Data.europa.eu
Data Quality Guidelines

August 2021

Publications Office
of the European Union

Data.europa.eu
Data Quality Guidelines

August 2021

For more information:

OP.C.4
Publications Office of the European Union

2, rue Mercier
L-2985 Luxembourg
LUXEMBOURG

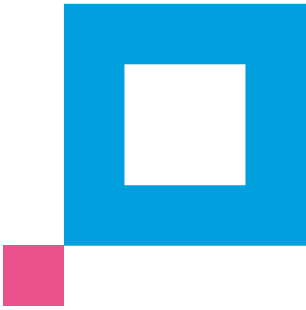
OP-DATA-EUROPA-EU@publications.europa.eu

Luxembourg: Publications Office of the European Union, 2021

This publication was developed as part of the 'Data quality guidelines for the publication of data sets in the EU Open Data Portal' project carried out by Fraunhofer FOKUS and financed by the ISA^2 programme.

PDF ISBN 978-92-78-42763-4 doi:10.2830/333095 OA-06-22-121-EN-N

Contents



Introduction

Data quality is fast becoming a hot topic, as demand for high-quality data continues to grow with a focus on data that is publicly available and can be easily reused for different purposes. Poor quality is a major barrier to data reuse. Some data cannot be interpreted due to ill-defined, inaccurate elements such as missing values, mismatches, missing data types, lack of documentation about the structure or format availability (HTML, GIF or PDF). Users find poor-quality data harder to understand and may use it less often. The data provider may even appear less reliable as a result.

For data to be easily reusable, data publishers must make sure it is easy to discover, analyse and visualise. Reusers must understand what the data is about and how it is defined or structured, and should preferably get the data in the format they need.

Data quality covers different aspects, for example consistency, conformity, completeness or documentation. The FAIR guiding principles for scientific data management and stewardship ([1]) provide a framework for grouping the different aspects of data quality. The framework consists of four dimensions – findability, accessibility, interoperability and reusability – and provides concrete metrics for each dimension. Data publishers should become acquainted with the FAIR principles before publishing data. It is also helpful to develop a data management plan (DMP) that outlines how data should be handled. A DMP addresses questions such as where to publish data, where to store metadata, which format to use and which standard to follow. This sort of plan will make publication easier.

Data needs to be carefully prepared before publication. Preparation is an interactive and agile process used to explore, combine, clean and transform raw data into curated, high-quality data sets. This process consists of six different phases (see Figure 1).

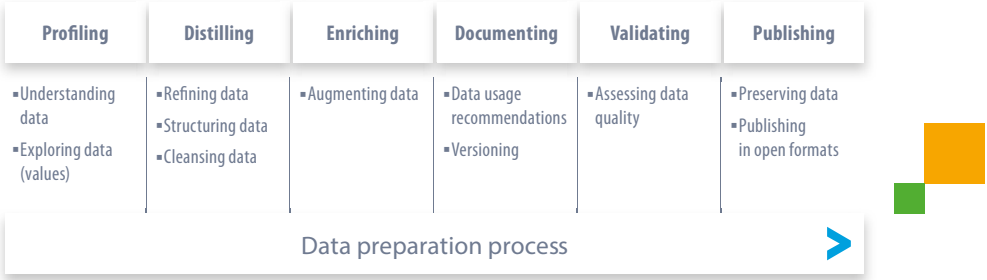


Figure 1. Data preparation process

([1]) https://www.go-fair.org/fair-principles/

By ensuring data of the highest quality along with data consistency, conformity and completeness, data providers help reusers to easily discover, reuse, analyse, visualise or process data for analytics and business intelligence and to contribute to increasing the transparency of EU data.

For these reasons, in 2019 the Publications Office of the European Union (the Publications Office) launched the 'Data quality guidelines for the publication of data sets in the EU Open Data Portal ([2])' project, aimed at analysing major quality issues and providing a set of recommendations for data providers from the EU and its Member States concerning the quality of data resources available through the EU Open Data Portal (EU ODP). The project ([3]) was carried out by Fraunhofer FOKUS (acknowledgements to Lina Bruns, Benjamin Dittwald and Fritz Meiners for their contributions) and consisted of the following three parts.

- Data profiling. Analysis of the data published by the EU institutions and bodies to identify the most common data quality issues.

This part consisted of two major steps. First, all metadata was assessed in an automated way against a set of criteria using the FAIR principles. This step was used to identify data sets of poor quality, which were analysed in depth in the second step. The second step was carried out manually and involved the analysis of 50 distributions from selected datasets. In contrast to step one, the second step focused on analysing the actual data. The data was checked for encoding issues, accessibility, compliance with standards and proper presentation of numbers and dates.

For more information about this part of the project please contact:
OP-DATA-EUROPA-EU@publications.europa.eu

- Data quality indicators and metrics. Identification of data quality dimensions, indicators and metrics to indicate how data quality can be measured.

This part consisted of two main tasks. Firstly, identifying data-quality indicators and metrics appropriate for assessing data quality, and secondly, developing mock-ups for a future data quality dashboard. The first task led to the identification of 12 relevant indicators for data quality across the four FAIR dimensions (see Figure 2).

([2]) On 21 April 2021 the EU Open Data Portal and the European Data Portal were consolidated into one single service and became data.europa.eu.

([3]) The project was financed by the ISA2 programme.

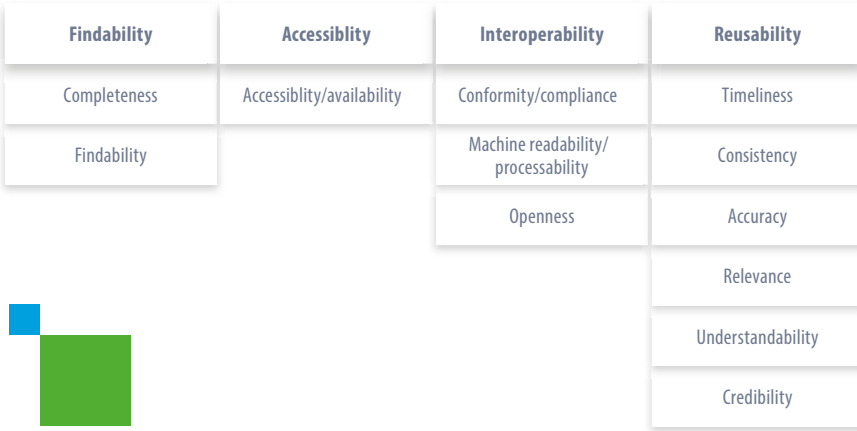


Figure 2. Overview of quality indicators grouped by FAIR dimensions

Metrics were also assigned for each indicator that show how to actually measure and quantify the quality indicators. In total, 42 metrics were described and illustrated with real data mostly taken from the EU ODP ([4]) (see Table 6).

For more information about this part of the project please contact:
OP-DATA-EUROPA-EU@publications.europa.eu

- Recommendations for delivering high-quality data. A set of recommendations for data providers from the EU and its Member States.

The current document is based on the outcome of Parts 1 and 2 and on a literature review. The recommendations are addressed to data providers to support them in preparing their data, developing their data strategy and ensuring data quality. It is composed of the following four parts.

1. **Recommendations for providing high-quality data.** The recommendations cover general aspects of quality issues regarding the findability, accessibility, interoperability and reusability of data (including specific recommendations for common file formats like CSV, JSON, RDF and XML).
2. **Recommendations for data standardisation (with EU controlled vocabularies) and data enrichment.**
3. **Recommendations for documenting data.**
4. **Recommendations for improving the 'openness level'.**

At the end of the publication the reader will find a glossary, a table with the overview of quality indicators and metrics, a checklist with the most important steps for improving the quality of data and metadata and a list of literature.

([4]) Please note that the interface has changed after the consolidation of EU Open Data Portal and the European Data Portal into data.europa.eu.

1. Recommendations for providing high-quality data

Introduction

The aim of this section is to provide quick and practical recommendations for data providers, allowing them to prepare and publish high-quality data sets.

It presents a set of best practices for data preparation, especially covering aspects of the data preparation process phase 'validating' (see Figure 3).

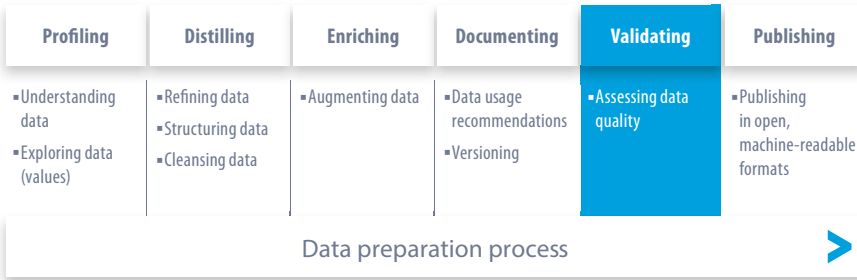


Figure 3. Data preparation process – Validating

An overview of universally applicable recommendations is given in Section 1.1, followed by format-specific recommendations in Section 1.2 addressing commonly used and open-data-appropriate (machine-readable and non-proprietary) file formats.

1.1. General recommendations

This section provides general recommendations to consider when publishing data. These recommendations apply to all kinds of data, regardless of the file format they are published in. The recommendations are grouped by the FAIR dimensions ([5]) of findability (Section 1.1.1), accessibility (Section 1.1.2), interoperability (Section 1.1.3) and reusability (Section 1.1.4). Each recommendation includes a description, screenshots and a reference to the respective metric, as well as helpful information about tooling and/or linkage to further relevant sources of information. File-format-specific recommendations for the machine-readable formats CSV, XML, RDF, JSON and APIs are covered in Section 1.2.

Before going on to the recommendations, there are two things you should consider in general if you are interested in publishing high-quality data: (i) make use of tooling, (ii) create a DMP.

([5]) https://www.go-fair.org/fair-principles/

(i) Make use of tooling

Data preparation is an ongoing, iterative and repetitive process. Most of the steps which should be performed within the data preparation process (see Figure 3) can be automated and supported with tools. If you are publishing data periodically, it might be worth investing in an 'extract, transform, load' (ETL) tool and related tools that support you in preparing and publishing high-quality data sets.

There are plenty of commercial tools that can help you prepare your data following the data preparation process (see Figure 3). A large number of solutions are available and the data preparation functions they offer are heterogeneous, so finding the right one might seem daunting. Data preparation functions are, for example, transforming, cleansing, blending, modelling and enriching data. Gartner Research has analysed 16 tools available from common vendors and classified them in a magic quadrant, identifying 'leaders', 'challengers', 'niche players' and 'visionaries' (see Figure 4), with strengths and cautions for each vendor ([6]). This assessment may help you to find the most appropriate tool for the task at hand. Gartner Research has also published the 'Market guide for data preparation tools' ([7]), in which the market is analysed and several products are introduced. Another report that lists and compares data preparation solutions is 'The Forrester Wave™: Data preparation solutions' ([8]).

Figure 4. Magic quadrant for data quality tools
Source: Gartner Research (2019a).

([6]) Gartner Research (2019a), 'Magic quadrant for data quality tools' (https://www.gartner.com/en/documents/3905769/magic-quadrant-for-data-quality-tools).

([7]) Gartner Research (2019b), 'Market guide for data preparation tools' (https://www.gartner.com/en/documents/3906957/market-guide-for-data-preparation-tools).

([8]) Little, C. (2018), 'The Forrester Wave™: Data preparation solutions', Forrester (https://www.forrester.com/report/The+Forrester+Wave+Data+Preparation+Solutions+Q4+2018/-/E-RES141619).

There are also some useful open-source tools that mostly focus on one concrete aspect of data preparation or that specialise in data quality issues within a certain file format, such as CSVLint ([9]) for CSV files or JSONLint ([10]) for JSON files. Another open-source tool is OpenRefine ([11]), which helps clean messy data and transform and extend data. Talend's Open Studio Line ([12]) is another open-source suite licensed under Apache. It is made up of components covering (big) data preparation and integration and data quality and uses machine-learning technology to perform data preparation tasks.

(ii) Create a data management plan

A DMP outlines how data is to be handled. It should establish where to publish data, where to store metadata, which format to use and which standard to follow. Answering these questions beforehand will make the publication process easier as it will be homogeneous and formalised. There is also a common standard for machine-actionable DMPs ([13]), and the FAIRification process provides some useful information you may wish to consider in your DMP ([14]).

1.1.1. Findability

1.1.1.1. Describe your data with metadata to improve data discovery

Dimension	Findability
Indicator	Completeness
Metrics	• Number of empty fields in metadata • Keywords assigned • Categories assigned • Temporal information given • Spatial information given

Metadata is descriptive data. Take for example an audio track: information regarding the artist and album is considered metadata, since this information is not part of the actual file. It is, however, very important when trying to find the file among others. Similarly, if a text document was missing its title, it would be very hard for users to discover the document. Complete and updated metadata is therefore vital for finding and using data. In addition, metadata can help users identify whether the information retrieved matches their request. A library of books would be of little use if the books were missing their key metadata information: author, title and ISBN. The same applies to data published online.

([9]) https://csvlint.io/
([10]) https://jsonlint.com/
([11]) https://openrefine.org/
([12]) https://www.talend.com/products/talend-open-studio/
([13]) https://github.com/RDA-DMP-Common/RDA-DMP-Common-Standard
([14]) https://www.go-fair.org/fair-principles/fairification-process/

Often, when publishing your data in a catalogue, some metadata fields are set as mandatory, which means that they have to be filled in before the data can be published. However, it is recommended that metadata fields that are not set as mandatory also be filled in. For the data publisher it does not take much effort to fill in these fields, and for data users complete metadata can be very beneficial. The more information given about data, the easier it is for users to find and to get a first understanding of, which in turn increases the chances that they will reuse it.

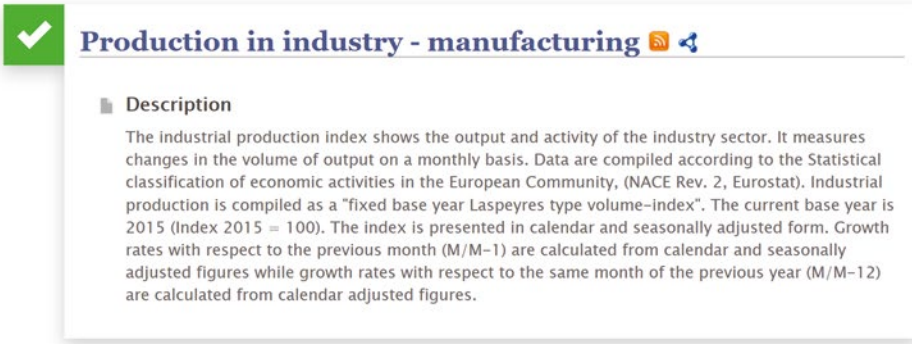
The following metadata information should be provided in order to increase the findability of data:

- title
- description
- keywords
- categories
- temporal information
- spatial information.

When filling in this metadata information, data publishers should make sure that the information given is as precise, accurate and helpful as possible. Keep in mind that a potential user has probably never seen your data before and needs to get a clear understanding of what your data is about.

Good example

This screenshot shows that a detailed description is given for the 'Production in industry – manufacturing' data set. This helps potential users to get an overview of what to expect in the data set.



Bad example

In this example, the description of the data set is the very similar to the data set's title and does not provide any helpful information. A user would have a hard time getting a grasp of what the 'Interest rates – monthly data' data set may contain.

Helpful links and tools

Title	Description	Link
What is metadata and why is it as important as data itself?	An online article from opendatasoft that provides helpful information about metadata (e.g. definition, purpose).	https://www.opendatasoft.com/blog/2016/08/25/what-is-metadata-and-why-is-it-important-data

1.1.1.2. Mark null values explicitly as such

Dimension	Findability
Indicator	Findability
Metrics	• Number of null values

Sometimes, data is simply not complete. However, a missing value is no reason for not publishing the data in question. In order to avoid confusion, the data provider should clearly mark missing values as null values. Users that are not familiar with the data can thus recognise that the data was not simply forgotten, because the null value serves as special marker indicating that the value does not exist. In other words, a null value is a visual representation of a missing value.

There are several ways of indicating a null value, for example by marking the missing value with 'NULL' or 'NA'. However, if you notice that within your data you have a high percentage of null values within one row or column, you should consider deleting the respective column or row as it probably does not bring any added value to data users.

The example below shows a CSV table with data about page visits. In the table labelled 'bad example', missing values are indicated by simply leaving fields empty. This is ambiguous and may lead to errors during further processing. In contrast, the table labelled 'good example' shows the same data, but with missing values clearly marked as such.

Bad example	Good example
Year; Visitors, Viewing time	Year; Visitors; Viewing time
2014;768954;00:03:18	2014;768954;00:03:18
2013;;00:02:59	2013;null;00:02:59
2013;822101;00:02:59	2012;792967;00:02:52
2011;721519;	2011;721519;null
2010;707402;00:03:50	2009;429430;00:03:16

1.1.2. Accessibility

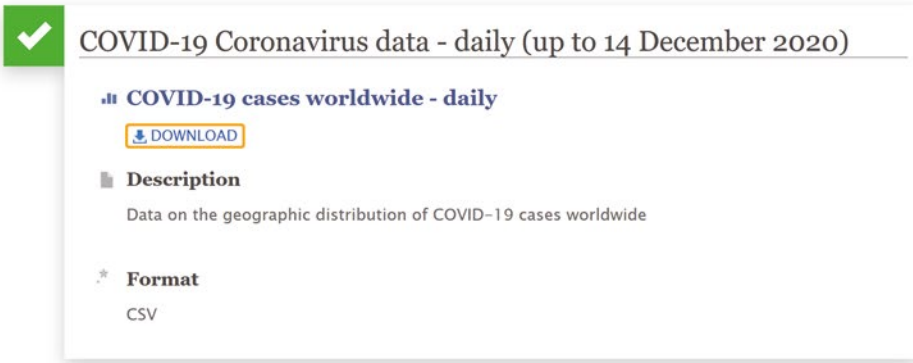
1.1.2.1. Publish data without restrictions

Dimension	Accessibility
Indicator	Accessibility
Metrics	• Downloadable without registration

One of the core principles of open data is its accessibility: data should be accessible and available to the widest range of users possible to avoid limiting its potential reuse. To allow easy consumption and further processing, no access restrictions should be in place, regardless of whether these require manual intervention (e.g. registration) or can be bypassed automatically (e.g. providing credentials). This also applies to the files themselves, for example encrypted archives. Keep in mind that any access restriction limits the number of potential data users and so, if possible, should be avoided.

Good example

This screenshot shows a data set which is directly downloaded when the user clicks on 'download'. No registration or password is needed.



Bad example

This example shows a data set which cannot be downloaded without a password. This hampers its reuse and is not in line with open data principles.

```
code:      "authentication_required"
error:     true
message:   "You must be logged in to access this resource"
```

Helpful links and tools

Title	Description	Link
Ten principles for opening up government information	Description of the core open data principles. Pay attention to Principle 4 'Ease of physical and electronic access'.	https://sunlightfoundation.com/policy/documents/ten-open-data-principles/

1.1.2.2. Provide an accessible download URL

Dimension	Accessibility
Indicator	Accessibility/availability
Metrics	• Download URL given • Download URL accessible

Data can only be reused by others if it is accessible. Typically, the main point of access is a download URL, which must be set in the metadata and be accessible, i.e. reachable via a browser. This means the data publisher must ensure that when a user clicks on the download URL provided, this URL functions properly and the user can directly download the data.

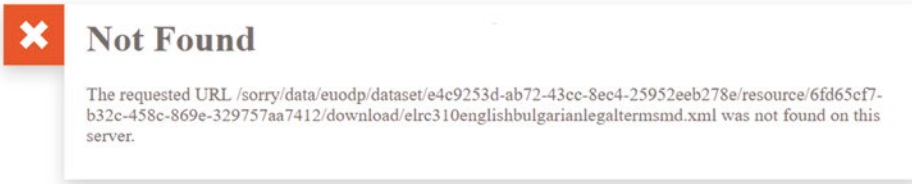
Good example

This screenshot shows three download URLs given for a data set, each pointing to a different file format. The download begins directly when the user clicks on the download button.

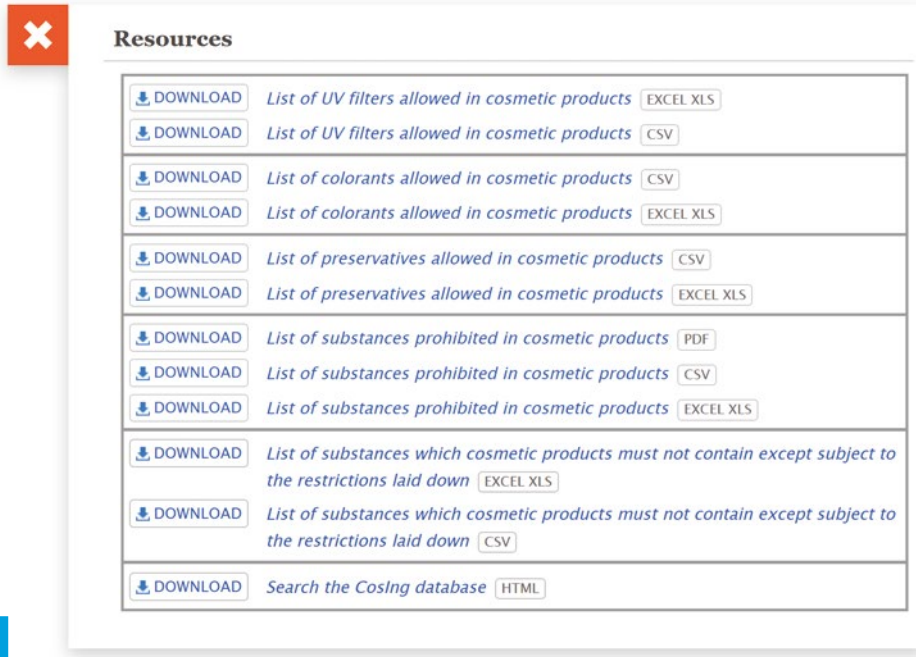
Bad examples

These screenshots show a download URL which redirects the user to another web page instead of initiating a file download.

As already mentioned, the download URL should be not only available but also accessible, meaning it should not return an error when clicking on it. An invalid download URL, for example, returns a 404 error (not found), which makes the data inaccessible for the user.

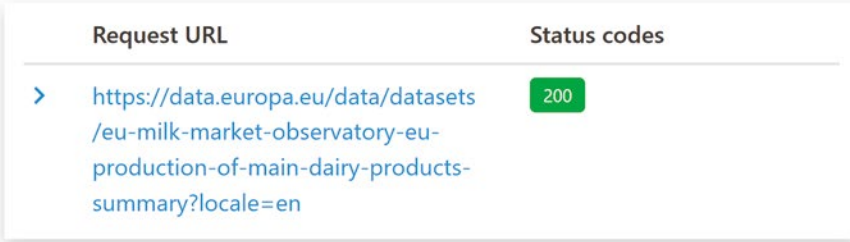


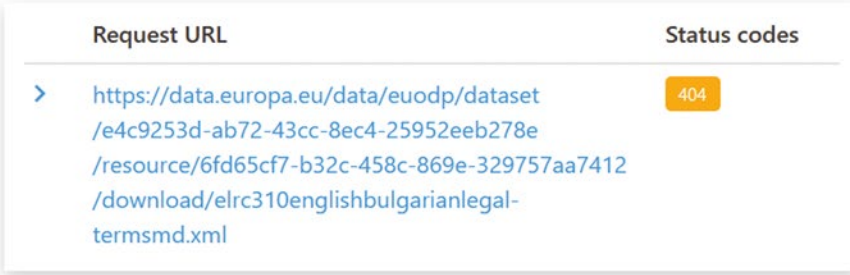
Another bad example is depicted in the following screenshot. Here, the data set includes several resources that are of a different nature. In this case, it is recommended that the data set be split into six individual data sets as shown, to make sure that the resources for each set cover the same content (possibly in different formats).



