and `acid-free paper`, or for URIs that refer to people, who, as we had seen, may have many namesakes.

The URI http://vocab.getty.edu/page/aat/300444127 identifies `adhesive-coated paper`, while http://vocab.getty.edu/page/aat/300311608 identifies the term `acid-free paper`; these terms are important in the identification, storage, preservation of paper-based artworks. Acid-free paper can be also labelled as `papel alcalino` in Portuguese,

in Ukrainian. Using http://vocab.getty.edu/page/aat/300311608 is very practical to connect catalogues of American, Portugese, Ukrainian and any other catalogues without the ambigouity of translation or understanding the type of paper we are talking about.

The URI https://isni.org/isni/0000000078519858 helps to resolve the `0000000078519858` numeric identifier; it refer to the most famous Taylor Swift.

3.6 Named entity recognition and disambiguation

We started this chapter with the example that in the United States alone, more than 38,000 men were named James Smith, and more than 32,000 women were named Maria Garcia; the number increases with the addition of further English- and Spanish-language territories. We have also shown that some generic name titles, like `Symphony No. 2`. can refer to a great many musical works or even more recorded or music sheet notations.

Named entity recognition and disambiguation (NERD) is the task of identifying and determining the meaning of named entities in a given context. It means that the text `Taylor Swift` is correctly recognised as the name of the American singer-songwriter born in 1989, or with the photographer or any other person with the same name.

NERD requires knowledge to connect the text `Machinist` correctly with either Michelle Zauner a.k.a. `Japanese Breakfast` or Lloyd Cole.

Table 3.2: Identifiers help to connect metadata to informative entities.

Subject	Predicate	Object
Machinist	is written by	Michelle Zauner
Japanese Breakfast	recorded	Machinist
Lloyd Cole	recorded	Machinist
Machinist	was released in	2001
`spotify:3OQ3DP6IzwE5KHdSp0pGJB`		Machinist
`spotify:13FGWUlqQpGugvEcnEUqou`		Machinist

Identifiers are unique names that help us connect data and metadata or connect predicates to named entities. The recording identifier `13FGWUlqQpGugvEcnEUqou` ensures that the Machinist song can be unambiguously selected if we create a Japanese Breakfast playlist on the Spotify platform, and for copyright royalty payments to Michelle Zauner; and at the same time, Machinist is never connected to Michelle Zauner or Japanese Breakfast.

High-quality identifiers are of utmost importance. In their absence, we rely on well-structured knowledge to deduce or infer the identity of a sound recording and its performer or author. For example, knowing that Machinist was recorded in 2001 when Michelle Zauner was 12, makes it unlikely that she is the performer. However, adding further information that she first started to play the guitar at the age of 15 (in the year 2004, later than 2001) and made her recorded debut in 2011 excludes this Machinist as hers.

We aim to create high-quality information resources that make such inference possible even without a prior successful identification; for example, a dress historian may find blue cooking aprons even if their color is recorded as `blue`, `blauw`, `kék`, , or , and the inventory book is not talking about an apron but `schort`, `kötény`, or . Such disambiguation can be a great tool in scientific research, or reduce the costs of copyright management.

3.6.1 Identity & Data Brokerage

> In principle data infrastructures can be linked directly together. Stable identifiers of digital entities on one infrastructure can be maintained on another to link infrastructures in one direction, or there can be reciprocal links to traverse infrastructures in either direction. […] An alternative to linking infrastructures is for a third party infrastructure to act as a broker between infrastructures. Wikidata is a collaboratively edited multilingual database hosted by the Wikimedia foundation, which can be used for this kind of data brokerage. (Meeus et al. 2022, p10)

The Dictionary of Archives Terminology identifiers use acid-free-paper for `acid-free paper`, while the Art & Architecture Thesaurus® Online (a globally used resource of the Getty Research Institute; in short: AAT) uses 300311608. Which is better? There is no answer for this question, it depends on your application. If you want to exchange data with another collection that already uses AAT, then using the same thesaurus offers the most reward with the least work. However, if you use AAT but you want to connect to a collection that uses the Dictionary of Archives Terminology, then you will have to find a way to reconcile acid-free-paper with 300311608.

Wikidata also identifies the different names, aliases, and potential identifiers of `acid-free paper` with the QID of `Q3178534` that resolves with https://www.wikidata.org/wiki/Q3178534. The reason why we use Wikidata QIDs whenever possible is that they offer a simple way to connect our users to many potential identifiers. By clicking to Q3178534, and scrolling down to Identifiers, you will find a links to several widely used thesauri.

3.7 The promise of the internet of data

> An essential process is the joining together of subcultures when a wider common language is needed. Often two groups independently develop very similar concepts, and describing the relation between them brings great benefits. […] A small group can innovate rapidly and efficiently, but this produces a subculture whose concepts are not understood by others. Coordinating actions across a large group, however, is painfully slow and takes an enormous amount of communication. The world works across the spectrum between these extremes, with a tendency to start small—from the personal idea—and move toward a wider understanding over time. […] The Semantic Web, in naming every concept simply by a URI, lets anyone express new concepts that they invent with minimal effort. Its unifying logical language will enable these concepts to be progressively linked into a universal Web. This structure will open up the knowledge and workings of humankind to meaningful analysis by software agents, providing a new class of tools by which we can live, work and learn together. (Berners-Lee, Hendler, and Lassila 2001)

Tim Berners-Lee is often credited as the inventor of the World Wide Web. His seminal, co-authored paper in 2001 envisioned the semantic graph that connects all knowledge and workings of humankind, supported by intelligent software agents. This promise was much more difficult to fulfill than the creation of the original World Wide Web, which allowed the accessible publication of hypertext documents (pages of illustrated text that cross-refer to other pages regardless of the server's physical location that stores the URL-referred connecting page). It goes well beyond the scope of our manual to describe the difficulties of working with the semantic web; one of the many reasons why it took two decades to become mainstream is partly the complex and expensive publication infrastructure needed and partly the shortage of skills in knowledge organisation. Wikipedia, Wikidata, and recently the Wikibase software as a free, stand-alone open-source product have contributed the most to democratising the semantic web.

Recalling the Turtle representation of a semantic statement:

```
<http://example.org/person/Mark_Twain>
   <http://example.org/relation/author>
   <http://example.org/books/Huckleberry_Finn> .
```

can be all represented by URIs:

```
<https://www.wikidata.org/wiki/Q7245>
   <https://www.wikidata.org/wiki/Property:P50>
   <https://www.wikidata.org/wiki/Q215410> .
```

Which resolves into : Mark Twain (Q7245) author (P50) Adventures of Huckleberry Finn (Q215410) .

Among the many advantages of this solution, one is resolving multi-language use.

- ☒ `Mark Twain (Q7245)` is connected to the international standard ISNI number 0000000077209145, and to the ID of the this particular author in numerous national library systems.
- ☒ `author (P50)` resolves for `author` in English, `szerző` in Hungarian, in Hindi, and in Greek; buy publishing this statement, you can connect with Indian or Greek sources even if you computer does not have such characters.
- ☒ `Adventures of Huckleberry Finn (Q215410)` connects to the French library catalogue item cb120369031 and 4311319-9 in the German national library system.

It is not only Wikidata (and Wikibase) that can provide a similar solution; in fact, for librarian, archivist, or musicological uses, there are better solutions available. But they all require specialist knowledge and expensive infrastructure. In the subsequent chapters, we introduce

Wikidata (see Chapter 4) and Wikibase (see Chapter 5; where we continue the explaining how to create the entries like the one for *Adventures of Huckleberry Finn*.) We believe that Wikidata offers the most democratic, least costly and most accessible platform to create an international consensus among researchers or collectors of a topic. Wikibase, the software that powers Wikidata, is the easiest, less costly start for an avantgarde group of collectors, a small research group, or a niche research interest group to start building a shared knowledge base.

4 Wikidata and Other Open Knowledge Graphs

A knowledge graph represents a network of real-world entities—such as objects, events, situations, or concepts—and illustrates their relationship.

Most companies and institutions work with a variety of information systems that are not well integrated. Information is located in different places, inside and outside the organisation, and cannot be accessed as a whole. In recent decades, it has become clear that unifying these information sources into central databases or data lakes is rarely a good solution. Creating such centralised data stores is very costly and requires a lot of organisation. By the time centralisation is completed and finished, it often becomes apparent that the knowledge requirements and the methodology for organising data have changed.

Here's where knowledge graphs come in. They can automatically integrate and present a unified view of diverse and initially unrelated data sources. For instance, in an enterprise, they can power initiatives like Customer 360. Moreover, knowledge graphs are ideal for implementing the Human-in-the-Loop (HITL) design principle in AI deployment. They offer a comprehensive view of the knowledge base that algorithms rely on, enhancing oversight and control.

4.1 Connect to Wikidata

Wikidata is a collaboratively edited multilingual knowledge graph hosted by the Wikimedia Foundation. It is a common source of open data that Wikimedia projects, such as Wikipedia and anyone else, can use under the CC0 public domain license[1]. As of early 2023, Wikidata had 1.54 billion item statements or small, verifiable scientific statements about our world[2].

Wikidata is a document-oriented database, focusing on items, which represent any kind of topic, concept, or object.

[1] CC0 enables scientists, educators, artists and other creators and owners of copyright- or database-protected content to waive those interests in their works and thereby place them as completely as possible in the public domain, so that others may freely build upon, enhance and reuse the works for any purposes without restriction under copyright or database law.

[2] We introduced the concept of statements as atomic knowledge carriers in **?@sec-identifiers-and-metadata**.

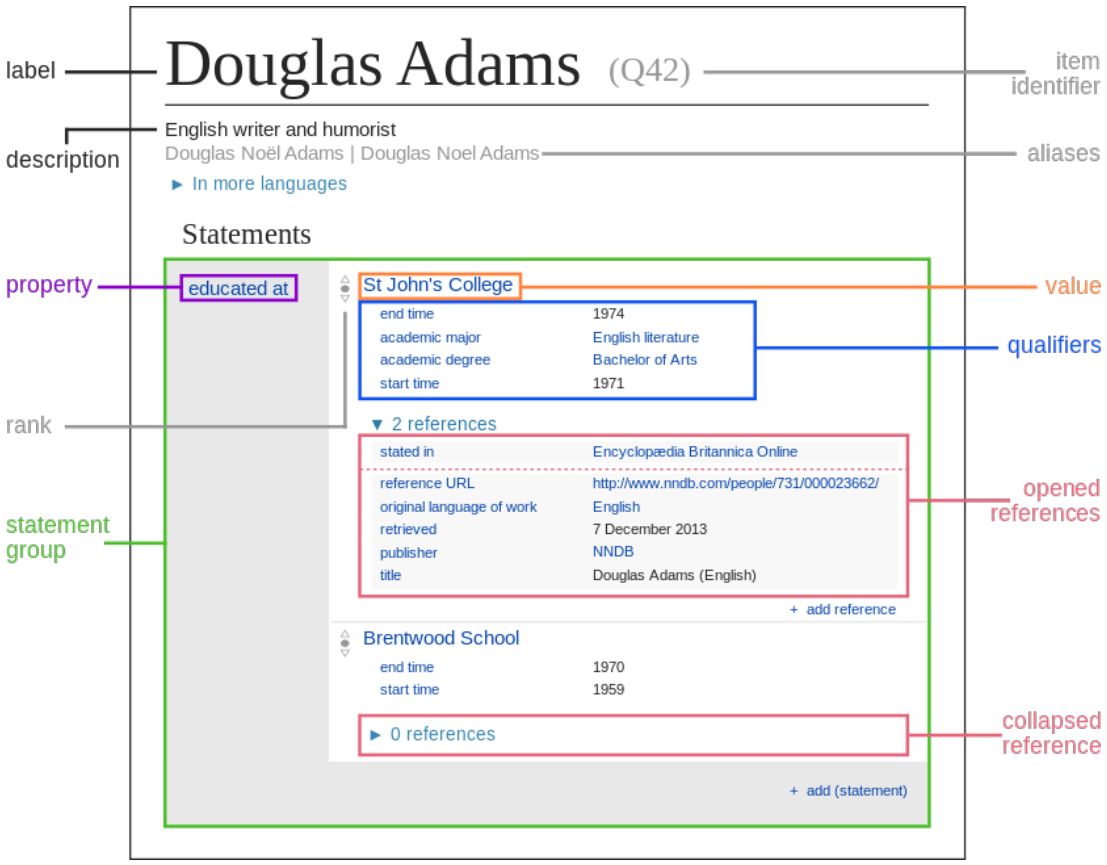


Figure 4.1: Wikidata is a document-oriented database. This document connects a lot of knowledge about the late English writer and humorist, Douglas Adams.

Knowledge graphs connect things in the real world, because their nodes—in Wikidata, the conceptual document—, represent people, objects, and their relationships as they are out there, and not as they are represented by an "ordinary" database . The Q42 document about the late English writer and humorist Douglas Adams connects facts about his life (birthday, place of birth, time of death), and connects him to his books, their translations, identifiers to look up these books, and so on.

Wikidata is a knowledge graph: it connects the concept of Douglas Adams (Q42), to the concept of his most quoted humorous episode from his world-famous *Hitchhiker's Guide to the Galaxy* (Q25169) , which is a similarly structured document about the five books of his series, which document is further connected in the graph to the concept of the books' Serbian translation (Q117279887).

Wikidata is not a database but a very useful system for filling up and keeping many databases in sync worldwide. If your own institutional or private library has a catalogue, you may have a copy of the *Hitchhiker's Guide to the Galaxy*; in this case, your catalogue is likely to have a

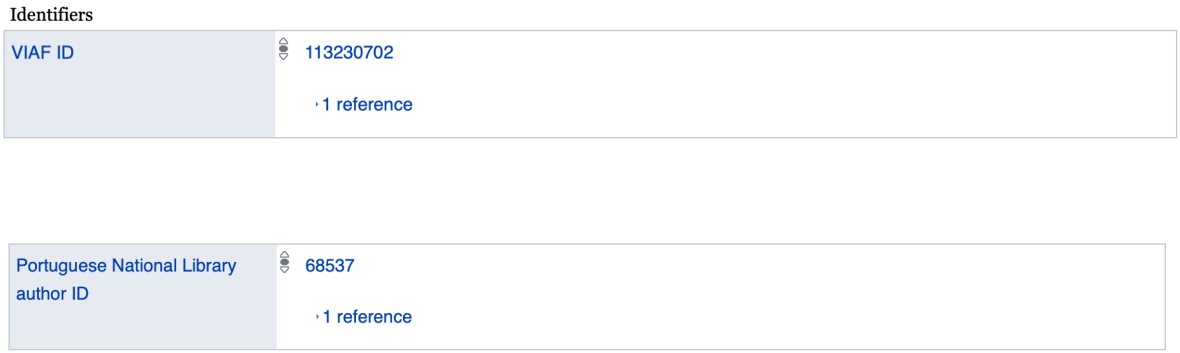
local, private identifier to your copy of the book. Imagine your little private catalogue, where you, like the editors of Wikidata, reserved the #42 entry to Douglas Adams' book.

ID | Author | Title |

|——-|——————————————————-|——————————————————-| | My-01 | Martell, Yann (Q13914) | *Life of Pi* (Q374204) | | ... | ... | ... | | My-42 | Adams, Douglas (Q42) | *Hitchhiker's Guide to the Galaxy (*Q25169) | | ... | ... | ... |

If you can connect your **My-42** entry of *Hitchhiker's Guide to the Galaxy* with the books' Wikidata entry Q25169, you can import a wealth of information into your private catalogue. Furthermore, if you connect the Wikidata item Q42 of the author Douglas Adams to your catalogue's own entry about the author, you can import a lot of additional knowledge, for example, information about his other works, or the end term of these books' copyright protection, after which they will become public domain and they will be free to copy and distribute.

In Wikidata, each item has a unique, persistent identifier, a positive integer number, prefixed with the upper-case letter Q, known as a "QID".Global information systems like to anchor authoritative information about people, books, musical works, and other important things to persistent identifiers. For example, in VIAF, the authority file that keeps information synchronised across national libraries, Douglas Adams' persistent identifier is 113230702, whereas in the Portuguese National Library it is 68537. Wikidata is particularly useful because it serves as an "identity broker", and this linking information can be retrieved directly from Douglas Adams' Q42 page.



4.1.1 Getting started with Wikidata

4.1.1.1 Global Identities

Mr and Mrs Barasits, a.k.a. `János Barasits` (1859-1935) and his wife, `Barasits, Jánosné`, born Pichler, Kornélia, were prominent postcard producers and publishers at the beginning of the 20th century. They produced plenty of beautiful postcards.

In the 1920s and 1930s, the authors' right (~copyright) protection of photographs and postcards was relatively short, only 15 years, so their postcards went into the public domain in terms of copying long ago. Plenty of their beautiful works are out there on the internet, but it is very hard to put them into a collection, because most databases know next to nothing about the identities of these creators and their creations.

Unfortunately, you cannot find their name in the most commonly used authority controls, i.e., VIAF or ISNI. Writing to VIAF is only possible via member institutions, and ISNI costs money. A temporary solution is to create a Wikidata QID for János Barasits (Q124423018), until somebody registers his name into VIAF. With this entry, it will be easier to find further postcards from them, or other information about them all over the world!

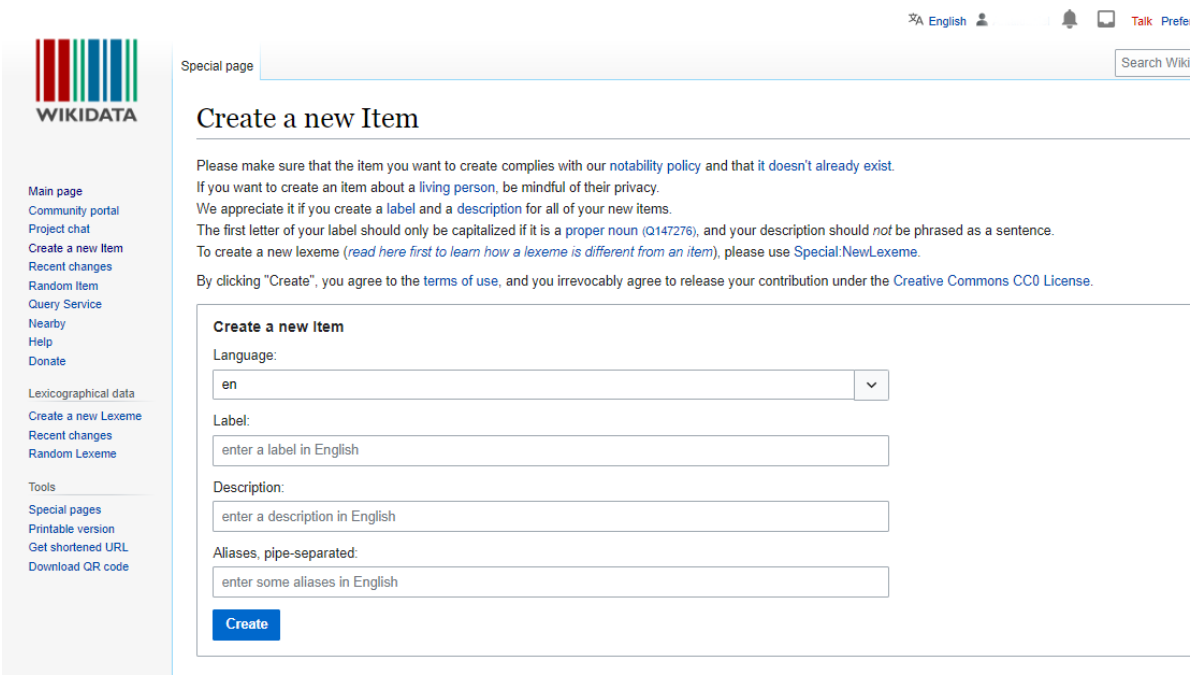
Writing in Wikidata is free for all and subject to community review. If you read this tutorial, please pledge to record new persons (or other items) into Wikidata, only if your knowledge is solid. You can verify the information needed through proper research.

4.1.1.2 Create a Wikidata Item

In this tutorial, you can learn how to create a new item on Wikidata. Countless web and AI applications and millions of people use Wikidata, so in the beginning it is recommended to not experiment with it in the live system. Wikidata has a Sandbox for practising. We recommend using it as a first step. If you work with Wikibase, particularly with Reprex's OpenCollections, you will have access to a similar sandbox. It will be prefilled with data, concepts, and properties suitable for your learning needs, often going beyond what you would find in the public Wikidata.

Let's see how you can create your own János Barasits item.

You can see how creating a new item looks like in the system:



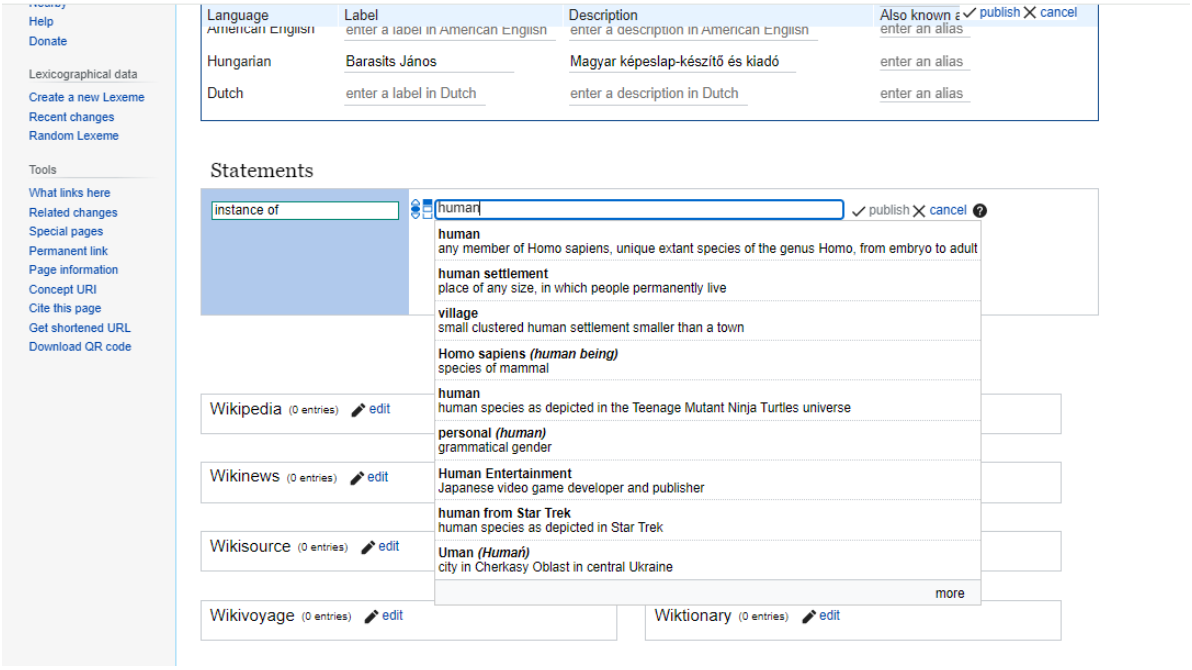
The first step in creating an item (in this case an item for János Barasits) is providing the two most important information for an item, which is the `Label` and the `Description`.

The `Label` is the name of the item (in our case the label of the item will be "János Barasits").

The `Description` contains a short explanation of our item (in our case the description for the item will be "Hungarian postcard maker and publisher).

`Aliases` are alternative names for the item.

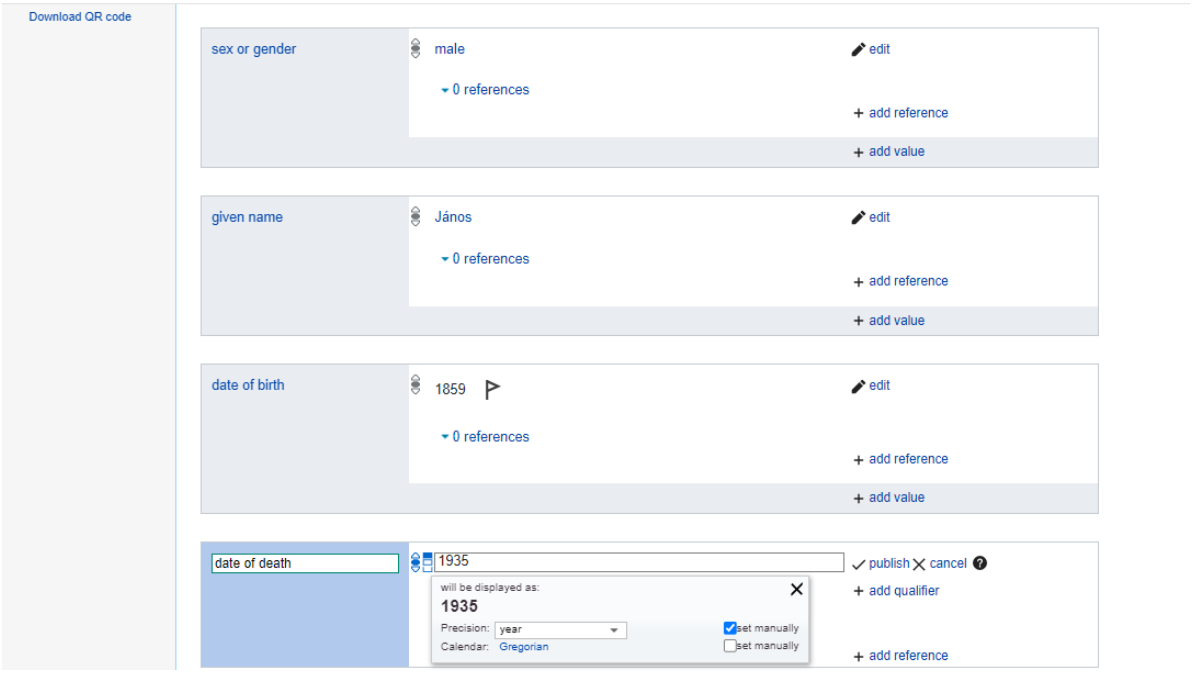
After creating the item with the basic information of `Label` and `Description` we can weave this information entry into the knowledge graph. At this point, `János Barasits` could be a person, it could be a book titled after the person, or a photo of the person. Connecting János Barasits to other entities, such as the concept of a human being, will allow other people and their computer systems to understand that we are talking about a person here.You can do that by creating "Statements". The property "instance of" defines the class our item is a particular example or member of. In this case we would like to make a statement about our item "János Barasits" defining with the property "instance of" that he is a member of the "human" class.