Helpful links and tools

Title	Description	Link
HTTP status check	This tool can be used to manually check whether the download link of your data is accessible or not. After the check, the tool states the status code of the link – the colour of the status code indicates whether the link works properly or not (open source).	https://httpstatus.io/





HTTP status codes	This site provides a list of all status codes and their meanings (open source).	https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml

1.1.3. Interoperability

1.1.3.1. Formatting of date and time

Dimension	Interoperability
Indicator	Conformity/compliance
Metrics	• Conformity of date formats

Data (and metadata) often contains dates and times. Depending on the regional conditions, there are different ways of stating dates, which can lead to confusion. The following example highlights the issue with ambiguous date formats: 01/02/2020 could mean either 1 February 2020 or 2 January 2020, depending on a country's customs.

Therefore, date and time should always be encoded as ISO 8601 (YYYY-MM-DD hh:mm:ss). If applicable, the time zone used should be stated. The time zone is always derived from Coordinated Universal Time (UTC).

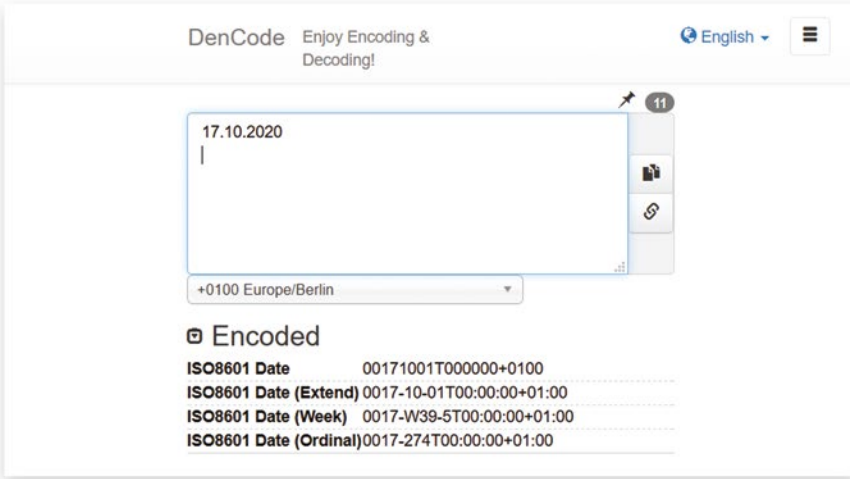
The examples below show a CSV table with data about page visits. In the bad examples, the time format does not follow a consistent schema, making it very hard to process correctly. In contrast, the good examples show the same data with all timestamps formatted using ISO 8601 encoding.

Bad example	**Good example**
Year; Visitors, Viewing time	Year; Visitors; Viewing time
2014;768954;3:18	2014;768954;00:03:18
2013;822101;00:02:59	2013;822101;00:02:59
2012;792967;0:02:52	2012;792967;00:02:52
2011;721519;03:44	2011;721519;00:03:44
2010;707402;3m:50s	2010;707402;00:03:50
2009;429430;3:16	2009;429430;00:03:16

Bad example	**Good example**
Start Date; End Date	Start Date; End Date
01.01.2014; 31.03.2014	2014-01-01; 2014-03-31
01.01.2014; 30.06.2016	2014-01-01; 2016-06-12

Helpful links and tools

Title	Description	Link
ISO standard for date and time	An introduction to ISO 8601 for date and time formats (open source / commercial).	https://www.iso.org/iso-8601-date-and-time-format.html
DenCode	ISO date and time generator, encoder and decoder. This tool helps you to convert your data into ISO 8601 formats (open source).	https://dencode.com/date/iso8601



1.1.3.2. Formatting of decimal numbers and numbers in the thousands

Dimension	Interoperability
Indicator	Conformity/compliance
Metrics	—

Data often contains numbers. In this section we do not want to give detailed information on how to handle different numeric types (integer, float, double), but rather recommendations on how to deal with numbers in a more general sense. For example, a comma is often used to separate whole numbers from decimals. This might cause problems, for example in a CSV file when the separator between the values is set as a comma. To avoid the unintended interpretation of a comma separating a whole number from a decimal, a dot should be used instead.

When dealing with large numbers, sometimes a thousand separator is used, for example a dot or white space. Again, this can lead to misinterpretation – especially when the data is being processed automatically – and might mean the user has to clean the data before they can reuse it. Thousand separators should therefore not be used.

Bad example	Good example
0,53	0.53
789.654	789654
789 654	789654
25.026,8	25026.8

1.1.3.3. Make use of standardised character encoding

Dimension	Interoperability
Indicator	Conformity/compliance
Metrics	• Character encoding issues

In order to make sure that characters are displayed correctly, and to ensure the greatest possible compatibility with applications processing data, a standardised character encoding should always be used. Typically, UTF-8 is the encoding of choice on the web. UTF-8 is a character encoding for Unicode, an international standard for the representation of all meaningful characters. With this, all characters, whether Latin alphabet or Japanese characters, are displayed correctly. To ensure that your data can be blended and reused with other data from international sources and to avoid problems during machine processing, it is helpful to use an internationally recognised and widely used character set encoding from the outset.

However, in general you should avoid using any special characters in your data, even if they are part of UTF-8. In doing so, backward compatibility with older systems is encouraged.

Depending on the program you are using, UTF-8 must be activated explicitly in the 'Save-As' dialogue. In Microsoft Excel and in LibreOffice Calc, for example, you can select the character encoding explicitly when saving a CSV file. If a different character set than UTF-8 is used in your data, it is essential to specify this in the metadata. DCAT-AP does not specify a dedicated field for this information. However, Inspire suggests adding this type of information to the 'media type' description ([15]).

([15]) https://ies-svn.jrc.ec.europa.eu/projects/metadata/wiki/INSPIRE_profile_of_DCAT-AP_-_Reference#Character-encoding

Bad example

This screenshot shows a data set which does not use UTF-8, as you can see in the text highlighted in yellow.

10 Trends Transforming Education as We Know It		2017-11-14T00:00:00+01:00
Back in the Game â€" Reclaiming Europeâ€™s Digital Leadership	Video Explainer	2017-11-13T00:00:00+01:00
Nord Stream 2 â€" Divide et Impera Again?	Avoiding a Zero-Sum Game	2017-10-27T00:00:00+02:00

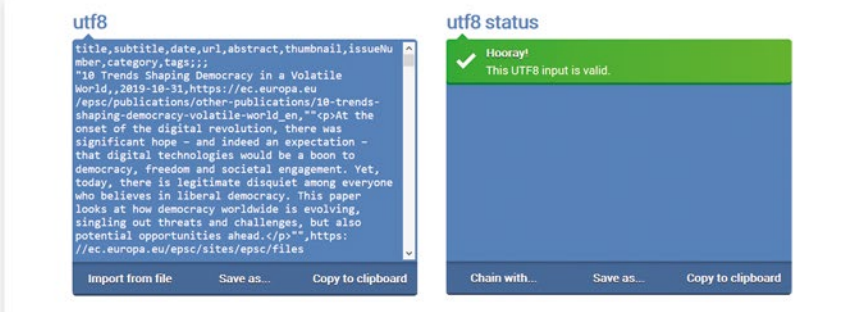
Good example

This screenshot shows the same data set, this time encoded in UTF-8.

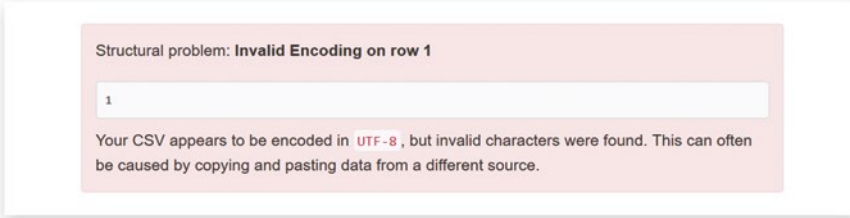
10 Trends Transforming Education as We Know It		2017-11-14T00:00:00+01:00
Back in the Game — Reclaiming Europe's Digital Leadership	Video Explainer	2017-11-13T00:00:00+01:00
Nord Stream 2 – Divide et Impera Again?	Avoiding a Zero-Sum Game	2017-10-27T00:00:00+02:00

Helpful links and tools

Title	Description	Link
UTF-8 validator	This online tool helps you check your input for valid UTF-8 encoding (open source).	https://onlineutf8tools.com/validate-utf8



CSVLint	You can use this tool to check whether your CSV file contains any encoding issues. If the tool detects that your CSV is encoded in UTF-8 but contains invalid characters, you will get an error message (open source).	https://csvlint.io



1.1.4. Reusability

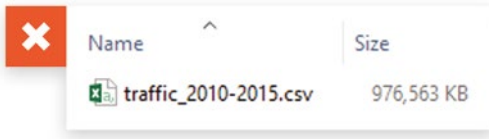
1.1.4.1. Provide an appropriate amount of data

Dimension	Reusability
Indicator	Relevance
Metrics	• Appropriate amount of data

Depending on the data to be published, the meaning of the term 'appropriate' can differ greatly. It is important to publish all relevant data, but caution should be taken not to blindly publish all available data without considering its usefulness. On the other hand, data publishers have to make sure that a sufficient amount of the data is published, so that there is enough context and users can derive value from it. It would be rather useless for data users to find a CSV file with only two lines.

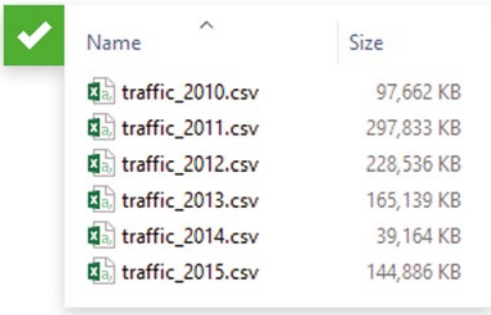
However, there is no clear indication of what an appropriate amount of data is, as this is highly dependent on the purpose a user has in mind. To find a good balance, you could start by asking yourself whether all the data you are about to publish really provides value to others. If not, you could think about reducing your data if it seems like a large amount. On the other hand, you could ask yourself if the amount of data you want to publish is sufficient for users to make sense of it and to add value, or if you should add more data or context.

Bad example



The file in the screenshot contains fictitious traffic data aggregated over the course of 6 years. In total, the file is nearly 1 GB in size. If users are only interested in data for 1 year, they still have to download the entire file.

Good example



In contrast, this screenshot shows the same data split by year. This way, the file size remains reasonable and users can download the exact files they need. Each file should be published in a separate data set.

1.1.4.2. Consider community standards

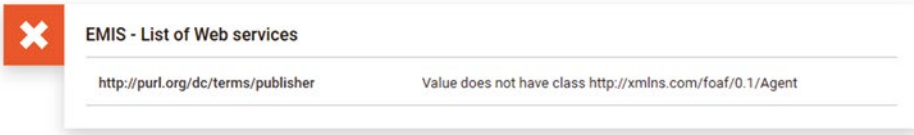
Dimension	Reusability
Indicator	Consistency
Metrics	• Compliance with community standards

Community standards are a powerful tool for ensuring conformity across files and formats of a common domain. Using community standards makes it easier to reuse data, as all data following the same standard looks similar – for example it is organised in a standardised way, the documentation follows a common template or a common vocabulary is used. Lots of different community standards exist, for example standards for specific domains such as climate and forecast, astrophysics or statistical data. But there are also non-domain-specific standards, such as DCAT-AP, a standard for storing data catalogue metadata.

Depending on the use case, there may be validators that aid in checking files against such a standard. Ensuring the compliance of files against community standards greatly helps reusability and eases further processing. To make sure that your data is being reused, you should consider using community standards.

Bad example

This screenshot shows a message from a SHACL validation which produced an error against the DCAT-AP community standard. More precisely, the value that was attached to the property *dcterms:publisher* was not of the required type.



Good example

This screenshot shows a data set with an XML resource that conforms to its schema.

Helpful links and tools

Title	Description	Link
FAIR list of community standards	List of community standards for various domains (open source).	https://www.go-fair.org/fair-principles/r1-3-metadata-meet-domain-relevant-community-standards/
SHACL validator	This online tool allows you to validate your RDF files against a given standard (open source).	https://shacl.org/playground/

1.1.4.3. Remove duplicates from your data

Dimension	Reusability
Indicator	Consistency
Metrics	• Freeness from duplicates

Each piece of data should be unique. Duplicate data is of no additional value. Instead, it lowers the quality of the data as it might cause errors during further processing. For example, a data user performing analytics on the data will receive biased results as some data are duplicates.

Examples

The table labelled 'bad example' shows a CSV file where some rows are duplicates. In contrast, the rows in the table labelled 'good example' are all distinct, and no row carries the same information as another one.

Bad example	Good example
Year; Visitors; Viewing time	Year; Visitors; Viewing time
2014;768954;00:03:18	2014;768954;00:03:18
2013;822101;00:02:59	2013;822101;00:02:59
2013;822101;00:02:59	2012;792967;00:02:52
2011;721519;00:03:44	2011;721519;00:03:44
2010;707402;00:03:50	2010;707402;00:03:50
2010;707402;00:03:50	2009;429430;00:03:16

Helpful links and tools

Most ETL tools provide functions for detecting missing data and handling null values.

1.1.4.4. Increase the accuracy of your data

Dimension	Reusability
Indicator	Accuracy
Metrics	• Percentage of accurate cells

Accuracy can be measured in many dimensions. What accuracy means specifically, how it is measured and what result is deemed acceptable always depend on the specific use case. For example, in CSV files, each cell of a column could be checked for accuracy against an encoding format, for example ISO 8601 for dates. The ratio between accurate and inaccurate cells could then give users a first impression of what to expect from the data and how difficult processing may be. Higher accuracy is typically an indicator of higher-quality data.

Examples

When evaluating the conformity of the 'Viewing time' column against ISO 8601 encoding, the table labelled 'bad example' would score an accuracy rating of 50 %, since half of the cells follow this time format. In contrast, the table labelled 'good example' would yield an accuracy score of 100 %, since all timestamps are correctly encoded.

Bad example	Good example
Year; Visitors; Viewing time	Year; Visitors; Viewing time
2014;768954;3:18	2014;768954;00:03:18
2013;822101;00:02:59	2013;822101;00:02:59
2012;792967;0:02:52	2012;792967;00:02:52
2011;721519;03:44	2011;721519;00:03:44
2010;707402;3m:50s	2010;707402;00:03:50
2009;429430;3:16	2009;429430;00:03:16

1.1.4.5. Provide information on byte size

Dimension	Reusability
Indicator	Accuracy
Metrics	• Content size accuracy

When publishing data, it is good to also provide information on the distributions' byte size. This information helps users and automated processes to anticipate what to expect before downloading the actual file. Also, this information enables filtering by size.

Bad example

This screenshot shows a distribution without the *dcat:byteSize* property set.

```
<dcat:Distribution rdf:about="https://example.org/sample-distribution">
  <dct:title> Sample Title</dct:title>
  <dct:language rdf:resource="http://publications. europa.eu/resource/authority/language/DEU"/>
  <dcat:downloadURL rdf:resource="https://example.org/download"/>
</dcat:Distribution>
```

Good example

This screenshot shows a distribution for which the *dcat:byteSize* property is set.

```
<dcat:Distribution rdf:about="https://example.org/sample-distribution">
  <dct:title>Sample Title</dct:title>
  <dct:language rdf:resource="http://publications.europa.eu/resource/authority/language/DEU"/>
  <dcat:downloadURL rdf:resource="https://example.org/download"/>
  <dcat:byteSize rdf:datatype="http://www.w3.org/2001/XMLSchema#decimal"> 12168</dcat:byteSize>
</dcat:Distribution>
```

1.2. Format-specific recommendations

1.2.1. CSV

Please check the general recommendations in Section 1.1, which also apply to CSV files.

1.2.1.1. Use a semicolon as a delimiter

Dimension	Interoperability
Indicator	Machine readability/processability
Metrics	• Processability of file format and media type

Even though the name 'CSV' (comma separated values) implies the use of commas as separators between each value, we recommend using semicolons instead. Commas are often used in the values themselves (for example when using decimal numbers). To avoid a comma being interpreted as a separator, it would need to be masked. Masking is not a problem in itself, but it can be a source of error if you overlook a comma that needs to be masked. Semicolons are used less often within the actual values and should thus be used as delimiters in CSV files.

The delimiter is always set between two values, and the last value in line is not followed by a delimiter as depicted in the examples. Make sure that there are no spaces or tabs on either side of the delimiters in the row.

Bad example	**Good example**
Year; Visitors; Viewing time;	Year; Visitors; Viewing time
2013; 822101;00:02:59;	2013;822101;00:02:59
2012;792967;00:02:52;	2012;792967;00:02:52
2011; 721519;00:03:44;	2011;721519;00:03:44
2010;707402;00:03:50;	2010;707402;00:03:50
2009;429430;00:03:16;	2009;429430;00:03:16

Helpful links and tools

Title	Description	Link
CSVLint	This online tool helps you to detect white space between delimiters and values (open source).	https://csvlint.io

1.2.1.2. Use one file per table

Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• Data following a given schema • Processability of file format and media type

Each CSV file should only contain one table. If the table to be published consists of several sheets, a CSV file should be created for each sheet. Different structuring would break table structure and hinder machine interpretability.

Bad example	Good example
File: View_And_Country_Statistics.csv	**File:** View_Statistics.csv
Year;Visitors;Viewing time	Year;Visitors;Viewing time
2014;768954;00:03:18	2014;768954;00:03:18
2013;822101;00:02:59	2013;822101;00:02:59
2012;792967;00:02:52	2012;792967;00:02:52
2011;721519;00:03:44	2011;721519;00:03:44
2010;707402;00:03:50	2010;707402;00:03:50
2009;429430;00:03:16	2009;429430;00:03:16
	File: Country_Statistics.csv
Country;Population;Capital	Country;Population;Capital
Germany;83149300;Berlin	Germany;83149300;Berlin
Finland;5517919;Helsinki	Finland;5517919;Helsinki
France;66993000;Paris	France;66993000;Paris
Spain;47100396;Madrid	Spain;47100396;Madrid
Italy;60262701;Rome	Italy;60262701;Rome

1.2.1.3. Avoid white space and additional information in the file

Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• Data following a given schema • Processability of file format and media type

It is important to ensure that the file only contains data which belongs to the actual table, like column headers and values of the relevant table entries. Often, tabular data is added, for example table titles and empty rows. This can give more visual clarity for human beings, but can lead to difficulties when automatically processing data, because blank lines and table titles are also interpreted automatically. This is illustrated in Figure 5 and Figure 6.

Figure 5 shows a spreadsheet which is well arranged for human beings, with a table title (blue) and blank lines (yellow). Figure 6 shows the same data in a text editor. The table title line (blue) and the blank lines (yellow) have been interpreted. As this can lead to failures in processing, additional content other than column headers and actual values, i.e. table titles and blank lines, should be avoided.

Retrieval statistics website XY, 2009–2014			
Year	Visitor	Viewing time	Viewing time per page
2014	768954	00:03:18	00:00:45
2013	822101	00:02:59	00:00:44
2012	792967	00:02:52	00:00:42

Figure 5. Blank lines and titles opened in a spreadsheet

Retrieval statistics website XY, 2009–2014
;;;
Year; Visitors; Viewing time per Visitor; Viewing time per page
;;;
2014;768954;00:03:18; 00:00:45
2013;822101;00:02:59; 00:00:44
2012;792967;00:02:52; 00:00:42

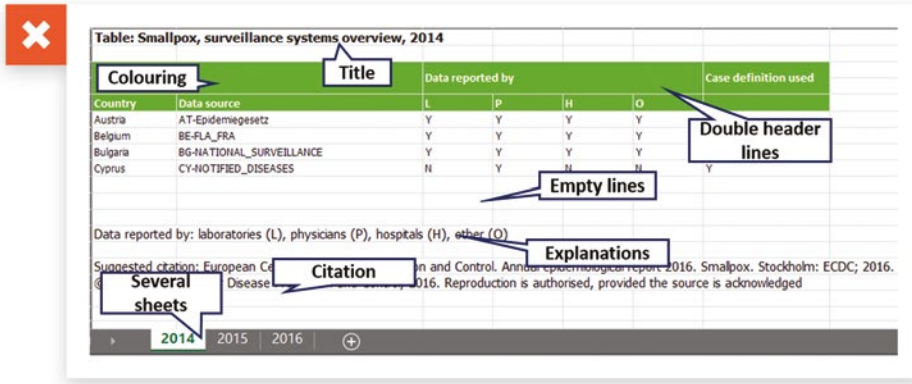
Figure 6. Interpretation of blank lines and titles in CSV files

Explanations, modification dates, sheet names, etc. are not part of a CSV file and should be listed in the metadata of the resulting data set.

NB: Do not confuse sheet names with column headers. The latter is part of the actual data and should thus be included in the first row.

Bad example

The following example contains some additional information and formatting next to the actual content data, which makes it difficult to automatically process the data. The issues are labelled with a text box.



How to address these issues

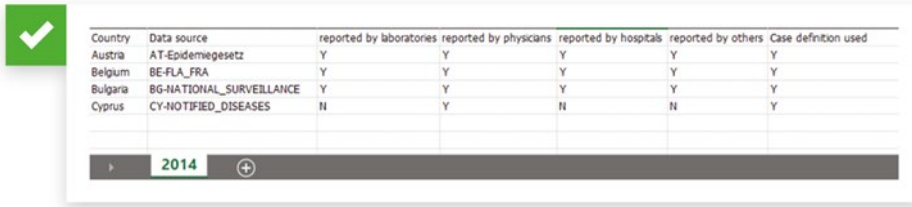
Title	Delete the title in the actual CSV file. Instead, the title is represented within the name of the distribution.

Double header	CSV files should only contain one header line. The good example below indicates how the double header line can be resolved in this case.
Empty lines	Delete all empty lines as they do not provide any extra value and make data processing difficult.
Explanations	Explanations can be very helpful for users to get a better understanding of your data, but do not put them directly in the CSV file. Instead, explanations and descriptions should be stored in suitable metadata properties, for example *dct:description*. Another option is to store the metadata in a dedicated document. This should then be linked to the data set containing the data to be documented.

Several sheets	A CSV file should only contain one sheet. To solve this issue, you could provide yearly data in a separate data set.

Good example

The good example below shows a cleared version of the same data. All additional information has been removed.

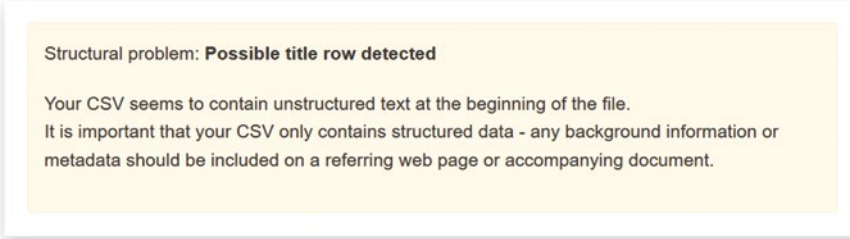


Country	Data source	reported by laboratories	reported by physicians	reported by hospitals	reported by others	Case definition used
Austria	AT-Epidemiegesetz	Y	Y	Y	Y	Y
Belgium	BE-FLA_FRA	Y	Y	Y	Y	Y
Bulgaria	BG-NATIONAL_SURVEILLANCE	Y	Y	Y	Y	Y
Cyprus	CY-NOTIFIED_DISEASES	N	Y	N	N	Y

2014

Helpful links and tools

Title	Description	Link
CSVLint	This online tool helps you to detect blank rows within your CSV file. It also checks whether your CSV contains a title (open source).	https://csvlint.io



1.2.1.4. Insert column headers

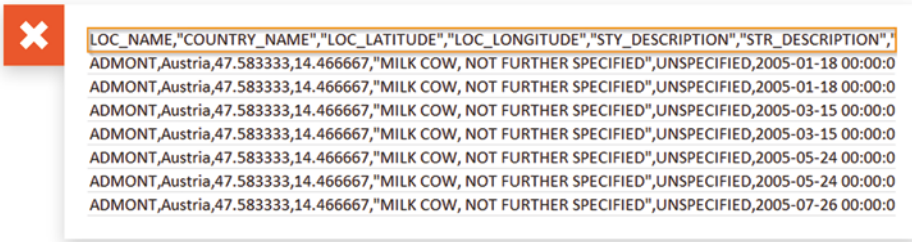
Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• Data following a given schema • Processability of file format and media type

Column headers should always be included in the first row of a CSV file. Without headers, it is difficult for users to interpret the meaning of the data. Therefore, it is also important that the column headers be chosen so that the meaning of the associated values can be clearly identified. There are no specific recommendations regarding headers made up of more than one word. Spaces are allowed in the headers as well as the actual fields.

The following bad example shows a CSV file with no headers. The good example depicts how a header line could look.

Bad example	Good example
	Year; Visitors; Viewing time
2014;768954;00:03:18	2014;768954;00:03:18
2013;822101;00:02:59	2013;822101;00:02:59
2012;792967;00:02:52	2012;792967;00:02:52
2011;721519;00:03:44	2011;721519;00:03:44
2010;707402;00:03:50	2010;707402;00:03:50
2009;429430;00:03:16	2009;429430;00:03:16

If the column headers are not self-explanatory, a corresponding explanation should be included in the metadata, for example in the field for description. Alternatively, the explanations can also be put into separate files and linked via the *foaf:page* property. Further recommendations on how to document data can be found in Part 3. The following example shows a CSV file with a header line that is not self-explanatory. In this case, it is useful for data users not familiar with the data set to have more information about the meaning of the headers. However, data publishers should pay particular attention to the labelling of their headers. If they are clear and understandable for everyone, providing additional explanations in metadata is not necessary.



```
LOC_NAME,"COUNTRY_NAME","LOC_LATITUDE","LOC_LONGITUDE","STY_DESCRIPTION","STR_DESCRIPTION",'
ADMONT,Austria,47.583333,14.466667,"MILK COW, NOT FURTHER SPECIFIED",UNSPECIFIED,2005-01-18 00:00:0
ADMONT,Austria,47.583333,14.466667,"MILK COW, NOT FURTHER SPECIFIED",UNSPECIFIED,2005-01-18 00:00:0
ADMONT,Austria,47.583333,14.466667,"MILK COW, NOT FURTHER SPECIFIED",UNSPECIFIED,2005-03-15 00:00:0
ADMONT,Austria,47.583333,14.466667,"MILK COW, NOT FURTHER SPECIFIED",UNSPECIFIED,2005-03-15 00:00:0
ADMONT,Austria,47.583333,14.466667,"MILK COW, NOT FURTHER SPECIFIED",UNSPECIFIED,2005-05-24 00:00:0
ADMONT,Austria,47.583333,14.466667,"MILK COW, NOT FURTHER SPECIFIED",UNSPECIFIED,2005-05-24 00:00:0
ADMONT,Austria,47.583333,14.466667,"MILK COW, NOT FURTHER SPECIFIED",UNSPECIFIED,2005-07-26 00:00:0
```

Helpful links and tools

Title	Description	Link
CSV on the web	W3C primer for the use of CSV on the web. Section 1.1 explains the structure of a CSV and refers to headers (open source).	https://w3c.github.io/csvw/primer/

1.2.1.5. Ensure that all rows have the same number of columns

Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• Data following a given schema • Processability of file format and media type

It is very important that each row has the same number of columns and thus follows the structure of a CSV. This means that each row should have the same number of delimiters. If one row is missing a value, this usually gets interpreted as 'null'. This can lead to erroneous processing of data.

If your CSV contains rows with a different number of columns, you should check whether there is an issue with incorrectly escaped values (e.g. a value contains a semi-colon which is not masked and thus gets interpreted as a delimiter).

Bad example	**Good example**
Year, Visitors	Year; Visitors; Viewing time
2014;768954;00:03:18;	2014;768954;00:03:18
2013;822101	2013;822101;00:02:59
2012;792967;00:02: 52;	2012;792967;00:02:52
2011;721519;00:03:44;	2011;721519;00:03:44
2010; 00:03:50	2010;707402;00:03:50
2009;429430;00:03:16;	2009;429430;00:03:16

Helpful links and tools

Title	Description	Link
GoodTables	GoodTables is a tool to validate tabular data and checks, for example whether all rows have the same number of columns (open source).	https://frictionlessdata.io/tooling/good-tables/#a-simple-example
CSVLint	This online tool helps to detect rows that contain a different number of columns (open source).	https://csvlint.io

1.2.1.6. Indicate units in an easily processable way

Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• Data following a given schema • Processability of file format and media type

Numeric values should follow the general recommendations given in Section 1.1. A value's unit should be stated in the relevant column header so that the unit becomes clear to the user. Additionally, the unit of measurement used in the data can be referenced in the corresponding *stat:dcat* metadata.

If the unit varies, a dedicated column for the unit should be used. Putting the unit directly behind the numeric value in one cell makes it harder for users to process the data. Ideally, the corresponding values from the controlled vocabulary ([16]) should be used.

Bad example

Ingredient	Amount
Carbohydrates	16g
Magnesium	2mg

Good example

Ingredient	Amount	Unit
Carbohydrates	16	g
Magnesium	20	mg

Better example

Ingredient	Amount	Unit
Carbohydrates	16	<http://publications.europa.eu/resource/authority/measurement-unit/GRM>
Magnesium	20	<http://publications.europa.eu/resource/authority/measurement-unit/MGM>

([16]) https://op.europa.eu/en/web/eu-vocabularies/at-dataset/-/resource/dataset/measurement-unit

1.2.2. XML

Please check the general recommendations in Section 1.1, which also apply to XML files.

1.2.2.1. Provide an XML declaration

Dimension	Reusability
Indicator	Consistency
Metrics	• Compliance with community standards

Each XML file should have a complete XML declaration. This contains metadata regarding the structure of the document and is important for applications to properly process the file. For example, information regarding XML version and character encoding are typically present in the declaration.

Bad example

This screenshot shows an XML without a declaration.

```
<fruits>
        <fruit>
                <type>Apple</type>
                <origin>Germany</origin>
                <drupe>true</drupe>
        </fruit>
</fruits>
```

Good example

This screenshot shows the same XML with a properly formatted declaration.

```
<?xml version="1.0" encoding="UTF-8"?>
<fruits>
        <fruit>
                <type>Apple</type>
                <origin>Germany</origin>
                <drupe>true</drupe>
        </fruit>
        <fruit>
                <type>Grape</type>
                <origin>Italy</origin>
                <drupe>false</drupe>
        </fruit>
</fruits>
```

1.2.2.2. Escape special characters

Dimension	Reusability
Indicator	Consistency
Metrics	—

When special characters are used in XML files they need to be escaped. This ensures a sound file structure and prevents applications used for processing the file from misinterpreting the data. Escaping is done by replacing them with the equivalent XML entities. An overview of the characters is shown in Table 1.

Table 1. Characters that need escaping in XML

	Escaped form	Replaced by
Ampersand	&amp;	&
Less than	&lt;	<
Greater than	&gt;	>
Quotes	&quot;	"
Apostrophe	&apos;	'

Bad example This screenshot shows an XML without escaping.

```
<fruit id="&1">
        <type>Apple <</type>
        <origin>> Germany</origin>
        <description>"Very tasty!"</description>
</fruit>
```

Good example This screenshot shows the same XML with properly escaped characters.

```
<fruit id="& amp;1">
        <type>Apple &lt;</type>
        <origin>&gt; Germany</origin>
        <description>&quot;Very tasty!&quot;</description>
</fruit>
```

Helpful links and tools

Title	Description	Link
XML Escape / Unescape	Online tool that escapes special characters in text so they can be used in XML (open source).	https://www.freeformatter.com/xml-escape.html

1.2.2.3. Use meaningful names for identifiers

Dimension	Reusability
Indicator	Consistency
Metrics	• Compliance with community standards

All identifiers, whether tags or attributes, should have meaningful names and should ideally not be used twice. There are no official recommendations regarding the spelling of the identifiers, so you can use, for example, camelCase or PascalCase. However, different forms should not be mixed together. Furthermore, special characters should not be used in the identifiers.

Bad example

This example shows XML with the 'fairtrade' identifier (i.e. the element's name) not being written using PascalCase or camelCase, making it harder to read by humans and thus prone to processing errors.

```
<fruits>
        <type>Apple</type>
        <origin>Germany</origin>
        <drupe>true</drupe>
        <fairtrade>true</fairtrade>
</fruits>
```

Good example

This screenshot shows XML with an identifier which consists of two words being concatenated via camelCase.

```
<fruits>
        <type>Apple</type>
        <origin>Germany</origin>
        <drupe>true</drupe>
        <fairTrade>true</fairTrade>
</fruits>
```

Helpful links and tools

Title	Description	Link
Title Case	This tool converts phrases consisting of multiple words into various case formats (open source).	https://titlecase.com/

1.2.2.4. Use attributes and elements correctly

Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• Data following a given schema • Processability of file format and media type

While there is no mandatory binding directive as to whether data should be encoded in elements or attributes, it has been established as best practice that information that is part of the actual data should be represented by elements. Metadata that contains additional information should instead be implemented as attributes. For example, in the snippet labelled 'good example', the 'id' is part of the metadata and thus an attribute of a 'fruit' type element. In the snippet labelled 'bad example', information has been encoded in attributes for which elements should have been used instead.

Bad example

This screenshot shows XML in which data has been encoded using attributes where elements would have been more suitable.

```
<fruit type="apple" drupe="true" id="1">
        <origin>Germany</origin>
</fruit>
```

Good example

This screenshot shows XML in which data and metadata have been encoded using elements and attributes correctly.

```
<fruit id="1">
        <type>Apple</type>
        <origin>Germany</origin>
        <drupe>true</drupe>
</fruit>
```

Helpful links and tools

Title	Description	Link
XML specification	W3C recommendations for XML (open source).	https://www.w3.org/TR/2006/REC-xml11-20060816/

1.2.2.5. Remove program-specific data

Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• Data following a given schema • Processability of file format and media type

XML, as with any open format, should always be independent of specific programs or tools used for processing the files. This allows the user to choose the tool they prefer for processing the data without having to sanitise it first.

Bad example

This screenshot shows XML which contains a version number of a hypothetical program that has been used for the creation or processing of the file. This information does not add anything to the data and should thus be removed.

```
<fruits>
 <fruit id="1">
        <type>Apple</type>
        <description>Very tasty</description>
 </fruit>
<fruits>
<createdWith version="1.0">myXmlTool</createdWith>
```

1.2.3. RDF

Please check the general recommendations in Section 1.1, which also apply to RDF files.

1.2.3.1. Use HTTP URIs to denote resources

Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• Data following a given schema • Processability of file format and media type

Resource IDs should be HTTP URIs, since ideally these allow direct access to the resource in question. They also make resources indexable by search engines, which enhances their findability. This only applies, however, if these identifiers are persistent and do not contain volatile information, for example credentials.

Bad example This screenshot shows a resource in RDF/XML which is not denoted via HTTP URI.

```
<vcard:hasAddress rdf:resource="myAddress">
```

Good example This screenshot shows a resource in RDF/XML which is denoted via HTTP URI.

```
<vcard:hasAddress
rdf:resource="http://www.w3.org/2006/vcard/ns#myAddress">
```

1.2.3.2. Use namespaces when possible

Dimension	Reusability
Indicator	Consistency
Metrics	• Compliance with community standards

While namespaces are not required for processing RDF, they reduce verbosity and file size. Similarly to the recommendations regarding plain XML, identifiers for classes should be written in PascalCase while identifiers for properties are typically written in camelCase.

Bad example RDF without namespaces and identifier conventions applied can be harder to read.

```
<rdf:rdf>
        <rdf:description rdf:about="http://myresource">
        <http://mynamespace#myproperty>Sample
                </http://mynamespace#:myproperty>
        </rdf:description>
</rdf:rdf>
```

Good example This screenshot shows the use of namespaces as well as conventions for class and property identifiers, which improves readability.

```
<rdf:RDF xmlns:myNamespace="http://myNamespace#">
        <rdf:Description rdf:about="http://myResource">

<myNamespace:myProperty>Sample</myNamespace:myProperty>
        </rdf:description>
</rdf:rdf>
```

Helpful links and tools

Title	Description	Link
Ontotext	Tool that allows import of structured data and conversion to RDF data. During the import namespaces can be defined. (commercial / open source).	https://www.ontotext.com/products/ontotext-platform/
Anzo	Platform that allows transformation of structured and semi-structured data into RDF graphs. Querying data and analysis thereof is then possible on the graph. (commercial / open source).	https://www.cambridgesemantics.com/product/
OpenRefine	OpenRefine is a refinement tool for cleaning data. It features a built-in exporter to generate RDF files (open source).	https://openrefine.org/
Trifacta Wrangler	Trifacta Wrangler is a suite of data preparation tools. It allows transformation of different formats, thereby cleaning and merging data. RDF is among the supported formats (commercial).	https://www.trifacta.com/products/wrangler-editions/#wrangler

1.2.3.3. Use existing vocabularies when possible

Dimension	Interoperability
Indicator	Conformity/compliance Machine readability/processability
Metrics	• DCAT-AP compliance of metadata • Conformity of file formats and licences • Conformity to access property values • Data following a given schema • Usage of controlled vocabularies

Existing vocabularies should be reused whenever possible. The Publications Office provides such vocabularies for use with DCAT-AP ([17]).

Bad example

This screenshot shows the licence of a data set referenced without using the controlled vocabulary. This makes further processing much harder and is error prone with regard to spelling.

```
<dcterms:license rdf:resource="http://CC_BY_4_0"/>
```

([17]) https://op.europa.eu/en/web/eu-vocabularies/

Good example

This screenshot shows the same licence being referenced using the controlled vocabulary published by the European Commission.

```
<dcterms:license rdf:resource=

http://publications.europa.eu/resource/authority/licence/CC_BY_4_0/>
```

Helpful links and tools

Title	Description	Link
EU Vocabularies	EU Vocabularies provides access to vocabularies managed by the EU institutions and bodies (open source).	https://op.europa.eu/en/web/eu-vocabularies
Ontorion	This tool is a plugin for Microsoft Excel 2010 and 2013 that can be used to import RDF data into Excel from a SPARQL endpoint, thereby converting RDF to XLS (open source).	https://www.cognitum.eu/semantics/Tools/SparqlExcelTools.aspx

1.2.4. JSON

Please check the general recommendations in Section 1.1, which also apply to JSON files.

1.2.4.1. Use suitable data types

Dimension	Interoperability
Indicator	Machine readability/processability
Metrics	• Processability of file format and media types

JSON permits the following data types.

- Null value (absence of a value), represented by the keyword 'null'.
- Boolean values, either true or false.
- Strings, where the masking of single characters works the same way as with CSV files.
- Numbers and simple sequences of the digits 0–9, optionally with a sign and/or decimal point.
- Lists, also called arrays, enclosed in square brackets, the individual elements separated by commas. Lists can also be empty.
- Objects, enclosed in curly brackets and containing any number of comma-separated key-value pairs.

For further processing it is important to use suitable data types. For example, numbers should be encoded using the number type, and Boolean values using the Boolean type. This prevents errors stemming from encoding prohibited values, for example a value other than 'true', 'false' or 'null' for Boolean fields.

Bad example

This screenshot shows a JSON file with various data types. All information has been encoded using strings, regardless of the underlying data type.

```
{
  "type": "apple",
  "fairTrade": "true",
  "amount": "5"
}
```

Good example

This screenshot shows the same JSON file, this time with dedicated data types where applicable.

```
{
  "type": "apple",
  "fairTrade": true,
  "amount": 5
}
```

Helpful links and tools

Title	Description	Link
JSONLint	This online tool checks whether your input is valid JSON (open source).	https://jsonlint.com

1.2.4.2. Use hierarchies for grouping data

Dimension	Interoperability
Indicator	Machine readability/processability
Metrics	• Processability of file format and media types

Instead of attaching all fields to the root JSON object, data should be semantically grouped. This improves readability by humans and can enhance performance when processing the file. Also, many tools allow collapsing objects and arrays, which allows users to quickly navigate the desired information.

Bad example

This screenshot shows a JSON file with grouped data. All information has been attached to the root object. For objects with a larger number of fields, this can quickly reduce readability.


```
{
  "type": "apple",
  "calcium": 6.0,
  "magnesium": 5.0,
  "zinc": 0.0
}
```


Good example

The screenshot shows the same JSON file with semantically grouped data.


```
{
  "type": "apple",
  "nutrients": {
     "calcium": 6.0,
     "magnesium": 5.0,
     "zinc": 0.0
}
```


1.2.4.3. Only use arrays when required

Dimension	Interoperability
Indicator	Machine readability/processability
Metrics	• Processability of file format and media types

Data should only be encoded into arrays if the size of the list is dynamic, i.e. not known beforehand or subject to change. If this is not the case, using explicit fields makes further processing easier. In addition, it cannot be guaranteed that the values in an array are always provided in the same order, which makes the data prone to erroneous interpretation.

Bad example

This screenshot shows a JSON file with array usage, but it is unclear what type of nutrients the values are referring to. Dedicated fields would have been more useful in this scenario.


```
{
  "type": "apple",
  "nutrients": [6.0, 5.0, 0.0]
}
```


Good example

This screenshot shows a JSON file in which array usage is useful.


```
{
  "type": "apple",
  "nutrients": {
     "calcium": 6.0,
     "magnesium": 5.0,
     "zinc": 0.0
  }
}
```


1.2.5. APIs

Please check the general recommendations in Section 1.1, which also apply to APIs.

1.2.5.1. Use correct status codes

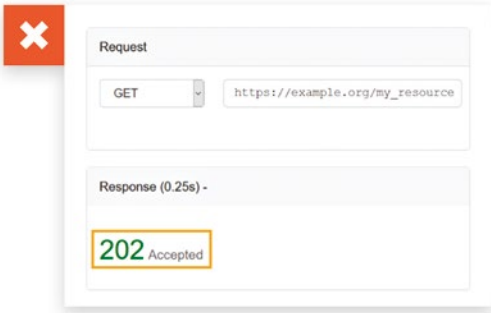
Dimension	Accessibility
Indicator	Accessibility/availability
Metrics	• Access URL accessible • Download URL accessible • Downloadable without registration

APIs are typically available via URLs, which should be available publicly without credentials. These URLs can be called via various methods defined in HTTP. In addition to the actual payload, each server also sends a status code when answering requests from clients. These codes provide information on whether the request was served flawlessly. For example, 200 indicates no problems, whereas 404 indicates that a resource was not found. An overview of the available methods and typically used status codes is shown in Table 2.

Table 2. Overview of methods and status codes

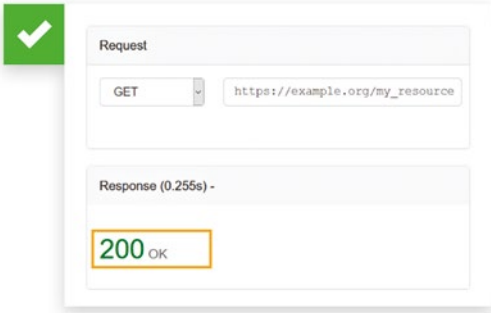
Name	Description	Statuscode	
		Flawless	Errors
GET	Retrieves a resource without altering it.	200	404
POST	Uploads a new resource to the server.	201	400, 401, 403
PUT	Replaces an existing resource with a new complete resource.	200, 204	400, 401, 403
PATCH	Replaces selected parts of a resource without replacing it entirely.	200, 204	400, 401, 403
DELETE	Deletes an existing resource from a server.	200, 204	400, 401, 403

Bad example



This screenshot shows a GET request on a resource. However, contrary to the HTTP standard, the status code '202 Accepted' is returned.

Good example



This screenshot shows a GET request on a resource. As intended by the HTTP standard, the correct status code '200 OK' is returned.

Helpful links and tools

Title	Description	Link
HTTP status codes	This site provides a list of all status codes and their meanings (open source).	https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml

1.2.5.2. Set correct headers

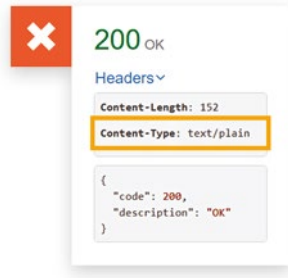
Dimension	Reusability
Indicator	Accuracy
Metrics	• File format accuracy • Content size accuracy

In addition to status codes the HTTP standard allows metadata to be encoded via headers. These are not part of the actual payload (i.e. website or resource) that is requested. However, information of interest to the consumers of the data can be encoded here. Of course, appropriate headers must be used. Also, the metadata encoded must be accurate and match the payload. A list of typical headers is shown in Table 3.

Table 3. Typical headers that are used in conjunction with APIs

Header	Description
Content-Type (server)	Indicates the payload's MIME ([18]) type.
Content-Length (server)	Indicates the size of the payload in bytes.
Content-MD5 (server)	Indicates the checksum of the payload. A checksum allows the user to check if the payload has been downloaded in its entirety and not been corrupted or changed during transmission.
Accept (client)	If an endpoint offers a payload in multiple formats, this header can be used by the client to indicate the desired format. Like the Content-Type header, a MIME type must be specified.

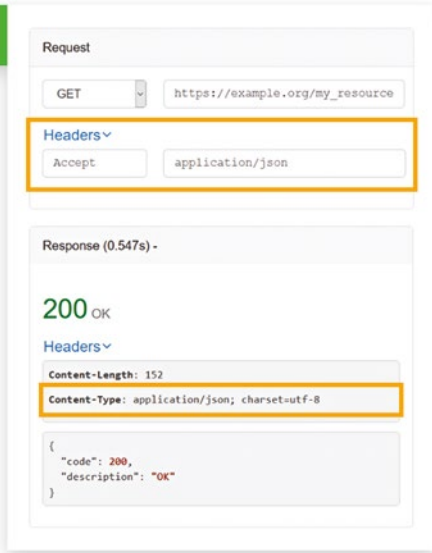
Bad example



This screenshot shows the two headers 'Content-Length' and 'Content-Type' returned for a GET request on a resource. However, the 'Content-Type' header is incorrect, since JSON has been returned instead of plain text.