Through the sandbox explore the different type of properties and statements. Add a few basic statement to your new item:

- János Barasits is a human—his gender was male—he was born in 1859 (with the precision of a year only)—he died in 1935.

It should look similar to this:



To see another example on how useful knowledge graphs can be consider the following. The four character, `1935` can be understood as a number for most readers, but such a data point without a defined meaning is useless. In a basic database you would see 1935 and know that it is a number. However, in knowledge graphs, like here on Wikidata, when we add the "metadata", and we connect 1935 to the definition of `date of death`, we add a meaning ("semantics") to the number 1935. Now, 1935 is not only a number but also a date of someone's death.

The definition of date of death is useful in itself, but in a knowledge base, we can do even more with this piece of information. With this information we can combine the fact that in Europe the copyright protection term of people's creation runs up to 70 years after their death. Thus, a knowledge base can infer the fact that currently János Barasits's postcards are out of copyright and they can be freely copied and distributed!

Here is a very basic Wikidata page for János Barasits. What is very important, is that we have a globally unique identifier, `Q124423018` that uniquely identifies him as a human. If you have a collection of postcards (digitals or analogue, vintage physical objects), connecting your own database with `Q124423018` will help you to import the knowledge of the expired copyright protection term; it will help you finding other out-of-copyright scanned copies of Barasits' postcards; it will be easier to connect to other collections that hold items from them, and so on.

4.1.2 Retrieve an item from Wikidata

Many internal enterprise resource systems or APIs use SQL, a 50+ year-old data query language. SQL is the lingua franca of relational database systems; you may be familiar with it. Can you query Wikidata in SQL?

Not exactly. It requires a different querying language, which was developed for knowledge graphs. It is called SPARQL because it is similar to SQL, but they are rather distant cousins.

While SQL is widely used, it does have a significant limitation: your query scripts are specific to one database system or API. What works in your internal catalogue may not function in another organisation's. If you've written a script to update your data from a specific web API, it doesn't guarantee that the script will be compatible with another API. Furthermore, it's not future-proof: if the API owner (or your database manager) makes even a slight adjustment to the system, you may need to modify or rewrite your retrieval code.

Remember, the significant advantage of Wikidata and other open knowledge graphs is that millions of people work on improvements and extensions daily. This means that an SQL request would be outdated every day. Instead of SQL, SPARQL queries do not look for cells in data tables, but they use intelligent knowledge to find the cells containing data about what you need. In SQL, you need to know which table contains people's birthdays and death dates to find out the year when János Barasits died. In SPARQL, you are looking for the cell that contains the date of death for the human known as János Barasits.

4.1.3 SPARQL basics

SPARQL, pronounced 'sparkle', is the standard query language and protocol for Linked Open Data and RDF databases. Having been designed to query a great variety of data, it can efficiently extract information hidden in non-uniform data and store it in various formats and sources. The SPARQL standard is designed and endorsed by the World Wide Web Consortium and helps users and developers focus on what they would like to know instead of how a database is organised. With SPARQL, you can access many large open knowledge resources, like the EU Open Data Portal (see here), the Eurostat data warehouse, or Wikidata (turorial here), or the knowledge basis of the Dutch heritage organisations, including the Rijksmuseum (see here).

Our data curators must be able to run SPARQL queries and make elementary modifications to them. Because we often import very large datasets, it would be very difficult to manually control every record on the graphical user interface. We use pre-written SPARQL queries (the data curator is expected to run via a simple URL link, perhaps modifying a class's QID or a property's PID) that serve as so-called *unit tests*. These queries programmed by Reprex allow simple tests like these:

- ☒ If the curator gave us 5432 person records, we have 5432 persons in the Reprexbase instance;
- ☒ If the gender breakup of a person's records is 2834:2598, the instance results in exactly the same persons of two genders (assuming that no third gender is used in the original data.)
- ☒ If we received data on Ján Levoslav Bella's Symphony in B minor, the publication year is 1982.

A simple SPARQL query looks like this:

```
SELECT ?a ?b ?c
WHERE
{
  x y ?a.
  m n ?b.
  ?b f ?c.
}
```

Suppose we want to list all children of the baroque composer Johann Sebastian Bach. Using pseudo-elements like in the queries above, how would you write that query?

Hopefully you got something like this:

```
SELECT ?child
WHERE
{
  #  child "has parent" Bach
  ?child parent Bach.
  # (note: everything after a '#' is a comment and ignored by WDQS.)
}
```

or this,

```
SELECT ?child
WHERE
{
  # child "has father" Bach
  ?child father Bach.
}
```

or this,

```
SELECT ?child
WHERE
{
  #  Bach "has child" child
  Bach child ?child.
}
```

The first two triples say that the ?child must have the parent/father Bach; the third says that Bach must have the child ?child. Let's go with the second one for now.

So what remains to be done in order to turn this into a proper WDQS query? On Wikidata, items and properties are not identified by human-readable names like "father" (property) or "Bach" (item). (For good reason: "Johann Sebastian Bach" is also the name of a German painter, and "Bach" might also refer to the surname, the French commune, the Mercury crater, etc.) Instead, Wikidata items and properties are assigned an identifier. To find the identifier for an item, we search for the item and copy the Q-number of the result that sounds like it is the item we are looking for (based on the description, for example). To find the identifier for a property, we do the same, but search for "P:search term" instead of just "search term", which limits the search to properties. This tells us that the famous composer Johann Sebastian Bach is `Q1339`, and the property to designate an item's father is `P:P22`.

And last but not least, we need to include prefixes. For simple WDQS triples, items should be prefixed with `wd:`, and properties with `wdt:`. (But this only applies to fixed values – variables don't get a prefix!)

Putting this together, we arrive at our first proper WDQS query:

```
SELECT ?child
WHERE
{
# ?child  father   Bach
  ?child wdt:P22 wd:Q1339.
}
```

Try the first query:

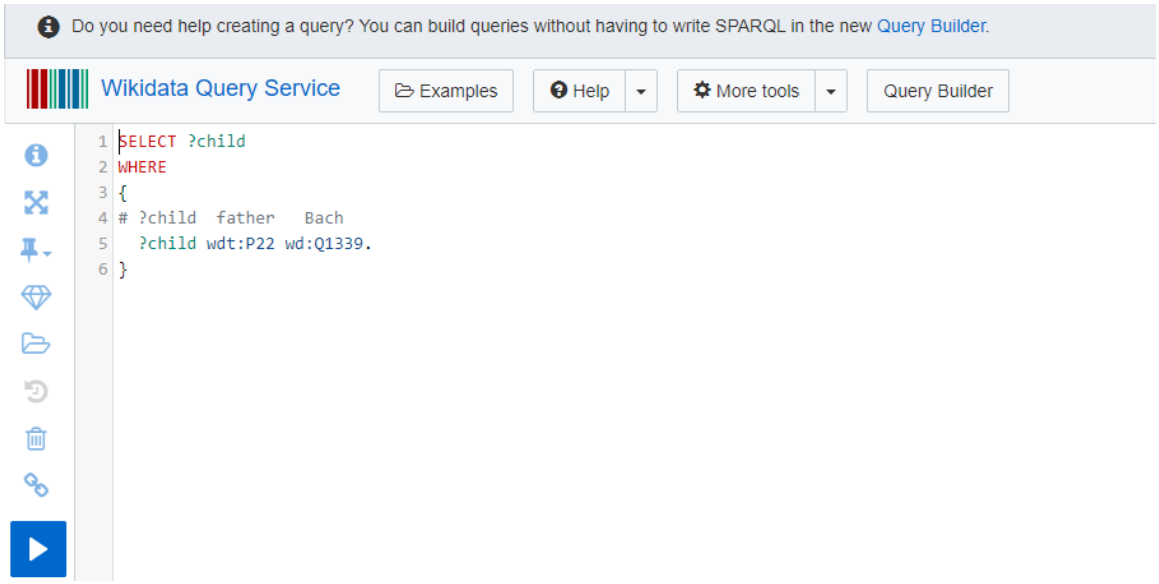


Figure 4.2: Click on the image to try it out live on the Wikidata SPARQL Endpoint. The query will run if you press the sign on the endpoint in the bottom left corner.

The first querry will provide you with identifiers, which is great if you are a programmer and you are wiring your database to Wikidata, but less impressive if you are getting familiar with SPARQL and you want to see clearly the fruits of your work.

Luckily, Wikidata has a human-friendly extension to SPARQL. If you add the following command to your query: `SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE]".` somewhere within the `WHERE` clause, you get additional variables: For every variable `?foo` in your query, you now also have a variable `?fooLabel`, which contains the label of the item behind `?foo`.

If you add this to the `SELECT` clause, you get the item as well as its label:

```
SELECT ?child ?childLabel
WHERE
{
# ?child  father   Bach
  ?child wdt:P22 wd:Q1339.
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE]". }
}
```

Figure 4.3: Click on the image to try it out live on the Wikidata SPARQL Endpoint. The query will run if you press the sign on the endpoint in the bottom left corner .

Try running that query – you should see not only the item numbers, but also the names of the various children.

child	childLabel
wd:Q57225	Johann Christoph Friedrich Bach
wd:Q76428	Carl Philipp Emanuel Bach
...	

4.1.4 Pre-filter Wikidata

When you work with OpenCollections or Wikibase, you may want to synchronize your knowledge graph with Wikidata. A straightforward way to do this is to import a part of the Wikidata knowledge graph into your instance.

Imagine you would like to copy the definition of a human, Béla Bartók, to your Wikibase instances. The following querry

```
SELECT DISTINCT ?itemLabel ?itemLabelLang ?itemDescription ?itemDescriptionLang ?aliases ?ali
  wd:Q83326 rdfs:label ?itemLabel ;
            schema:description ?itemDescription .
  OPTIONAL {
```

```
    wd:Q83326 skos:altLabel ?aliases .
    BIND(LANG(?aliases) AS ?aliasesLang)
  }
  BIND(LANG(?itemLabel) AS ?itemLabelLang)
  BIND(LANG(?itemDescription) AS ?itemDescriptionLang)
  FILTER(?itemLabelLang IN ("en", "de", "hu", "sk", "lt", "bg"))
  FILTER(?itemDescriptionLang IN ("en", "de", "hu", "sk", "lt", "bg"))
  FILTER(?aliasesLang IN ("en", "de", "hu", "sk", "lt", "bg"))
}
```

Try it out

- ☒ You can modify the querry. In Line 3, the `wd:Q83326` identifies the QID for Béla Bartók. Try it out with `wd:``28104185`!
- ☒ We asked the labelling in six languages. You can use `IN ("en", "de")` or even `IN ("de")` if you want to reduce the number of languages or change the language codes.

You would like to copy property definitions to your Wikibase instance. The following code will provide you the necessary information (without additional statements) about the property `wd:P31`—a very important property for data modelling.

```
SELECT ?property ?propertyLabel ?dataType ?propertyDescription ?lang ?alias WHERE {
  VALUES ?property { wd:P31 }  # Replace these IDs with the property IDs you are interested i
  ?property a wikibase:Property .
  ?property wikibase:propertyType ?dataType .

  # Fetch labels in the specified languages
  ?property rdfs:label ?propertyLabel .
  BIND(LANG(?propertyLabel) AS ?lang)
  FILTER(?lang IN ("en", "fr", "sk", "hu", "bg", "lt"))  # Replace these your languages
  BIND(IF(?lang = "en", 1, 2) AS ?labelRank)

  # Fetch descriptions in the specified languages
  OPTIONAL {
    ?property schema:description ?propertyDescription .
    FILTER(LANG(?propertyDescription) IN ("en", "fr", "sk", "hu", "bg", "lt"))
    FILTER(LANG(?propertyDescription) = ?lang)  # Ensure matching languages
  }
  # Fetch aliases in the specified languages
  OPTIONAL {
  ?property skos:altLabel ?alias .
  FILTER(LANG(?alias) IN ("en", "fr", "sk", "hu", "bg", "lt"))
```

```
  FILTER(LANG(?alias) = ?lang)  # Ensure matching languages
  }

}
ORDER BY ?labelRank ?lang
```

Try it out

The same query without aliases

- ☒ Try it with replacing the property value to `wd:P434`.
- ☒ Change the language codes for labelling. If a certain label does not exist on Wikidata in one of the languages, you will get no label.

Imagine you would like to work with the biographical data of photographers connected to Hungary. The following query can show you who has information on Wikidata. You may decide to import this information and use it as a starting point.

```
# Photographers: citizens of Hungary

SELECT ?item ?itemLabel  ?givenNameLabel ?lastnameLabel ?birthdate ?deathdate ?nationalityLa
    ?item wdt:P31 wd:Q5 .                # instance of human
    ?item wdt:P106/wdt:P279* wd:Q33231.  # occupation,subclass of occupation photographer
    ?item wdt:P27 wd:Q28.                # country of citizenship is Hungary
    optional { ?item wdt:P735 ?lastname . }
    optional { ?item wdt:P734 ?givenName . }
    optional { ?item wdt:P569 ?birthdate . }
    optional { ?item wdt:P570 ?deathdate . }
    optional { ?item wdt:P27 ?nationality . }

  SERVICE wikibase:label { bd:serviceParam wikibase:language "en,hu" }
}

order by ?itemLabel
```

Try it out . Beware, that Wikidata is huge, and query may take minutes to run; you often get an error message that your query run out of resources. Then try again.

Or similarly, with composers connected to Slovakia:

```
# Composers: citizens of Slovakia

SELECT ?item ?itemLabel  ?givenNameLabel ?lastnameLabel ?birthdate ?deathdate ?nationalityLa
    ?item wdt:P31 wd:Q5 .                # instance of human
    ?item wdt:P106/wdt:P279* wd:Q36834.  # occupation or subclass of occupation that is compo
    ?item wdt:P27 wd:Q214.               # country of citizenship is Slovakia
    optional { ?item wdt:P735 ?lastname . }
    optional { ?item wdt:P734 ?givenName . }
    optional { ?item wdt:P569 ?birthdate . }
    optional { ?item wdt:P570 ?deathdate . }
    optional { ?item wdt:P27 ?nationality . }

  SERVICE wikibase:label { bd:serviceParam wikibase:language "en,sk,de,hu" }
}

order by ?itemLabel
```

Try it out

5 Wikibase and Enterprise Knowledge Graphs

In the previous chapter, we introduced the idea of an open knowledge graph that connects knowledge curated by many people and organisations.

We have shown how valuable an open knowledge graph, like Wikidata, can be in reducing a private database's data curation, data control, and other related costs. Can we rely on similar knowledge graphs that are more specific to our professional domain and have more nuanced information than Wikidata? What if we want to keep music rights management databases or music distribution inventories updated and prefilled with data? Do we want to connect reliable, science-based data to our internal ESG systems?

Private enterprise knowledge graphs are usually made for precisely this purpose. Wikidata was originally created to support the increasingly automated corrections of the vast, open-source Wikipedia encyclopedias.

Encyclopedias have a limit of notability: they do not want to store information about every human living on Earth, but only those whose lives and work are notable enough to be interesting for the general public and who are living anyway in the public eye. (It would be unethical and even illegal to connect personal data about private individuals who do not wish to go out to the public space.) A private knowledge graph can connect information about all writers as rightsholders or their heirs, if they are deceased, to pay out royalties wherever they live.

5.1 The promise of the semantic web

> An essential process is the joining together of subcultures when a wider common language is needed. Often two groups independently develop very similar concepts, and describing the relation between them brings great benefits. […] A small group can innovate rapidly and efficiently, but this produces a subculture whose concepts are not understood by others. Coordinating actions across a large group, however, is painfully slow and takes an enormous amount of communication. The world works across the spectrum between these extremes, with a tendency to start small—from the personal idea—and move toward a wider understanding over time. […] The Semantic Web, in naming every concept simply by a URI, lets anyone express new concepts that they invent with minimal effort. Its unifying logical language will enable these concepts to be progressively linked into a universal Web. This structure will open up the knowledge and workings of humankind to meaningful

> analysis by software agents, providing a new class of tools by which we can live, work and learn together. (Berners-Lee, Hendler, and Lassila 2001)

Tim Berners-Lee is often credited as the inventor of the World Wide Web. His seminal, co-authored paper in 2001 envisioned the semantic graph that connects all knowledge and workings of humankind, supported by intelligent software agents[1].

This promise was much more difficult to fulfil than the creation of the original World Wide Web, which allowed the accessible publication of hypertext documents (pages of illustrated text that cross-refer to other pages regardless of the server's physical location that stores the URL-referred connecting page).

It goes well beyond the scope of our manual to describe the difficulties of working with the semantic web. One of the many reasons why it took two decades to become mainstream is partly the need for complex and expensive publication infrastructure and partly the shortage of skills in knowledge organisation. Wikipedia, Wikidata, and recently the Wikbase software as a free, stand-alone open-source product have contributed the most to democratising the semantic web.

Recalling the Turtle representation of a semantic statement:

```
<http://example.org/person/Mark_Twain>
   <http://example.org/relation/author>
   <http://example.org/books/Huckleberry_Finn> .
```

In the semantic web,

Mark Twain (the person) *created* (the verb) *Huckleberry Finn* (the book as object)

can be all represented by URIs, in which case anybody, including a software agent can read this statement. Substituting for `http://example.org/`, a part of the WWW namespace that is reserved for examples (and will never be allocated for any user), we can write:

```
<https://www.wikidata.org/wiki/Q7245>
   <https://www.wikidata.org/wiki/Property:P50>
   <https://www.wikidata.org/wiki/Q215410> .
```

Which resolves into : `Mark Twain (Q7245)` *author (P50)* `Adventures of Huckleberry Finn (Q215410)` .

Among the many advantages of this solution, one is resolving multi-language use.

[1] A part of this text is repeated from (**collections?**) for readability.

- ☒ `Mark Twain (Q7245)` is connected to the international standard ISNI number 0000000077209145, and to the ID of the this particular author in numerous national library systems.
- ☒ `author (P50)` resolves for `author` in English, `szerző` in Hungarian, in Hindi, and in Greek; buy publishing this statement, you can connect with Indian or Greek sources even if you computer does not have such characters.
- ☒ `Adventures of Huckleberry Finn (Q215410)` connects to the French library catalogue item cb120369031 and 4311319-9 in the German national library system.

It is not only Wikidata (and Wikibase) that can provide a similar solution; in fact, for librarian, archivist, or musicological uses, there are better solutions available. But they all require specialist knowledge and expensive infrastructure.

To paraphase Tim Berners-Lee from the previous larger quote, *"Coordinating actions across a large group, however, is painfully slow and takes an enormous amount of communication"*, for example, it took the world's archivists 10 years of hard work to come up with a better conceptual model for connected records in archives. On Wikibase, *"… a small group can innovate rapidly and efficiently, but this produces a subculture whose concepts are not understood by others"*. Wikibase can be thought of a local, private Wikidata; if it reaches a critical size, it can be connected to Wikidata for a global reach and a higher level of international consensus. Eventually, for specialists needs, one may develop a more customised set of definitions and relationships (a so-called ontology), for example, for handling problems with copyright data management. But Wikibase provides the easiest, less costly start for an avantgarde group to share knowledge and build a shared knowledge base.

5.2 Wikibase

Wikibase is the software that runs Wikidata. Wikidata evolved into a central hub on the web of data and one of the largest existing knowledge graphs, with more than 100 million items maintained by a community effort. Since its launch, an impressive 1.3 billion edits have been made by 20,000+ active users. Today, Wikidata contains information about a wide range of topics such as people, taxons, countries, chemical compounds, astronomical objects, and more. This information is linked to other key data repositories maintained by institutions such as Eurostat, the German National Library, the BBC, and many others, using 6,000+ external identifiers. The knowledge from Wikidata is used by search engines such as Google Search, and smart assistants including Siri,Alexa, and Google Assistant in order to provide more structured results.

While one of the main success factors of Wikidata is its community of editors, the software behind it also plays an important role. It enables the numerous editors to modify a substantial data repository in a scalable, multilingual, collaborative effort.

Wikibase is a software system that help the collaborative management of knowledge in a central repository. It was originally developed for the management of Wikidata, but it is available now for the creation of private, or public-private partnership knowledge graphs. Its primary components are the *Wikibase Repository*, an extension for storing and managing data, and the *Wikibase Client* which allows for the retrieval and embedding of structured data from a Wikibase repository. It was developed by Wikimedia Deutschland.

The data model for Wikibase links consists of "entities" which include individual "items", labels or identifier to describe them (potentially in multiple languages), and semantic statements that attribute "properties" to the item. These properties may either be other items within the database, or textual information.

> **Note**
>
> **Wikidata** itself is a gigantic *Wikibase instance*. Their user interface is similar, but depending on what the administrator of your Wikibase instance allows you to do, you are likely to have more freedom to edit certain elements, like properties, than on Wikidata. Wikidata must protect the integrity of one of the world's largest knowledge systems, and does not allow editing access to certain elements.

Wikibase has a JavaScript-based user interface, and provides exports of all or subsets of data in many formats. Projects using it include Wikidata, Wikimedia Commons,[5] Europeana's Eagle Project, Lingua Libre,[6] FactGrid, and the OpenStreetMap wiki.[7]

5.3 Populating a Wikibase

Wikibase is an open knowledge base or universe when installed. We start populating it with some **items**. In the Wikidata data model, items are similar to things, and classes are also defined as items.

> **Note**
>
> A *sandbox* instance is a Wikibase instance designated for learning, testing, experimenting. Reprex has created several sandbox instances for onboarding our data curators and for educational purposes. Please see Chapter 6 for getting an account on such an instance.

5.3.1 Creating entities or items

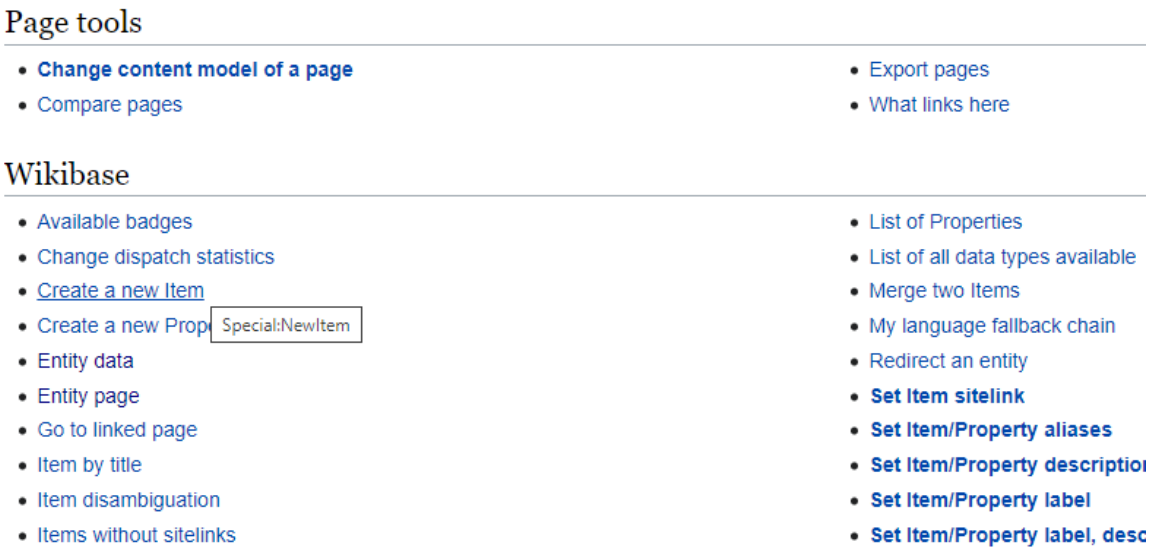


Figure 5.1: Special pages Wikibase Create a new item

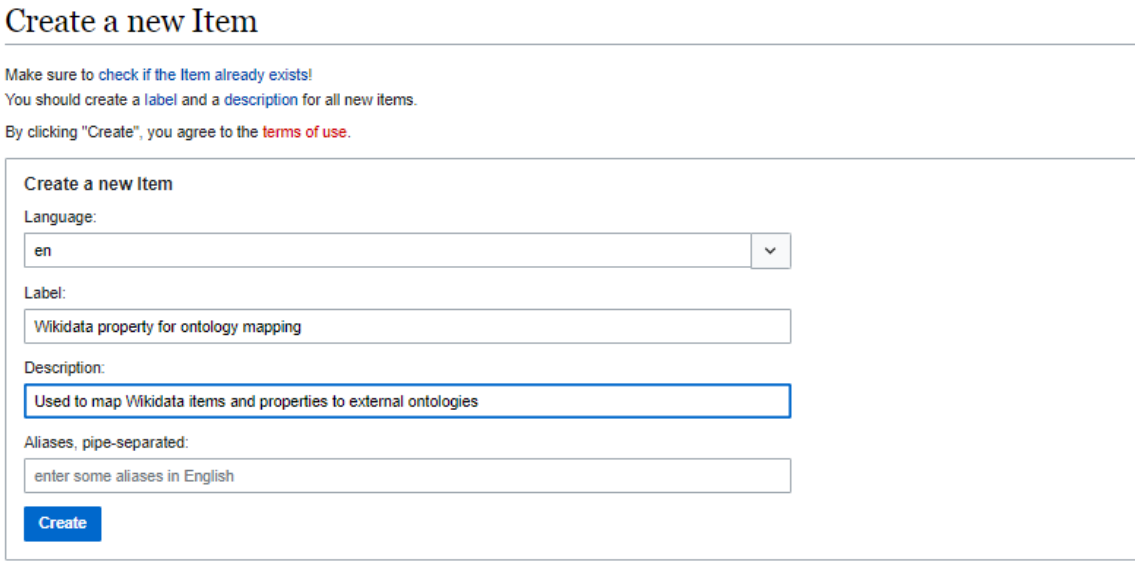


Figure 5.2: Identical to Wikidata: you must fill out at least the main Label of the item, and a description. We use English (en) as the master language for international cooperations.

Suppose you want to make an item or property entity multi-lingual. In that case, you must add at least a new label or description via the Special Pages on the Graphical User Interface (i.e., using your browser.) If you work with our import-export tool or the API, you can set labels and descriptions in several languages in one command.