([18]) https://www.iana.org/assignments/media-types/media-types.xhtml

Good example



This screenshot shows the two headers 'Content-Length' and 'Content-Type' returned for a GET request on a resource. Note that the 'Accept' header has been sent with the request, indicating the desired format of the resource.

Helpful links and tools

Title	Description	Link
HTTP headers	W3C RFC about HTTP headers (open source).	https://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html

1.2.5.3. Use paging for large amounts of data

Dimension	Reusability
Indicator	Relevance
Metrics	• Appropriate amount of data

Requesting large amounts of data can easily create high loads on the server. In some cases, not all data is required, or not all at once. In order to reduce this load and increase response times, pagination should be used when applicable. This means slices of data are served instead of an entire data set. The client can state in the request which slice to retrieve, as well as its size. This is typically achieved using the parameters shown in Table 4.

Table 4. Pagination using offset and limit parameters

Parameter	Behaviour
Offset	Specifies the resource from which to start counting.
Limit	Specifies how many resources shall be retrieved.

Good example

This screenshot shows an exemplary call to an API supporting pagination. The offset is five and the limit is three, therefore the results 6, 7 and 8 are returned.

```
// https://demo.ckan.org/api/3/action/package_list?limit=3&offset=5
{
  "help" : "https://demo.ckan.org/api/3/action/help_show?name=package_list" ,
  "success" : true ,
  "result" : Array [3] [
    "02a8c314-e726-44fb-88da-2e535e788675" ,
    "02p-expenditure-over-25k-apr-19" ,
    "02p-expenditure-over-25k-feb-20"
  ]
}
```

Helpful links and tools

Title	Description	Link
Postman	Tool for making HTTP requests (commercial / open source).	https://www.postman.com/

1.2.5.4. Document the API

Dimension	Reusability
Indicator	Understandability
Metrics	• Description of data given • Documentation of data given

APIs should be specified as thoroughly as possible. This includes available paths, returned formats and status codes. If an API allows file uploading, the expected payload should also be stated. Examples help potential users in using APIs. One standard used to describe APIs is OpenAPI. It allows either JSON or YAML to be used for describing APIs.

Example

An example of an OpenAPI specification for an API serving data about fruit can be seen in the screenshot below.

```
openapi: 3.0.0

info:
  version: "1.0"
  title: fruit-service

paths:
  /fruit/{type}/{format}:
    get:
      summary: Returns statistics about
                  the requested fruit
      operationId: getStatistics
      parameters:
          - name: type
          in: path
          description: Type of fruit
          required: true
          schema:
            type: string
      responses:
        '200':
          description: JSON
          content:
            application/json:
              schema:
                $ref: "#/fruitSchema"
        "404":
          description: Resource not found
```

Helpful links and tools

Title	Description	Link
OpenAPI Specification	Specification of the OpenAPI format (commercial / open source).	https://swagger.io/specification/
Swagger editor	An online editor for creating and validating OpenAPI specifications (commercial / open source).	https://swagger.io/tools/swagger-editor/
Swagger UI	An online visualiser for displaying OpenAPI specifications (commercial / open source).	https://swagger.io/tools/swagger-ui/

2. Recommendations for data standardisation (with EU controlled vocabularies) and data enrichment

Introduction

With the ever increasing volume of data on the web, standardisation is becoming more and more relevant. Data which must be converted to a common format before processing hinders further usage. Data standardisation increases processability.

Enrichment is the concept of linking data from external sources to existing data sets. This data can come from, among others, public authorities or open knowledge bases. Linking data can increase its value by creating new relationships and thus allowing new kinds of analysis. For example, if the database of a car authority containing licence plates and models is enriched and made interoperable with data about where the cars are registered, insights can be gained into which manufacturers are preferred in certain parts of the country.

Standardisation and enrichment are both part of the enriching process (see Figure 7).

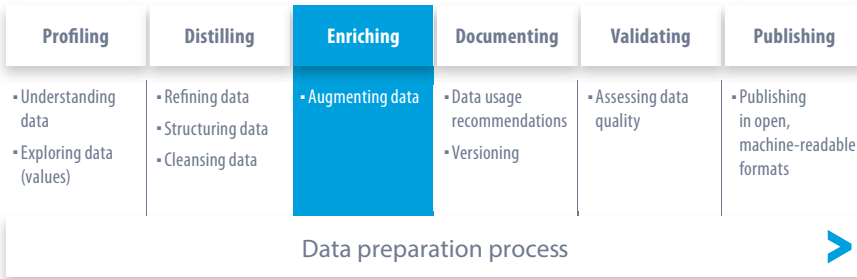


Figure 7. Data preparation process – Enriching

The aim of this section is to give data providers actionable recommendations which enable them to publish data sets with a high level of standardisation and enrichment.

Section 2.1 contains recommendations on how to reuse concepts from controlled vocabularies. Another aspect of reusing controlled vocabularies is the harmonisation of labels, which is introduced in Section 2.2. Section 2.3 focuses on recommendations regarding dereferencing label translations. Finally, Section 2.4 gives recommendations on how to link and augment data.

2.1. Reuse unambiguous concepts from controlled vocabularies

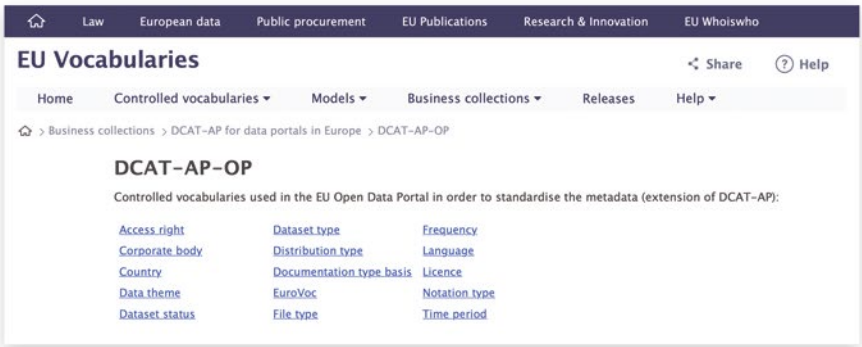
This section covers recommendations on achieving a high level of standardisation and on enriching data. A higher level of data standardisation can be achieved by integrating RDF vocabularies such as lists of authorities, taxonomies, classifications or terminologies into the data. These controlled vocabularies describe, identify and organise the concepts unambiguously in their area of expertise and can be reused to harmonise or augment the data.

In RDF vocabularies, each concept is identified by a unique resource identifier (URI), enabling any system to refer to it unambiguously. This is important, as it allows these concepts to be referenced from anywhere once they have been published on the web. These references then form a web of linked data, i.e. the semantic web ([19]). Using URIs that are only valid and/or unique within a certain namespace would fail to achieve this.

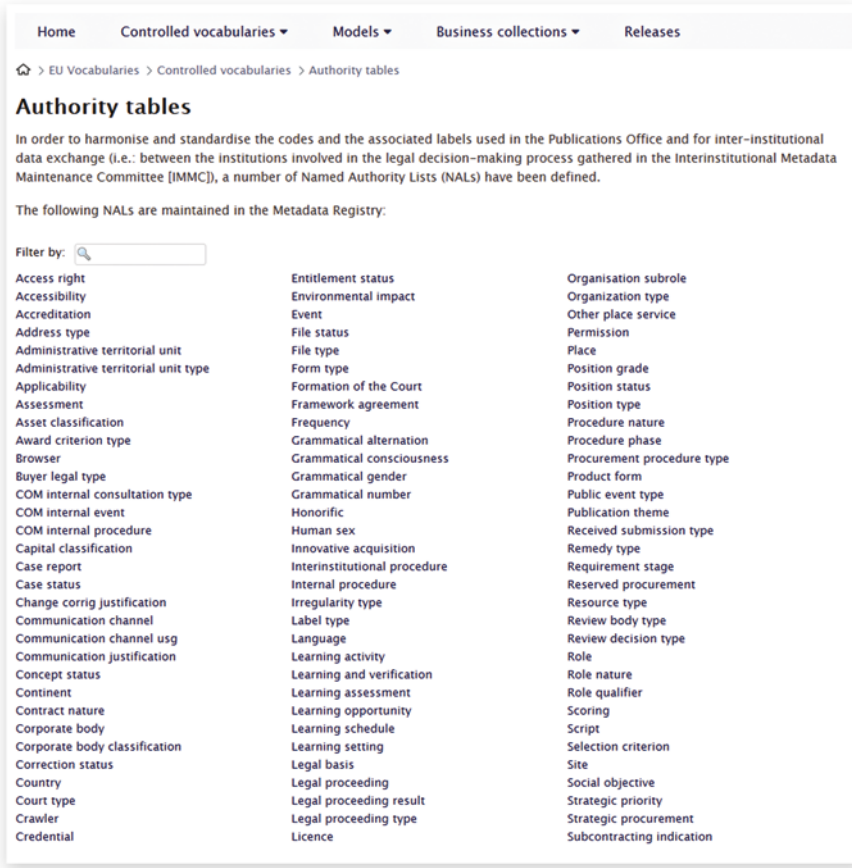
Example

The European multilingual classification of skills, competences, qualifications and occupations (ESCO) works as a dictionary, describing, identifying and classifying professional occupations, skills and qualifications relevant for the EU labour market and education and training. Those concepts are reused in different online platforms to use ESCO for services such as matching jobseekers to jobs on the basis of their skills or suggesting training to people who want to reskill or upskill.

The Publications Office maintains a number of EU Vocabularies and Authority Tables used in data.europa.eu in order to standardise the metadata (extension of DCAT-AP), as can be seen in the screenshots below.



([19]) https://www.w3.org/standards/semanticweb/



2.2. Harmonise the tables

Instead of hardcoding labels into data, these labels can be referenced by unique identifiers, i.e. URIs. This means that if those labels change, the reference does not need to be adjusted, reducing the burden of maintenance for data providers.

Example

The example below shows a sample of data from Erasmus statistics:

STUDENT_ID	HOME_INSTITUTION_CDE	HOME_INSTITUTION_	STUDENT_AGE_VALUE	STUDENT_GENDER_CDE	STUDENT_NATIONALITY
X1	E ALICANT01	ES	21	F	ES
X3	D KOLN07	DE	25	F	DE
X6	SF TURKU01	FI	21	F	FI

The value provided in the 'student nationality' or 'home institutions' fields can be standardised based on the Country table. Instead of encoding the country code (here: ES, DE or FI), the corresponding unique identifiers for these countries can be provided and additional data can be derived from the country identifier, such as the country label or the country ISO code (two or three letters).

HOME_INSTITUTION_CTRY_	Unique Identifier (Country Table)	Label (Country Table)	ISO 3166-2 (Country Table)	ISO 3166-3 (Country Table)
ES	http://publications.europa.eu/resource/authority/country/ESP	Spain	ES	ESP
DE	http://publications.europa.eu/resource/authority/country/DEU	Germany	DE	GER
FI	http://publications.europa.eu/resource/authority/country/FIN	Finland	FI	FIN

2.3. Dereference the translation of a label

Once the labels are indicated by the unique identifiers from the controlled vocabularies, the URIs can be dereferenced. This allows the label to be resolved in any language supported by the controlled vocabulary.

Example

The example illustrates the 'meter' concept in the Measurement Unit table. The 'meter' concept is represented by different preferred labels (prefLabel) in the different EU official languages. Assigning the 'meter' concept <http://publications.europa.eu/resource/authority/measurement-unit/MTR> URI into your data set enables the automatic dereferencing of the different language versions and offers enhanced access to your data. In addition, even if one translation is updated in the Authority Table, there is no need to update the 'meter' concept. The URI will automatically dereference the right value from the table.

The two screenshots below show the 'meter' concept in RDF (top) from the corresponding Measurement Unit table from the EU Vocabularies website (bottom).

```
<skos:Concept rdf:about="http://publications.europa.eu/resource/authority/measurement-unit/MTR" at:deprecated="false">
    <skos:inScheme rdf:resource="http://publications.europa.eu/resource/authority/measurement-unit"/>
    <at:authority-code>MTR</at:authority-code>
    <at:op-code>MTR</at:op-code>
    <atold:op-code>MTR</atold:op-code>
    <dc:identifier>MTR</dc:identifier>
    <at:start.use>1952-07-23</at:start.use>
    <skos:prefLabel xml:lang="bg">метър</skos:prefLabel>
    <skos:prefLabel xml:lang="cs">metr</skos:prefLabel>
    <skos:prefLabel xml:lang="da">meter</skos:prefLabel>
    <skos:prefLabel xml:lang="de">Meter</skos:prefLabel>
    <skos:prefLabel xml:lang="el">μέτρο</skos:prefLabel>
    <skos:prefLabel xml:lang="en">metre</skos:prefLabel>
    <skos:prefLabel xml:lang="et">meeter</skos:prefLabel>
    <skos:prefLabel xml:lang="fi">metri</skos:prefLabel>
    <skos:prefLabel xml:lang="fr">mètre</skos:prefLabel>
    <skos:prefLabel xml:lang="ga">méadar</skos:prefLabel>
    <skos:prefLabel xml:lang="hr">metar</skos:prefLabel>
    <skos:prefLabel xml:lang="hu">méter</skos:prefLabel>
    <skos:prefLabel xml:lang="it">metro</skos:prefLabel>
    <skos:prefLabel xml:lang="lv">metri</skos:prefLabel>
    <skos:prefLabel xml:lang="lt">metras</skos:prefLabel>
    <skos:prefLabel xml:lang="mt">metru</skos:prefLabel>
    <skos:prefLabel xml:lang="nl">meter</skos:prefLabel>
    <skos:prefLabel xml:lang="pl">metr</skos:prefLabel>
    <skos:prefLabel xml:lang="pt">metro</skos:prefLabel>
    <skos:prefLabel xml:lang="ro">metru</skos:prefLabel>
    <skos:prefLabel xml:lang="sk">meter</skos:prefLabel>
    <skos:prefLabel xml:lang="sl">meter</skos:prefLabel>
    <skos:prefLabel xml:lang="es">metro</skos:prefLabel>
    <skos:prefLabel xml:lang="sv">meter</skos:prefLabel>
</skos:Concept>
```

Concept scheme

Measurement unit

Authority Table

URI: http://publications.europa.eu/resource/authority/measurement-unit

About | Browse content | Documentation

Filter by:

Code	Label	Valid since	Valid until	Predecessor	Successor	Definition
2N	decibel	1952-07-23				The decibel (dB) is a uni or the logarithm of a rat
3C	manmonth	1952-07-23				The manmonth is a unit
AD	byte	1952-07-23				The byte is a unit of infc
AMP	ampere	1952-07-23				The ampere (A) is the ba charges, such as electro
BAR	bar	1952-07-23				The bar (bar) is a non-SI 000 Pa, 10^5 Pa, which is
BIT	bit	1952-07-23				The bit (b), short for "bi computing and digital c no decision at the lowes
BQL	becquerel	1952-07-23				The becquerel (Bq) is a c radioactive material in w
C34	mole	1952-07-23				The mole (mol) is the ba chemical substance that
C45	nanometre	1952-07-23				The nanometre (nm) is a (0.000000001 m).
CDL	candela	1952-07-23				The candela (cd) is the b intensity, in a given dire intensity in that directio

2.4. Linking and augmenting your data

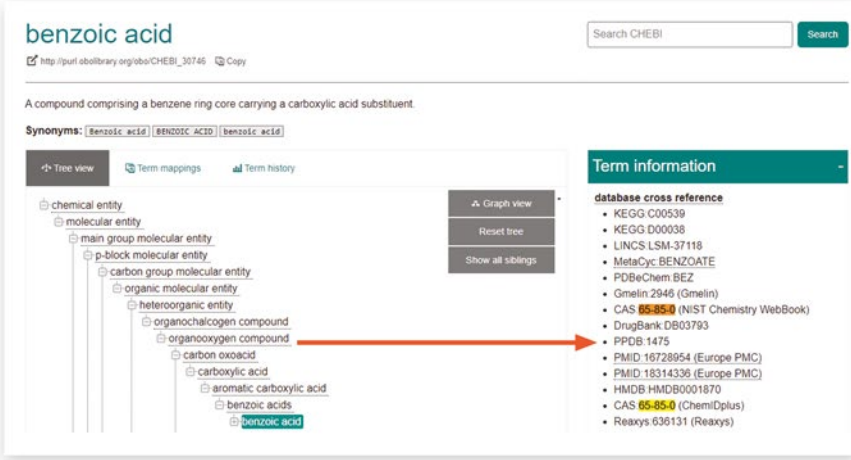
Consistent use of unique identifiers also allows linkage and augmentation with external data. This adds value to existing data by linking to new concepts or aspects of existing data. Optimal usage of controlled vocabularies can be achieved using a four-star data format such as RDF or JSON-LD.

Example

The screenshot below shows a dataset, which lists the names of common cosmetic ingredients with their corresponding chemical abstract registry number (CAS number).

Reference number	Chemical name / INN	Name of Common Ingredients Glossary	CAS Number	EC Number
1	Benzoic acid and its sodium salt	BENZOIC ACID; SODIUM BENZOATE	65-85-0 / 532-32-1	200-618-2 / 208-534-8
1a	Salts of benzoic acid other than th	AMMONIUM BENZOATE / BUTYL BENZ(	1863-63-4 /2090-05-3 / 582-25-2	217-468-9 / 218-235-4 /
2	Propionic acid and its salts	PROPIONIC ACID / AMMONIUM PROPI(	79-09-4 / 17496-08-1 / 4075-81-4	201-176-3 / 241-503-7 /
3	Salicylic acid and its salts	SALICYLIC ACID, CALCIUM SALICYLATE,	69-72-7[1]/ 824-35-1[2]/ 18917-8	200-712-3[1]/ 212-525-4
4	Hexa -2,4-dienoic acid and its salt	SORBIC ACID / CALCIUM SORBATE / SO	110-44-1 / 7492-55-9 / 7757-81-5	203-768-7 / 231-321-6 /
7	Biphenyl -2-ol	O-PHENYLPHENOL	90-43-7	201-993-5
9	Inorganic sulphites and hydrogen	SODIUM SULFITE / AMMONIUM BISULF	7757-83-7 / 10192-30-0 / 10196-(	231-821-4 / 233-469-7 /
11	Chlorobutanol	CHLOROBUTANOL	57-15-8	200-317-6
12	4 -Hydroxybenzoic acid and its sal	4-HYDROXYBENZOIC ACID / METHYLPA	99-96-7 /99-76-3 /36457-19-9 /1(	202-804-9 /202-785-7 /2

Linking the CAS number with the corresponding value in the *Chemical Entities of Biological Interest dictionary* (ChEBI) would augment the data set with new derived data (synonyms, standardised identifiers and cross references). ChEBI is a freely available dictionary of molecular entities focused on 'small' chemical compounds and is shown in the screenshot below.



For example, CAS number 65-85-0 (benzoic acid) has the identifier CHEBI:30746 represented by the URI <http://purl.obolibrary.org/obo/CHEBI_30746> (indicated by the red arrow).

Another example is illustrated by two data sets published in JSON-LD – 'Pesticide (New)' and 'Pesticide-EPPO'. They contain data from the collection of the single active substances and their maximum residue levels (MRLs) related to foodstuffs intended for human or animal consumption in the European Union.

The <http://data.europa.eu/dph/id/pesticides/product/0110020> data set, which is shown in the screenshot below, corresponds to the foodstuff product 'Orange'.


```
{
        "@id" : "http://data.europa.eu/dph/id/pesticides/product/0110020",
        "@type" : "http://data.europa.eu/dph/def/pesticides#Product",
        "hasParentProduct" :
"http://data.europa.eu/dph/id/pesticides/product/0110000",
        "productType" :
"http://data.europa.eu/dph/id/pesticides/productType/4",
        "label" : [ {
          "@language" : "cs",
          "@value" : "Pomeranče"
        }, {
          "@language" : "mt",
          "@value" : "Larinġ"
        }, {
          "@language" : "pt",
          "@value" : "Laranjas"
        }, {
          "@language" : "pl",
          "@value" : "Pomarańcze"
        }, {
          "@language" : "nl",
          "@value" : "Sinaasappelen"
        }, {
          "@language" : "lv",
          "@value" : "Apelsīni"
        }, {
          "@language" : "it",
          "@value" : "Arance dolci"
        }, {
          "@language" : "lt",
          "@value" : "Apelsinai"
        }, {
          "@language" : "hr",
          "@value" : "Naranča"
},
```

```
{
            "@language" : "fr",
            "@value" : "Oranges"
          }, {
            "@language" : "en",
            "@value" : "Oranges"
          }, {
            "@language" : "el",
            "@value" : "Πορτοκάλια"
          }, {
            "@language" : "fi",
            "@value" : "Appelsiinit"
          }, {
            "@language" : "es",
            "@value" : "Naranjas"
          }, {
            "@language" : "et",
            "@value" : "Apelsinid"
          }, {
            "@language" : "de",
            "@value" : "Orangen"
          }, {
            "@language" : "la",
            "@value" : "Citrus sinensis"
          }, {
            "@language" : "da",
            "@value" : "Appelsiner"
          },{
            "@language" : "da",
            "@value" : "Appelsiner"
          }, {
            "@language" : "hu",
            "@value" : "Narancs"
          }, {
            "@language" : "bg",
            "@value" : "Портокали"
          }, {
            "@language" : “sk”,
            "@value" : “pomaranče”
          }, {
            "@language" : "ro",
            "@value" : "Portocale"
          }, {
            "@language" : "sv",
            "@value" : "Apelsiner"
          }, {
            "@language" : "sl",
            "@value" : "Pomaranče"
          } ]
}
```

This foodstuff product contains 'Fenoxicarb pesticide residues' identified by the URI <http://data.europa.eu/dph/id/pesticides/substance/299>. This relationship is shown in the screenshot of the MRL data set below (the 'Orange' foodstuff is highlighted in blue).


```
{
        "@id" : "http://data.europa.eu/dph/id/pesticides/mrlHst/100283",
        "@type" :
"http://data.europa.eu/dph/def/pesticides#MaximumResidueLevel",
        "applicationDate" : "2008-09-01T00:00:00",
        "mrlValue" : "2",
        "product" : "http://data.europa.eu/dph/id/pesticides/product/0110020",
        "residue" : "http://data.europa.eu/dph/id/pesticides/substance/299"
}
```


All information regarding this pesticide can be retrieved by dereferencing the URI highlighted in green. This yields the data shown in the screenshot below:


```
{
        "@id" : "http://data.europa.eu/dph/id/pesticides/substance/299",
        "@type" : [ "http://data.europa.eu/dph/def/pesticides#Residue",
"http://data.europa.eu/dph/def/pesticides#Substance" ],
        "hasSubstanceCode" : "S029900",
        "isLegislatedBy" : [
"http://data.europa.eu/dph/id/pesticides/legalResource/publication-473",
"http://data.europa.eu/dph/id/pesticides/legalResource/publication-518",
"http://data.europa.eu/dph/id/pesticides/legalResource/publication-1" ],
        "isMemberOf" :
"http://data.europa.eu/dph/id/pesticides/substance/299",
        "label" : [ {
          "@language" : "pl",
          "@value" : "Fenoksykarb"
        }, {
          "@language" : "it",
          "@value" : "Fenoxicarb"
        }, {
          "@language" : "es",
          "@value" : "Fenoxicarb"
        }, {
          "@language" : "ro",
          "@value" : "Fenoxicarb"
        }, {
          "@language" : "bg",
          "@value" : "Феноксикарб"
        }, {
          "@language" : "pt",
          "@value" : "Fenoxicarbe"
        }, {
          "@language" : "fi",
          "@value" : "Fenoksikarbi"
},
```

```
{
            "@language" : "lt",
            "@value" : "Fenoksikarbas"
          }, {
            "@language" : "sk",
            "@value" : "Fenoxykarb"
          }, {
            "@language" : "lv",
            "@value" : "Fenoksikarbs"
          }, {
            "@language" : "cs",
            "@value" : "Fenoxykarb"
          }, {
            "@language" : "sv",
            "@value" : "Fenoxikarb"
          }, {
            "@language" : "hu",
            "@value" : "Fenoxikarb"
          }, {
            "@language" : "de",
            "@value" : "Fenoxycarb"
          }, {
            "@language" : "hr",
            "@value" : "Fenoxycarb"
          }, {
            "@language" : "nl",
            "@value" : "Fenoxycarb"
          }, {
            "@language" : "mt",
            "@value" : "Fenoxycarb"
          }, {
            "@language" : "fr",
            "@value" : "Fenoxycarb"
          }, {
            "@language" : "da",
            "@value" : "Fenoxycarb"
          }, {
            "@language" : "en",
            "@value" : "Fenoxycarb"
          }, {
            "@language" : "et",
            "@value" : "Fenoksükarb"
          }, {
            "@language" : "sl",
            "@value" : "Fenoksikarb"
          }, {
            "@language" : "el",
            "@value" : "Φενοξυκάρμπ (Fenoxycarb)"
          } ]
},
```

The 'Pesticide-EPPO' data set contains cross references between the entities contained in the 'Pesticides (New)' data set and the items in the EPPO Global Database. More specifically, the data set links the instances of the 'Pesticides (New) Product' class to the possible corresponding EPPO Global Database items, which enables a five-star ranking.

Helpful links and tools

Title	Description	Link
EU Vocabularies and Authority Tables	EU Vocabularies and Authority Tables have been developed for the Publications Office in order to facilitate the exchange of data between the different information systems of the EU institutions (legislation, calls for tender, etc.) and describe data sets (open source).	https://op.europa.eu/en/web/eu-vocabularies/authority-tables https://op.europa.eu/en/web/eu-vocabularies/dcat-ap-op
OpenRefine (OntoText)	Tool for cleaning and extending data from external sources (open source).	https://openrefine.org/ https://openrefine.org/documentation.html https://github.com/OpenRefine/OpenRefine
Cleaning data with OpenRefine	This article describes how to discover inconsistencies in data and how to diagnose the accuracy of data with OpenRefine ([20]).	https://doaj.org/article/3ccd075407a4481c85c0d00d65a003c0
OntoRefine	ata transformation tool that can be used for converting tabular data into RDF (commercial / open source).	http://graphdb.ontotext.com/documentation/free/loading-data-using-ontorefine.html#ontorefine-overview-and-features

([20]) De Wilde, M., van Hooland, S. and Verborgh, R. (2013), 'Cleaning data with OpenRefine', *The Programming Historian*, 1 August 2013, Editorial Board of the Programming Historian, United Kingdom.

3. Recommendations for documenting data

Introduction

More and more data is published on the web every day. However, in order to improve interoperability and ease further processing, this data has to be documented. This way users can know what to expect with regard to both syntax (i.e. structure) and semantics (i.e. content). In addition to improving data quality for users, documentation can enhance the value of data, as misinterpretation of data becomes less likely when context is provided. This document covers aspects relevant to step four of the data preparation process, which is shown in Figure 8.

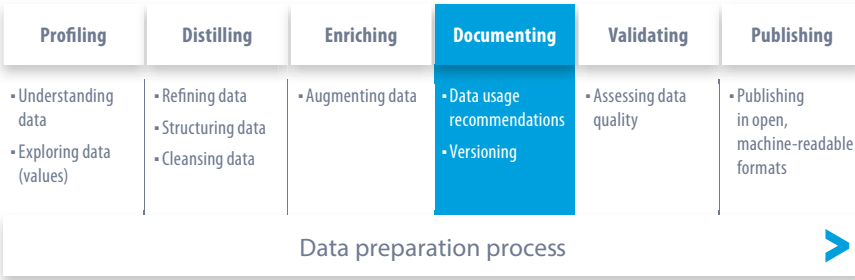


Figure 8. Data preparation process – Documenting

The aim of this section is to give actionable recommendations covering the tasks involved in documenting data, aided by tools. This includes documenting structure and meaning, as well as proper versioning.

A general recommendation on where to publish documentation is given in Section 3.1. Section 3.2 contains recommendations on using schemas to document data structures. In addition to the structure, the meaning of data should also be documented, which is covered in Section 3.3. Section 3.4 contains recommendations on the various aspects of documenting data changes.

3.1. Publish your documentation

The topics covered in this section include describing data structures, i.e. the internal representation of files, and tracking changes of data. Developing a DMP before publishing data is crucial for achieving a coherent data structure. The plan should cover aspects such as expected/targeted data models, whether raw data will be used and how the data will be processed. Section 1.1 contains more information about DMPs.

Regardless of the format or file type used for documenting data, it is vital that this documentation be published alongside the data, ideally in a separate distribution. This distribution should then be linked to the data itself via the *dct:conformsTo* property of data sets / distributions specified by the DCAT-AP ([21]) standard. This applies regardless of the file format used.

Example

This screenshot shows a distribution using the *dct:conformsTo* property to link to another distribution containing a schema specifying the data's structure.

```
<rdf:Description rdf:about="http://data.europa.eu/distribution/truck_data">
  <rdf:type rdf:resource="http://www.w3.org/ns/dcat#Distribution"/>
  <dcterms:conformsTo rdf:resource="http://data.europa.eu/distribution/truck_data_schema"/>
  <dcterms:title xml:lang="en">Truck parking static data</dcterms:title>
  <dcterms:format rdf:resource="http://publications.europa.eu/resource/authority/file-type/XML"/>
</rdf:Description>
```

3.2. Use schemas to specify data structure

Despite format standards specifying the internal structure with regard to syntax and permitted keywords and identifiers, the data publisher can choose the way data is written to a file (i.e. serialised). For further processing, however, this serialisation must be known to the user by means of data schemas. Instead of expecting the user to download and analyse the data, the serialisation schema can also be specified separately, often in a dedicated format. The following sections provide descriptions of these schema languages and an overview of schema specifications for the most commonly used formats, namely JSON, XML, CSV and RDF.

3.2.1. How to specify JSON data structures

The schema language used for JSON files is called JSON Schema ([22]). The schemas are JSON files themselves, but contain information describing a data structure that can be resembled as JSON. Data providers should publish a JSON schema that specifies the JSON structure along with their data.

([21]) https://joinup.ec.europa.eu/solution/dcat-application-profile-data-portals-europe/releases
([22]) https://json-schema.org/

Example

An example of an OpenAPI specification for an API serving data about fruit can be seen in the screenshot below.

```
{
 "$schema": "http://json-schema.org/draft-07/schema#",
 "description": "Employee data",
 "type": "object",
 "properties": {
  "name": {
   "type": "string"
  },
  "age": {
   "type": "number",
            "minimum": 0,
            "maximum": 100
  }
 }
}
```

Helpful links and tools

Title	Description	Link
JSON Schema	Vocabulary that allows users to annotate and validate JSON documents (open source).	https://json-schema.org/
JSON schema generator	Online tool that generates a schema from existing JSON data (open source).	https://jsonschema.net

3.2.2. How to specify XML data structures

There are multiple schema languages for specifying the structure of XML files, for example RELAX NG ([23]) and Schematron ([24]). XSD (XML Schema Definition Language) is recommended by the W3C and thus also endorsed in this document. An XSD file itself consists of XML. It is made up of two parts: structures ([25]) and data types ([26]). As the names suggest, the former defines the structural part of XSD whereas the latter defines data types that can be used in XSD. Overall, XSD specifies exactly which elements/attributes are allowed and what data type the content must have. It is also possible to specify patterns to check for the correctness of data formats, such as postal

([23]) https://relaxng.org/
([24]) http://schematron.com/
([25]) https://www.w3.org/TR/xmlschema11-1/
([26]) https://www.w3.org/TR/xmlschema11-2/

codes, during validation. Data providers should publish XSD schemas that specify the XML structure alongside their data.

Example

The screenshot shows an XSD schema which specifies the structure of sample data from the fruit domain. For example, it states that the 'drupe' value can either be true or false, not unknown.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
 <xs:element name="fruit">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="type"/>
    <xs:element ref="description"/>
    <xs:element ref="drupe"/>
   </xs:sequence>
   <xs:attribute name="id" use="required" type="xs:integer"/>
  </xs:complexType>
 </xs:element>
 <xs:element name=”type” type="xs:NCName"/>
 <xs:element name=”description” type="xs:string"/>
 <xs:element name=”drupe” type="xs:boolean"/>
</xs:schema>
```

Helpful links and tools

Title	Description	Link
Liquid Studio	XML schema editor which allows generation of XSD files from existing XML (commercial).	https://www.liquid-technologies.com/xml-schema-editor
XMLFox	XML editor that features XSD validation (commercial / open source).	https://www.xmlfox.com/

3.2.3. How to specify CSV data structures

Frictionless Data ([27]) have developed a CSV table schema expressible in JSON. This means that the structure to which a CSV file must cohere is described in a JSON file. At the time of writing, dedicated tooling support for creating Frictionless Data schemas is only available as libraries for various programming languages. However, since Frictionless Data is specified using JSON, any text editor with JSON support can be used for this task.

([27]) https://frictionlessdata.io/

The UK National Archives ([28]) have also published a CSV schema language ([29]) that can be used to describe the content of CSV files. It can be used to specify, among other things, the number of columns, whether values are mandatory or optional and what data range applies.

Data providers should publish schemas in either the Frictionless Data or National Archives formats which specify the CSV table alongside the data.

Example

This screenshot shows the Frictionless Data schema for fictional employee data. Note the restrictions: department names can only consist of capital letters and the numbers 1 to 4, and employees are either retired or not.

```
{
 "fields": [
  {
   "name": "name",
   "type": "string",
   "description": "Employee's name"
  },
  {
   "name": "department",
   "type": "string",
   "description": "Department ID"
          "constraints": {
    "pattern": "[A-Z]{1,4}"
   }
  },
  {
   "name": "retired",
   "type": "boolean",
   "description": "Employee status"
  }
 ]
}
```

([28]) https://www.nationalarchives.gov.uk/

([29]) http://digital-preservation.github.io/csv-schema/csv-schema-1.1.html

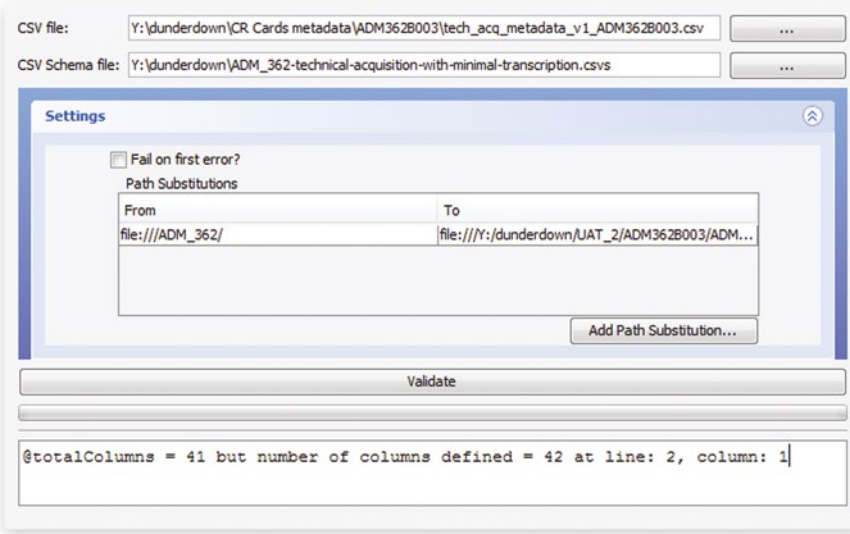
Example

This screenshot shows the National Archives CSV schema for fictional employee data. It contains the same restrictions as in the previous example.

```
version 1.1
@separator ";"
@totalColumns 3
name: unique notEmpty
department: regex("[A-Z]{1,4}")
retired: is("yes") or is("no")
```

Helpful links and tools

Title	Description	Link
CSV Validator	Cross-platform desktop application, command-line utility and programming library suitable for validating CSV files against the National Archives CSV schema.	https://digital-preservation.github.io/csv-validator/



3.2.4. How to specify RDF data structures

The prime way of defining the structure of RDF graphs is using ontologies. The structure of RDF can also be specified using schemas. SHACL ([30]) (Shapes Constraint Language) is a powerful concept that allows validation against these schemas. It specifies a syntax that can be used to define conditions incoming RDF must cohere with. Data

([30]) https://www.w3.org/TR/shacl/

providers should publish SHACL shape files that specify the RDF structure in addition to the actual data.

Example

This screenshot shows the SHACL shape file that specifies personal data. Note the constraint that the date of birth must be earlier than the date of death.

```
@prefix schema: <http://schema.org/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

schema:PersonShape
      a sh:NodeShape ;
      sh:targetClass schema:Person ;
      sh:property [
            sh:path schema:givenName ;
            sh:datatype xsd:string ;
            sh:name "given name" ;
      ] ;
      sh:property [
            sh:path schema:birthDate ;
            sh:lessThan schema:deathDate ;
            sh:maxCount 1 ;
      ] .
```

If the sample data is validated against this SHACL file, the following report is generated. As expected, the mismatch between birth date and death date is detected as a violation.

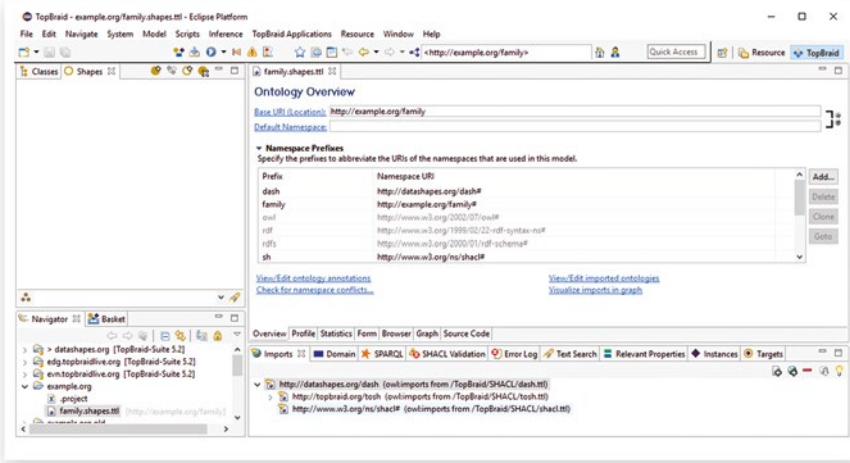
```
[
      a sh:ValidationResult ;
      sh:resultSeverity sh:Violation ;
      sh:sourceConstraintComponent sh:LessThanConstraintComponent ;
      sh:sourceShape _:n703 ;
      sh:focusNode <http://example.org/ns#Bob> ;
      sh:resultPath schema:birthDate ;
      sh:value “1971-07-07” ;
      sh:resultMessage “Value is not < value of schema:deathDate” ;
] .
```

The examples in this section are adapted from those provided by SHACL Playground ([31]).

([31]) https://shacl.org/playground/

Helpful links and tools

Title	Description	Link
TopBraid Composer	Standalone SHACL validator (commercial).	https://www.topquadrant.com/products/topbraid-composer/
SHACL Playground	Web-based SHACL validation tool (open source).	https://shacl.org/playground/



3.2.5. How to specify APIs

APIs are not files themselves, but serve data on the web, accessible by URL. For users to be able to easily use an API, it must be thoroughly documented. Data providers should document not only the structure of served data, but also how this data can be accessed on the web. Depending on the API's protocol, different documentation methods may be applicable. HTTP APIs should be documented according to the OpenAPI ([32]) standard. This allows, among other operations, specification of URLs, HTTP status codes and structure of payloads (i.e. what the served data looks like). OpenAPI specifications can be written in either JSON or YAML. Recommendations on good API design are given in Section 1.2.

The following aspects of an API should be specified:

- URLs and endpoints;
- the protocol(s) of the endpoints (e.g. HTTP, FTP);
- access methods (e.g. HTTP methods, status codes);
- ways to alter results (e.g. query parameters, HTTP headers).

Additionally, the semantic meaning of the served data should be explained.

([32]) https://www.openapis.org/

Example

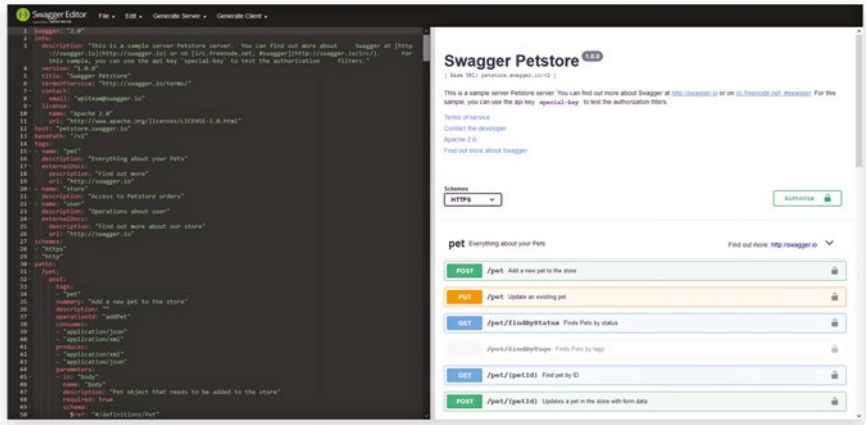
This screenshot shows a truncated snippet of an OpenAPI specification that defines the EU ODP's API for retrieving a data set ([33]). Aside from ensuring a sound structure with all mandatory fields set, the specification should be complete and exhaustive with regard to the aspects mentioned above. Meaningful descriptions and summaries help grasp the semantic meaning of the data served.