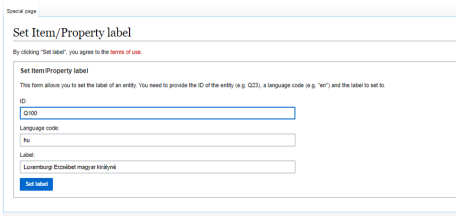


Figure 5.3: You can reach this form via the Special Pages Wikibase Set Item/Property label or et Item/Description link.

5.3.2 Creating properties

Properties are describing relationships between items. You can create them similary to items, but navigating to Special pages Wikibase Create a new property (*not item*). Properties are far more important than items, because they define the rules of the knowledge base. The type of relationships will allow our artificial intelligence applications to make deductive or inductive new discoveries and expand our knowledge.

In our introduction to Wikidata (Section 4.1), you found exactly the same graphical interface to work with items as on Wikibase, but on the public Wikidata instance of Wikibase, you cannot find an add new property button.

> **Note**
>
> On Wikidata, you are not allowed to create new properties: they are created after a consultation with the Wikidata community. The addition of properties determines who the knowledge graph will work in the future.

Needless to say that when you work with a Wikibase instance, you should be also very careful with properties. While changing items usually requires domain-specific knowledge, which you likely possess if you work on an instance, the property sometimes requies knowledge about the information or data model of the instance.

Not always: some properties are self-explanatory and very easy to create and maintain. For example, the adition of identifiers to other data systems is straightforward. Adding properties that define family relationships (which have their logical rules) requires more careful planning.

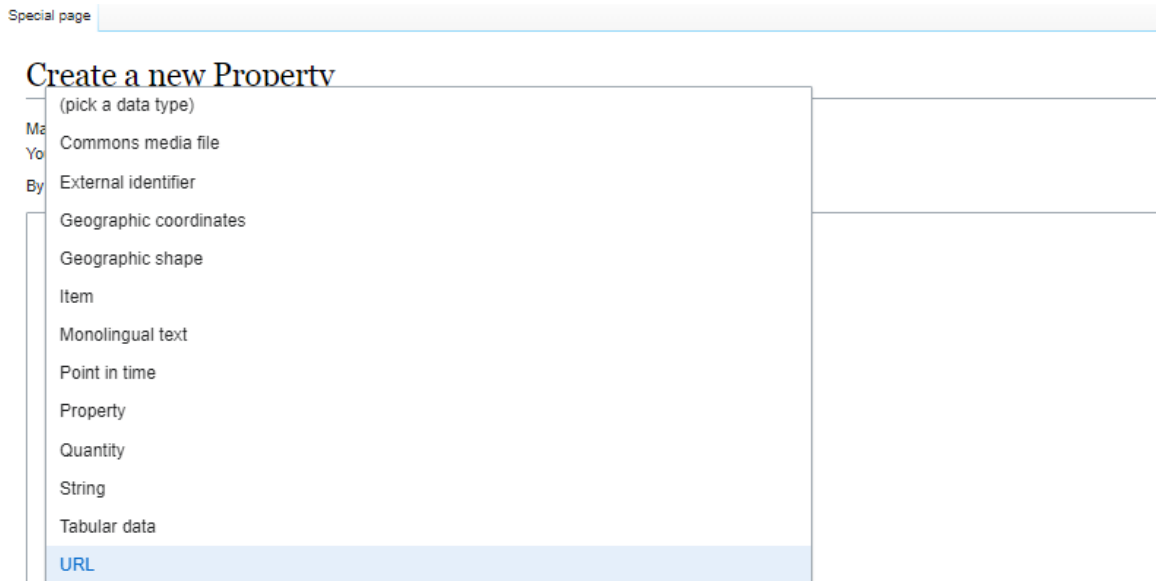


Figure 5.4: Properties have an extra field that you must fill out: the type of expected data type.

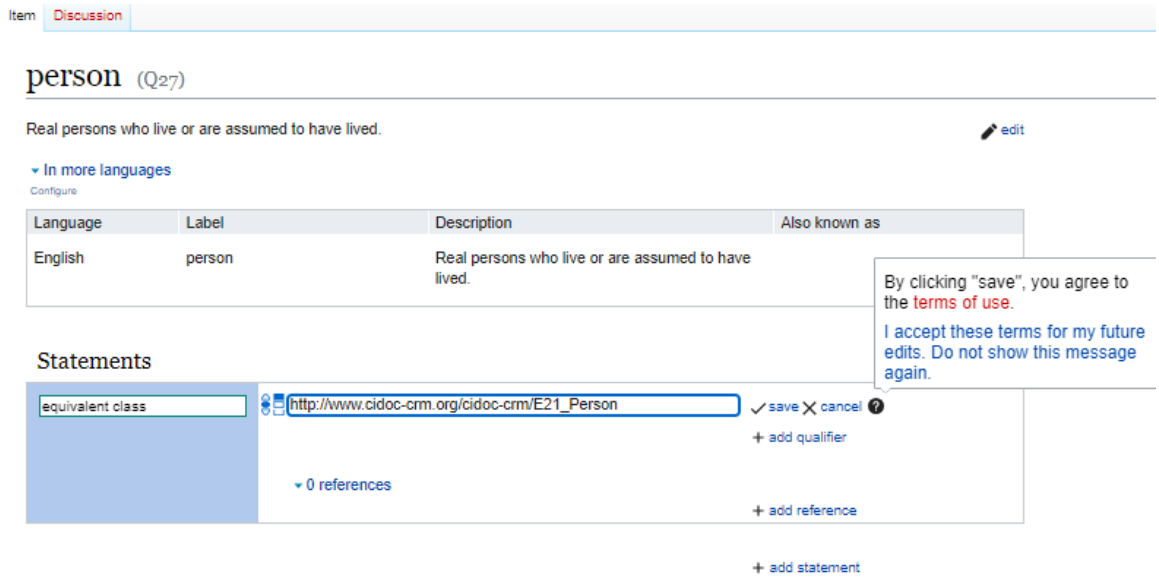
Properties have expected data types:

- Use a URL for connecting to other ontologies, data models (and add persistent URIs)
- Use *item* for entities that you want to weave together in the knowledge graph.
- Use literal values like *string* that for data that will be entered, but not will be placed on a graph.

For example, if you add *Mai Manó* as a string, it will be recorded, but you cannot connected with the works of Mai Manó, the photographer. If you create an entity (item) for `Mai Manó`, you will be able to link this entity to the works of Mai Manó, to his children, to his house.

5.3.3 Adding statements

Now we are ready to start to build an intelligent knowledge base. We connect the **person** item in our Wikibase via the **equivalent class** property to the E21_Person definition of the CIDOC CRM. This will allow us to export our knowledge base to a standard museological graph.



In this case, the `equivalent class` property only accepts URLs. The URI of the CIDOC definition of `E_21 Person` takes the format of a URL so you can enter it here, but a simple string like E21 would not be allowed.

> **Note**
>
> Adding statements is exactly the same procedure on Wikibase as on Wikidata (which is a gigantic Wikibase instance itself.) The only difference is that you can only use properties (or items) that exist on the Wikibase instance or Wikidata. Because Wikibase instances usually should have a different knowledge coverage, some properties and items are not available on others.

5.3.4 Synchronize with Wikidata

In our case, we want to be able to pre-fill data from Wikidata, and then, eventually suggest changes in the public Wikidata. This requires adding statements about Wikidata equivalent properties and items when applicable.

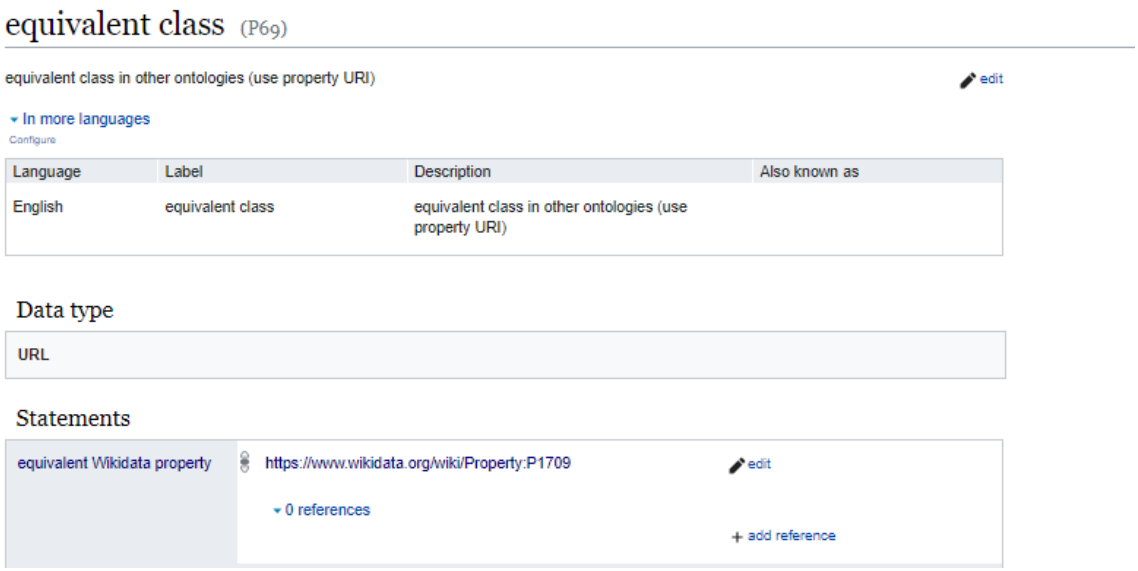


Figure 5.5: We created a special property, equivalent Wikidata property, to link the P69 property definition in our Wikibase instance to Wikidata's equivalent P1709. This will allow us synchronisation among the public Wikidata and our Wikibase.

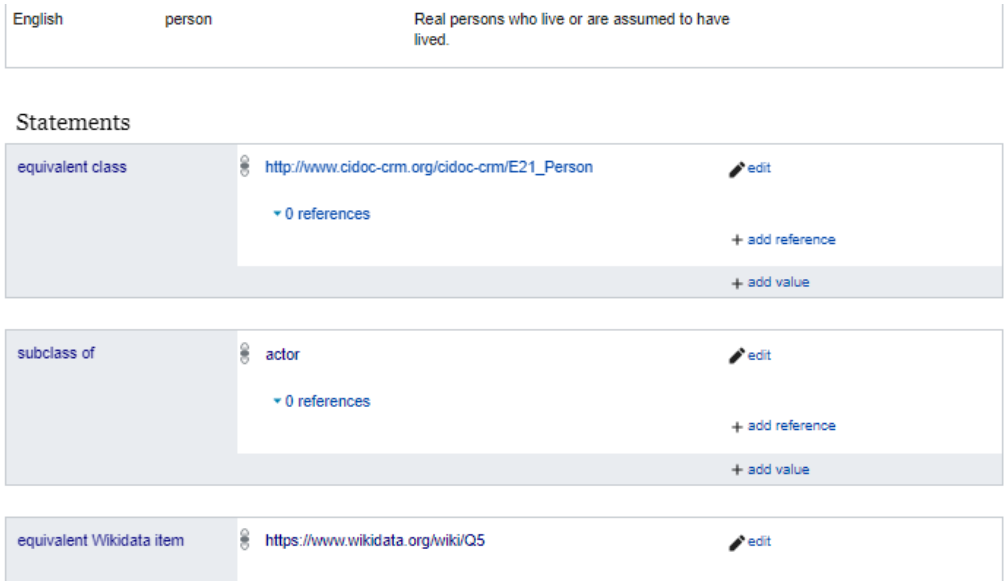
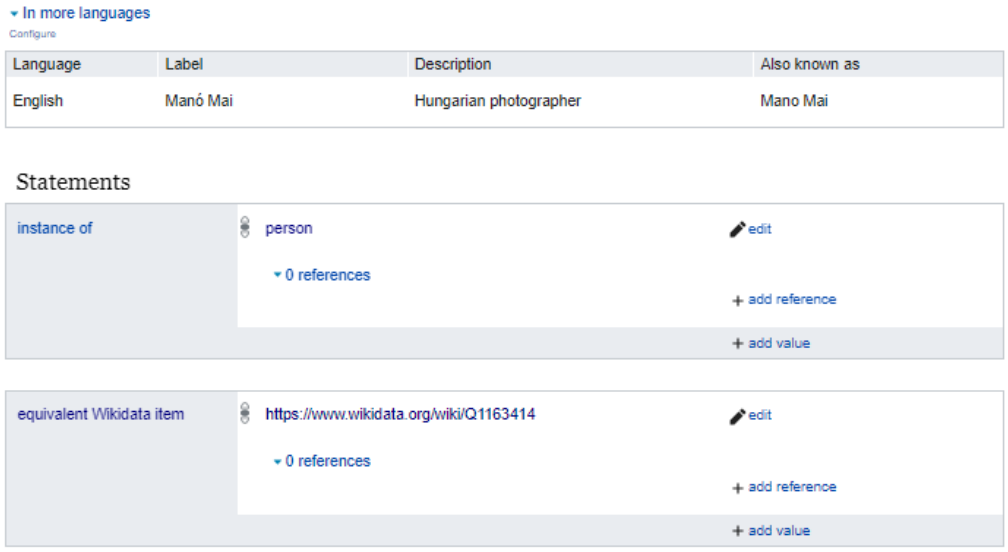


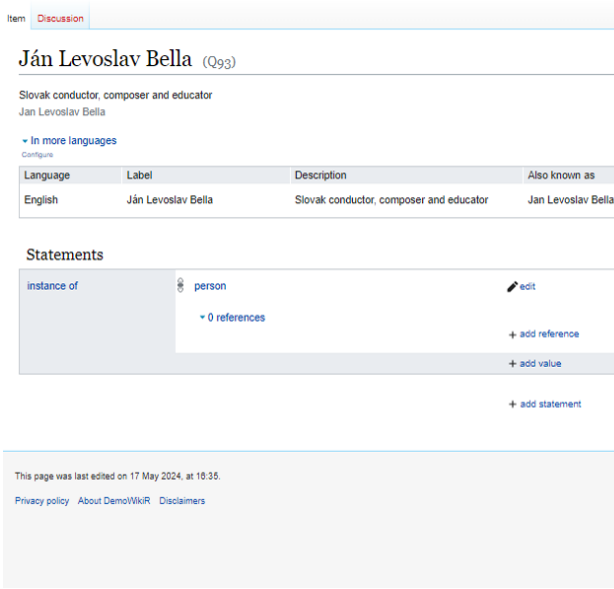
Figure 5.6: For items (and classes are defined as items in Wikibase, just like instances of persons), we created a special property equivalent Wikidata item to keep the Person entity (see above its creation) synchronized with Wikidata's Q5 item.

Let us put this all together and create a bibliographic entry. Here we will use a slight deviation from CIDOC, and use the instance of property (equivalently defined in our Wikibase with Wikidata) for class inheritance. When we create a new entity (Manó Mai), we will define this entity as an instance of a `person`. Persons have birth date, family members, they can create new creative works. In ontologies and in RDF we call these abstract concepts classes.



We immediately record that our entries about *Manó Mai*, the great photographer, should be talking about the same person as Wikidata's Q1163414 document item.

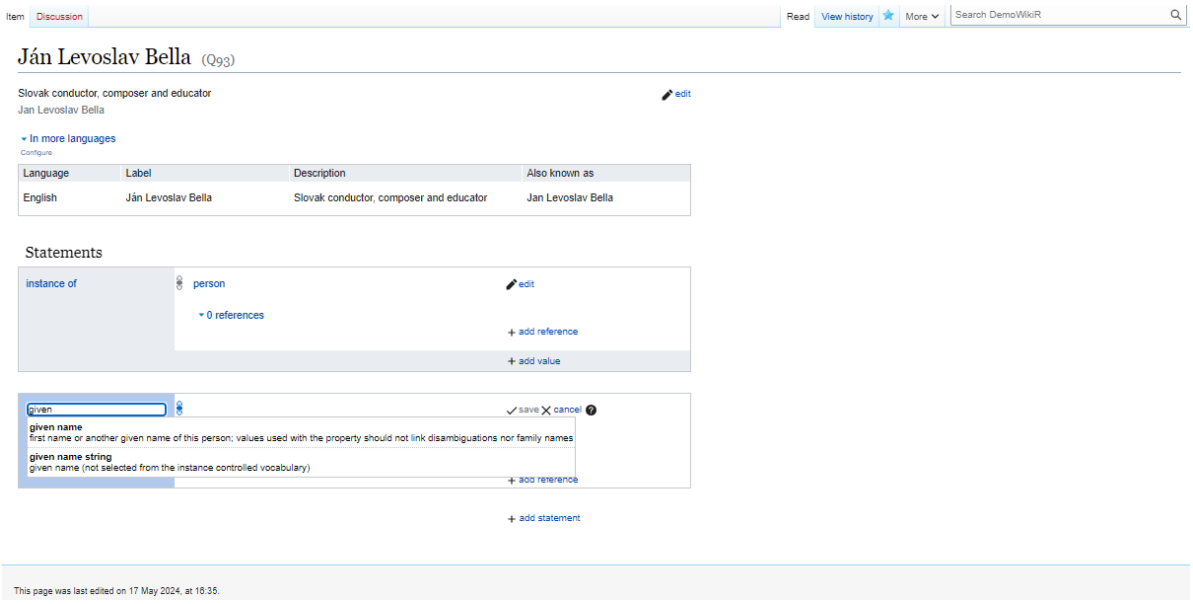
5.4 Good practices



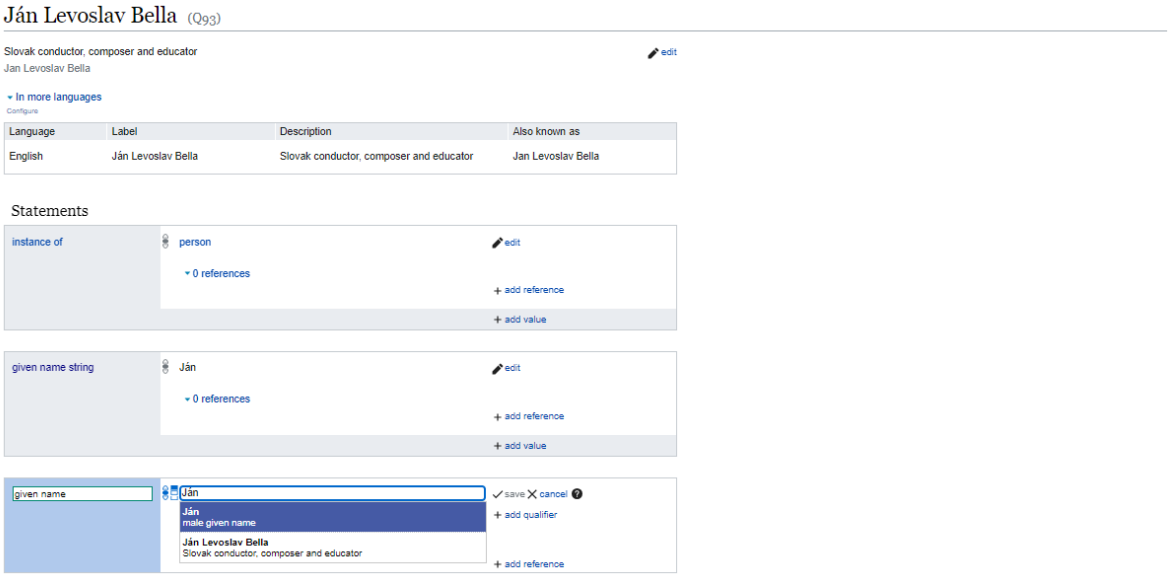
Let us consider the creation of an entry for the Slovak composer,

5.4.1 Use of name strings or controlled vocabularies

In this case, we would like to code the `given name` property to `Ján`. We can do it in two ways: - add the string *Ján* without further control, or, - add `Ján` as a controlled string (an item *datatype* on Wikibase.)



Unless we can import comprehensive datasets, usually data enrichment is a second step. In such cases, we import first to a `name string` property given names, locations, venue names, and other important nodes of our knowledge graph.



The use of controlled vocabularies makes filtering the database easier, and reduces the likelihood of errorneous entries. In the Wikidata data model, we can add a taxonomical class to such controlled vocabulary items. By coding `Ján` as an instance of the `Slovak make given name`, we can later search composers or persons easer by this name given name or we can infer that the composer was born as a man.

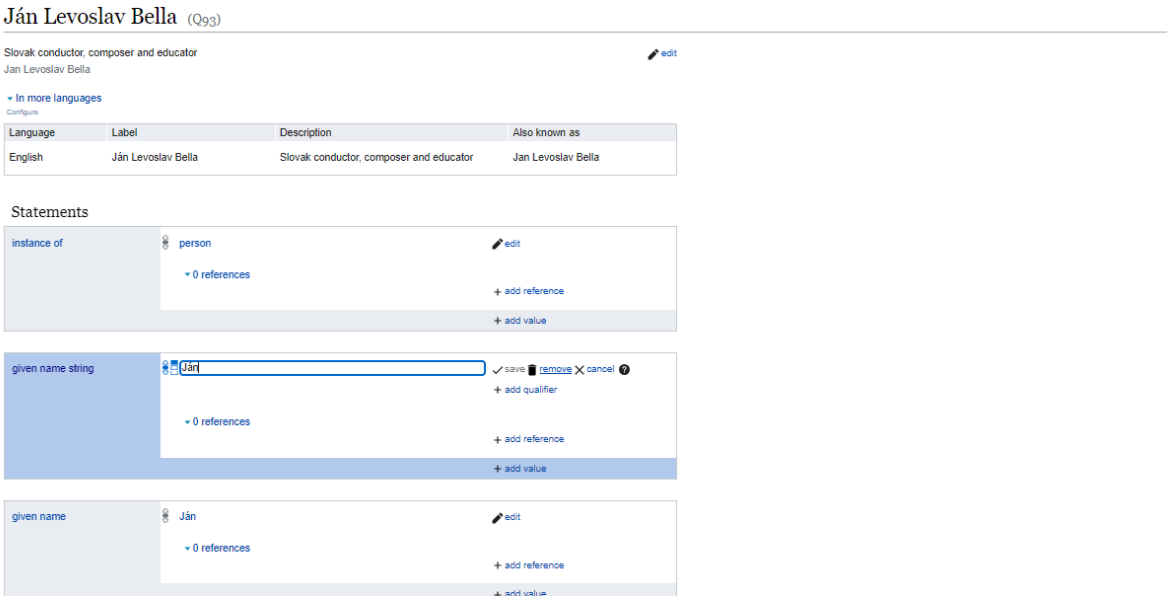


Figure 5.7: The use of controlled vocabularies (and items) have many advantages.

In this case, we would like to code the `given name` property to `Ján`. We can do it in two ways: add `Ján` as a controlled string (an item *datatype* on Wikibase) add the string *Ján* without further control.

Coding `Ján`. to Ján must be done with the knowledge of the data curator. We can only make this coding if we know that the string *Ján* came from a `given name` (or equivalent) database table column, if indeed it comes from a database of Slovak persons. This is one of the reasons why our `bots`, i.e., automated importing tools will map given names first to the `given name string` property.

Similar name string properties:

- ☐ `location of first performance (string)`, `location of creation (string)`: The strings `Bratislava`, `Bratislava, CS`, `Bratislava, SK`, `Bratislava, Austria-Hungary` or map to the `item:Bratislava`.
- ☒ `location of first performance`, `location of creation`: locations must be items of the class `city`, `town`, `village` (they all have their regional and country entities), or `region` (they have their country) or `country`. The city item Bratislava contains the knowledge that this is the current capital city of the Slovak Republic, and it is a former town in Czechoslovakia and Austria-Hungary (it has 232 statements which enrich the concept of `Bratislava`), and it is connected to lists like List of people from Bratislava.
- ☐ `venue of first performance (string)`: the string *Jesuit Church of St. Francis Xavier* will need to be matched to a venue item

☒ `venue of first performance`: Jesuit Church of St. Francis Xavier, Skalica, Slovakia as a venue item, which can be a class of building, or an atelier, or a concert hall within a building.

☐ `event of the first performance (string)`: `Prague Spring International Music Festival`—this is not a venue but a festival event.

- `event of the first performance`: Prague Spring International Music Festival is a repeating event, and it has its own entity among music festivals.

5.5 The EU Knowledge Graph

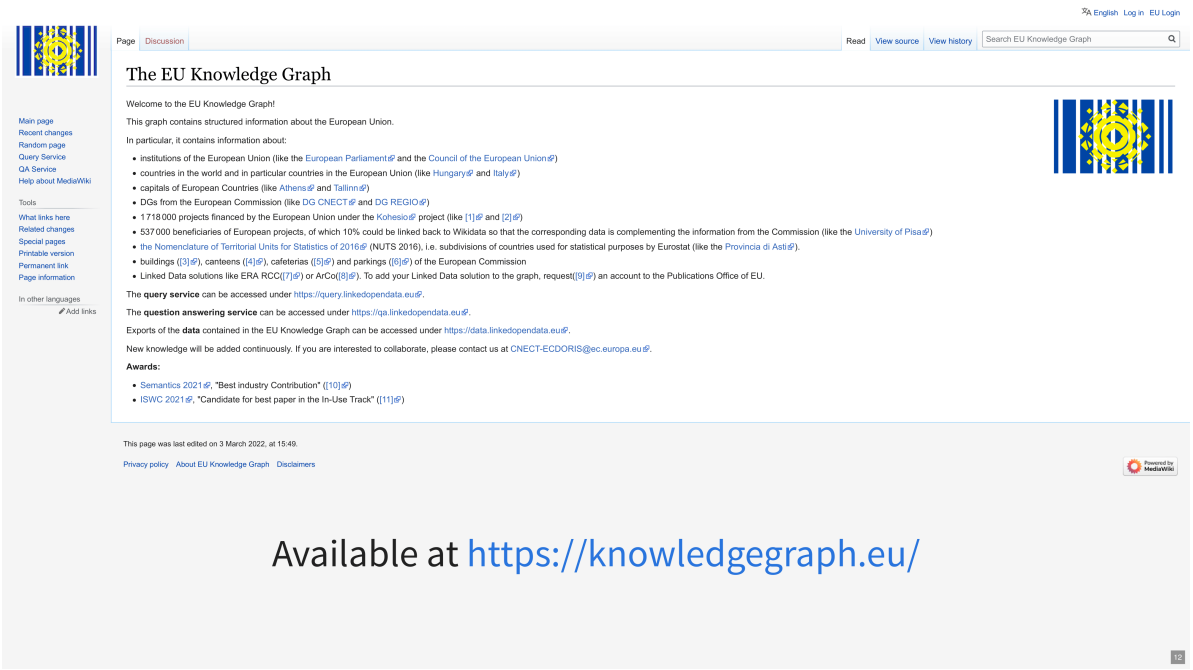


Figure 5.8: EU Academy Course: EU Knwoledge Graph

Because of the success of Wikidata, many projects and institutions are looking into Wikibase, the software that runs Wikidata. They aim to reuse the software to construct institutional or cross-institutional, domain-specific knowledge graphs. Several factors make Wikibase attractive:

☒ the fact that it is a well-maintained open-source software;

☒ there is a rich ecosystem of users and tools around it;

- ☒ Wikimedia Deutschland (WMDE), the maintainer of Wikibase, has made considerable investments in optimising the software's use outside of Wikidata or other Wikimedia projects;
- ☒ The EU Knowledge Graph runs on Wikibase;
- ☒ The *EU Academy* and the *EU Open Data Portal* actively disseminate good practices and know-how on its implementation in cross-institutional data-sharing programs.

Our OpenCollections instances are prepared with a similar mindset to the creation of the EU Knowledge Graph. We pre-populate a Wikibase instance from Wikidata about many institutional, geographical or biographical facts of the domain (Diefenbach, Wilde, and Alipio 2021), or with elements of the Wikidata data model and its compatibility classes with other ontologies.

5.6 EU Academy Course on Wikibase

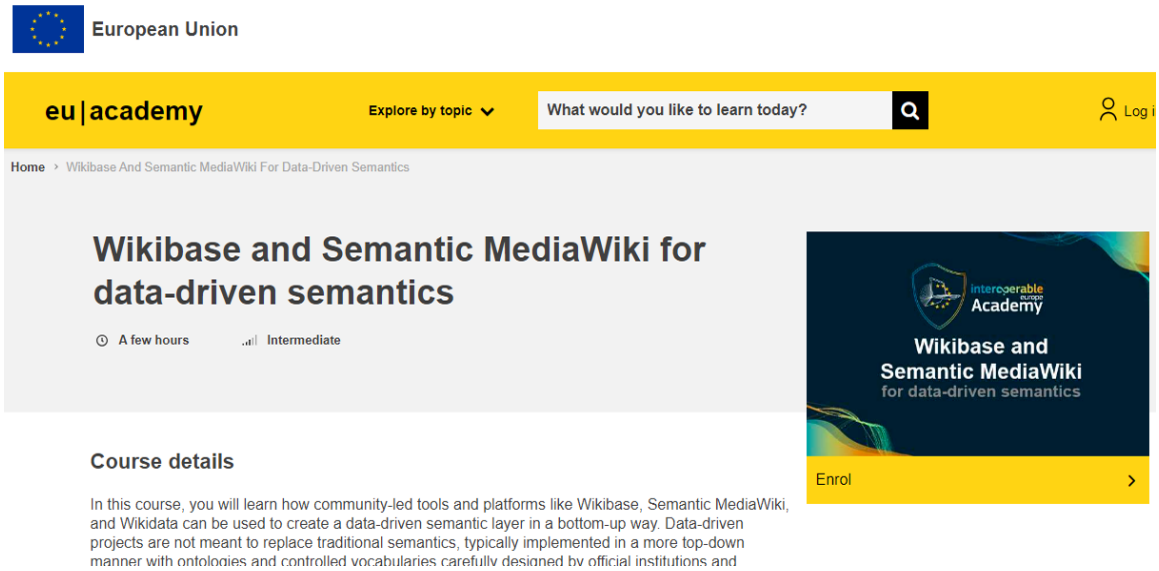


Figure 5.9: The EU Academy course on using Wikibase and Semantic MediaWiki

Target audience

Policymakers, public administrators, data maintainers, IT professionals.

Learning objectives

- Pros and cons of using Wikibase/SMW for your dataspace

- Lessons learnt from projects already using Wikibase/SMW instances
- Practical know-how about setting up a new Wikibase/SMW from scratch
- What should be on Wikidata vs in a local Wikibase/SMW
- Comparison between Wikibase and SMW

Offered by

This content is offered by the European Commission. The European Commission is the European Union's politically independent executive arm. It is alone responsible for drawing up proposals for new European legislation, and it implements the decisions of the European Parliament and the Council of the European Union.

6 Reprex's Sandbox

6.1 Create an Account

Depending on the type of MediaWiki+Wikibase instance you are using, you may need to create an account to access the site. The process may be less or more strict, depending on how much private data the instance holds.

1. Access Reprex's Sandbox Environment. Beware, we have multiple instances, so *access the instance with its URL where you have an invitation.*
2. On this page, select **Request Account**.

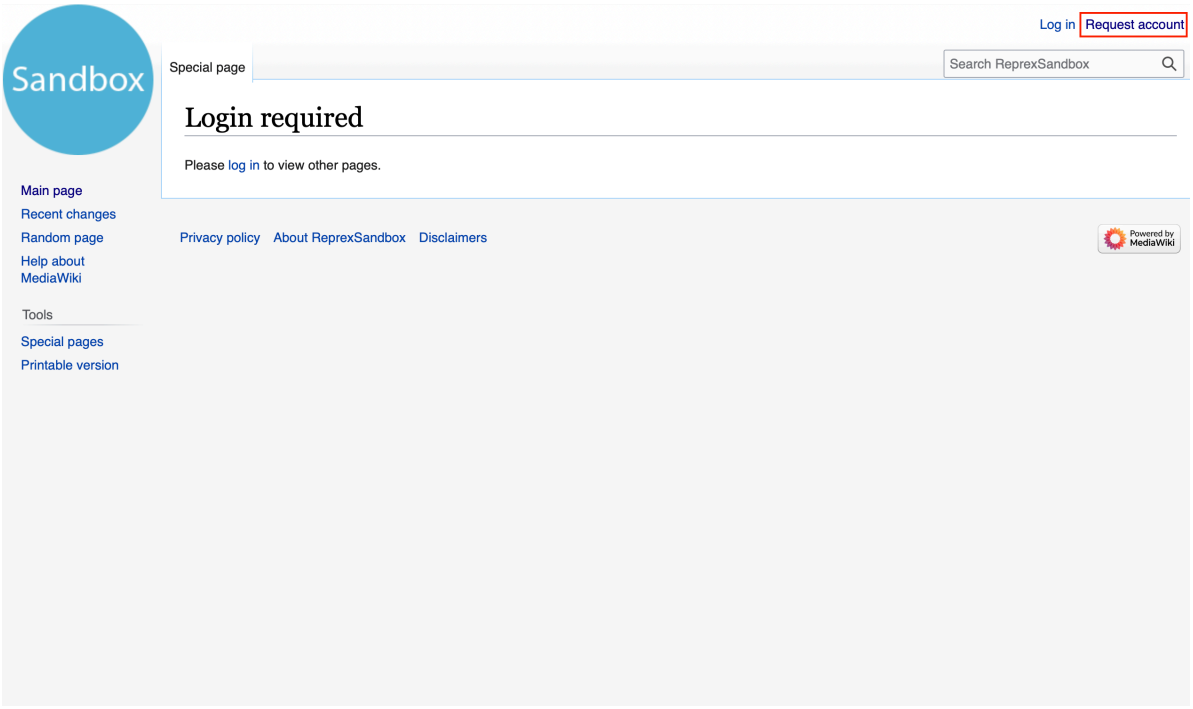


Figure 6.1: Request account - beware, we maintain several sandbox and live production instances, you must navigate to the one wher eyou really want to have this account.

3. On the next page, type in your chosen **username** and your **email address**. For a username, use a professional one that is similar to what you use on Keybase, Github, etc. Then confirm by clicking **request account** again.

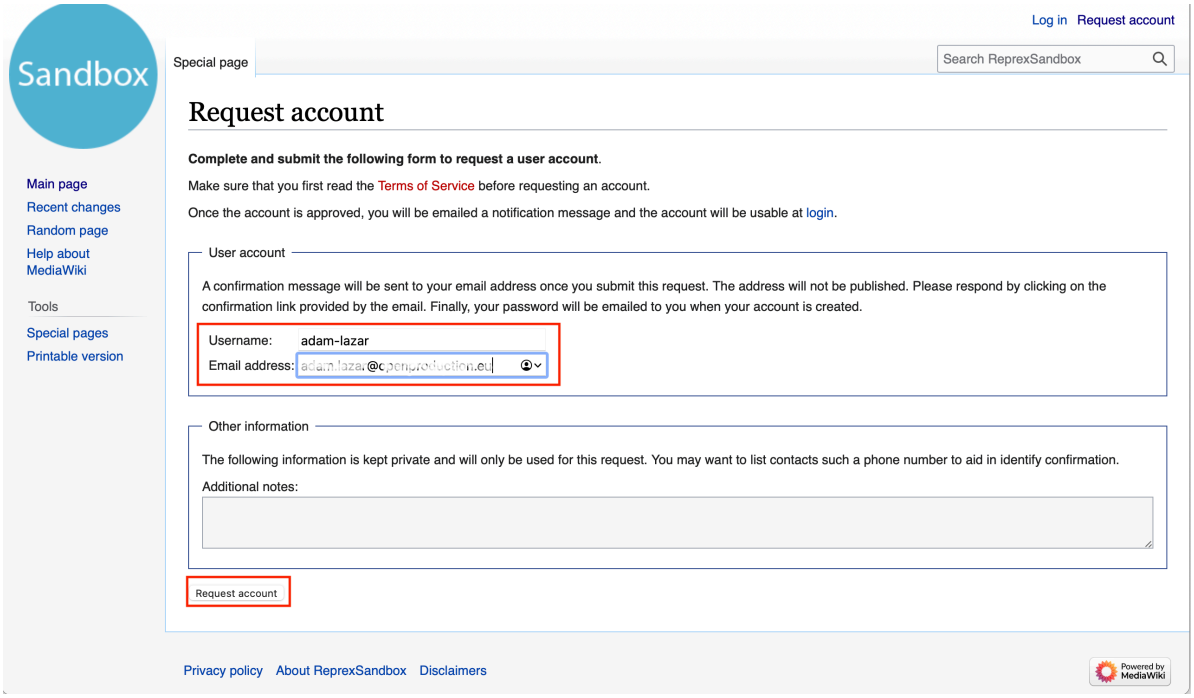


Figure 6.2: Usernames on Wikibase always start wih a capital letter, i.e., Janedoe, or Jane.doe, Or Jane.Doe.

4. Check your email inbox now. You should receive an email with a confirmation link. Click on this confirmation link. (The machine-generated email may easily go to the spam box.)

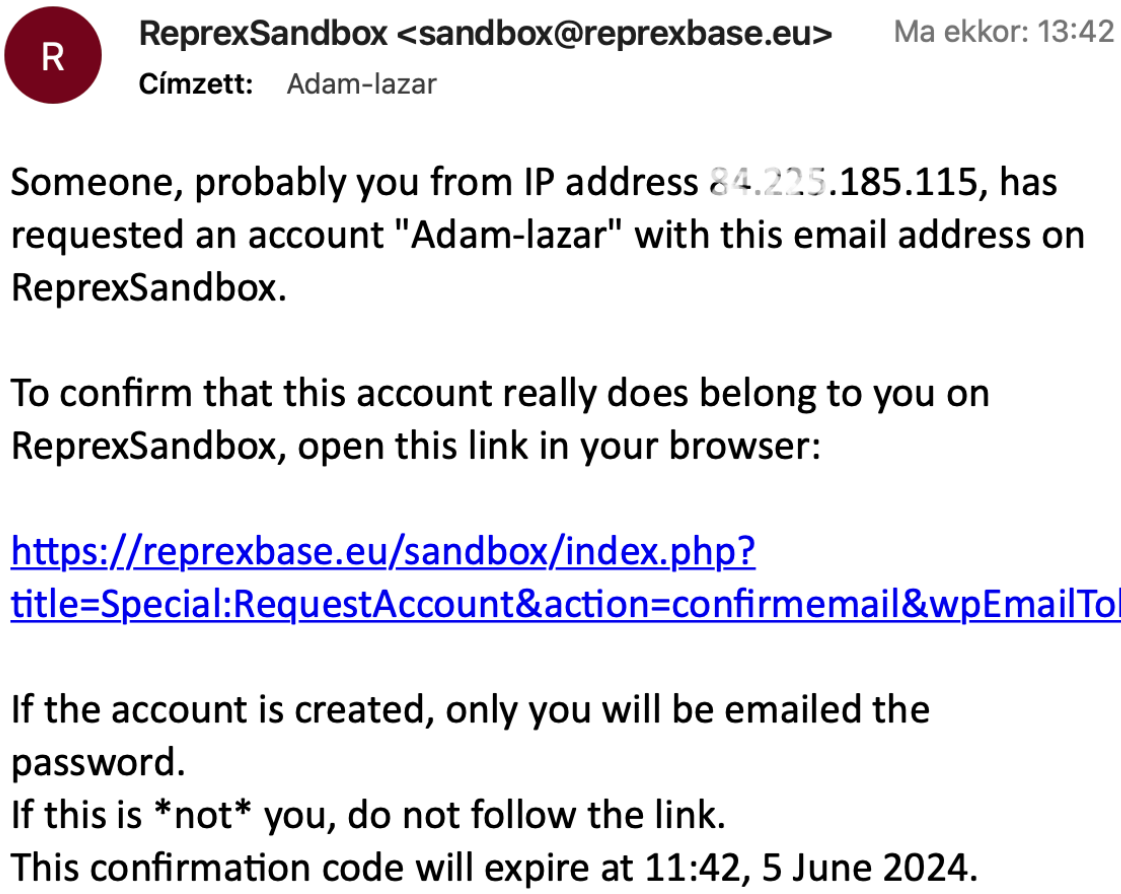
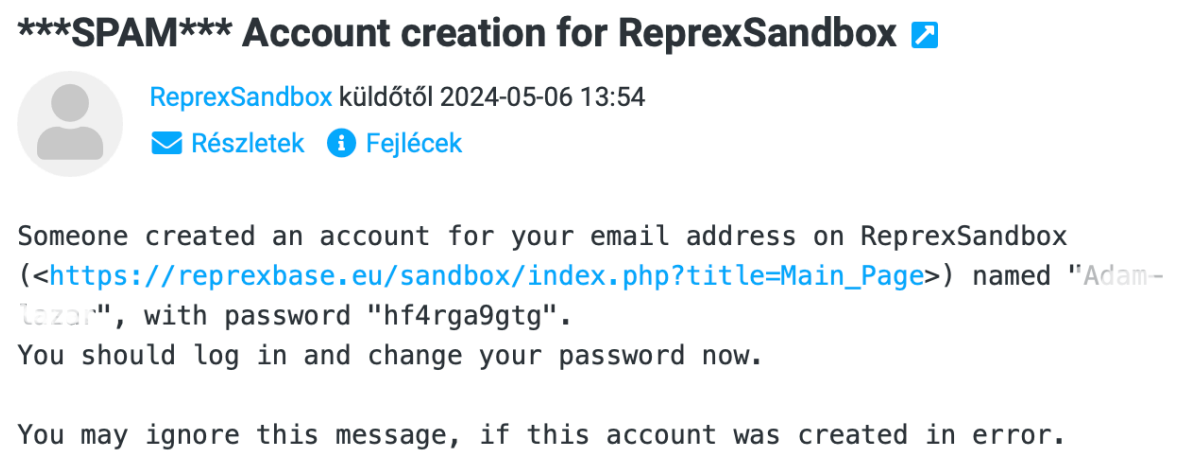


Figure 6.3: Often the confirmation mail ends up in your spam.

5. After you confirm your account request, the administrators of the Wikibase instance will evaluate it. Then, you will receive another email with your login credentials, including your temporary password.



SPAM Account creation for ReprexSandbox

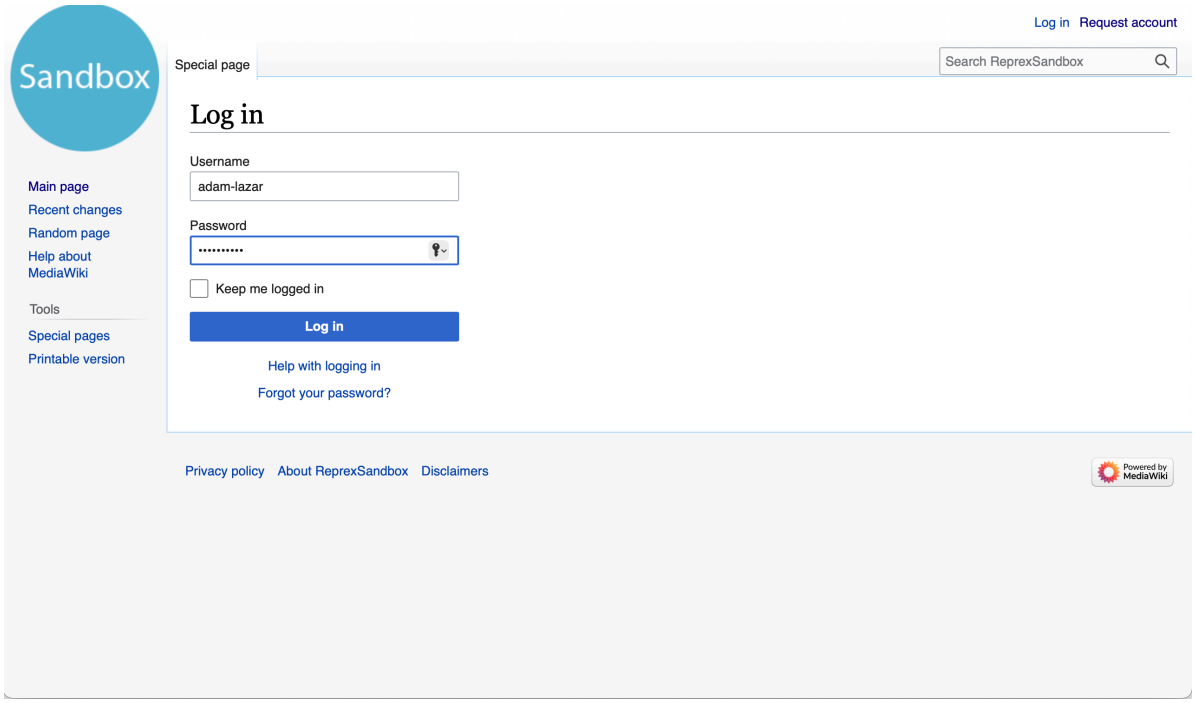
ReprexSandbox küldőtől 2024-05-06 13:54

Részletek Fejlécek

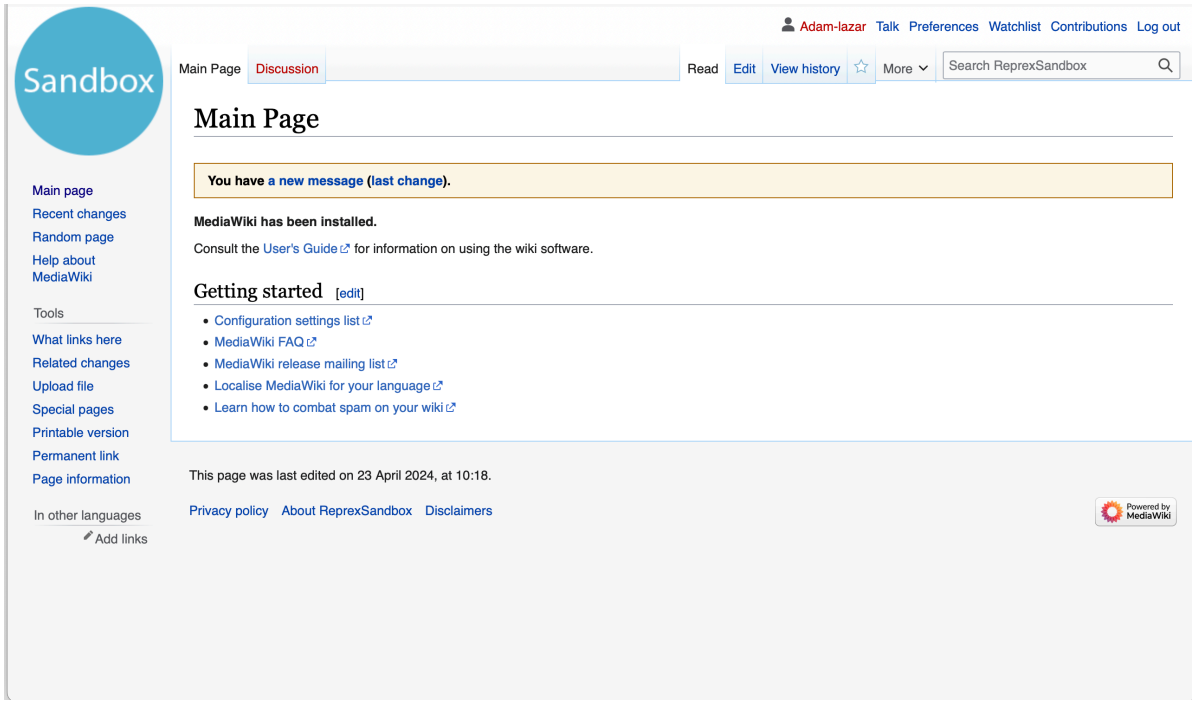
Someone created an account for your email address on ReprexSandbox (<https://reprexbase.eu/sandbox/index.php?title=Main_Page>) named "Adam-lazar", with password "hf4rga9gtg".
You should log in and change your password now.

You may ignore this message, if this account was created in error.

6. Revisit the sandbox page and log in. On the login page, type your username and the temporary password you received, then click **Log In**. You will be automatically taken to the next page, where you must change your password by typing your new permanent password. Provide your new password, then confirm it.



7. All done; you are now logged in to your account.



6.2 Editing data

6.3 Weaving Data Into the Knowledge Graph

Just because we edit data in a Wikibase instance, it will not necessarily be more usable than a spreadsheet or a simple local database. If we add `42` without a context, such as age or the number of tracks, these two numeric characters will be only literal numbers. We can increase knowledge by making every point of information a node in the knowledge graph, an edge where new information can flow in.

In Wikibase, we call these nodes entities. If we make Albert Einstein an entity instead of the string Albert Einstein, we will be able connect knowledge about his life, his scientific work, proofs, photographs of his lectures, and other forms of knowledge.

When we start importing information into a knowledge graph or begin editing and enriching information within the graph, we are faced with a crucial decision. We must determine which data points, such as cells in an original spreadsheet, or database table, or financial ledger, should be elevated to the status of nodes in the graph. These nodes, or entities, have the potential to develop their own relationships, thereby enriching the overall knowledge graph. Understanding this decision-making process empowers us to effectively utilize Wikibase for our data management needs.

6.3.1 Improving relational databases

When the aim is to improve the data quality, content, or timeliness of a relational database system, the first and most essential candidates to become entities are the database's primary and secondary keys. To recall our simple example from Section 4.1,

ID	Author	Title
My-01	Martell, Yann (Q13914)	*Life of Pi* (Q374204)
…	…	…
My-42	Adams, Douglas (Q42)	*Hitchhiker's Guide to the Galaxy* Q25169)
…	…	…

> If you can connect your My-42 entry with Q25169 on Wikidata, you can import a wealth of information into your private catalogue. And if you add Q42 to the author Douglas Adams, you can import a lot of knowledge, for example, information about his other works or the end of the copyright protection term of these books, after which they will become public domain and free for copying and distribution.

Intuitively, in Wikibase, this means that we "conceptualise" authors and their books. The person known as *Douglas Adams* becomes a human, a creator, and a writer, with all the properties that writers have… such as books. The *Hitchhiker's Guide to the Galaxy* will turn from string into the concept of a `Book`. As soon as we state that this is a book, not merely a text, we can start adding book-specific knowledge to the `Hitchhiker's Guide to the Galaxy` book entity: ISBN number, first publication date, translations. And what is most important, we can connect this entity with the author, `Douglas Adams`, who is no longer just one of the many people who are known by this name, but the person who wrote quiet humorous books.

Conceptualisation is possibly manually, as we have shown in Section 4.1; but usually we do this after data modelling with bulk importing. You tells us what is your data about: books and author, and we import them as `Books` and `Authors`, so that we can start to look for more information about these books and authors in various knowledge systems.

6.3.2 Improving spreadsheet databases

Smaller organisations often do not use relational databases; instead, they use Excel or OpenOffice spreadsheets maintained by workers, often for decades. Turning such spreadsheets into knowledge base elements is similar to working with a relational database, but sometimes, it is a smaller and more difficult task.

Well-organised spreadsheets can be good databases because spreadsheet applications like Excel, OpenOffice, or Google Spreadsheet allow the use of primary and secondary keys by connecting worksheets and the creation of pivot tables.

The key challenge with spreadsheets is identifying the Things that should become entities. What is your spreadsheet about? Buildings? Then, addresses and building names should become entities and nodes in the graph. Addresses keep changing, building geometries keep changing, and new additions are built or demolished. Street names change. Even city names change; cities merge and divide.