```
paths:
  '/data set/{data setId}.rdf':
    get:
      summary: Retrieve data set in RDF/XML
      description: >
        Return the full content of the data set in RDF
      operationId: getData set
      parameters:
        - name: data setId
          in: path
          description: data set identifier
          required: true
          style: simple
          explode: false
          schema:
            type: string
      responses:
        '200':
          description: OK
          content:
            application/rdf+xml:
              schema:
                type: string
        '404':
          description: not found
          content:
            text/html:
              schema:
                type: string
```

([33]) https://app.swaggerhub.com/apis/EU-Open-Data-Portal/eu-open_data_portal/0.8.0

Helpful links and tools

Title	Description	Link
Swagger	Tooling that aids in editing and validating OpenAPI specifications (commercial / open source).	https://swagger.io/



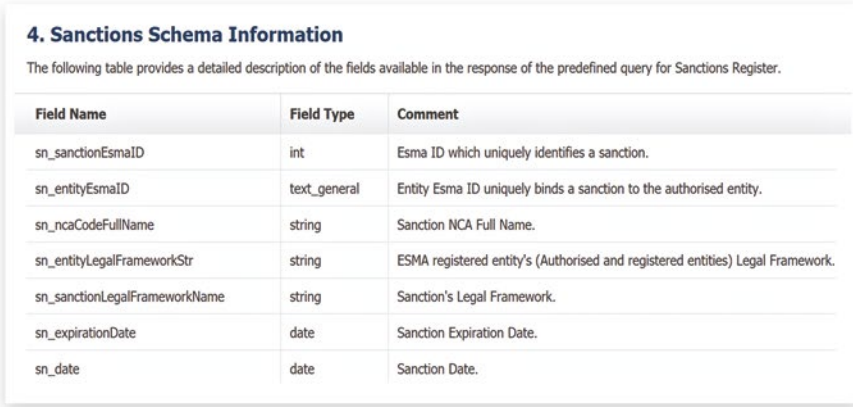
OpenAPI specification	Standard that defines OpenAPI (commercial / open source).	https://swagger.io/specification/

3.3. Document the semantics of data

Depending on its complexity, publishing a schema is not always sufficient. While a schema describes the syntax and structure, it does not explain the semantics of data. A description of the individual properties of a data structure helps users interpret and reuse data correctly and in the way intended by the data provider.

Example

The first screenshot shows a data set which links both the schema and the semantic description of its data. Note that all three link to a dedicated distribution, which contains links for accessing the files. The second screenshot shows a snippet of the HTML document of the semantic documentation.



4. Sanctions Schema Information

The following table provides a detailed description of the fields available in the response of the predefined query for Sanctions Register.

Field Name	Field Type	Comment
sn_sanctionEsmaID	int	Esma ID which uniquely identifies a sanction.
sn_entityEsmaID	text_general	Entity Esma ID uniquely binds a sanction to the authorised entity.
sn_ncaCodeFullName	string	Sanction NCA Full Name.
sn_entityLegalFrameworkStr	string	ESMA registered entity's (Authorised and registered entities) Legal Framework.
sn_sanctionLegalFrameworkName	string	Sanction's Legal Framework.
sn_expirationDate	date	Sanction Expiration Date.
sn_date	date	Sanction Date.

Helpful links and tools

Title	Description	Link
Sphinx	Tool for creating documentation. Supports, among others, HTML, PDF and plain text formats (open source).	https://www.sphinx-doc.org/en/master/index.html
Read the Docs	Open-source hosting service for documentation, for example those generated using Sphinx (open source).	https://readthedocs.org/

3.4. Document data changes

Data is likely to change over time. For example, schedules for public transport may be updated during roadworks, and if a new politician is elected their name may be added to the list of elected representatives. It is important to document all such changes. More precisely, users must know that data has changed, what has changed and where to find other versions of the data. This section contains recommendations covering all three aspects.

3.4.1. Adopt a data set release policy

When you have to update your data, it is important to consider the following questions.

- What constitutes a change in the data set?
- What is the impact of the new release: is it a major or minor change in the data?
- What is the importance of the change from the reuser's erspective?

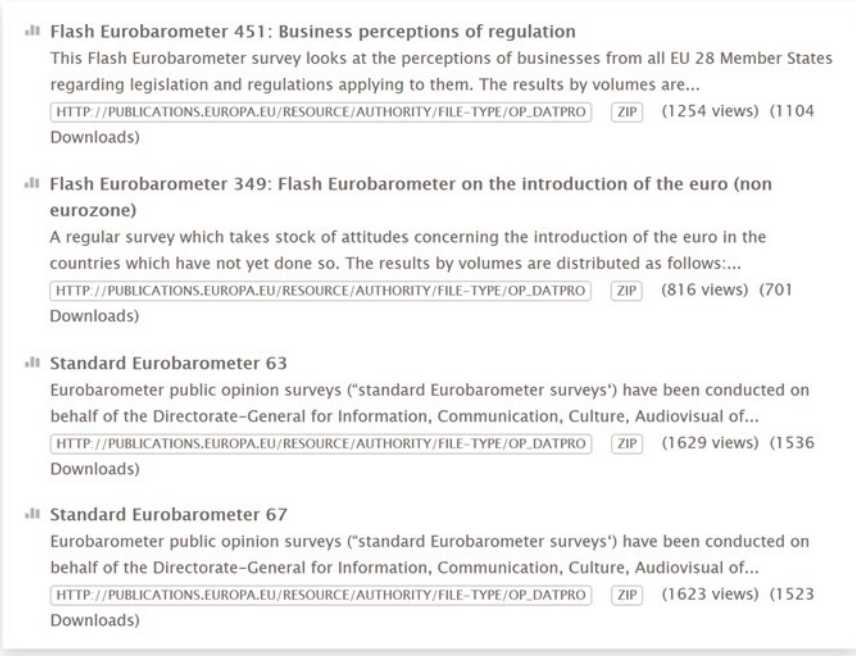
The data set release policy can be defined in the DMP. Steps include, among others, defining the file naming convention, release number and update frequency. Section 1.1 contains recommendations for creating DMPs.

3.4.2. Differentiate between a major and a minor release of a data set

If a new instance of a data set is different from its predecessor it can be considered as a new major release, meaning it is recommended that a new entry for this data set be created in the data catalogue. If the change in the data is minor and does not impact the reuser, it is recommended that the data set description be updated in the data catalogue.

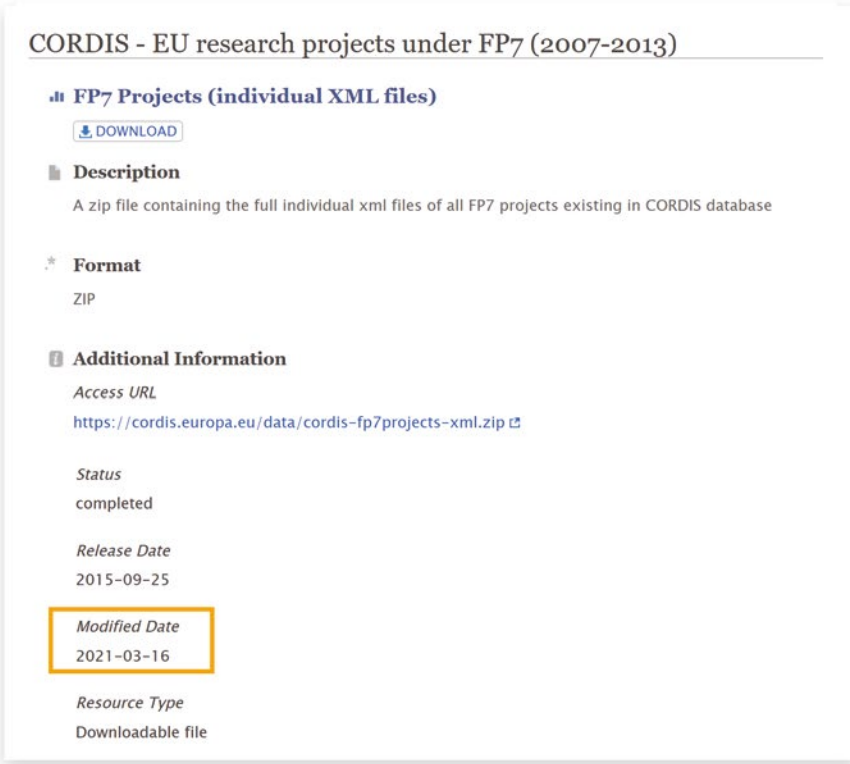
Example

Eurobarometer studies monitor public opinion in the European Union Member States and candidate countries. The survey results are regularly published in official reports. Each data set is part of a collection (Eurobarometer) and results in a succession of generated data sets. Each data set in the collection is identified and versioned.



Example (minor change)

In the example below, the date of modification has been updated after the data have been updated.



CORDIS - EU research projects under FP7 (2007-2013)

FP7 Projects (individual XML files)

DOWNLOAD

Description

A zip file containing the full individual xml files of all FP7 projects existing in CORDIS database

Format

ZIP

Additional Information

Access URL

https://cordis.europa.eu/data/cordis-fp7projects-xml.zip

Status

completed

Release Date

2015-09-25

Modified Date

2021-03-16

Resource Type

Downloadable file

Example (major change)

In the screenshot below, a new version of a data set has been added under the resources (version 1.1) as well as the documentation (XSD schema 1.1).

3.4.3. Indicate a data set's version (release) number

There are a multitude of conventions concerning when and how to increment version numbers. In the spirit of standardisation, it is advisable to adhere to commonly used specifications when choosing version numbers.

One such standard is called semantic versioning ([34]). It states that version numbers must consist of three digits, separated by dots, for example '1.2.3'. The first digit declares the major version, the second digit the minor version and the last digit the patch version.

Other methods of versioning exist, for example using digital object identifiers (DOIs). A new DOI is assigned for each version of a document. The DOIs are generated and maintained by central authorities in order to guarantee the uniqueness of the numbers.

The *owl:versionInfo* property should be used to indicate the version of a data set. Additionally, the *dct:modified* property should be used to state the date of the latest modification of the data set or distribution.

([34]) https://semver.org/

Example

The screenshot shows the same data set with the *owl:versionInfo* and *dct:modified* property set. The former is specified using semantic versioning.

```
<rdf:Description rdf:about="http://data.europa.eu/dataset/honorific">
  <dcterms:title xml:lang="en">Honorific Named Authority List</dcterms:title>
  <owl:versionInfo> 1.2.1 </owl:versionInfo>
  <dcterms:modified rdf:datatype="http:/ /www.w3.org/2001/XMLSchema#datetime">2020-01-08T08:39:53</dcterms:modified>
  <dcterms:issued rdf:datatype-="http://www.w3.org/200 l/XMLSchema#datetime">2016-11-18</dcterms:issued>
</rdf:Description >
```

3.4.4. Describe what has changed

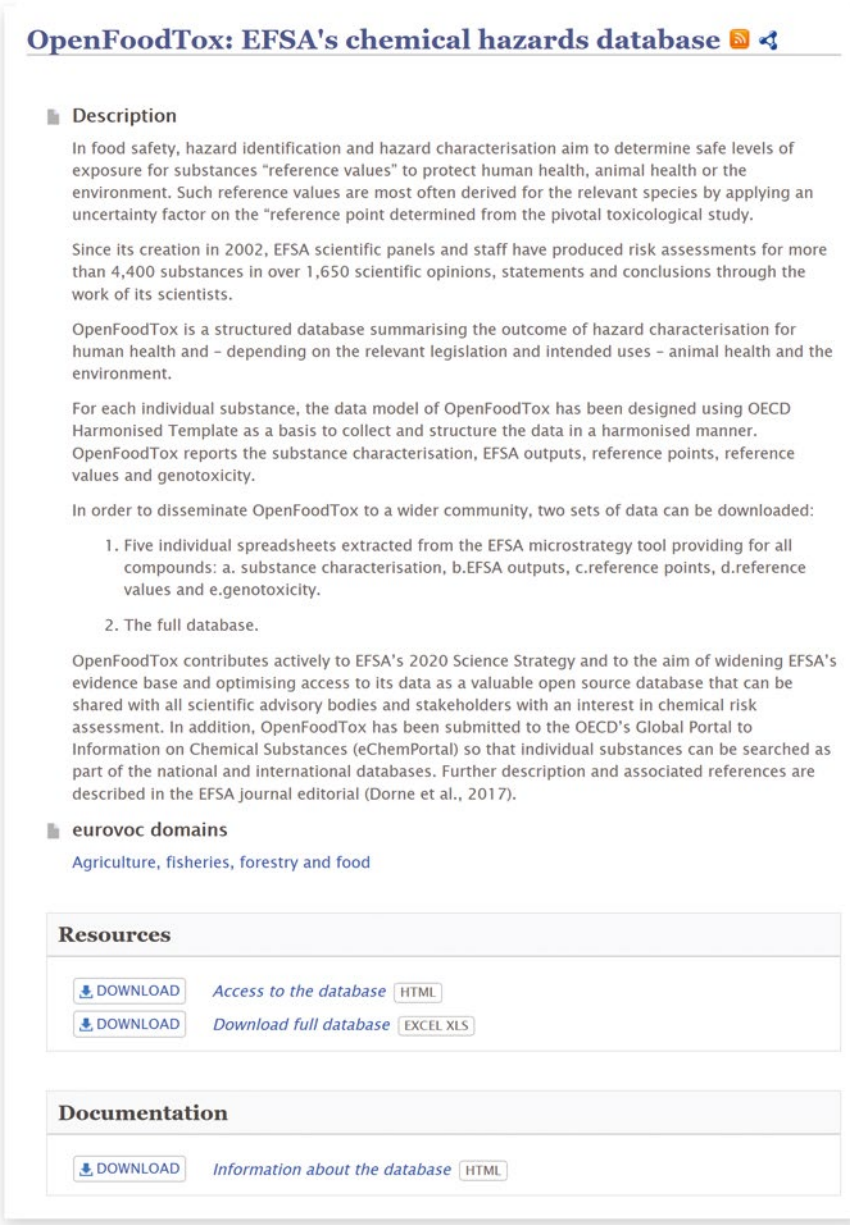
As stated earlier, it should not only be indicated that data has changed, but also what has changed. This is ideally documented in a separate document, which should be linked via the *foaf:page* property of a data set or distribution.

Example

This screenshot shows a data set with the *foaf:page* property set.

```
<rdf:Description rdf:about="http://data.europa.eu/dataset/honorific">
  <dcterms:title xml:lang="en">Honorific Named Authority List</dcterms:title>
  <owl:versionInfo> 1.2.1 </owl:versionInfo>
  <dcterms:modified rdf:datatype="http:/ /www.w3.org/2001/XMLSchema#datetime">2020-01-08T08:39:53</dcterms:modified>
  <foaf:page rdf:resource="http://data.europa.eu/document/honrific-documentation"/>
  <dcterms:issued rdf:datatype-="http://www.w3.org/200l/XMLSchema#datetime">2016-11-18</dcterms:issued>
</rdf:Description >
```

This screenshots shows the properties *adms:identifier*, *dct:modified* and *adms:versionNotes*.



OpenFoodTox: EFSA's chemical hazards database

Description

In food safety, hazard identification and hazard characterisation aim to determine safe levels of exposure for substances "reference values" to protect human health, animal health or the environment. Such reference values are most often derived for the relevant species by applying an uncertainty factor on the "reference point determined from the pivotal toxicological study.

Since its creation in 2002, EFSA scientific panels and staff have produced risk assessments for more than 4,400 substances in over 1,650 scientific opinions, statements and conclusions through the work of its scientists.

OpenFoodTox is a structured database summarising the outcome of hazard characterisation for human health and – depending on the relevant legislation and intended uses – animal health and the environment.

For each individual substance, the data model of OpenFoodTox has been designed using OECD Harmonised Template as a basis to collect and structure the data in a harmonised manner. OpenFoodTox reports the substance characterisation, EFSA outputs, reference points, reference values and genotoxicity.

In order to disseminate OpenFoodTox to a wider community, two sets of data can be downloaded:

1. Five individual spreadsheets extracted from the EFSA microstrategy tool providing for all compounds: a. substance characterisation, b.EFSA outputs, c.reference points, d.reference values and e.genotoxicity.
2. The full database.

OpenFoodTox contributes actively to EFSA's 2020 Science Strategy and to the aim of widening EFSA's evidence base and optimising access to its data as a valuable open source database that can be shared with all scientific advisory bodies and stakeholders with an interest in chemical risk assessment. In addition, OpenFoodTox has been submitted to the OECD's Global Portal to Information on Chemical Substances (eChemPortal) so that individual substances can be searched as part of the national and international databases. Further description and associated references are described in the EFSA journal editorial (Dorne et al., 2017).

eurovoc domains

Agriculture, fisheries, forestry and food

Resources

DOWNLOAD *Access to the database* HTML

DOWNLOAD *Download full database* EXCEL XLS

Documentation

DOWNLOAD *Information about the database* HTML

URI

http://data.europa.eu/88u/dataset/openfoodtox-efsa-s-chemical-hazards-database

Identifier

10.5281/zenodo.780543

DOI

10.5281/zenodo.780543

Landing Page

https://doi.org/10.5281/zenodo.780543

Release Date

2017-09-27

Modified Date

2020-04-15

Version

Version 3 10.5281/zenodo.3693783

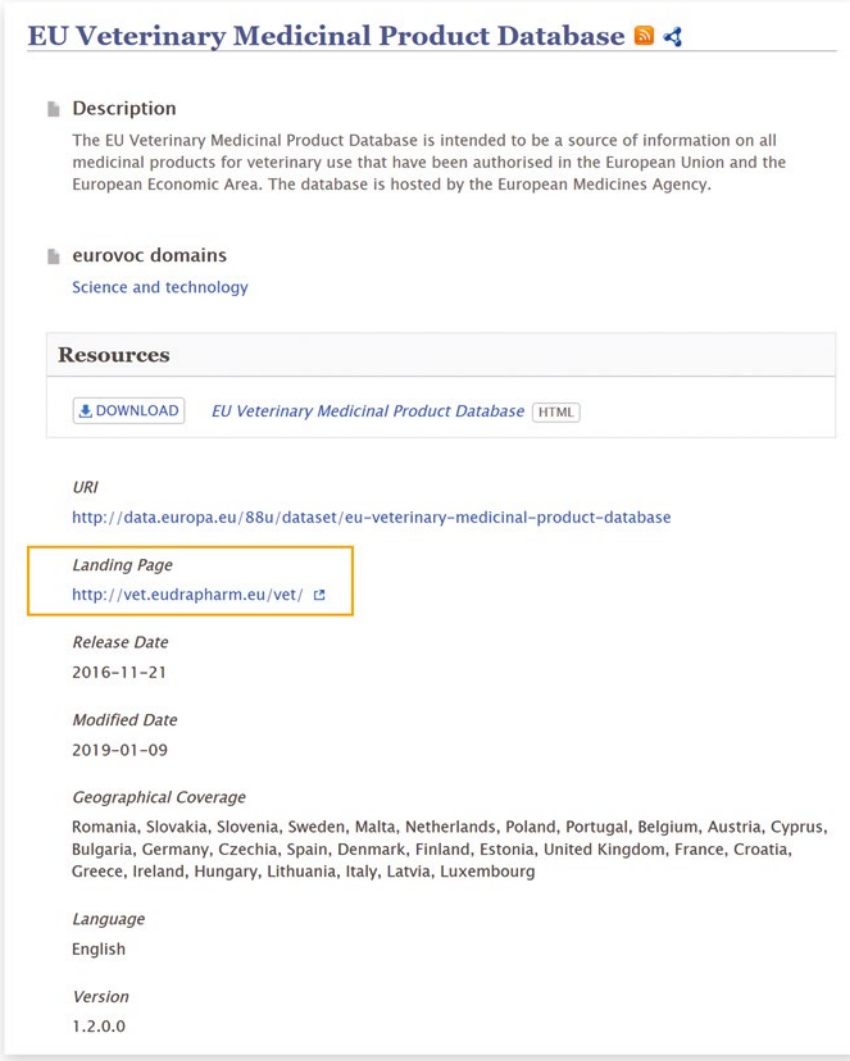
Version notes

This version replaces version 2 to include EFSA Opinions, Statements and Conclusions up to November 2019

If there are multiple versions of a data set, the landing page should point to the latest version of the data.

Example

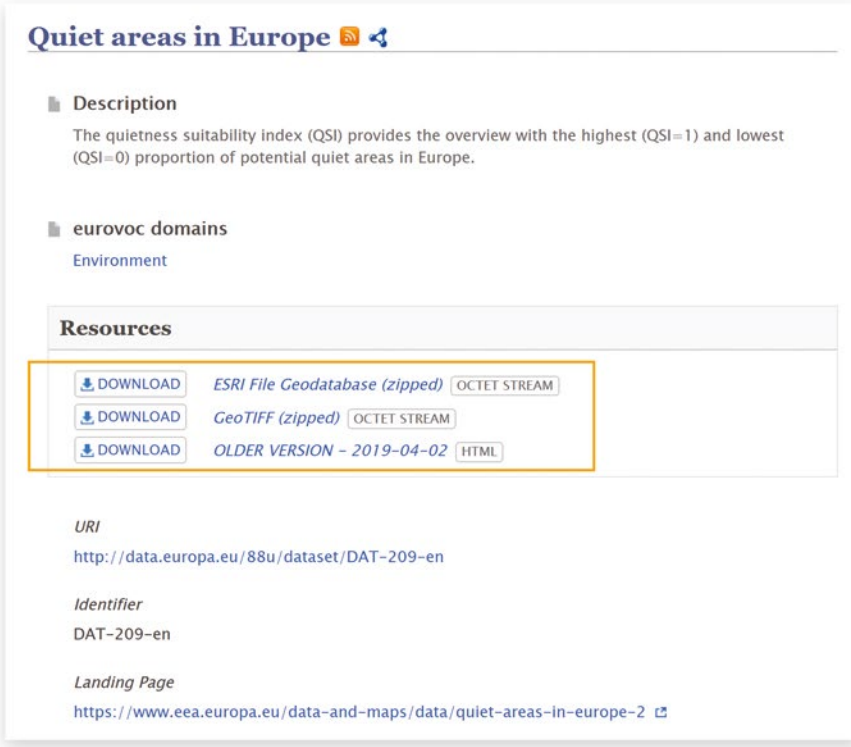
The screenshot below shows a data set with the *dcat:landingPage* property set.



Even if data sets are expressed in different file formats, they are still manifestations of the same work. A new data format of a data set should be released by adding a new distribution to the data set and changing the minor version number. For any changes to the data itself, a new major version of the data set should be created. In any case, it is important to update the *dct:modified* and *owl:versionInfo* properties.

Example

The screenshot below shows a data set with data being published in multiple formats.



The changes made to a data set can be documented using a changelog – a text file that contains a list of the changes made between versions of a file (or multiple files) in a structured and chronologically ordered way. Keywords like 'added', 'changed' and 'removed' help distinguish the types of changes made. One standard of structuring a

changelog is called Keep a Changelog ([35]), which uses Markdown ([36]) for formatting. A command line tool is available for managing changelogs in this way ([37]).

Example

The screenshots below show an example of a changelog formatted using Markdown. The raw text file is depicted on the left. The Markdown has been rendered using the Dillinger online tool ([38]), as can be seen on the right. The example features the semantic versioning mentioned above.

```
# ChangeLog

## 1.1.1 (2020-03-08)

**Fixed:**
* Hotfix for OpenApi Yaml file load
* Close models and data sets after use

**Removed:**
* Deprecated API endpoints
```

ChangeLog

1.1.1 (2020-03-08)

Fixed:

- Hotfix for OpenApi Yaml file load
- Close models and datasets after use

Removed:

- Deprecated API endpoints

1.1.0 (2020-03-06)

Added:

- Configuration of verticle instances and worker pool size

Helpful links and tools

Title	Description	Link
Dillinger	Online Markdown editor with preview (open source).	https://dillinger.io/
DoltHub	Version control for databases (commercial / open source).	https://www.dolthub.com/

([35]) https://keepachangelog.com/en/1.0.0/
([36]) https://daringfireball.net/projects/markdown/
([37]) https://github.com/churchtools/changelogger
([38]) Rendered using https://dillinger.io/

3.4.5. Release one data set per table

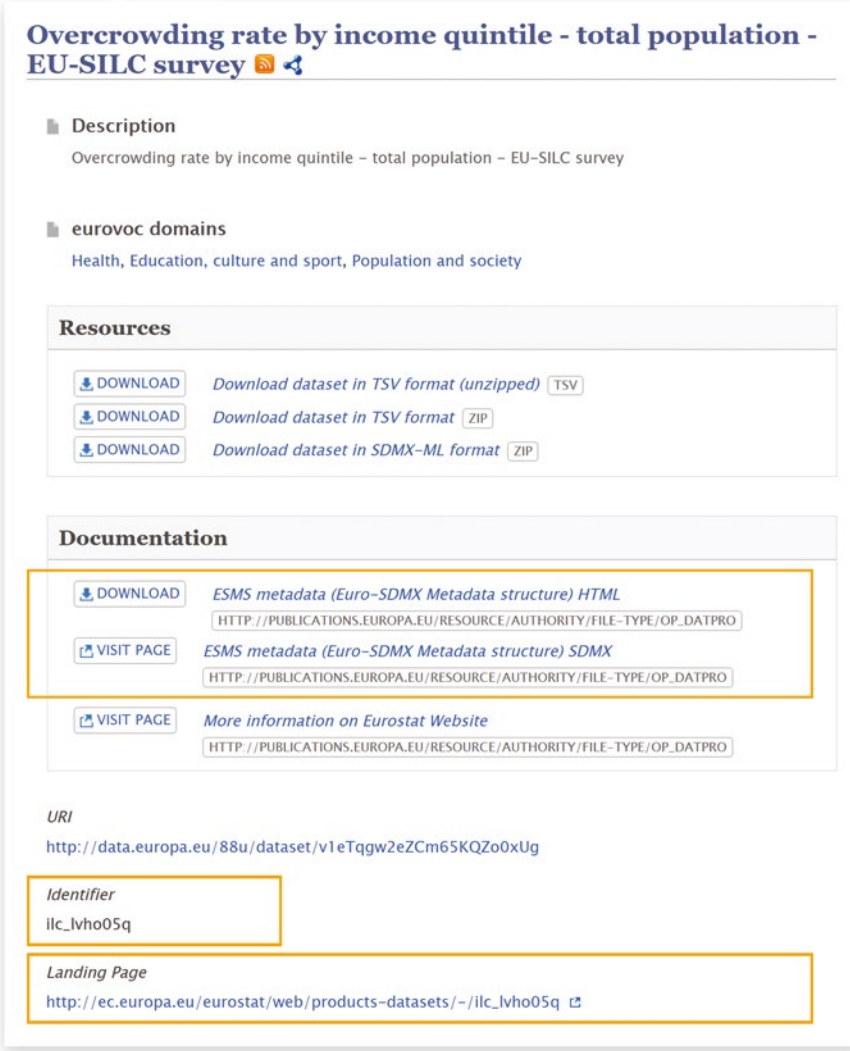
For tabular data, each sheet should be published as a new data set. This maintains a clear distinction between data and makes the data easier to process. Some formats, like CSV, do not even feature the concept of multiple tables per file.

Example

A statistical data publisher's policy is to publish one (major) data set per table. Data sets are updated twice a day, at 11.00 and 23.00. As statistics are updated on continuous basis, the publisher provides only one access URL referring to the last update of the data set. The same data set is expressed in different file formats (manifestations) without any difference between their actual content. Each data set:

- is identified by a unique identifier;
- is supplemented by reference metadata describing the statistical concepts and methodologies used to collect and generate the data and providing information about data quality;
- has machine-readable (SDMX) and human-readable (HTML) documentation;
- provides a link to the landing page of the product data set in the data provider website.

The screenshot below shows a data set with a unique identifier, machine- and human-readable documentation and a landing page.

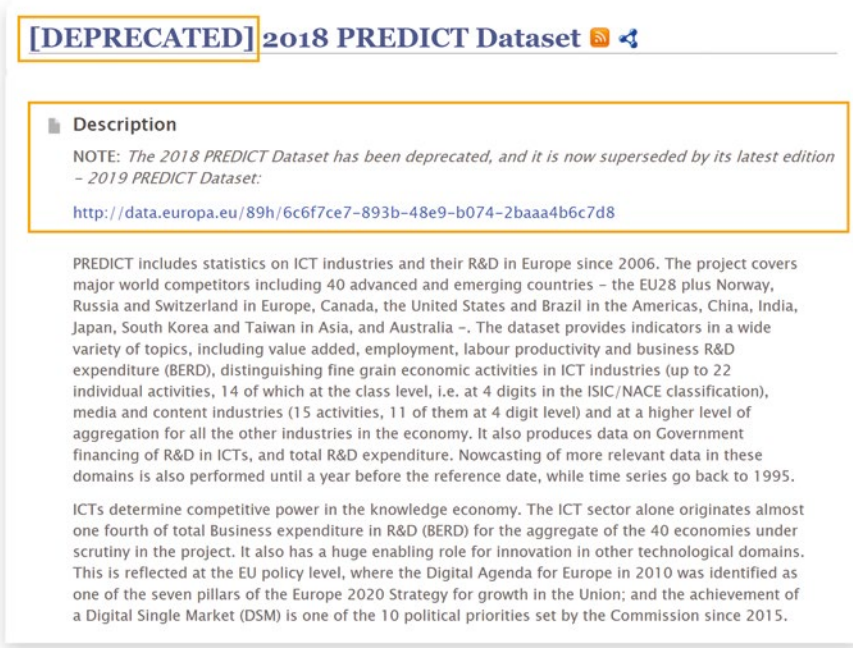


3.4.6. Deprecate old versions

If new, updated versions of data are published, the older versions should be marked as deprecated and the new version should be linked to from the deprecated version. This allows users to quickly identify old data and subsequently find the newest data.

Example

Predict includes statistics on ICT industries and their research and development in Europe since 2006. It is published on a yearly basis, with one data set per year. As soon as the latest version is published the previous version is deprecated, and a link referring to the updated data set is added in the description, as shown in the screenshot below.



[DEPRECATED] 2018 PREDICT Dataset

Description

NOTE: *The 2018 PREDICT Dataset has been deprecated, and it is now superseded by its latest edition – 2019 PREDICT Dataset:*

http://data.europa.eu/89h/6c6f7ce7-893b-48e9-b074-2baaa4b6c7d8

PREDICT includes statistics on ICT industries and their R&D in Europe since 2006. The project covers major world competitors including 40 advanced and emerging countries – the EU28 plus Norway, Russia and Switzerland in Europe, Canada, the United States and Brazil in the Americas, China, India, Japan, South Korea and Taiwan in Asia, and Australia –. The dataset provides indicators in a wide variety of topics, including value added, employment, labour productivity and business R&D expenditure (BERD), distinguishing fine grain economic activities in ICT industries (up to 22 individual activities, 14 of which at the class level, i.e. at 4 digits in the ISIC/NACE classification), media and content industries (15 activities, 11 of them at 4 digit level) and at a higher level of aggregation for all the other industries in the economy. It also produces data on Government financing of R&D in ICTs, and total R&D expenditure. Nowcasting of more relevant data in these domains is also performed until a year before the reference date, while time series go back to 1995.

ICTs determine competitive power in the knowledge economy. The ICT sector alone originates almost one fourth of total Business expenditure in R&D (BERD) for the aggregate of the 40 economies under scrutiny in the project. It also has a huge enabling role for innovation in other technological domains. This is reflected at the EU policy level, where the Digital Agenda for Europe in 2010 was identified as one of the seven pillars of the Europe 2020 Strategy for growth in the Union; and the achievement of a Digital Single Market (DSM) is one of the 10 political priorities set by the Commission since 2015.

3.4.7. Link versions of a data set

New versions or adaptions of a data set should use the *dct:isVersionOf* property to link to other versions of the data set. However, the property *dct:source* should be used to link to the original data set. Since this relationship is bidirectional, the original data set can use the *dct:hasVersio*n property to link to the new data set.

Example original data set

This screenshot shows a data set referencing a different version using the property *dct:hasVersion*. Note the use of the *adms:versionNotes* property giving a description of the current version.