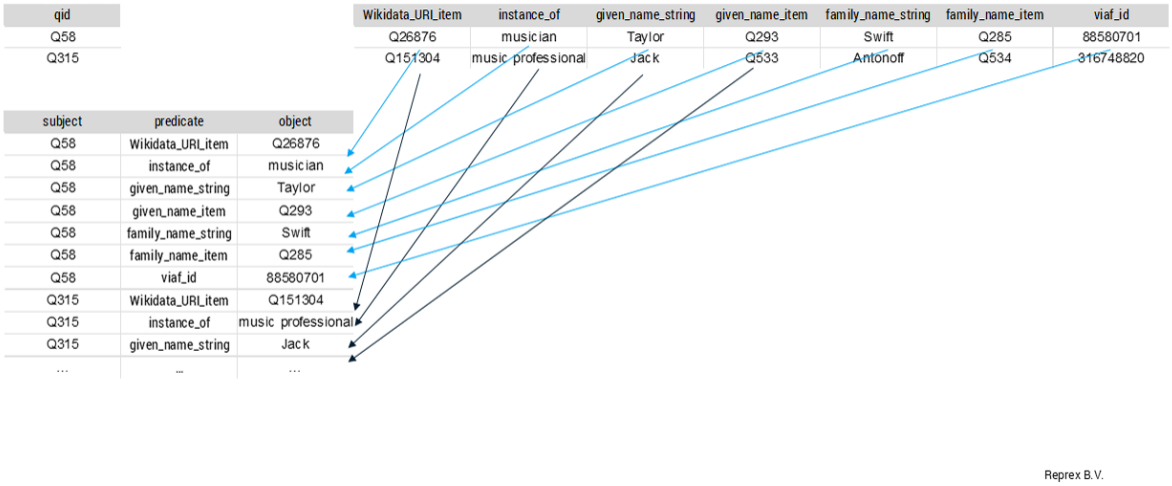
6.3.3 Improving annotated text, legal documents, lab notes, regulatory filings

6.3.4 Creating new indicators

7 Bulk import

7.1 Organise your data

In the Chapter 2 chapter on tidy data we have shown one of the advantages of a tidy dataset: it can be pivoted into a sequence of triple-form semantic statements. This is possible because tidy format is unambiguous: we always know that a number or string (value) belongs to its observational subject (in the rows) and the measured property variable (in the columns). In other words, the meaning of a cell is unambiguous, because we know the subject (from the rows) and the predicate (from the column headings.)



In the Chapter 3 chapter we have seen that naming is hard. Weather we are talking about people, objects, or table variables, it is difficult to come up with good names. Most programmers and open-source communities apply variable naming conventions.

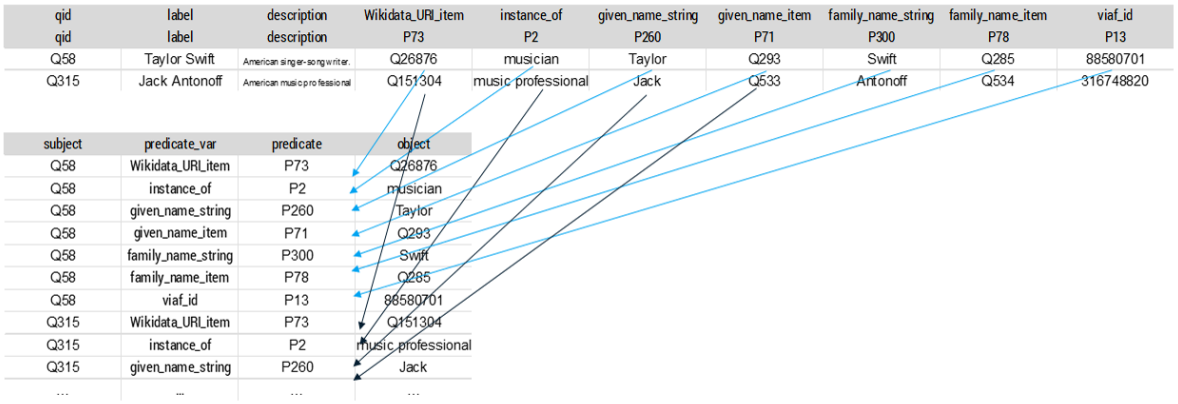
We apply the `snake case` convention, which creates variables like `given_name_string`. We make these names tidier with grouping semantic elements into the beginning or ending of the variable name; this way the variable name can be filtered easily.

When importing into Wikibase, we need to know what should be the `type` of the imported data. Shall we use a string for `Taylor` and `Swift`, or we want to create an entity for the name (variants) of `Taylor`?

- ☒ The Q number (`QID`) is unique to each Wikibase, including your Wikibase or Wikidata. The `qid` variable should always contain your QID.
- ☐ If you want to connect your knowledge base with the public Wikidata and other wiki products, use the `Wikidata URI item` property to record the Wikidata QID, too. For example, in our demo wikibase the QID of Taylor Swift (the singer) is Q58, whereas on Wikidata it is Q26876.
- ☒ Apply the `_string` ending if the variable should be imported as a string; in this case, it cannot form a node in the knowledge graph (no further relations can be made.) The `given_name_string` should import `Taylor` as a character string.
- ☒ Apply the `_item` ending if the variable should be imported as as entity—in Wikibase, these entities are called items.The `given_name_item` should import Taylor (Q3981665) as an item that can be the node in the knowledge graph. You can connect further statements (elements of knowledge) to nodes and therefore nodes can be made intelligent. For example, Taylor (Q3981665) states that this is a unisex name, and it is wrong to assume that most Taylors are women.
- ☒ Apply the `_date` ending if the variable should be imported as date; for example `inception_date`, `birth_date`. Dates are points in time, and they must be converted to a time type. (See: Dates.)
- ☒ Apply the `_id` ending if the variable should be imported as an actionable or not actionable external identifier; for example, `viaf_id`, or `my_database_id`. If the ID is actionable, like VIAF or ISNI, we will make them actionable, making sure that `88580701` as a `viaf_id` points to https://viaf.org/viaf/88580701.
- ☒ Apply the `_url` ending if the variable should be imported as an URL, and not an actionable URI; for example `official_website_url`.
- ☒ Apply the `_monotext` for monolingual text types.
- ☐ We have not yet written code to import geographical information, music notation, and some special data types.

7.1.1 Correspondence

Applying dual headings can help to map your column variables into Wikibase properties easily while pivotting into longer format.



qid	label	description	Wikidata_URI_item	instance_of	given_name_string	given_name_item	family_name_string	family_name_item	viaf_id
qid	label	description	P73	P2	P260	P71	P300	P78	P13
Q58	Taylor Swift	American singer-songwriter.	Q26876	musician	Taylor	Q293	Swift	Q285	88580701
Q315	Jack Antonoff	American music professional	Q151304	music professional	Jack	Q533	Antonoff	Q534	316748820

subject	predicate_var	predicate	object
Q58	Wikidata_URI_item	P73	Q26876
Q58	instance_of	P2	musician
Q58	given_name_string	P260	Taylor
Q58	given_name_item	P71	Q293
Q58	family_name_string	P300	Swift
Q58	family_name_item	P78	Q285
Q58	viaf_id	P13	88580701
Q315	Wikidata_URI_item	P73	Q151304
Q315	instance_of	P2	music professional
Q315	given_name_string	P260	Jack
...	...	...	...

8 OpenCollections

`OpenCollections` aims to provide a high degree of technical, syntactic, and semantic interoperability among the data systems of the partners in the data-sharing space. It imports data (or data maps) into a graph format, which is optimal for using heterogeneous data sources. Our innovative solutions aim to make this complex process as fast and weightless as possible.

The system is built around Wikibase, an information management software developed by Wikipedia based on the MariaDB relational database management system. Wikibase manages the world's most extensive open knowledge graph, Wikidata, and enables users to work in many natural languages with little or no IT or information science knowledge. Many use cases, including the creation of the EU Knowledge Graph, inspired us because Wikibase has a much lower learning need than more optimised graph database management systems.

`OpenCollections` improves the Wikibase experience with automated data-importing components with suitable job aids for users and exporting tools into more complex graphs that can provide data for training trustworthy AI systems.

- ☒ We understand the importance of compatibility. That's why we provide tools for mass importing data and schematic information from existing relational database management systems like MySQL, PostgreSQL, or simpler, spreadsheet-based data sources. This reassures our users that OpenCollections can seamlessly integrate with their existing systems, providing a secure and confident data management experience.
- ☒ We provide training and job aids for manual data processes to keep partners' domain-level experts in the loop and provide human agency and oversight for trustworthy AI systems.
- ☒ We create a model supported by automation that translates the data held in Wikibase to standard machine-actionable ontologies like CIDOC, EDM, RiC, and DCAT-AP.

8.1 Going Beyond Wikibase

Our system is inspired by the WB-CIDOC model developed at the University of Helsinki for translating knowledge stored in Wikibase into the statements described with the CIDOC ontology used by intelligent cultural heritage systems (Kesäniemi, Koho, and Hyvönen 2022). CIDOC is a modern, events-based ontology that allows building trustworthy inference and deduction AI engines.

The WB-CIDOC provides rules for writing data into Wikibase in a way that translates correctly into an event-based model, but we find its use counter-intuitive and laborious for domain expert data curators.

Most domain experts would think that a biographical entity of `Albert Einstein` should have a birthday property with the date of `March 14, 1879`, while an event-based ontology would create first an abstract event, the `Birth of Albert Einstein`, with a timespan of `March 14, 1879, 0:00` to `23.59`. It is far easier to search for parallel events in this time window or connect further information— like persons present at birth, certificates created, etc.—than to connect this information to a simple, literal date.

Domain-level experts like copyright specialists, ESG experts, musicologists, bank professionals, and other users usually need formal computer- or information science training and find the entity-based approach closer to real-world experience. We design our knowledge-base instances with hooks for more complex knowledge-base ontologies. This allows our users to review the information in a natural, entity-based format; our intelligent applications translate the information to more complex structures, such as event-based conceptual models, to allow more reasoning capacity for our AI systems.

8.1.1 Translation to more complex data models

Our system is inspired by the WB-CIDOC model developed at the University of Helsinki for translating knowledge stored in Wikibase into the statements described with the CIDOC ontology used by intelligent cultural heritage systems (Kesäniemi, Koho, and Hyvönen 2022). CIDOC is a modern, events-based ontology that allows building trustworthy inference and deduction AI engines.

The WB-CIDOC provides rules for writing data into Wikibase in a way that translates correctly into an event-based model, but we find its use counter-intuitive and laborious for domain expert data curators.

Most domain experts would think that a biographical entity of `Albert Einstein` should have a birthday property with the date of `March 14, 1879`, while an event-based ontology would create first an abstract event, the `Birth of Albert Einstein`, with a timespan of `March 14, 1879, 0:00` to `23.59`. It is far easier to search for parallel events in this time window or connect further information— like persons present at birth, certificates created, etc.—than to connect this information to a simple, literal date.

> **Note**
>
> Domain-level experts like copyright specialists, ESG experts, musicologists, bank professionals, and other users usually need formal computer- or information science training and find the entity-based approach closer to real-world experience. We design our knowledge-

base instances with hooks for more complex knowledge-base ontologies. This allows our users to review the information in a natural, entity-based format; our intelligent applications translate the information to more complex structures, such as event-based conceptual models, to allow more reasoning capacity for our AI systems.

8.1.2 Record-keeping and retention

National archives play a crucial role in preserving the collective memory and history of a nation. Connecting national archives to institutional enterprise record-keeping systems has many advantages.

1. **Contextualising institutional or enterprise records**: Private organisations and users cannot copy all legally or historically relevant documents in their inventory. Connecting to memory institutions, such as records or legal databases, allows one to find precedents and understand one and one's own historical records in context without the need to hoard information on an excessive scale. Just the way we do not need to burden our office bookshelves with bilingual dictionaries or printed copies of changing regulations, we can further lower the burden by making our records system compatible with national records.
2. **Record retention and public archiving** is a regulated process that serves as the foundation of many business processes' regulatory or assurance oversight. Businesses often must deposit copies of legally important disclosures and certificates at public bodies. Larger institutions, primarily if they work for the public benefit, usually have a legal mandate to place some of their documents into a public archive. Private persons and companies often donate documents to such archives when they want to be credited with their work, intellectual property, or the value of their activities.

Because OpenCollections is based around a document-based database, it is very well suited to support document exchanges between private institutions (e.g., the exchange of technical and delivery documentation along the supply chain), public institutions (e.g., the exchange of public documents), and public-private exchanges.

We provide mappings, software tools and training to apply Records in Context (RiC), a novel ontology released in 2023 after over a decade's work to replace the four international standards on archiving. The last international standards on the topic were created before the commercial Internet; RiC provides backwards compatibility to millennia of historical records, corporate document inventories, and physical data vaults on one hand, and opens up the use of modern knowledge graphs to link information in the archives with your documents in use. RiC is the gateway to corporate and institutional textual big data.

8.1.3 Data catalogues, and the meaning of data tables

Following the DCAT-AP specification of the EU Open Data Portal and Stat-DCAT-AP to offer full compatibility with European statistical portals and open data portals, we translate information about datasets, data codes and structures, and variable descriptions. This translation works with few limitations for global resources beyond Europe. It connects corporate or institutional datasets and accounts with statistical and national accounts data from public sources, offering unparalleled ease in creating economic or sustainability-controlling applications.

8.1.4 Collections and inventories

References

Allamanis, Miltiadis, Barr, Earl T., Bird, Christian, and Sutton, Charles. 2015. "Suggesting Accurate Method and Class Names." In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, 38–49. Bergamo, Italy. https://dl-acm-org.proxy.uba.uva.nl/doi/abs/10.1145/2786805.2786849.

Berners-Lee, Tim, Roy T. Fielding, and Larry M. Masinter. 2005. "Uniform Resource Identifier (URI): Generic Syntax." Request for Comments RFC 3986. Internet Engineering Task Force. https://doi.org/10.17487/RFC3986.

Berners-Lee, Tim, James Hendler, and Ora Lassila. 2001. "The Semantic Web." Scientific American, Incorporated.

Dallas, Costis. 2016. "Digital Curation Beyond the 'Wild Frontier': A Pragmatic Approach." *Archival Science* 16 (4): 421–57. https://doi.org/10.1007/s10502-015-9252-6.

Data Documentation Initiative. 2020. "DDI Lifecycle (3.3) Documentation." https://ddi-lifecycle-documentation.readthedocs.io/en/latest/index.html.

Diefenbach, Dennis, Max de Wilde, and Samantha Alipio. 2021. "Wikibase as an Infrastructure for Knowledge Graphs: The EU Knowledge Graph." In *ISWC 2021*. Online, France. https://hal.science/hal-03353225.

Harpring, Patricia, and Murtha Baca. 2016. "19. Art Vocabulary: Categorizing Works of Art." In *Handbuch Sprache in Der Kunstkommunikation*, edited by Heiko Hausendorf and Marcus Müller, 425–54. Berlin, Boston: De Gruyter. https://doi.org/doi:10.1515/9783110296273-020.

Hartmann, Thomas, Sarven Capadisli, Franck Cotton, Richard Cyganiak, Arofan Gregory, Benedikt Kämpgen, Olof Olsson, Heiko Paulheim, Joachim Wackerow, and Benjamin Zapilko. 2024. "DDI-RDF Discovery Vocabulary. A Vocabulary for Publishing Metadata about Data Sets (Research and Survey Data) into the Web of Linked Data." Edited by Thomas Hartmann, Richard Cyganiak, Joachim Wackerow, and Benjamin Zapilko. W3C. https://rdf-vocabulary.ddialliance.org/discovery.html.

Kesäniemi, Joonas, Mikko Koho, and Eero Hyvönen. 2022. "Using Wikibase for Managing Cultural Heritage Linked Open Data Based on CIDOC CRM." In *New Trends in Database and Information Systems*, edited by Silvia Chiusano, Tania Cerquitelli, Robert Wrembel, Kjetil Nørvåg, Barbara Catania, Genoveva Vargas-Solar, and Ester Zumpano, 542–49. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-15743-1_49.

Meeus, Sofie, Wouter Addink, Donat Agosti, Christos Arvanitidis, Bachir Balech, Mathias Dillen, Mariya Dimitrova, et al. 2022. "Recommendations for interoperability among infrastructures." *Research Ideas and Outcomes* 8 (October). https://doi.org/10.3897/rio.8.e96180.

Paskin, N. 1999. "Toward Unique Identifiers." *Proceedings of the IEEE* 87 (7): 1208–27. https://doi.org/10.1109/5.771073.

Paskin, Norman. 2003. "Identification and Metadata." In *Digital Rights Management: Technological, Economic, Legal and Political Aspects*, 26–61. Lecture Notes in Computer Science 2770. Berlin: Springer.

Pomerantz, Jeffrey. 2015. *Metadata*. The MIT Press Essential Knowledge Series. Cambridge, MA, USA: MIT Press.

UNECE. 2014. "Generic Statistical Information Model. GSIM V2.0 Documents. UNECE Statswiki." 2014. https://statswiki.unece.org/display/gsim/GSIM+v2.0+documents.

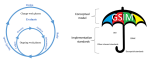
Vardigan, Mary, Pascal Heus, and Wendy Thomas. 2008. "Data Documentation Initiative: Toward a Standard for the Social Sciences." *International Journal of Digital Curation* 3 (1): 107–13.

A Question Bank Items In Wikibase

In this guide you are going to learn how to feed different types of questions and information related to them to Wikibase.

First, you will read about how to load into Wikibase different type of questions. Second, you will learn about different types of information and how to link them to your question. Last, you will see how to add different translations to a question.

The concept behind the workflows follows the DDI-Lifecycle framework. The DDI lifecycle model is designed to support the documentation and management of data throughout its entire lifecycle, ensuring that data can be effectively shared, reused, and preserved. DDI-Lifecycle is not yet available in RDF. Some elements of DDI are standardised; others are not. Whenever possible, we use the standardised *DDI-RDF Discovery Vocabulary*; when no such ontology is present, we create our interpretation of DDI Hartmann et al. (2024).



DDI is part of the **General Statistical Information Model** (GSIM)(UNECE 2014), which accompanies the **Generic Statistical Business Process Model** (GSBPM) as an international standard model that "describes and defines the set of business processes needed to produce official statistics." We use the conceptualisation of GSIM so that our results will be similar in quality to official statistics; of course, similar processes allow us to create products that combine well with official statistical products.

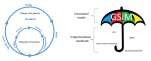


Figure A.1: The two implementation standards, DDI and SDMX ensures that our observatories can work together with statistical offices, or respectable social sciences data repositories.

Applying the *Data Documentation Initiatve* (DDI) will ensure that we will remain compatible with official statistical microdata and metadata services and other social sciences archives, like GESIS, the official data archive of all European Commission-mandated survey research dating back over 50 years (Vardigan, Heus, and Thomas 2008).

A.1 Need for Questions

The need for questions arises from the fact that you want to collect some data systematically, either in an online or face-to-face questionnaire, in a structured interview, or in making data requests to an API or a form to record repeated answers to this question. After statistical manipulations, such as summarising and averaging, the responses will create empirical variables in a dataset. You can rely on existing data and expand your knowledge utilising already collected (open) data if you use the same questions that other researchers or statisticians have used before you.

Without addressing the theory of data harmonisation, these are the steps you are likely to make:

1. For example, if you need data on reusable plastic bags, you need to find a widely shared definition of `plastic`, `reusability` and `bags`.
2. You should consult a database or a question bank to find out if others have already asked about reusable plastic bags.
3. If you formulate a questionnaire using the exact wording and answering options as earlier surveys on attitudes to reusable plastic bags, you will be able to compare the results. So, you need access to question forms (question texts) and answer options.
4. If you work on an international project or want a global comparison, you will need to ensure that the question texts and answer options are translated very similarly and understood equally in different languages.
5. As a practical last step, the responses must be coded the same way as in international data repositories; for example, female respondents are coded with `F` in most statistical data repositories, even if the word 'female' may start with a different letter in many languages.

Our question bank is designed to be searchable by concepts, question types, question labels or question texts.

A.2 Question Types

What is a survey question after all? DDI organises questions into 3+1 hierarchical levels.

- ☒ `Question`: A question and its answer options formulated in at least one natural language, for example, `Please tell me to what extent you agree or disagree with the following statement: "I trust that products carrying the EU ecolabel are environmentally-friendly." [ ] Totally agree [ ], Tend to agree [ ], Tend to disagree [ ], Totally disagree [ ], DK`. In this example DK refers to declined to comment on the question, or a refusal to answer this question.

- ☒ `QuestionItem`: the concept (i.e., unemployment, or plastic bags) being measured by the question, text for the question, response domain information, clarifying instructions, external aids (clarifying objects used in presenting the question to the respondent), Input and Output Parameters and Bindings, allowed response cardinality and estimation of the time required to respond.
- ☒ `QuestionBlock`: This structure is intended to bundle together a set of questions (items and/or grids) that have meaning only about a specified object expressed as the evaluation material. This form of question set is common in educational testing where a text, image, or other material is provided, and the respondent is asked questions specific to the material. For example, a portion of a play script is provided, and the respondent is asked questions concerning the dialogue and/or stage directions provided in the script. Note that the intent of QuestionBlock is not to bundle together a set of questions that are commonly used together or used in a specified order.
- ☒ `QuestionGroup` is only for administrative purposes.

A.2.1 Model question

One other way to make questions and resulting responses and their statistically processed variables comparable is to ask questions about different concepts in a same way? A standard quesiton in a Cultural Access and Participation Survey is:

`How many times in the past 12 months have you been to ... ... a concert? ... cinema? ... church?`

While people may have recollection biases about the 12 months, and may use a bit differently the concept of concert or cinema, because of the same syntax, context, we can assume that their responses are comparable. In this case, `How many times in the past 12 months have you been to ...` is a model question.

A model question is a question template that can create simple questions or question items in question grids or blocks.

- URI: Q127
- label: `Trust in EU ECOLABEL [model]`
- questionText (description): `Please tell me to what extent you agree or disagree with the following statement: "I trust that products carrying the EU ecolabel are environmentally-friendly." [ ] scale`

And a question based on a model question, taken from

- QID: https://reprexbase.eu/demowiki/index.php?title=Item:Q111
- label: `Trust in EU ECOLABEL`

- questionText (description): `Please tell me to what extent you agree or disagree with the following statement: "I trust that products carrying the EU ecolabel are environmentally-friendly." [ ] Totally agree [ ], Tend to agree [ ], Tend to disagree [ ], Totally disagree [ ], DK`
- variable representation: scale representation base type
- study (DDI): Eurobarometer 88.1 (2017)

Our model questions follow one of the following formats:

- `questionText (description), [ ] concept`, where the question connects to a concept, such as environmental protection (Q131).
- `questionText (description), [ ] scale`, where the answers are on a scale, for example Estimated number of employees in FTE [model] (Q123)
- `questionText (description), [ ] category`, where the answer options are categories, for example: Reduced use of single use plastic bags [model] (Q112)
- `questionText (description), [ ] ranking`, where the respondent has to create a rank from the answer options, for example: Important environmental issue [model] (Q143)
- `questionText (description), [ ] concept, [ ] scale,` where beside the model question there are other sub-questions as well, for example: Important for reduction of plastic [model] (Q128)

Based on the DDI-Lifecycle model we could generate differently structured model questions, and if there will be user need, we will introduce further question templates.

The DDI-Discovery ontology requires the questions to take this format:

> *Please tell me to what extent you agree or disagree with the following statement: "I trust that products carrying the EU ecolabel are environmentally-friendly." [ ] Totally agree [ ], Tend to agree [ ], Tend to disagree [ ], Totally disagree [ ], DK.*

This is a good representation to for an existing questionnaire, but it is not really good for a questionbank, because in some cases, the agreement scale may be a 3-level, in others, a 5-level agreement scale:

> *Please tell me to what extent you agree or disagree with the following statement: "I trust that products carrying the EU ecolabel are environmentally-friendly." [ ], Agree [ ], [ ], Disagree [ ], DK*

We can argue that the responses are still comparable, but `[ ] Totally agree [ ], Tend to agree [ ]` should be added together for a broader `[ ] Agree` category if one survey uses the 5-scale version of the response scale while the other uses the 3-scale (agree, disagree, decline) version.

This is why we separately record the model question, the subquestions and the answer options.

A.2.2 Simple, Multiple Choice and Matrix Questions

Different question types have different elements. Some questions consist of one question, others have group questions and several sub-questions.

A question might consists of the following elements:

`[Model question] + [Question Items] + [Answer options]`

- All elements should be added separately to the Wikibase
- The format changes based on the type of the question:
  - simple question: `[Model question] + [Answers]`
  - multiple choice question: `[Model question] + [Question Items]`
  - matrix question: `[Model question] + [Question Items] + [Answer options]`

Let's see how to load into Wikibase:

- Simple Questions
- Matrix Questions
- Multiple Choice Questions

A.2.2.1 Simple Questions

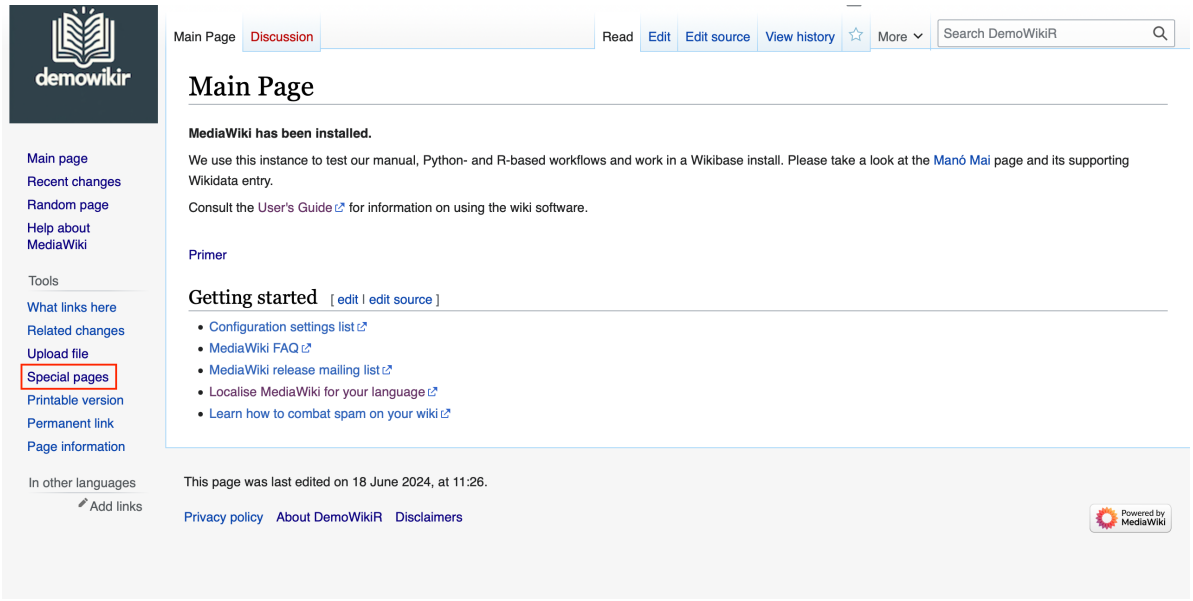
In case of Simple Questions there's only one question.