```
<dcat:Dataset rdf:about="http://data.europa.eu/dataset/government-data">
  <dct:title xml:lang="en">Government Data</dct:title>
  <dct:hasVersion rdf:resource="http://data.europa.eu/dataset/general-government-data"/>
  <adms:versionNotes xml:lang="en">Initial release</adms:versionNotes>
</dcat:Dataset>
```

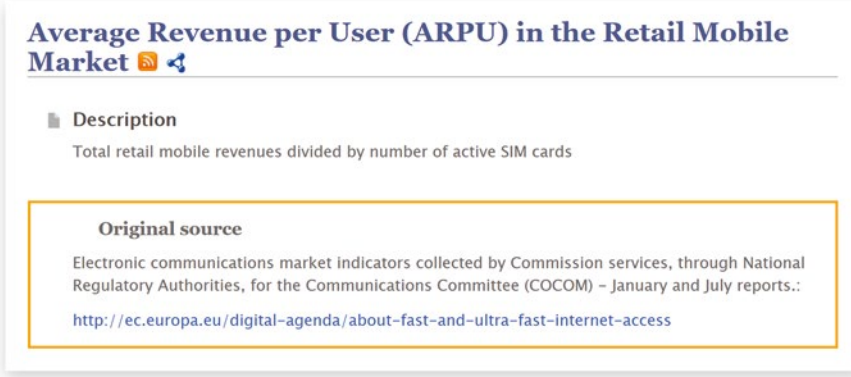
Example derived data set

This screenshot shows a data set referencing the original version it has been derived from using the property *dct:isVersionOf*.

```
<dcat:Dataset rdf:about="http://data.europa.eu/dataset/general-government-data">
  <dct:isVersionOf rdf:resource= "http://data.europa.eu/dataset/general-data"/>
  <dct:title xml:lang="en">General Government Data</dct:title>
</dcat:Dataset>
```

Example

This screenshot shows a data set which links to the original source and parent data set. This is done in both the description and the metadata properties.



Parent dataset

This dataset is part of of another dataset:

http://digital-agenda-data.eu/datasets/digital_agenda_scoreboard_key_indicators

eurovoc domains

Science and technology, Economy and finance

URI

http://data.europa.eu/88u/dataset/NaUJDKaulkIWX0YFtDz86Q

Identifier

mob_arpu

Alternative Title

Average Revenue per User (ARPU) in the Retail Mobile Market

Landing Page

http://semantic.digital-agenda-data.eu/codelist/indicator/mob_arpu

Type of Dataset

Statistical

Release Date

2014-05-22

Modified Date

2015-07-27

Temporal Coverage From

2010-01-01

Geographical Coverage

Slovakia, Slovenia, Sweden, Netherlands, Poland, Portugal, Romania, Belgium, Austria, Cyprus, Bulgaria, Germany, Czechia, Spain, Denmark, Finland, Estonia, United Kingdom, France, Hungary, Greece, Italy, Ireland, Luxembourg, Lithuania, Malta, Latvia

Language

English

Catalogue

European Union Open Data Portal

source

http://ec.europa.eu/digital-agenda/about-fast-and-ultra-fast-internet-access

is part of

http://data.europa.eu/88u/dataset/digital-agenda-scoreboard-key-indicators

Helpful links and tools

Title	Description	Link
Data Versioning WG	Research Data Alliance working group (open source).	https://www.rd-alliance.org/groups/data-versioning-wg
Research Data Alliance best practices	Principles and best practices in data versioning for all data sets, big and small (open source).	https://www.rd-alliance.org/group/data-versioning-wg/outcomes/principles-and-best-practices-data-versioning-all-data-sets-big

4. Recommendations for improving the openness level

Introduction

The objective of this section is to help data publishers achieve the highest possible openness level for their data, with a special emphasis on the publishing phase of the data preparation process (see Figure 9).

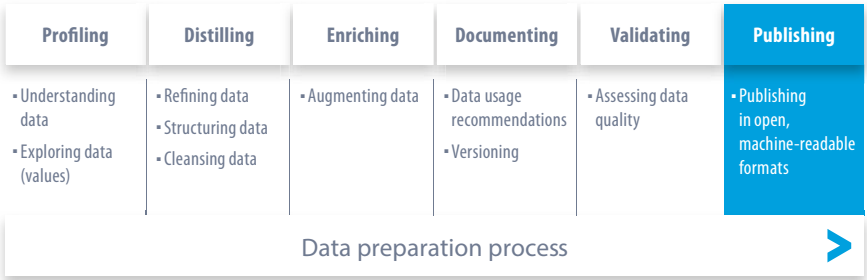


Figure 9. Data preparation process – Publishing

In Section 4.1 the five-star model for measuring openness of data is introduced. The following sections contain recommendations on how to achieve each level of the model.

4.1. Five-star model

Openness is of particular importance when publishing data. It directly affects users' ability to reuse and process data, and thus the value of data. In this section, openness is discussed with regard to file formats.

Tim Berners-Lee's five-star model ([39]), which was developed in 2001, is an attempt to provide a scale for measuring the openness of data. Data can achieve a maximum of five stars, indicating the highest level of openness. The ranks are cascading, meaning that in order to comply with a certain rank, the criteria of the preceding ranks must also be met. Regardless of actual data quality, the first star is awarded for using an open licence. If data usage is restricted by a proprietary licence its quality is rendered meaningless. In order to achieve a second star, the chosen file format must be (semi-)

([39]) https://5stardata.info

structured. A table stored as CSV is much easier to process than an image in which a table is depicted. Next, usage of non-proprietary formats is required for a three-star rating. Using URIs as identifiers for resources is required for a four-star rating. The decisive characteristic for achieving the full five stars is linking data together to provide context. An illustration of this hierarchy is shown in Figure 10. The following sections contain recommendations for acquiring all five stars.

Figure 10. Cascading steps of the five-star model with exemplary file formats
Source: https://5stardata.info/en/

4.2. Use structured data (one → two stars)

As mentioned above, the first star is awarded for using an open licence. To achieve a two-star rating, data must be structured. Table 5 in Section 4.6 gives an overview of the common formats and indicates whether they are machine readable or not. Based on this, the recommended formats for data publishers are RDF, XML, JSON and CSV. Section 1.2 describes how to achieve well-structured data in these formats. Recommendations are given on how to construct well-formed files, as well as an overview of tooling support.

Example

This screenshot shows a data set which contains both PDF and XLS files. PDF is a format suitable for human reading. However, data publishers should make sure that they also publish their data in a machine-readable format to enable others to easily process the data. To achieve a two-star rating, data must be published in a machine-readable format (or any other structured data format).

Helpful links and tools

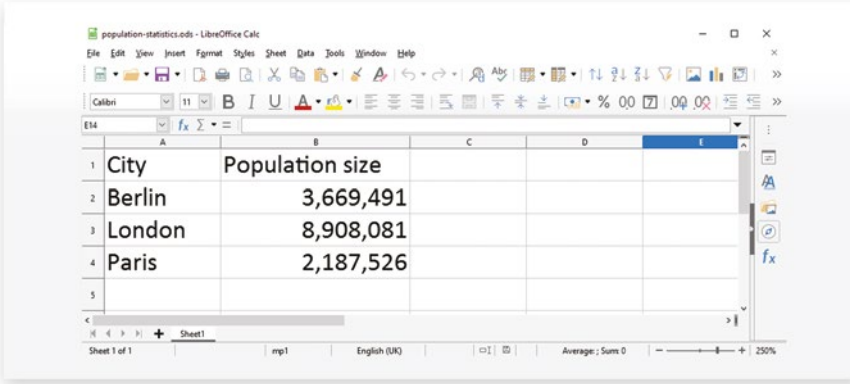
Title	Description	Link
PDF to XLS	Free online tool for extracting tables from PDF into XLS files (open source).	https://pdftoxls.com/
PDFTables	Paid online tool with an API for extracting tables from PDF into XLS, CSV, XML or HTML files (commercial).	https://pdftables.com/
Coenterprise tableau	ETL suite that supports PDF content extraction into CSV files during the data preparation phase (commercial).	https://www.coenterprise.com/solutions/data-analytics/

4.3. Use a non-proprietary format (two → three stars)

Using a machine-readable format is key to achieving a high openness level. However, some formats, like XLS, are proprietary, which means that a certain piece of software – in this case Microsoft Excel – is needed to fully process the file. Often, this kind of software is not freely available. As accessibility for everyone is a core principle of open data, proprietary file formats are not the correct choice. Thus, to receive the third star, a non-proprietary file format such as ODS must be used. Table 5 in Section 4.6 gives an overview of which formats are non-proprietary.

Example

This screenshot shows tabular data in ODS format, opened in the non-proprietary application LibreOffice.



Example

```
1 City;Population size
2 Berlin;3669491
3 London;8908081
4 Paris;2187526
```

This screenshot shows tabular data in CSV, an open text-based format.

Helpful links and tools

Title	Description	Link
LibreOffice	Open-source office suite supporting OpenDocument formats (open source).	https://www.libreoffice.org/
OpenOffice	Open-source office suite supporting OpenDocument formats (open source).	http://www.openoffice.org/
Microsoft Office	Proprietary office suite which supports OpenDocument formats from the 2013 version (commercial).	https://www.office.com/
OnlyOffice	Desktop and web-based collaborative office suite (commercial).	https://www.onlyoffice.com/en/
Recommended formats	List of open formats recommended by the UK Data Service (open source).	https://www.ukdataservice.ac.uk/manage-data/format/recommended-formats

4.4. Use URIs to denote things (three → four stars)

Three-star data is easily processable, but isolated and hard to reference by others. In order to achieve a four-star rating, URIs must be used to denote things. Of course, the file itself should also be resolvable by a URI. The recommendation in this section focuses on using URIs in the data itself.

'Things' refers to resources or concepts within the data. For example, a city would be a concept that could be denoted by the URI <http://cities.org/berlin>, instead of the plain identifier 'Berlin'. In contrast, numbers, such as a population size, do not need to be denoted as URIs. Things not considered a resource are called 'literals', the difference being that literals only acquire meaning when used in conjunction with resources. Numbers, Boolean values (true and false) and dates have little meaning on their own and are thus literals. RDF graphs are made up of triples, consisting of a subject, predicate and object. Subjects and predicates must always be resources, whereas objects can either be resources or literals.

In order to replace identifiers with URIs, a first step can be looking at existing controlled vocabularies and knowledge bases to see if the concepts already have widely adopted URIs. These are covered in the next section. If none exist, the authority publishing the data can publish its own ontology in order to define concepts that have not been specified elsewhere.

Example

The first triple (yellow) consists of only resources, whereas the second triple (green) contains a literal (the population number). They could be read as 'Berlin is in Germany' and 'Berlin has the population size 3 669 491' respectively.

```
<http://cities.org/Berlin> <http://conjunctions.org/isIn> <http://countries.org/Germany>
<http://cities.org/Berlin> <http://population.org/size> 3669491
```

The triples that make up RDF graphs are stored in dedicated databases called triple stores. They can then be queried using SPARQL, a query language similar to SQL.

URIs can not only be used in RDF files though. All formats in which resources and concepts are denoted by an identifier can make use of URIs.

Example

This screenshot shows the city population CSV file from earlier. Here, the city names have been replaced with referenceable URIs.

```
1 City;Population size
2 http://cities.org/Berlin ;3669491
3 http://cities.org/London ;8908081
4 http://cities.org/Paris; 2187526
```

URIs should be unique on the web. This means that if two pieces of data have the same URI, they mean the same thing. Additionally, using URIs allows other data providers to link to the data, which is required for achieving the five-star rating covered in the next section.

Example

This screenshot shows the same data as in the CSV example above, albeit as RDF. Note that all referenceable data is denoted with a URI (yellow boxes). The only exceptions are the population numbers, which are literals (red boxes) and are not referenceable (and do not need to be).

```
<rdf:RDF>
-<rdf:Description rdf:about="http://example.org/berlin">
    <rdf:type rdf:resource="http://example.org#PopulationStatistic"/>
    <dcterms:location rdf:resource="http://publications.europa.eu/resource/authority/place/DEU_BER"/>
    <dbpedia:populationTotal rdf:datatype="http://www.w3.org
    /2001/XMLSchema#nonNegativeInteger">3669491</dbpedia:populationTotal>
  </rdf:Description>
-<rdf:Description rdf:about="http://example.org/london">
    <rdf:type rdf:resource="http://example.org#PopulationStatistic"/>
    <dcterms:location rdf:resource="http://publications.europa.eu/resource/authority/place/GBR_LON"/>
    <dbpedia:populationTotal rdf:datatype="http://www.w3.org
    /2001/XMLSchema#nonNegativeInteger">8908081</dbpedia:populationTotal>
  </rdf:Description>
-<rdf:Description rdf:about="http://example.org/paris">
    <rdf:type rdf:resource="http://example.org#PopulationStatistic"/>
    <dcterms:location rdf:resource="http://publications.europa.eu/resource/authority/place/FRA_PAR"/>
    <dbpedia:populationTotal rdf:datatype="http://www.w3.org
    /2001/XMLSchema#nonNegativeInteger">2187526</dbpedia:populationTotal>
  </rdf:Description>
</rdf:RDF>
```

Helpful links and tools

Title	Description	Link
ConverterToRdf	W3C list of tools that help convert various files to RDF format (open source).	https://www.w3.org/wiki/ConverterToRdf
OpenLink Virtuoso	Open-source triple store (commercial).	https://virtuoso.openlinksw.com/
SPARQL specification	W3C SPARQL 1.1 specification (open source).	https://www.w3.org/TR/sparql11-overview/

4.5. Use linked data (four → five stars)

The main benefit of using URIs to denote things is that it makes information referenceable. Since the web is based mainly on HTTP, URIs are not only unique IDs, but also directly resolvable, thereby pointing to the resource. The next step is to actually link these pieces of information together in order to create linked data. A semantic graph, also known as a knowledge graph, can only be constructed using RDF format. A graph that is constructed this way can be traversed by resolving, i.e. dereferencing, the HTTP URIs. This means data can be inferred and more relations can be discovered. Data is enriched by adding URI references to other sources. Links can be established, for example, to the controlled vocabularies published by the Publications Office or DBpedia ([40]). The topic of enrichment by using controlled vocabularies and open knowledge bases like DBpedia is covered in Part 2. Using RDF and linking data are required to achieve the full five-star rating.

Example

This screenshot shows the same data as the example in the previous section. Here, a property has been added which links the locations to their representations in DBpedia, thereby creating linked data.