> **Note**
>
> For a clearer definition, see the disco:Question class.

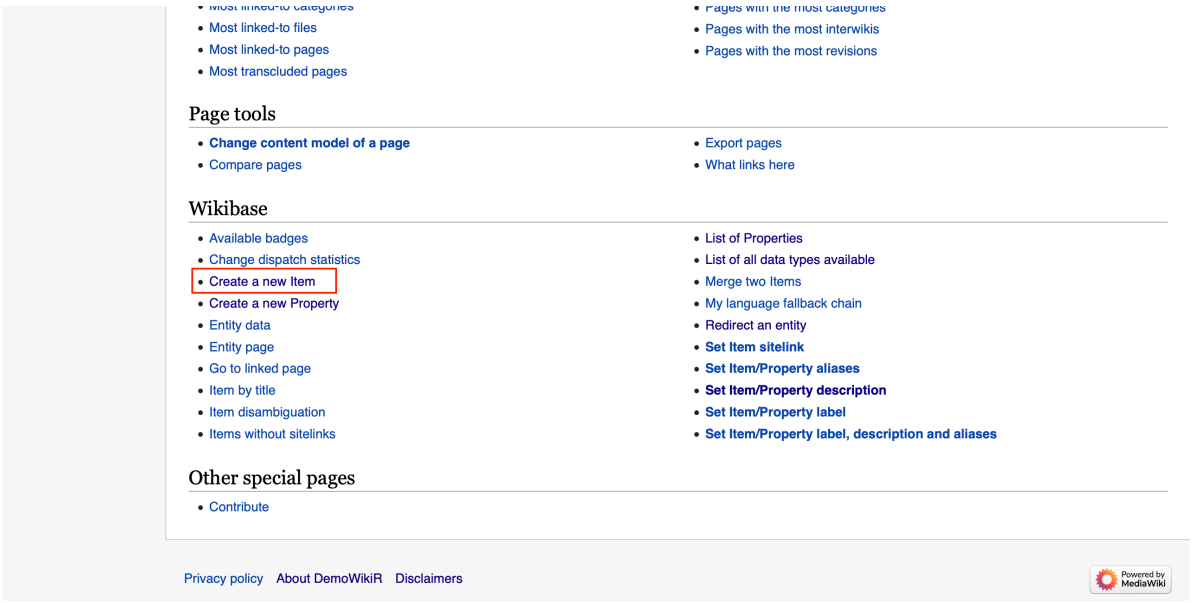
QD12	Please tell me to what extent you agree or disagree with the following statement: "I trust that products carrying the EU ecolabel are environmentally-friendly." Model Question	
	(READ OUT - ONE ANSWER ONLY)	
	Totally agree	1
	Tend to agree	2
	Tend to disagree	3
	Totally disagree	4
	DK Answer Options	5
	NEW	
	ASK ALL	

The format of a simple question is: `[Model question] + [Answer options]`

Let's see how to create a simple question entry in Wikibase. Go to "Special Pages"

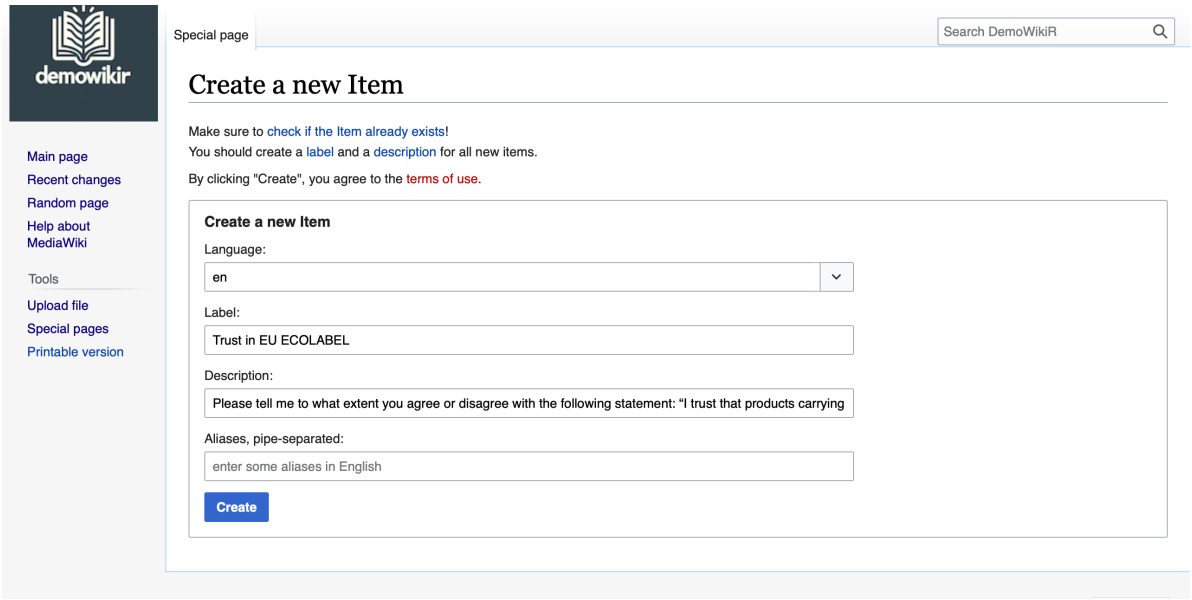


Scroll down and select: `Create a New Item`



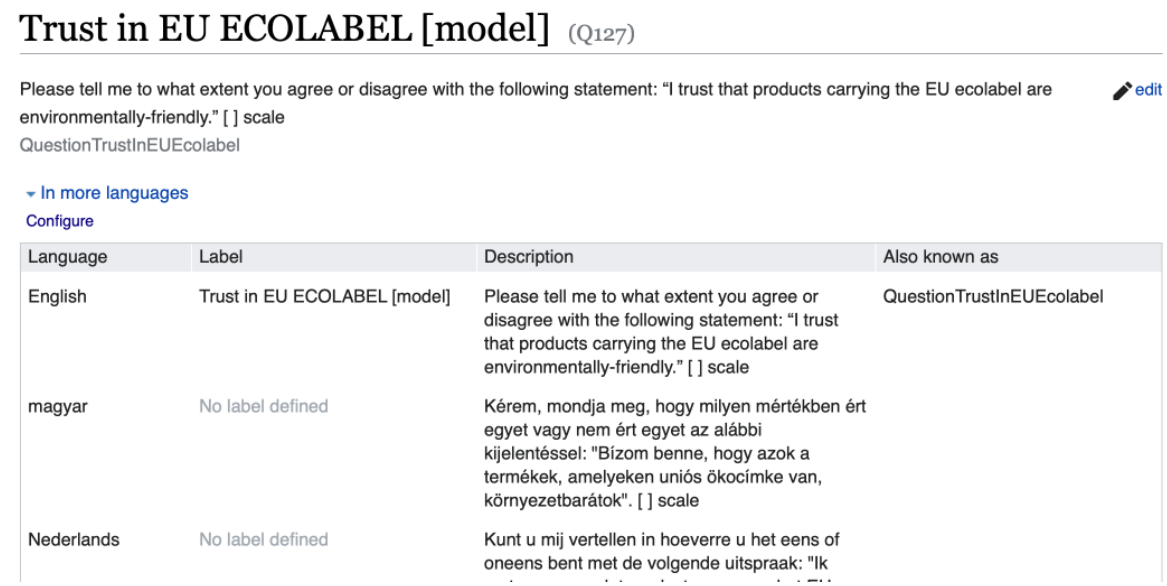
Fill the form with the question's data:

- Language Choose the language (en)
- Label - Give a short name for the question
- Description - Enter the question itself in the format specified above.
- Aliases - leave it empty



Click `Create`.

The question now is created on Wikibase.



Trust in EU ECOLABEL [model] (Q127)

Please tell me to what extent you agree or disagree with the following statement: “I trust that products carrying the EU ecolabel are environmentally-friendly.” [ ] scale edit

QuestionTrustInEUEcolabel

In more languages

Configure

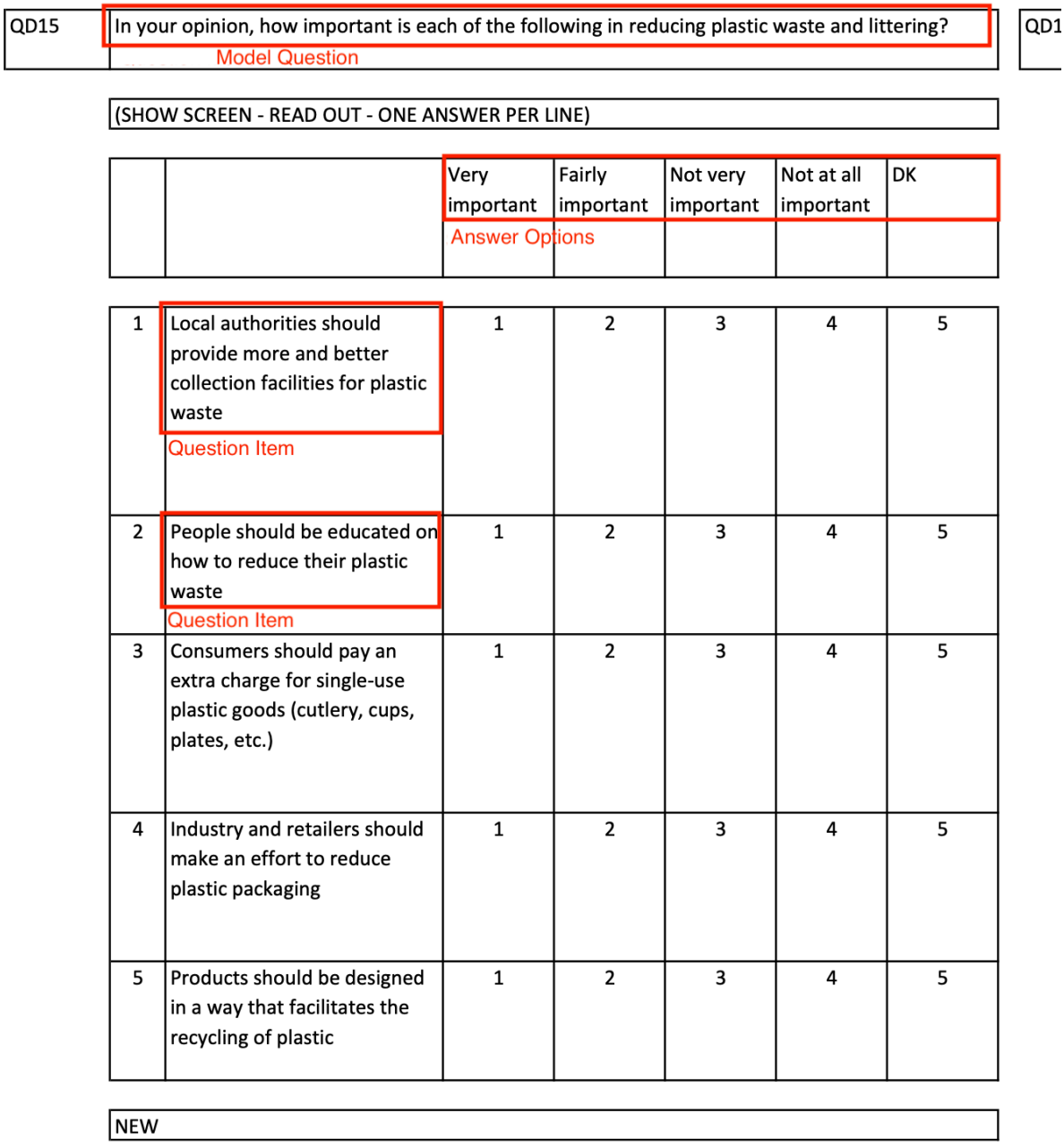
Language	Label	Description	Also known as
English	Trust in EU ECOLABEL [model]	Please tell me to what extent you agree or disagree with the following statement: “I trust that products carrying the EU ecolabel are environmentally-friendly.” [ ] scale	QuestionTrustInEUEcolabel
magyar	No label defined	Kérem, mondja meg, hogy milyen mértékben ért egyet vagy nem ért egyet az alábbi kijelentéssel: "Bízom benne, hogy azok a termékek, amelyeken uniós ökocímke van, környezetbarátok". [ ] scale	
Nederlands	No label defined	Kunt u mij vertellen in hoeverre u het eens of oneens bent met de volgende uitspraak: "Ik	

> **Note**
>
> Note: The system assigns a unique ID to every entry. In our example the ID is Q111.

A.2.2.2 Matrix Questions

Matrix questions have:

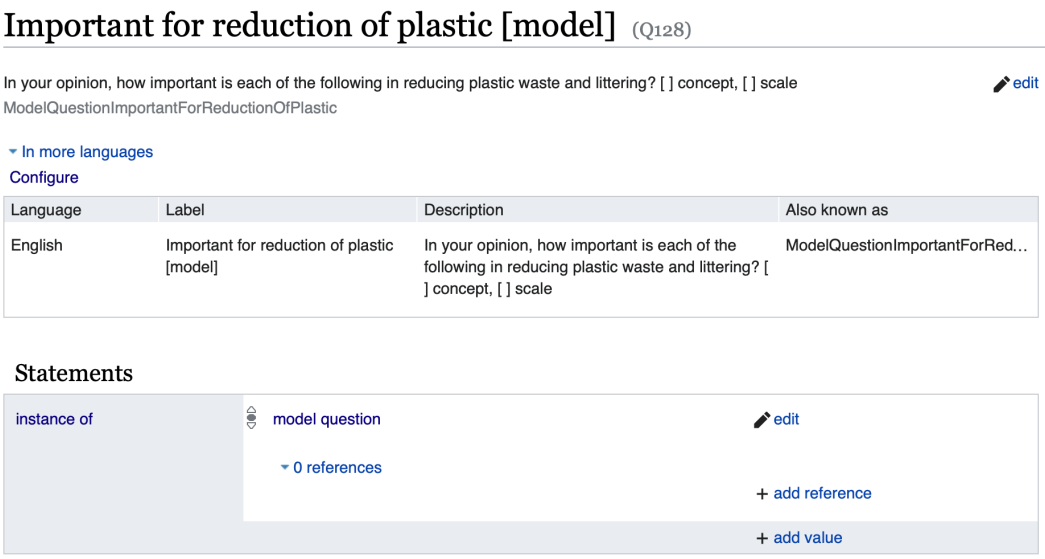
- a model question
- several question items
- answer options



QD15	In your opinion, how important is each of the following in reducing plastic waste and littering? Model Question						QD1
(SHOW SCREEN - READ OUT - ONE ANSWER PER LINE)							
		Very important	Fairly important	Not very important	Not at all important	DK	
		Answer Options					
1	Local authorities should provide more and better collection facilities for plastic waste Question Item	1	2	3	4	5	
2	People should be educated on how to reduce their plastic waste Question Item	1	2	3	4	5	
3	Consumers should pay an extra charge for single-use plastic goods (cutlery, cups, plates, etc.)	1	2	3	4	5	
4	Industry and retailers should make an effort to reduce plastic packaging	1	2	3	4	5	
5	Products should be designed in a way that facilitates the recycling of plastic	1	2	3	4	5	
NEW							

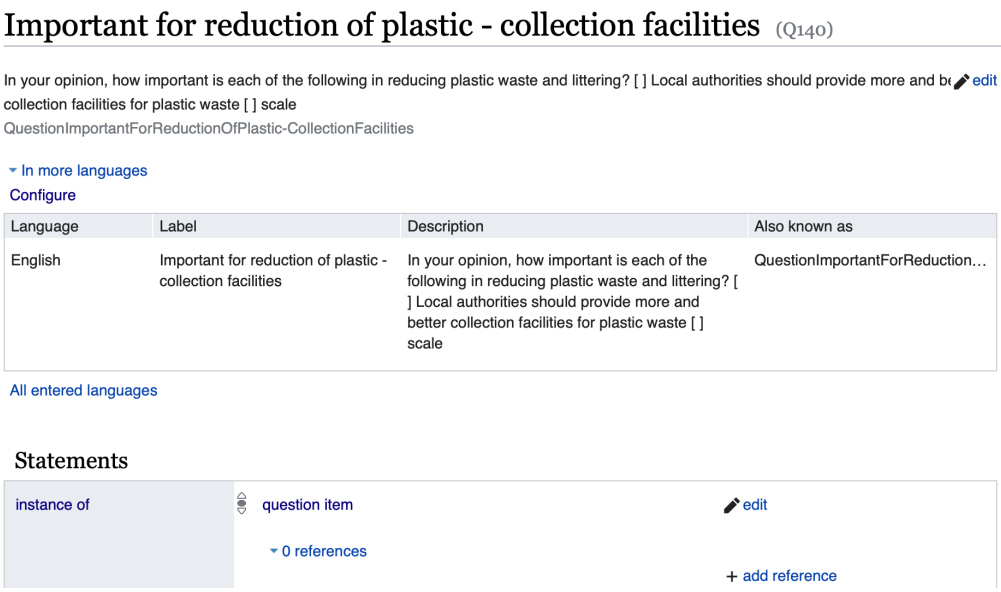
Figure A.2: This question is taken from the question block D (QD) from the Eurobarometer 88.1 study.

Following the functions "Special Pages" "Create New Item" you should feed into Wibikbase separately the model question and the questions items.



Q128 is the model question, which follows the structure:

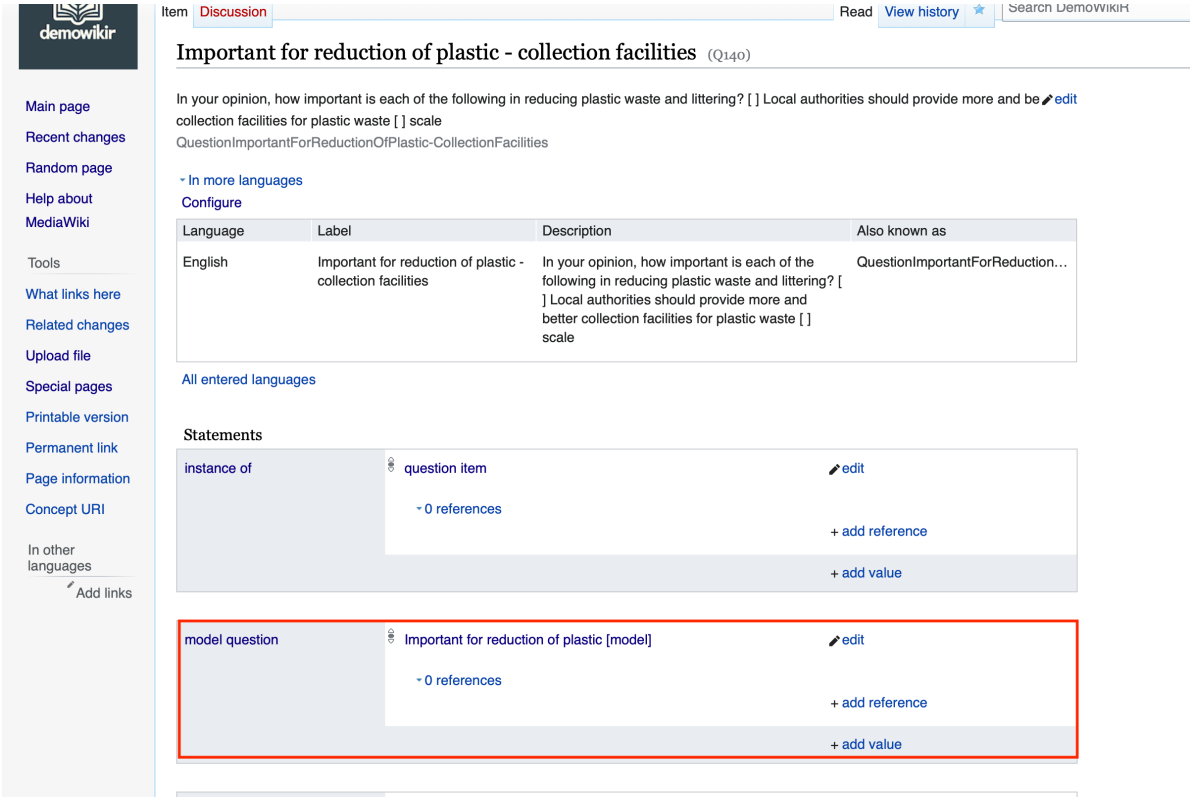
- Language Choose the language (en)
- Label - *questionName + [model]* - `Important for reduction of plastic [model]`
- questionText (description)
  - `In your opinion, how important is each of the following in reducing plastic waste and littering?`
  - `[ ] concept` - stands for the question items
  - `[ ] scale` - stands for the answer options, which follow a scale
- Aliases - leave it empty



Q140 is a question item, which follows the structure:

- Language Choose the language (en)
- label: `Important for reduction of plastic - collection facilities`
- questionText (description)
  - model question - `In your opinion, how important is each of the following in reducing plastic waste and littering?`
  - question item: `[ ] Local authorities should provide more and better collection facilities for plastic waste`
  - `[ ] scale` - stands for the answer options, which follow a scale
- Aliases - leave it empty

> When creating a "question item", using statements, always connect the "question item" to the "model question"



A.2.2.3 Multiple Choice Questions

Multiple Choice questions have:

- a model question
- several question items

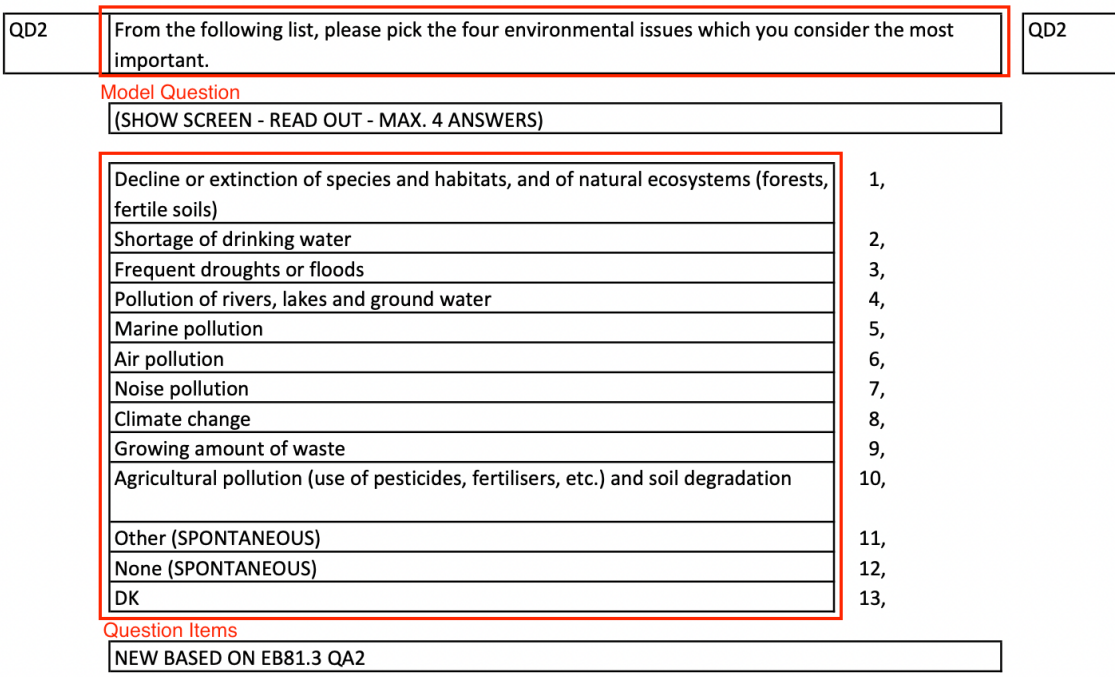
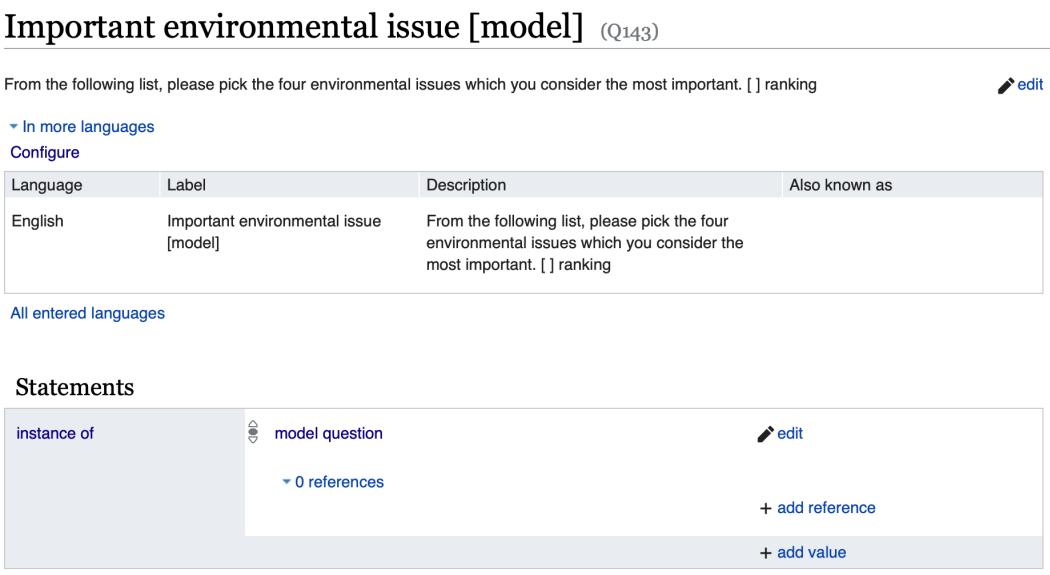


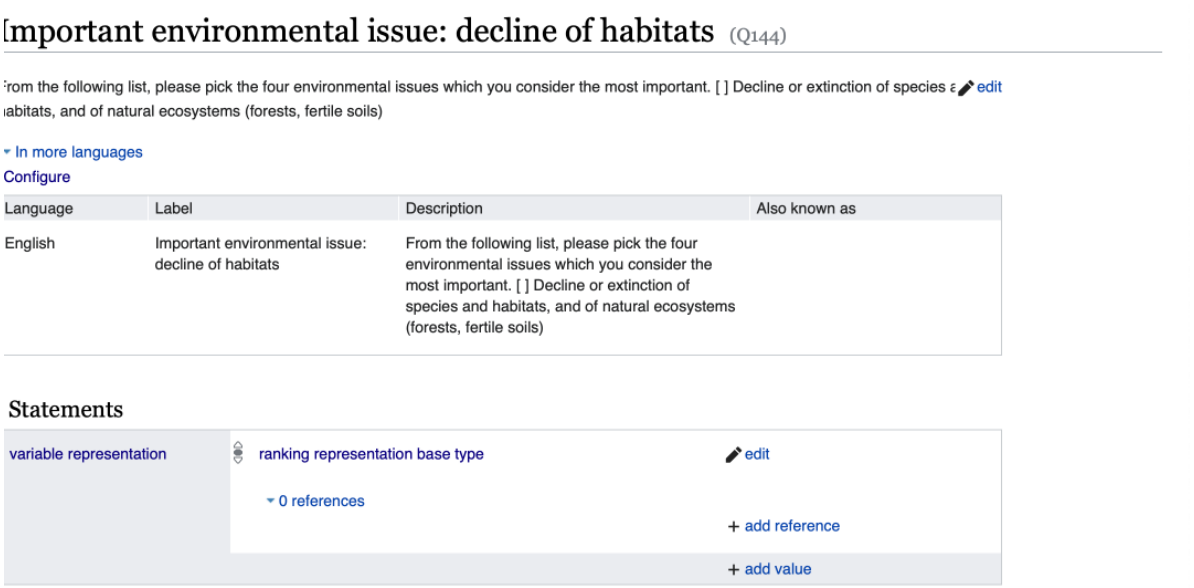
Figure A.3: This question is taken from the question block D (QD) from the Eurobarometer 88.1 study.

Following the functions "Special Pages" "Create New Item" you should feed into Wibikbase separately the model question and the questions items.



Q143 is the model question, which follows the structure:

- Language - Choose the language (en)
- Label - *questionName + [model]* - `Important environmental issue``[model]`
- questionText (description)
  - `From the following list, please pick the four environmental issues which you consider the most important.`
  - `[ ] ranking` - stands for the question items
- Aliases: leave it empty



Q144 is a question item, which follows the structure:

- Language Choose the language (en)
- label: `Important for reduction of plastic - collection facilities`
- questionText (description)
  - model question - `From the following list, please pick the four environmental issues which you consider the most important.`
  - question item - `[ ] Decline or extinction of species and habitats, and of natural ecosystems (forests, fertile soils)`
- Aliases - leave it empty

> When creating a "question item", using statements, always connect the "question item" to the "model question"

A.3 Add Metadata Statements to your Questions

Using Wikibase's "statements" feature you can link different type of information to your questions.

You need to add further metadata statements to the question bank item. Metadata is a statement about the data. We are adding standard, basic statements in subject, predicate, and object (triplet) format to each question bank item.

IN the following this guide explain how to add information about:

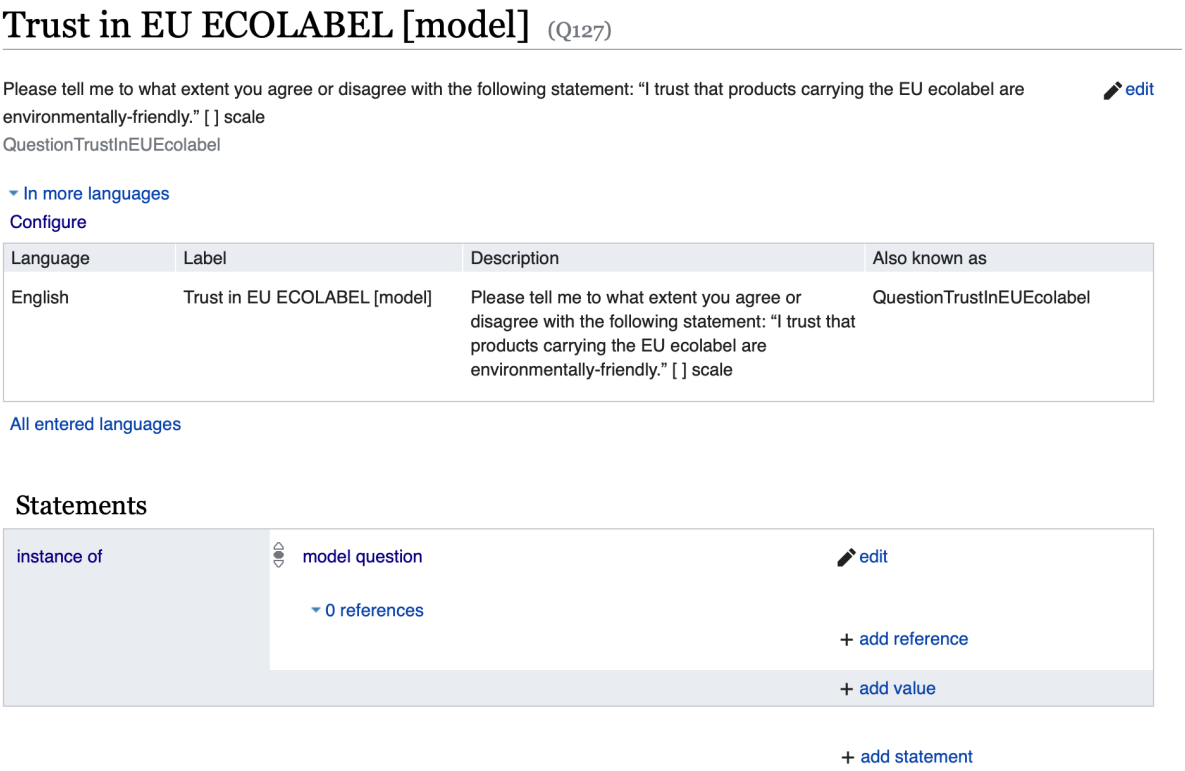
- questionnaire classes
- variable representation (P265): a DDI-Lifecycle category for the creation of variables from the answer options, for example
- study (DDI) P270: the *study* where you can find this (model) question (item). In DDI, a study represents the process by which a data set was generated or collected (in a survey). For example, Eurobarometer 88.1 (2017) Q139
- related survey concept (P267): a concept that a study (group), question (group) or question item aims to measure, for example environmental protection (Q131).

A.3.1 Questionnaire Classes

Let's start by specifying the entry we created as `model question` or `question item`.

Specify the entries created as `model question` or `question item`.

- Select `+add statement`.
- Using the instance of property, which is defining the taxonomical class of the entered item (in this case, a question.)
- In case of model questions, define them as `model question`.
- In case of question item, define them as `question items`.



In the case of questions items, always link them to the appropriate model questions using the model question (P266) property.

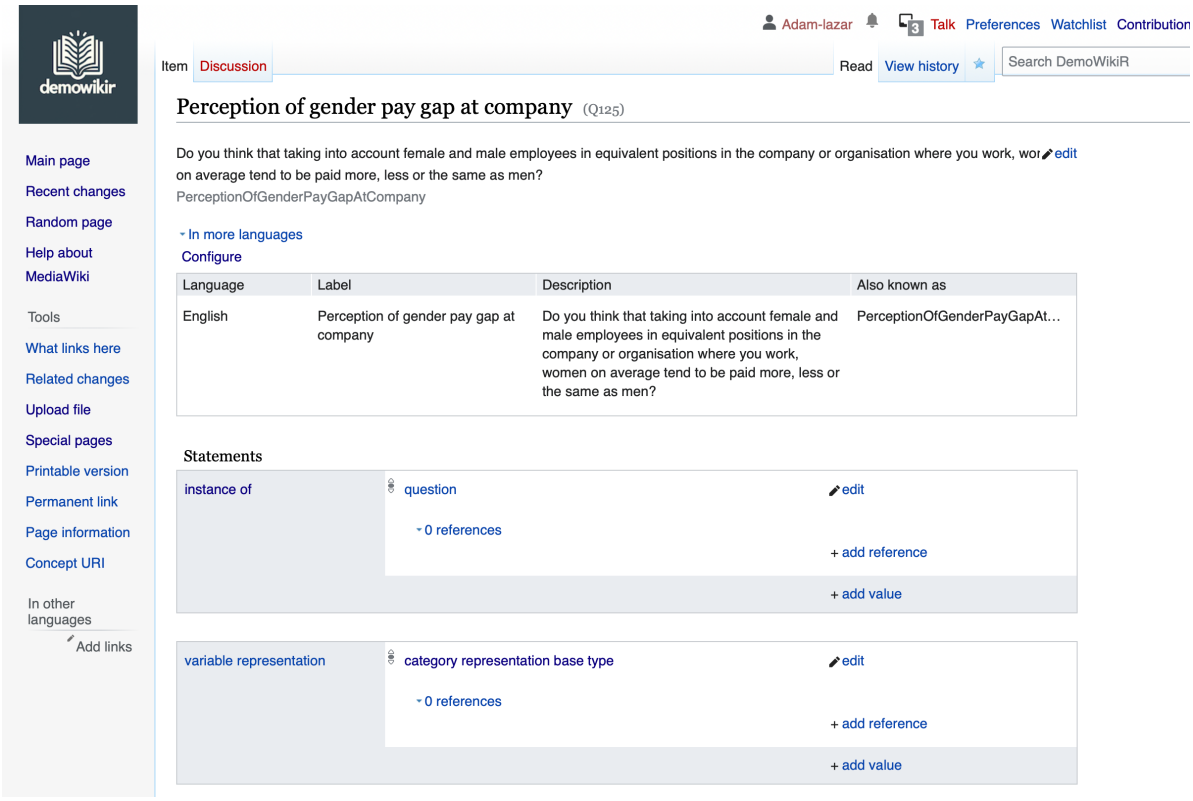
A.3.2 Variable Representation

When the questionnaire will be filled out in a raw dataset, each response of a question(item) will be translated into a variable. We need to define how we want to represent those answers in the resulting output dataset. (See DDI 3.3 (2020) documentation - Variable Value Representation and Question Response Domain)

Using statements you can define the representation of the variables. You can choose from the following categories:

- ☒ category representation base type: if the answers are categories (for example: `[ ] Female`, `[ ] Male`, `[ ] Prefer not to say`)

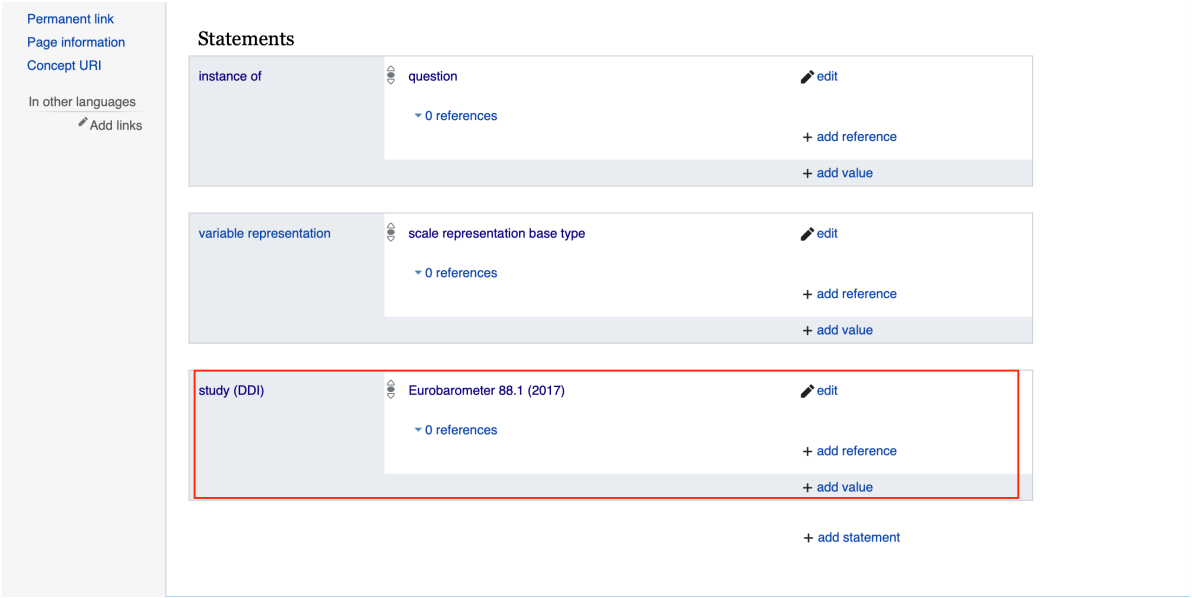
- ☒ category representation base type with a **scale**: if the answers are categories and follow a scale (for example: `[] Very important`, `[ ] Fairly important`, `[ ] Not very important`, `[ ] Not at all important`.)
- ☒ ranking representation base type: the respondant must rank the answer options, like 1st, 2nd, 3rd, etc.
- ☒ numeric variable representation base type: the answer should be a number, for example, the age of the respondant as an integer number or a postal code in a country where postal codes contain only numeric digits, f.e., `1051`.
- ☒ textual variable representation base type: the answer should be some text, for example, and open answer, or a geographical location typed as a simple text, for example, `Bratislava`.



A.3.3 Define the source study

> **Tip**
>
> For further details, please check the disco:Study class.

With the study (DDI) P270 property you must link as a statement the *study* where you found the (model) question (item).

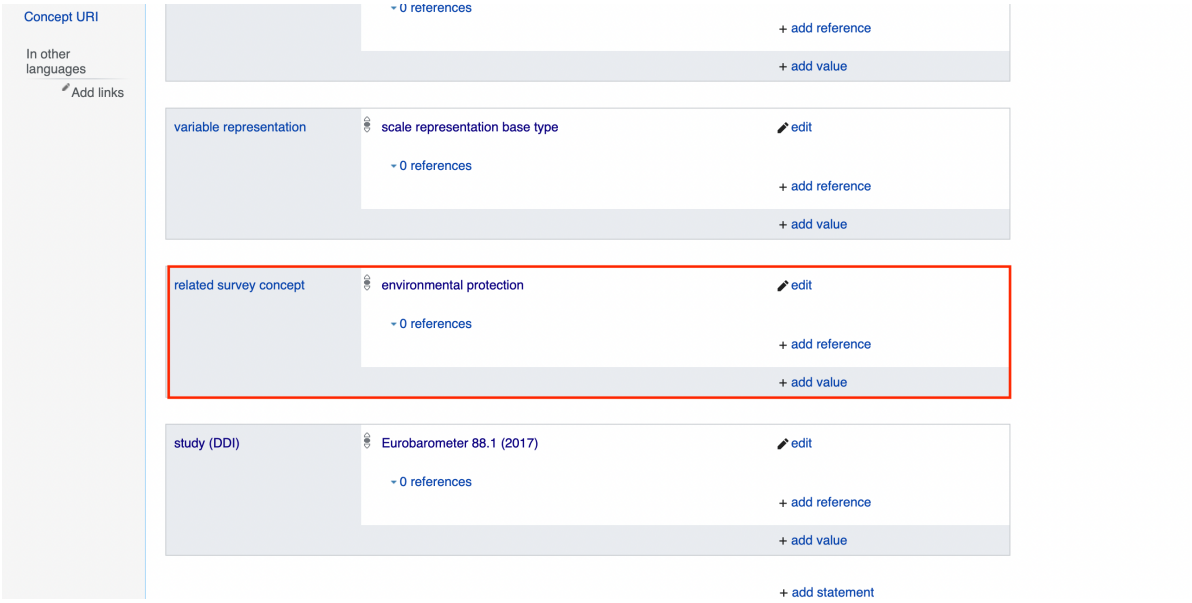


An example for a study: Eurobarometer 88.1 (2017) Q139

> **Note**
>
> Note: If the study is not yet in Wikibase, you can create an entry for it using the `Create a New Item` function.

A.3.4 Add related concept

With the related survey concept (P267) property you can link concepts, that a study (group), question (group) or question item aims to measure, for example environmental protection (Q131).



Where are the related concepts coming from?

1. The best case is that you use a widely accepted conceptualisation (ontology item) of your domain. For example, we took the study (Q149) concept from the DDI-Discovery (disco) ontology. You can connect statements of equivalence to a well-defined ontology via equivalent class (P69). In other words, our Q149 entity is equivalent to DDI's `Study`.

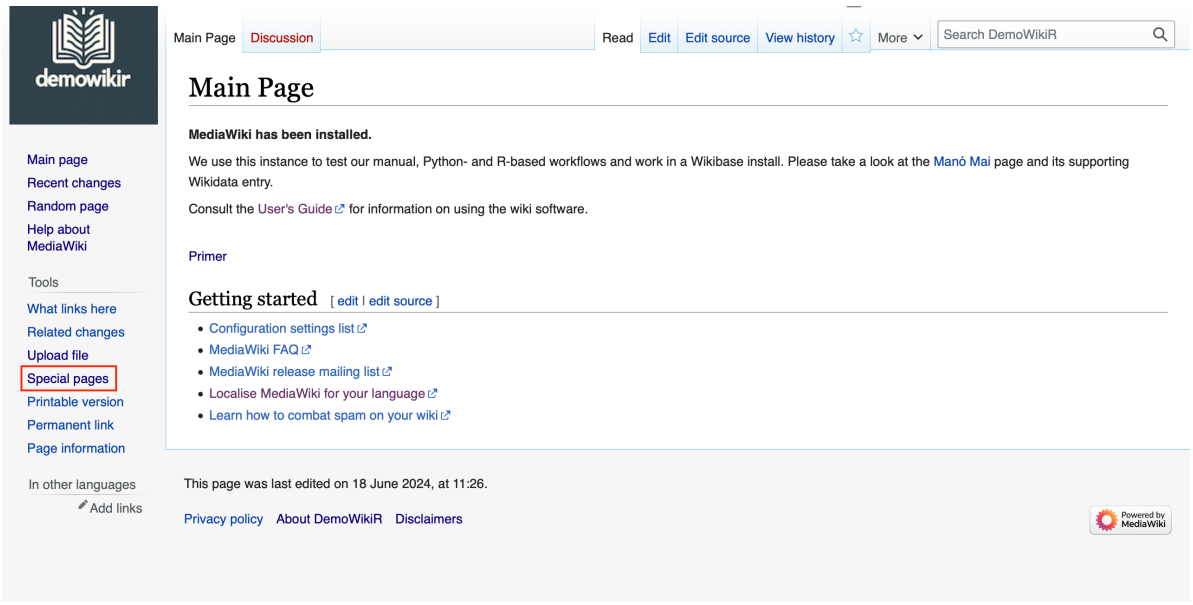
2. If there are no accepted ontologies or you are uncertain, it is a very good practice to use a concept definition from Wikidata. Even in the case of an ontological definition, adding the Wikidata QID is a great idea because Wikidata connects equivalent definitions across various domains' ontologies. You can make a statement about an equivalent Wikidata URI (for an item) by Wikibase URI (P73). See, for example plastic (Q148), Wikibase URI (P73), https://www.wikidata.org/wiki/Q11474, meaning that our `plastic` definition is equivalent to the Wikidata definition of `plastic`.

3. You can create your definition if you are still looking for a suitable definition in an accepted ontology or on Wikidata. For this, you should create a definition in Wikibase (as a new item.) See, for example, model question (Q126), which is our own proprietary definition until we find a more consensual one.

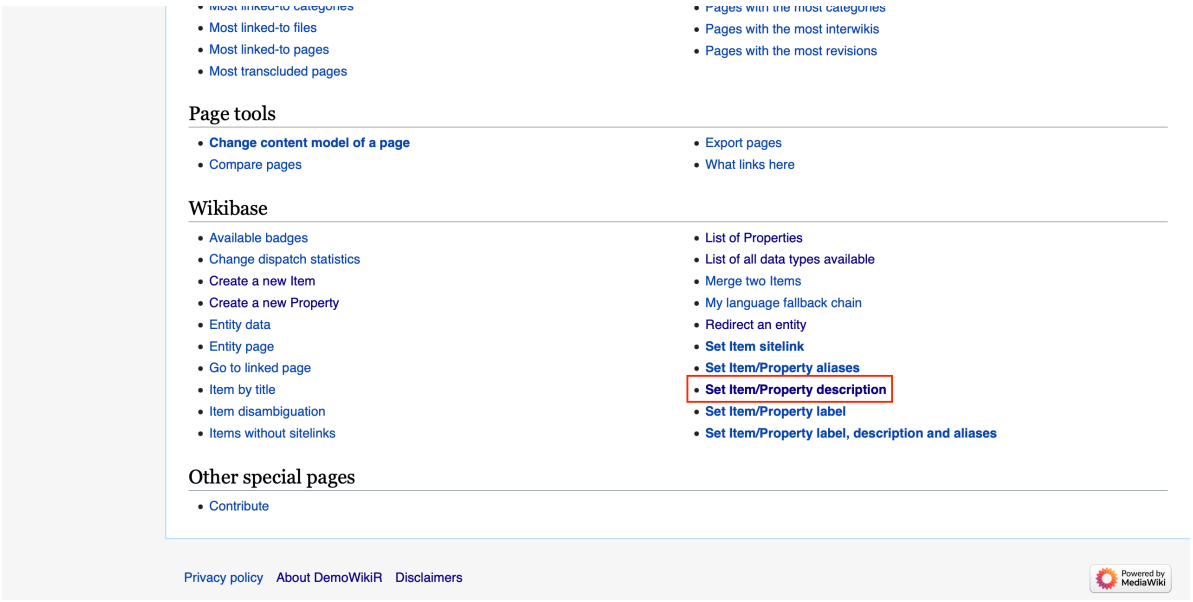
A.4 Add the questionText translations

On Wikibase you can add different language versions to the same question.

To do so, go to "Special Pages"



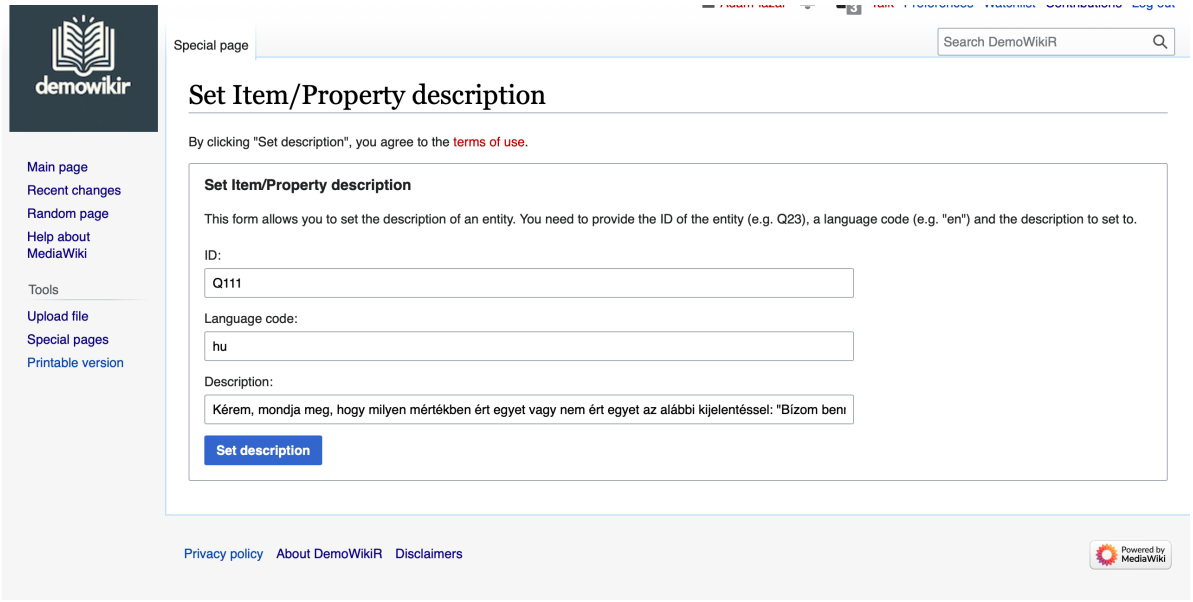
Scroll down and select: “Set Item/Property Description”



Fill the form:

- ID - The QiD of the question
- Language code - the new language you want to input the question

- Description - The question itself in the new language



Select “Set Description”.

The entry is now updated with another language.

B Variables in Music Databases

We use two information models:

- SDMX for statistical concepts
- DDI for the documentation of microdata

When repeatedly querying a music API, such as the Spotify API, we carry out an automated survey in which the respondent is not a human but a machine. Sending the same query to a well-designed API will yield a comparable answer to our question sent to the same API yesterday or tomorrow.

There are still many similarities with survey harmonisation: you would usually like to combine the data from the API with other data sources, in which case you still have to harmonise the concepts, the labelling, the translations, and the coding of your responses (processed in a dataset into variables.)

The previous Appendix introduces the key concepts and practices of survey harmonisation. When working with APIs, you do not need to harmonise question texts in human languages, because you harmonise them in a marchine-readable query language (for example, in SQL or SPARQL.) The rest of the data harmonisation workflow is the same.

B.1 String versus item

- Slovakia (Q79) is a well-defined node in our Wikibase graph.
- `Slovakia` as a string is not well-defined; it can only be understood if we add `"Slovakia"@en` a reference to the natural language of the string.

Whenever possible, we want to refer to well-defined nodes in the knowledge graph. For example, our entry Slovakia (Q79) states that it is equivalent with Slovakia (Q214) on Wikidata, and Wikidata connects plenty of metadata to this concept: the geographical boundaries, the fact that it is an independent state since 1993, it predecessors, capital, etc.

Our aim is to have a rich and standardised description to each variable, and as much as possible, to very constant (or attribute.) *Katarína Kubošiová* is a Slovak singer-songwriter, also known as *Katarzia*. To avoid any ambigouity with other people potentially called *Katarína Kubošiová* or *Katarzia*, we would like to refer to her with a globally unique identifier. Her ISNI identifier (ISNI:) is isni: 0000000467220673, which identifies her with global clarity.

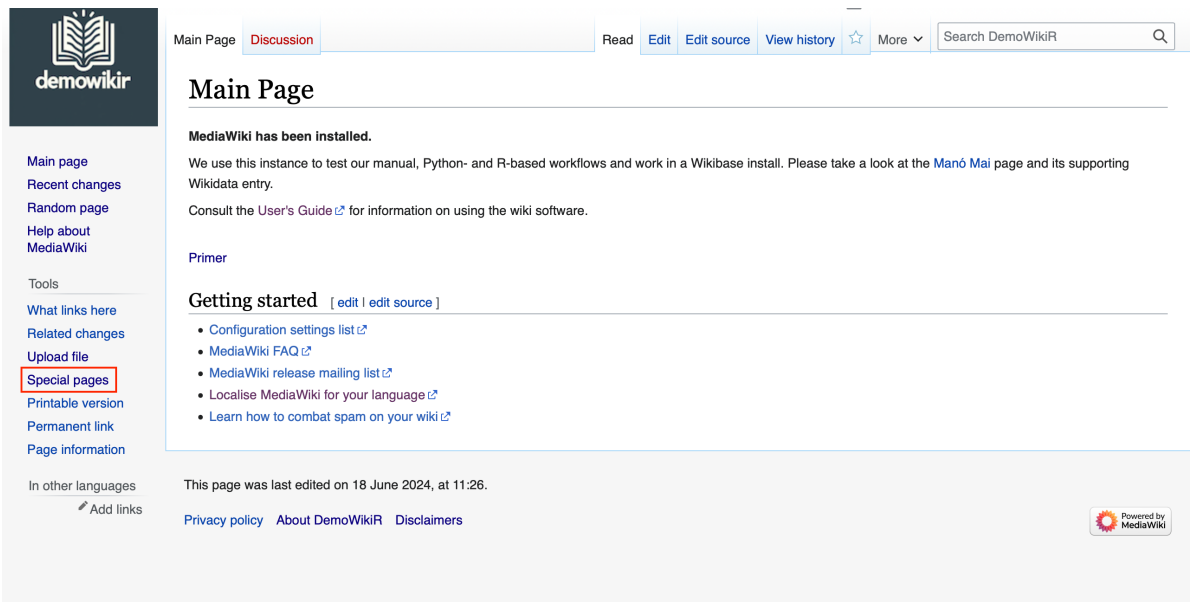
The metadata enrichment is possible to make data points into nodes. For example, if we conceptualise `Slovakia` into a node, than we can connect to this node sound recordings (regardless if they have a Slovak or English-language title) if they were registered with the Slovak national ISRC registrant's `SK` prefix. We can connect `Katarína Kubošiová`, `Katarzia`, `SK`, `Slovakia` in a graph to the concept of *Slovakia* with less or more clarity; in this case, for example, defining that a sound recording was registered in Slovakia, or the artist known as *Katarzia* was born in Slovakia or sung in the Slovak language.

B.1.1 1. Access Wikibase

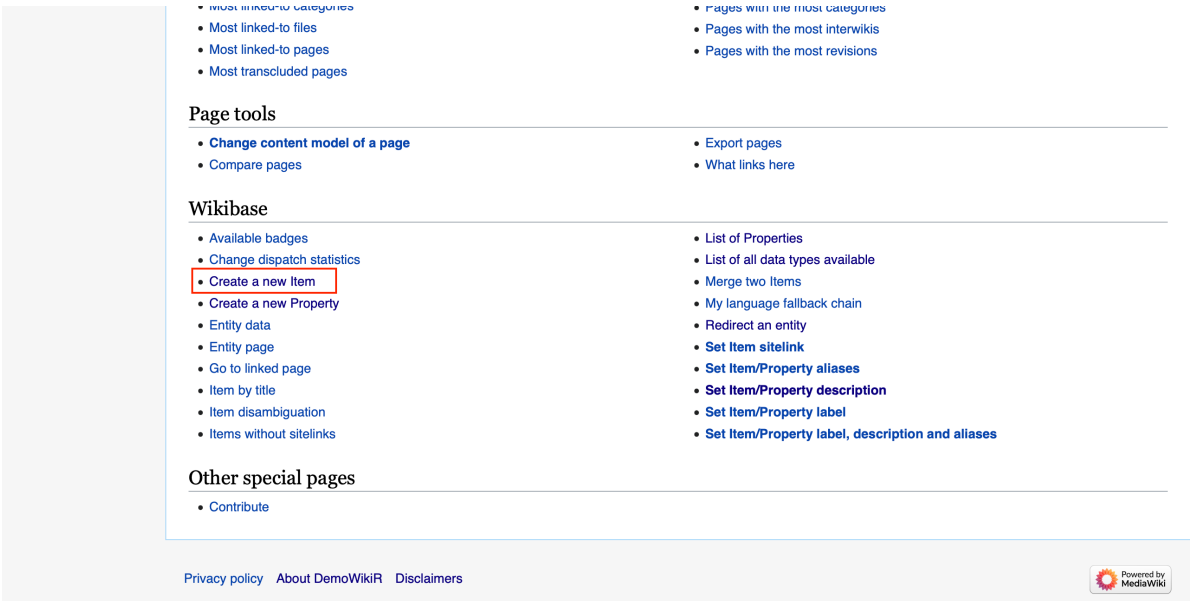
Login in with you account to Wikibase.

B.1.2 2. Create a New Item

Go to `Special Pages`

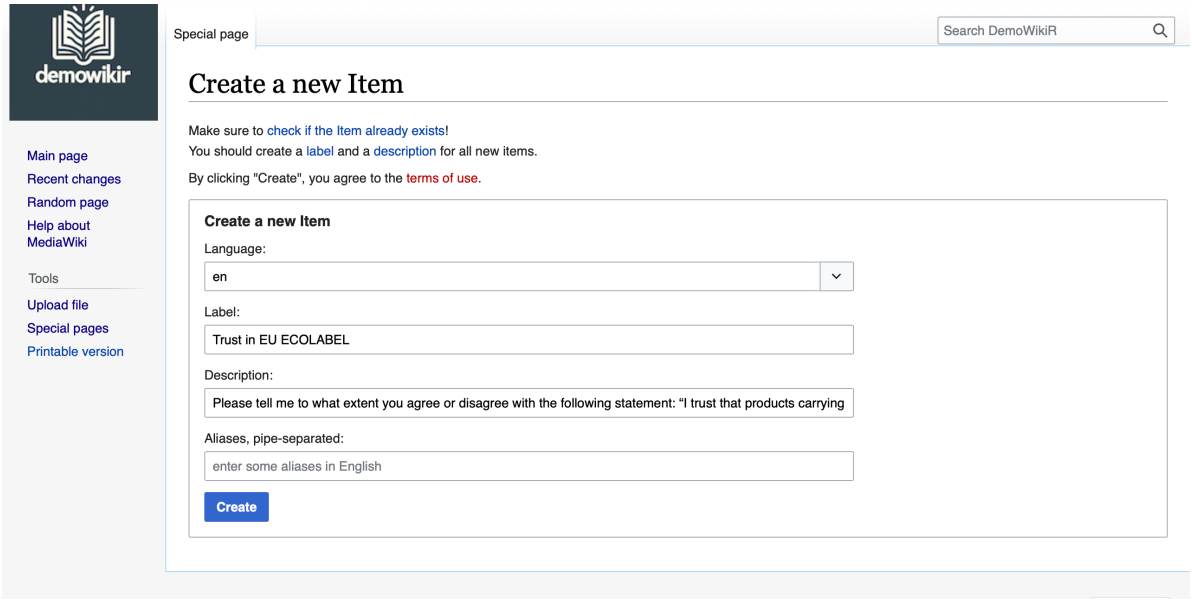


Scroll down and select: `Create a New Item`



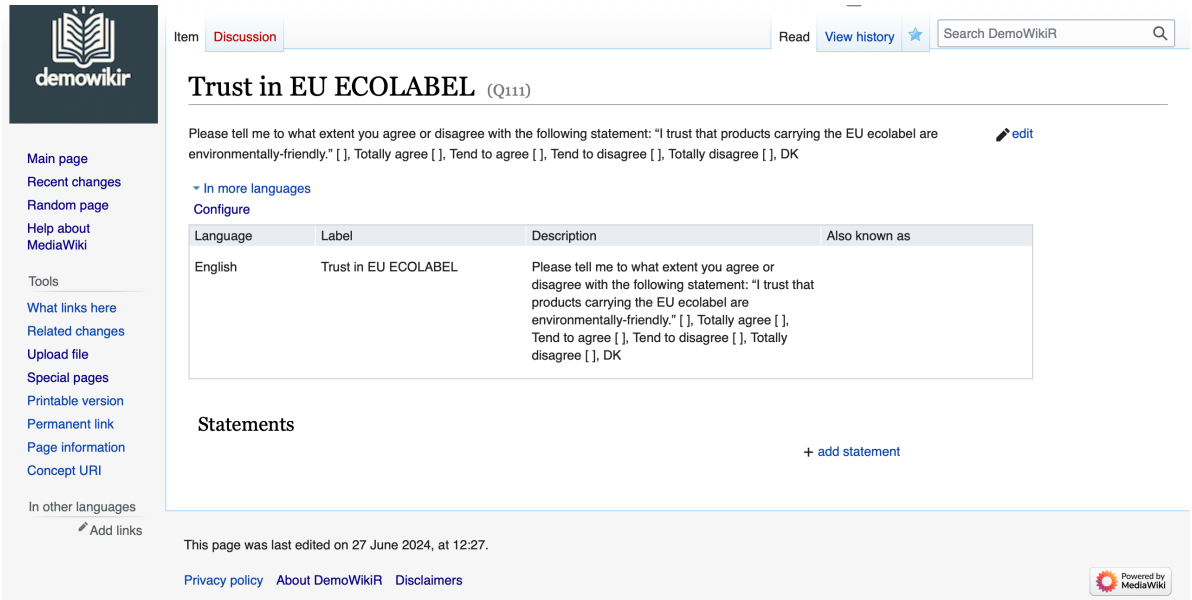
Fill the form with the item's data:

- Language - Choose the language (en)
- Label - Give a short name for the node, for example, *Katarína Kubošiová*
- Description - Enter the item description, for example *Singer-songwriter born in the Slovak Republic*
- Aliases - you can add *Katarzia* or any other known names here.



Click `Create`.

The item now is created on Wikibase. For each concept that you want to use in your research, its documentation should be present. For key persons, names, musical works, it is also advisable to have an item defined.



> **Note**
>
> Note: The system assigns a unique ID to every entry. For example, in our system, the ID of *Ján Levoslav Bella* (Slovak conductor, composer and educator), aslo known under the alias with no Slovak special characters as *Jan Levoslav Bella* is Q93. With Q93 you cannot make the mistake of confusing the fact that *Ján Levoslav Bella* is the same person as *Jan Levoslav Bella*.

B.1.3 3. Add Metadata Statements

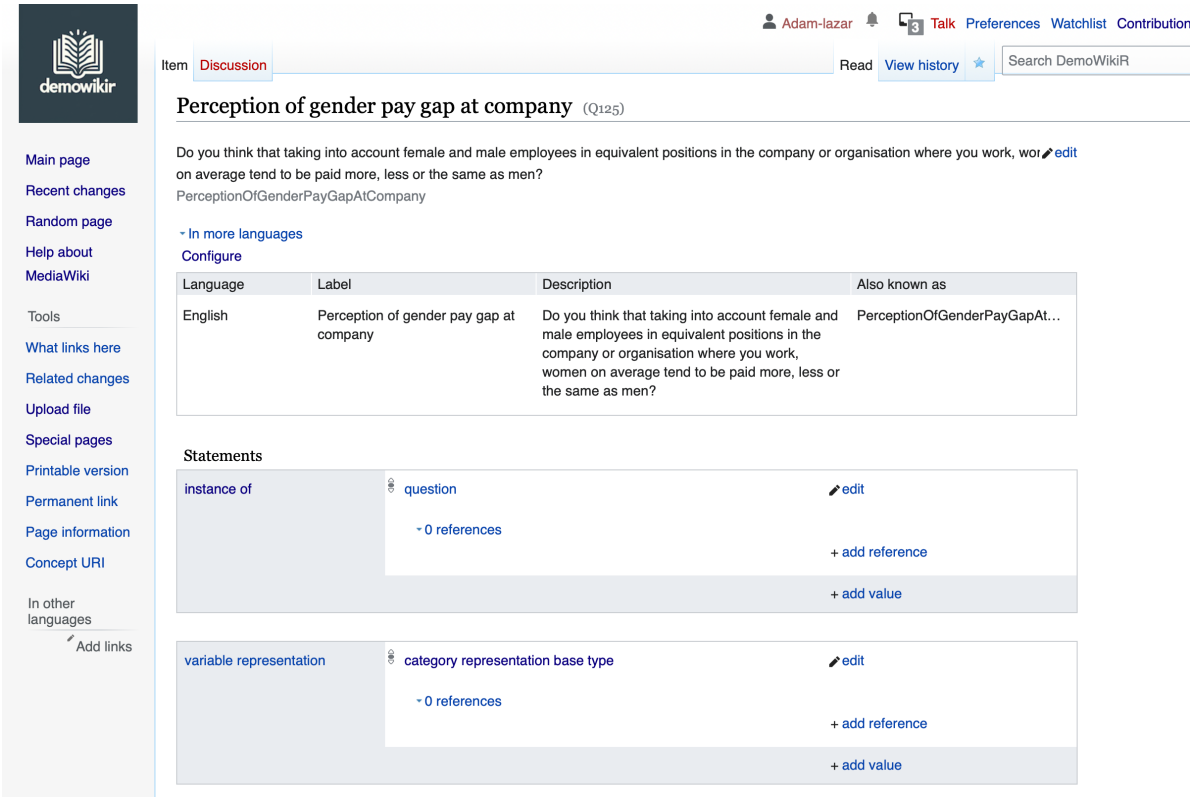
You need to add further metadata statements to the question bank item. Metadata is a statement about the data. We are adding standard, basic statements in subject, predicate, and object (triplet) format to each question bank item.

B.1.3.1 Variable Representation

DDI has standard variable representation definitions. When a questionnaire will be filled out in a raw dataset, or data will be systematically queried from and API, each response will be translated into a variable. We need to define how we want to represent those answers in the resulting output dataset. (See DDI 3.3 (2020) documentation - Variable Value Representation and Question Response Domain)

Using statements you can define the representation of the variables. You can choose from the following categories:

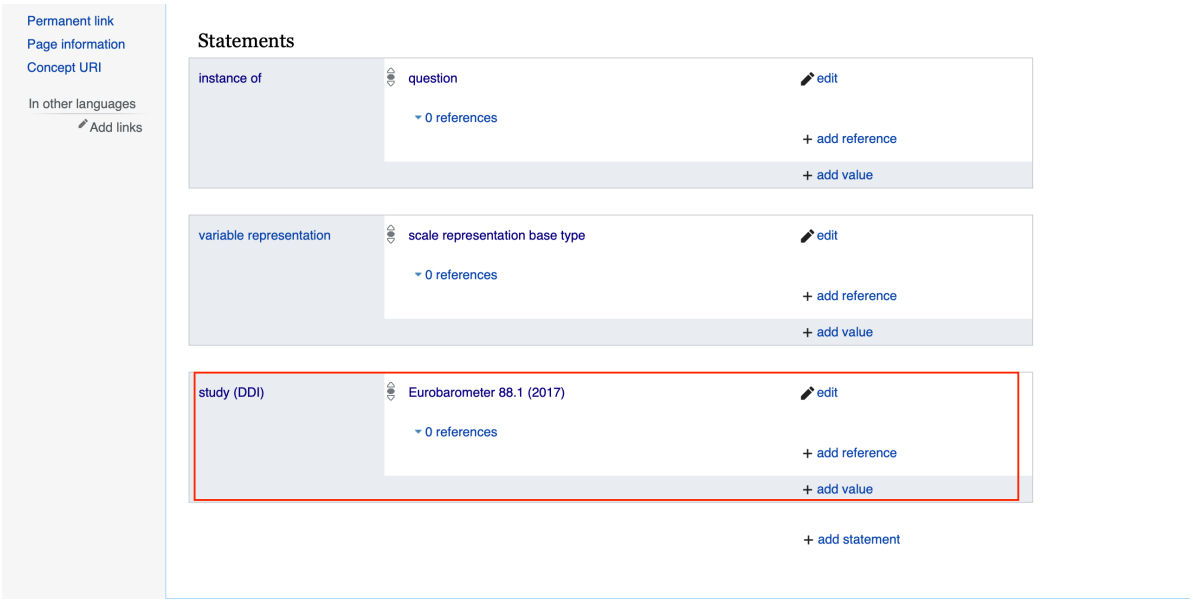
- ☒ category representation base type: if the answers are categories (for example: `[ ] Female`, `[ ] Male`, `[ ] Prefer not to say`)
- ☒ category representation base type with a **scale**: if the answers are categories and follow a scale (for example: `[] Very important`, `[ ] Fairly important`, `[ ] Not very important`, `[ ] Not at all important`.)
- ☒ ranking representation base type: the respondant must rank the answer options, like 1st, 2nd, 3rd, etc.
- ☒ numeric variable representation base type: the answer should be a number, for example, the age of the respondant as an integer number or a postal code in a country where postal codes contain only numeric digits, f.e., `1051`.
- ☒ textual variable representation base type: the answer should be some text, for example, and open answer, or a geographical location typed as a simple text, for example, `Bratislava`.



B.1.3.2 Define the source study

> **Tip**
>
> For further details, please check the disco:Study class.

With the study (DDI) P270 property you must link as a statement the *study* where you found the concept definition. If it was a formal ontology, or Wikibase, use different properties (see below).



An example for a study: Eurobarometer 88.1 (2017) Q139

> **Note**
>
> Note: If the study is not yet in Wikibase, you can create an entry for it using the `Create a New Item` function.

B.1.3.3 Add related concept

Where are the related concepts coming from?

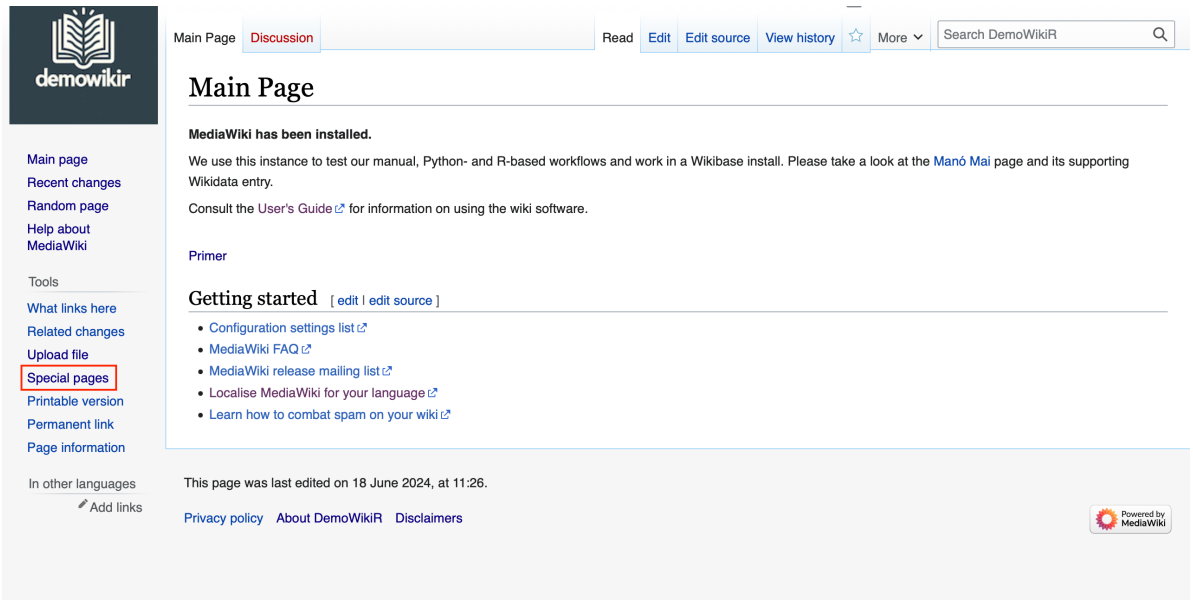
1. The best case is that you use a widely accepted conceptualisation (ontology item) of your domain. For example, we took the duration (Q132) concept from the Music Ontology. You can connect statements of equivalence to a well-defined ontology via equivalent class (P69). In other words, our Q149 entity is equivalent to Music Ontology's `duration` (in short: `mo:duration`.)
2. If there are no accepted ontologies or you are uncertain, it is a very good practice to use a concept definition from Wikidata. Even in the case of an ontological definition, adding the Wikidata QID is a great idea because Wikidata connects equivalent definitions across various domains' ontologies. You can make a statement about an equivalent Wikidata URI (for an item) by Wikibase URI (P73). See, for example duration (Q132), Wikibase URI (P73), https://www.wikidata.org/wiki/Q16038819, meaning that our `plastic` definition is equivalent to the Wikidata definition of `plastic` (which has a QID of `Q16038819` on Wikidata, and it received the QID of `Q132` on our Wikibase instance.)

3. You can create your definition if you are still looking for a suitable definition in an accepted ontology or on Wikidata. For this, you should create a definition in Wikibase (as a new item.) See, for example, model question (Q126), which is our own proprietary definition until we find a more consensual one.

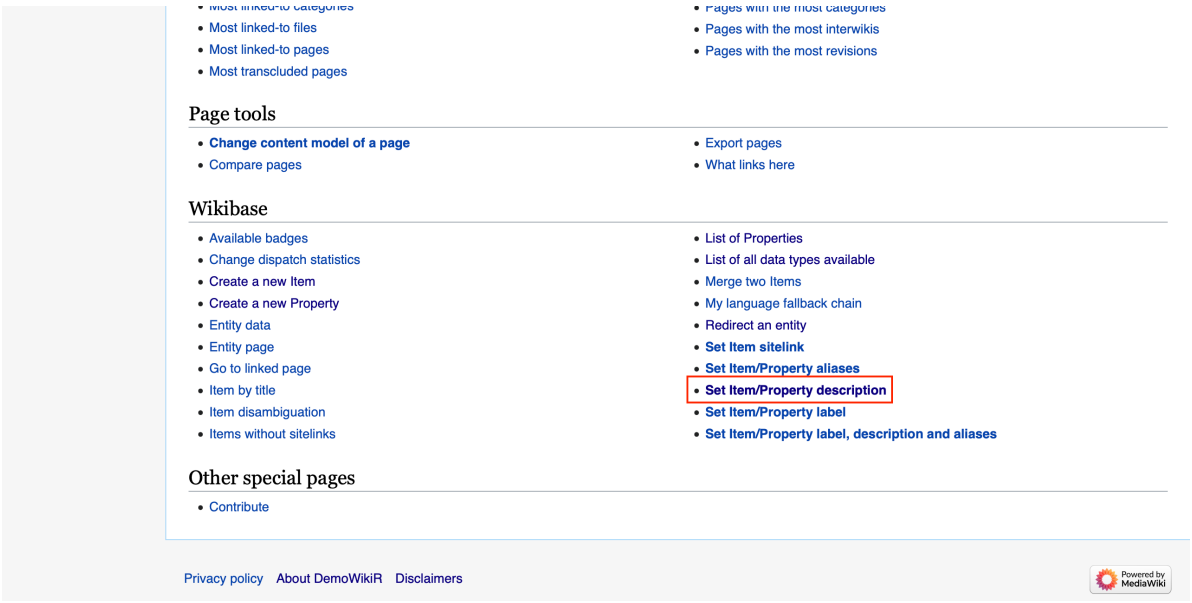
B.1.4 Add national langugage translations to your concept

On Wikibase you can add different language versions to the same question.

To do so, go to `Special Pages`

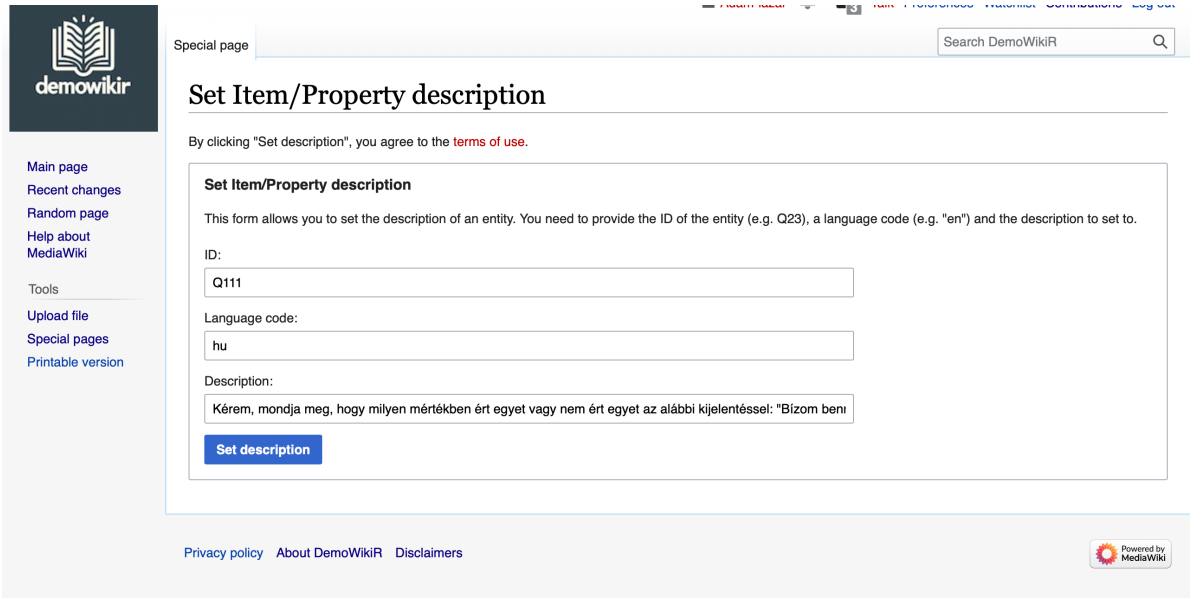


Scroll down and select: `Set Item/Property Description`



Fill the form:

- ID - The QiD of the question (for example, if you want to add a Dutch description to *Ján Levoslav Bella*, i.e., Slovak conductor, composer and educator, you must reference Q93.
- Language code - the new language you want to input the question, in this case, `nl`.
- Description - Write a short definition (up to 250 characters) in the new language.



Select “Set Description”.

The entry is now updated with another language label or description.