```
<rdf:RDF>
-<rdf:Description rdf:about="http://example.org/berlin">
    <rdf:type rdf:resource="http://example.org#PopulationStatistic"/>
    <dcterms:location rdf:resource="http://publications.europa.eu/resource/authority/place/DEU_BER"/>
    <owl:sameAs rdf:resource="http://dbpedia.org/resource/Berlin"/>
    <dbpedia:populationTotal rdf:datatype="http://www.w3.org
    /2001/XMLSchema#nonNegativeInteger"> 3669491 </dbpedia:populationTotal>
  </rdf:Description>
-<rdf:Description rdf:about="http://example.org/london">
    <rdf:type rdf:resource="http://example.org#PopulationStatistic"/>
    <dcterms:location rdf:resource="http://publications.europa.eu/resource/authority/place/GBR_LON"/>
    <owl:sameAs rdf:resource="http://dbpedia.org/resource/London"/>
    <dbpedia:populationTotal rdf:datatype="http://www.w3.org
    /2001/XMLSchema#nonNegativeInteger">8908081 </dbpedia:populationTotal>
  </rdf:Description>
-<rdf:Description rdf:about="http://example.org/paris">
    <rdf:type rdf:resource="http://example.org#PopulationStatistic"/>
    <dcterms:location rdf:resource="http://publications.europa.eu/resource/authority/place/FRA_PAR"/>
    <owl:sameAs rdf:resource="http://dbpedia.org/resource/Paris"/>
    <dbpedia:populationTotal rdf:datatype="http://www.w3.org
    /2001/XMLSchema#nonNegativeInteger">2187526</dbpedia:populationTotal>
  </rdf:Description>
</rdf:RDF>
```

([40]) https://wiki.dbpedia.org/

Helpful links and tools

Title	Description	Link
EU Vocabularies and Authority Tables	EU Vocabularies and Authority Tables have been developed for the Publications Office in order to facilitate the exchange of data between the different information systems of the EU institutions (legislation, calls for tender, etc.) and describe data sets (open source).	https://op.europa.eu/en/web/eu-vocabularies/authority-tables https://op.europa.eu/en/web/eu-vocabularies/dcat-ap-op
DBpedia	Linked data version of Wikipedia contents (open source).	https://wiki.dbpedia.org/
OpenRefine	With an RDF plugin this tool can import data in formats like CSV, JSON, and XML and map this data to an existing ontology (open source).	https://openrefine.org/
Cleaning data with OpenRefine	This article describes how to discover inconsistencies in data and how to diagnose the accuracy of data with OpenRefine ([41]).	https://doaj.org/article/3ccd075407a4481c85c0d00d65a003c0

4.6. File formats and their achievable openness level

The table below shows a list of commonly used formats along with information on whether they are machine readable and proprietary. The right-hand column indicates the number of stars that can be obtained when using this format for data publishing. The formats were selected based on the analysis performed in the data profiling phase. Ideally, the formats highlighted in green should be used. If this is not possible, formats from the yellow section should be used. Resorting to formats highlighted in red should be avoided, as only a one-star rating can be achieved with these.

([41]) De Wilde, M., van Hooland, S. and Verborgh, R., 'Cleaning data with OpenRefine', *The Programming Historian*, 1 August 2013, Editorial Board of the Programming Historian, United Kingdom, 2013.

Table 5. File formats and their achievable openness level

Format	Non-proprietary	Machine readable	Achievable stars
RDF	Yes	Yes	✪ ✪ ✪ ✪
XML	Yes	Yes	✪ ✪ ✪
JSON	Yes	Yes	✪ ✪ ✪
CSV	Yes	Yes	✪ ✪ ✪
ODS	Yes	Predominantly	✪ ✪ ✪
XLSX	Yes	Predominantly	✪ ✪ ✪
XLS	No	Predominantly	✪ ✪
TXT	Yes	Predominantly	✪*
HTML	Yes	Predominantly	✪*
PDF	Yes	No	✪
DOCX	Yes	No	✪
ODT	Yes	No	✪
PNG	Yes	No	✪
GIF	No	No	✪
JPG/JPEG	No	No	✪
TIFF	No	No	✪
DOC	No	No	✪

* Strictly according to the 5-star model, this format would have to be rated with three stars, since the data may well be designed to be machine-readable. However, we only give one star because this format was not originally intended to represent machine-readable but humand-readable content. Representing machine-readable content in this format does not meet best practice and is therefore not recommended by the authors.

Glossary

Accessibility

The degree to which required data can be accessed by data users, possibly including authentication and authorisation.

API (application programming interface)

An API is a programming interface. It is provided by a software system and allows other programs to communicate with this system.

APIs are often provided by data publishers and allow programs or apps to read the data directly over the web. To do this, the app sends a query to the API for the required data. The advantage of providing data via an API is that the entire data set does not need to be downloaded – it is possible to provide only the required data. This also ensures that the data is up to date.

Array

Arrays are list-like types of objects that represent a collection of elements that can be selected by corresponding indices.

Attribute

In the XML description language, an attribute represents a name–value pair that is part of a day. An attribute can only occur once per day and can only contain individual values.

Backward compatibility

Backward compatibility is the capacity of a hardware or software to interact with data and interfaces from earlier versions of the system or with other systems.

Boolean (values/type)

Boolean is a data type that can only contain one of the two possible values 'true' and 'false'.

camelCase

Spaces and special characters can hinder the automated processing of data. Therefore, it is advisable to group identifiers consisting of multiple words into one. In camelCase typography, the first character of each word is capitalised, except the first one. This is independent of the type of word.

Character encoding

Character encoding translates between characters and bytes through an encoding system.

Client

A client may be understood as an instance consuming data and can be a person or a computer. Typically, the client requests resources from a server. For example, a browser

loading a website would be considered a client, with the website being provided by the server.

CSV (comma-separated values)

CSV is a standard format for structured data. Because of its simplicity, openness and machine readability, CSV is often used for publishing open data.

data.europa.eu

The official portal for European Union data providing a single point of access to open data from international, EU, national, regional, local and geo data portals (https://data.europa.eu/en).

Data blending

Data blending is the process of merging data from different sources into one functioning data set.

Data catalogue

A data catalogue combines metadata with data management and search tools to improve data findability and to serve as an inventory and overview of possible uses for data.

Data cleansing

Data cleansing or data cleaning is the process of detecting and removing incorrect and/or inconsistent data from a record set.

Data preparation

Data preparation is the process of collecting, cleaning and consolidating data to create a consistent data set that can be used for analysis.

Data provider

The data provider is defined as the entity that provides content via a platform accessible to users. Decisions on the publication, terms of use and formats reside with the data provider.

Data set

A data set is a quantity of data that is related in content. A data set usually contains one or more resources, for example covering different formats, and metadata describing the content of the resources.

Data user

Data users are natural or legal persons who are entitled to use the data provided by the data provider for their own purposes and who are responsible for doing so in accordance with the conditions of use.

DCAT-AP (Data Catalogue Vocabulary Application Profile for Data Portals in Europe)

DCAT-AP is a standard based on the DCAT developed by the W3C and used for defining and structuring metadata for data sets from public authorities. It defines metadata fields and ranks them by importance, i.e. mandatory, recommended and

optional. For example, data sets must have a title, but providing a version is optional. For the greatest level of compatibility with users this standard should be followed as closely as possible.

DMP (data management plan)

A DMP is a written document that specifies what data is expected to be produced or acquired in a research project, how large the data set will be, how it will be analysed and described, how it will be stored and how it will be published and preserved.

Element

In XML, an element is a field containing data. An element is defined using tags and can also contain attributes.

Endpoint

An endpoint is a remote computing device that interacts with a network to which it is connected. Examples of endpoints are desktops, laptops and smartphones. Endpoints are vulnerable to cybercriminal activity.

Escaping

Escaping means making characters usable in data that are otherwise reserved for formatting. It is done by replacing the characters with specific codes. Without escaping, these characters would be interpreted as markup, which could break syntax validity.

EU ODP (European Union Open Data Portal)

Up until 21 April 2021 (when the European Data Portal and the European Data Portal were consolidated to become data.europa.eu – see glossary entry above), the EU ODP provided, via a metadata catalogue, a single point of access to data from the EU institutions, agencies and bodies for anyone to reuse.

Findability

The degree to which metadata and data is easy to find for humans and computers.

FAIR principles

The FAIR principles for scientific data management and stewardship published in *Scientific Data* ([42]) aim at enhancing the findability, accessibility, interoperability and reuse of digital assets.

GET request

In HTTP a GET request is a method for requesting a resource from a server.

Header

The term header refers to supplementary information of a file or protocol. For example, in CSV files a header line indicates variable names (and type/format if applicable) to be found in each column. In HTTP, headers allow a client or server to transmit supplementary information with a request.

([42]) Wilkinson, M., Dumontier, M., Aalbersberg, I. et al, 'The FAIR guiding principles for scientific data management and stewardship', *Scientific Data*, Vol. 3, Article No 160018, Macmillan Publishers Limited, 2016 (https://rdcu.be/cfaVN).

HTTP (hypertext transport protocol)

HTTP is one of the core technologies of the internet. It defines methods and status codes used for sending data between clients and servers.

ID

An ID is a unique identifier for a related set of data. Consecutive numbering is often used for this purpose. A URI is also a kind of ID.

Inspire

The infrastructure for spatial information in the European Community (Inspire) is an initiative of the European Commission that aims to create a European spatial data infrastructure for the purposes of a common environmental policy.

Interoperability

The degree to which data can be integrated with other data and interoperates with applications or workflows for analysis, storage and processing.

JSON

JSON is a powerful format that is well suited to data exchange between different applications. It can handle complex data structures, is easy to read for both humans and machines and is independent of platform and programming language.

Literal

In the context of RDF, a literal denotes a simple data value. Only RDF objects may be literals. Unlike RDF resources, these are not encoded with a URI and thus cannot be referenced from outside their 'own' triple. Literals are often used for data that loses its meaning outside its own triple, for example people's names.

Machine readability

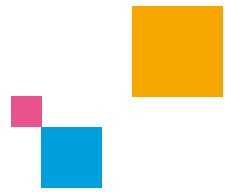
In principle, all data that can be interpreted by software is machine readable. In the context of open data this usually means data formats that enable further processing. The underlying data structure and corresponding standards must be publicly available and should be fully published and available free of charge.

Masking

Masking means hiding characters in data that may otherwise be interpreted incorrectly. For example, if commas were used as separators in a CSV file, commas in the data themselves would need to be masked.

Metadata

Metadata is used for the acquisition and description of a data set in a structured form. For example, metadata contains information about the content, title or format of a record. In short, metadata is data about data or references to the actual data. Metadata usually follows a certain schema which provides mandatory and optional information about the data set.



Namespace

Namespaces are used to prevent name conflicts in projects by ensuring that objects have unique identifiable names.

Null value

A null value indicates the complete absence of data. This should not be confused with an empty character string or the numeric value 0, since these contain actual information. A null value is therefore rather to be understood as an unknown value.

Payload

A payload is the transmitted data that contains the actual content. Metadata and HTTP headers (if applicable) are not part of the payload.

PascalCase

Spaces and special characters in identifiers can complicate data processing. If identifiers consist of several words, it is recommended that words be combined into one. In PascalCase notation the initial letters of each word are capitalised to facilitate human readability. This happens irrespective of word class, i.e. even verbs and adjectives begin with a capital letter.

RDF (resource description framework)

RDF is a model for storing data and metadata. It stores linked data in the form of triples.

Resource

In the context of RDF, a resource is defined as a data unit that can be related to other resources. A resource is usually unambiguously referenceable. The subject and predicate are resources and the object can be either a resource or a literal.

Resource ID

A resource ID or resource identifier is typically a string of characters used to reference and identify a resource.

Reusability

The degree to which data is optimised to be reused for replication and/or combination in a different setting. Reusability is achieved through well-specified metadata and data.

Server

A server provides data. Clients can send a request to the server, upon which the requested data is sent back to the client. For example, a website residing on a server on the internet can be loaded by a browser, i.e. the client.

Status code, HTTP

An HTTP status code is a standardised numeric value that provides information about the success of an HTTP request. All values within certain number ranges have a similar meaning, while the concrete numbers give a more precise differentiation. All codes in the range from 400 to 500 indicate errors on the client side. For example, code 403

shows that the request was not authorised, while 404 indicates that a resource is not available.

String

A string is a data type that is used to represent text. It includes characters and can include spaces and numbers.

Tag

In XML, a tag is the designation of a data unit. A keyword enclosed in arrow brackets marks the opening tag (<example>). The same keyword preceded by an arrow bracket and a slash and a closed by an arrow bracket marks the closing tag (</example>).

Triple

In RDF, a triple is the combination of a subject, a predicate and an object. This combination represents a unit of meaning. In RDF data is always stored in the form of triples. The corresponding database is called a triplestore.

URI (uniform resource identifier)

A URI is a unique reference to a resource. It can consist of letters and/or numbers; spaces are not allowed. A URI can point directly to the location of the resource, for example when using a network address (URL).

URL (uniform resource locator)

A URL is a subtype of URI. In contrast to a URI, a URL always points to a resource that can be found, so it is both identifier and address at the same time. Internet addresses or email addresses are URLs, for example.

UTF-8

UTF-8 is a widely used way of representing characters. Especially in connection with special characters, this type of storage ensures the greatest possible compatibility with other programs. It is the encoding of choice on the web.

Validator

A validator checks the syntactical correctness of code.

W3C

The World Wide Web Consortium is an international community for standardisation on the World Wide Web.

XML (Extensible Markup Language)

XML is a file format used for storing hierarchically structured data. It was designed to be machine readable and readable by humans.

Overview of quality indicators and metrics

Table 6. Overview of quality indicators and metrics

FAIR dimension	Indicator	Description	Metric	Can be aggregated throughout several data sets?	Data/ metadata	QN/QL (*)	Calculation	Used by other portal?	Relevance ranking
Findability	Completeness	The data is complete if it includes all items needed to represent the entity. Often related to null values in literature. At the metadata level, completeness indicates how much meta information is available for the given data set. Metadata should describe the resource as fully as possible.	Number of null values	Yes	Data	QN	Percentage	—	Medium
			Number of empty fields in metadata	Yes	Metadata	QN	Percentage	—	Medium
			Data set identifier re-solves to a digital object	Yes	Metadata	QN	Binary	—	Medium
	Findability	Data sets should be discoverable for both humans and computers. The findability of a data set depends on the description in the metadata: the better the data is described, e.g. through the usage of controlled vo-cabularies and keywords, the easier it is for users to find the data.	Keywords assigned	Yes	Metadata	QN	Binary	EDP	Medium
			Categories assigned	Yes	Metadata	QN	Binary	EDP	Medium
			Temporal information given	Yes	Metadata	QN	Binary	EDP	Medium
			Spatial information given	Yes	Metadata	QN	Binary	EDP	Medium
			Link to other data	Yes	Metadata	QN	Binary	GARDIAN	Low
Accessibility	Accessibility/ availability	Accessibility describes whether the content of the portal or the resources can be retrieved by a human or computer without any errors or access restrictions. Accessibility can be distinguished in two ways. For a human reader, the main issue is cognitive accessibility. For a computer, the main issue is physical accessibility.	Access URL accessible	Yes	Data	QN	Binary	EDP, GARDIAN	High
			Landing page accessible	Yes	Data	QN	Binary	—	Medium
			Download URL given	Yes	Data	QN	Binary	EDP, GARDIAN	High
			Download URL accessible	Yes	Data	QN	Binary	EDP	High
			Downloadable without registration	Yes	Data	QN	Binary	—	Medium
			Access authorisation information given	Yes	Metadata	QN	Binary	EDP	High
			Usage of controlled access right vocabulary	Yes	Metadata	QN	Binary	EDP	Medium

FAIR dimension	Indicator	Description	Metric	Can be aggregated throughout several data sets?	Data/ metadata	QN/QL (*)	Calculation	Used by other portal?	Relevance ranking
Interoperability	Conformity/ compliance	The data and metadata conform if they follow accepted standards, e.g. for capture, publication and description. An example could be the conformity of certain metadata values (URLs, emails), but also the overall compliance of the metadata with DCAT-AP. Valid date formats within the data or metadata also indicate conformity.	DCAT-AP compliance of metadata	Yes	Metadata	QN	Binary	EDP, GARDIAN	Medium
			Conformity of file formats and licences	Yes	Data	QN	Binary	—	Low
			Conformity of access to property values	Yes	Metadata	QN	Binary/ percentage	—	Low
			Conformity of date formats	Yes	Both	QN	Binary/ percentage	—	Low
			Conformity of email addresses	Yes	Both	QN	Binary/ percentage	—	Low
			Conformity of licences	Yes	Metadata	QN	Binary	—	Low
			Character encoding issues	Yes	Data	QN	Percentage	—	Low
			Data following a given schema	Yes	Data	QN	Binary	—	Low
	Machine readability/ processability	This indicator assesses the extent to which the data and metadata are machine interpretable, i.e. the extent to which they can be understood and handled by automated processes.	Processability of file format and media type	Yes	Data	QN	Binary	EDP	Medium
			Usage of controlled vocabularies	Yes	Both	QN	Binary/ percentage	EDP	Medium
	Openness	The openness of data is of crucial relevance for the concept of open data ([43]). Data is considered to be open if the resources are available in a non-proprietary format and can be used under an open licence.	Openness of file format and media type	Yes	Data	QN	Binary	EDP, GARDIAN	Medium
			Licence information given	Yes	Metadata	QN	Binary	EDP, GARDIAN	High
			Openness of licence	Yes	Metadata	QN	Binary	—	Medium
			Correctness of licence	Yes	Metadata	QN	Binary	EDP	Medium

([43]) Sunlight Foundation (2017), 'Ten principles for opening up government information' (https://sunlightfoundation.com/policy/documents/ten-open-data-principles/).

FAIR dimension	Indicator	Description	Metric	Can be aggregated throughout several data sets?	Data/ metadata	QN/QL (*)	Calculation	Used by other portal?	Relevance ranking
Reusability	Timeliness	Metadata and data are timely if they are up to date and represent the actual and current situation. This means that as soon as a change occurs in the real world, the data and metadata have to be modified too. However, the assessment of timeliness of data is not trivial as it is hard to automatically understand from the content if it is historical or real-time data. Thus, it is not easy to tell the requirements of timeliness in an automated way.	Update information given	Yes	Metadata	QN/QL	Binary	—	Medium
			Creation date given	Yes	Metadata	QN/QL	Binary	EDP	Medium
			Modification date given	Yes	Metadata	QN/QL	Binary	EDP	Medium
			Temporal information given	Yes	Metadata	QN/QL	Binary	EDP	Medium
	Consistency	Data and metadata are consistent if they do not contain any contradictions. Examples of contradictions would be a data set containing multiple and contradictory licence statements or modification dates that are earlier than creation dates. Contradiction might especially occur if data is combined from different sources.	Number of non-admissible values	Yes	Both	QN	Binary/ percentage	—	Low
			Semantic distance	Yes	Metadata	QN	Percentage	—	Low
			Compliance with community standards	Yes	Both	QN	Binary	GARDIAN	Low
			Freeness from duplicates	Yes	Data	QN	Binary/ percentage	—	Low

FAIR dimension	Indicator	Description	Metric	Can be aggregated throughout several data sets?	Data/ metadata	QN/QL (*)	Calculation	Used by other portal?	Relevance ranking
Reusability	Accuracy	Metadata isaccurate if the description of the content is as precise as possible, so that potential users get a realistic idea of the data and are able to quickly assess its relevance for their own contexts. Although this depends on the user's perception, there are some metadata values that can be checked automatically in terms of semantic accuracy: information given about file format and content size can be compared with the actual file format of the resource and its real-world size.	File format accuracy	Yes	Metadata	QN	Binary/ percentage	—	Low
			Content size accuracy	Yes	Metadata	QN	Binary/ percentage	—	Low
			Percentage of accurate cells	Yes	Data	QN	Percentage	—	Low
	Relevance	Data is only of use if it is relevant and of interest to the potential user. Thus, the data set should only contain the information necessary to support the task at hand. Relevance describes the extent to which the data is helpful and applicable, and the extent to which the amount of data is appropriate. This indicator is highly dependent on the user's perception and the task at hand.	Appropriate amount of data	No	Data	QL	—	—	Limited

FAIR dimension	Indicator	Description	Metric	Can be aggregated throughout several data sets?	Data/ metadata	QN/QL (*)	Calculation	Used by other portal?	Relevance ranking
Reusability	Understandability	Data and metadata are understandable if they are clear and comprehensible to the user. After studying the data and metadata, no ambiguities should remain. This indicator is highly dependent on the user's perception and their expert knowledge in the domain concerned. The understandability rating may increase if certain contextual information is provided, such as a description of the data, a title and keywords. However, in the end it depends on the user whether the data is actually comprehensible or not.	Description of data given	Yes	Metadata	QN/QL	Binary	—	Low
			Title given	Yes	Metadata	QN/QL	Binary	—	Low
			Keywords assigned	Yes	Metadata	QN/QL	Binary	EDP	Low
			Documentation of data given	Yes	Metadata	QN/QL	Binary	GARDIAN	Low
	Credibility	Data is considered credible if it is based on trustworthy sources. Credibility describes the extent to which 'data has attributes that are regarded as true and believable by users' ([44]). Thus, this indicator is highly dependent on the user's perception. Still, the credibility and trustworthiness of the data may increase if certain contextual information is provided, such as information about the original publisher, the contact point and the data set owner.	Contact point given	Yes	Metadata	QN/QL	Binary	EDP	Low
			Data set publisher given	Yes	Metadata	QN/QL	Binary	EDP	Low
			Data set creator given	Yes	Metadata	QN/QL	Binary	—	Limited

([44]) Iso25000.com, 'ISO/IEC 25012: Quality of Data Product' (http://iso25000.com/index.php/en/iso-25000-standards/iso-25012?limit=5&limitstart=0).
(*) Quantitative/qualitative

Checklist for publishing high-quality data

Make your data FAIR(1)

Define data management plan

Before collecting your data

- Data characteristics: structure, modelling
- Data documentation
- Data quality assurance
- Data sharing
- Data storage and dissemination

Assess data quality

Reusability, accessiblity

- Validate and clean your data
  - > Remove duplicates
  - > Mark null values
- Standardise your data
  - > Format date, time, numbers
  - > Check character encoding

Document your data

Reusability

- Document schemas and data models
- Document data changes and versions
- Document semantics of data
- Document data validation report
- Document validation rules

Enrich your data

Interoperability

- Reuse concepts from controlled vocabularies
- Harmonise labels
- Dereference translation of labels
- Link and augment your data

Publish your data

Reusability, accessiblity

- Offer direct access (accessible download URL)
- Use machine-readable open formats
- Publish under an open licence
- Publish documentation about your data
- Use URIs and linked data

Facilitate data discovery

Findability

- Describe your data with rich metadata

(1) https://www.go-fair.org/fair-principles/

List of figures

List of tables

Bibliography

Auer, S., Lehmann, J., Maurino, A., Pietrobon, R., Rula, A. and Zaveri, A. (2012), 'Quality assessment for linked data: a survey', *Semantic Web 1*, IOS Press, (http://www.semantic-web-journal.net/system/files/swj773.pdf).

Batini, C., Cappiello, C., Francalanci, C. and Maurino, A. (2009), 'Methodologies for data quality assessment and improvement', *ACM Computing Survey*, Vol. 41, No 3, pp. 16–52 (http://dimacs-algorithmic-mdm.wdfiles.com/local--files/start/Methodologies%20for%20Data%20Quality%20Assessment%20and%20Improvement.pdf).

Canova, L., Iemma, R., Morando, F., Orozco Minotas, C., Torchiano, M. and Vetrò, A., (2016), 'Open data quality measurement framework: definition and application to open government data', *Government Information Quarterly*, Vol. 33, No 2, Elsevier, pp. 325–337 (https://www.sciencedirect.com/science/article/pii/S0740624X16300132).

data.europa.eu, *Metadata Assessment Methodology* (https://www.europeandataportal.eu/mqa/methodology?locale=en#).

De Wilde, M., van Hooland, S. and Verborgh, R. (2013), 'Cleaning data with OpenRefine', *The Programming Historian*, Editorial Board of the Programming Historian, United Kingdom (https://doaj.org/article/3ccd075407a4481c85c0d00d65a003c0).

Duval, E. and Ochoa, X. (2009), 'Automatic evaluation of metadata quality in digital repositories', *International Journal on Digital Libraries*, Vol. 10, pp. 67–91 (https://link.springer.com/article/10.1007/s00799-009-0054-4).

European Commission (2014), *Training Module 2.2. – Open data & metadata quality* (https://www.europeandataportal.eu/sites/default/files/d2.1.2_training_module_2.2_open_data_quality_en_edp.pdf).

European Commission (2018), *Turning FAIR into Reality – Final report and action plan from the European Commission expert group on FAIR data*, Publications Office of the European Union, Luxembourg (https://op.europa.eu/en/publication-detail/-/publication/7769a148-f1f6-11e8-9982-01aa75ed71a1).

Gartner Research (2019a), 'Magic quadrant for data quality tools' (https://www.gartner.com/en/documents/3905769/magic-quadrant-for-data-quality-tools).

Gartner Research (2019b), 'Market guide for data preparation tools' (https://www.gartner.com/en/documents/3906957/market-guide-for-data-preparation-tools).

Hare, J. (2016), 'What is metadata and why is it as important as data itself?', opendatasoft (https://www.opendatasoft.com/blog/2016/08/25/what-is-metadata-and-why-is-it-important-data).

Iso25000.com, 'ISO/IEC 25012: Quality of data product' (http://iso25000.com/index.php/en/iso-25000-standards/iso-25012?limit=5&limitstart=0).

Kubler, S., Le Traon, Y, Neumaier, S., Robert, J. and Umbrich, J. (2018), 'Comparison of metadata quality in open data portals using the Analytic Hierarchy Process', *Government Information Quarterly*, Vol. 35, No 1, Elsevier (https://www.sciencedirect.com/science/article/pii/S0740624X16301319).

Little, C. (2018), 'The Forrester Wave™: data preparation solutions', Forrester (https://www.forrester.com/report/The+Forrester+Wave+Data+Preparation+Solutions+Q4+2018/-/E-RES141619).

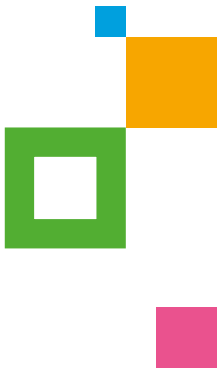
Lnénicka, M. and Máchová, R. (2017), 'Evaluating the quality of open data portals on the national level', *Journal of Theoretical and Applied Electronic Commerce Research*, Vol. 12, No 1, Universidad de Talca (https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-18762017000100003).

Neumaier, S. (2015), 'Open data quality: assessment and evolution of (meta-)data quality in the open data landscape', thesis (https://www.data.gv.at/wp-content/uploads/2016/02/Sebastian_Neumaier_MSc_2015.pdf).

Reiche, K. J. (2013), 'Assessment and visualization of metadata quality for open government data', thesis (https://www.inf.fu-berlin.de/inst/ag-se/theses/Reiche13-metadata-quality.pdf).

Strong, D. M. and Wang, R. Y. (1996) 'Beyond accuracy: what data quality means to data consumers', *Journal of Management Information Systems*, Vol. 12, No 4, Spring, pp. 5–33 (http://mitiq.mit.edu/Documents/Publications/TDQMpub/14_Beyond_Accuracy.pdf).

Sunlight Foundation (2017), 'Ten principles for opening up government information' (https://sunlightfoundation.com/policy/documents/ten-open-data-principles/).



List of topics

(section number in brackets)

- Make use of tooling whenever possible (1)
- Develop a data management plan (1)
- Describe your data with metadata to improve data discovery (1)
- Mark null values explicitly as such (1)
- Publish data without restrictions (1)
- Provide an accessible download URL (1)
- Consider ISO standards for formatting date and time (1)
- Use a dot to separate whole numbers from decimals (1)
- Do not use a thousand separator(1)
- Make use of a standardised character encoding (1)
- Provide an appropriate amount of data (1)
- Consider community standards (1)
- Remove duplicates from your data (1)
- Increase the accuracy of your data (1)
- Provide information on byte size (1)
- Make use of controlled vocabularies to standardise data (2)
- Link relevant data sets (2)
- Use knowledge bases for enrichment (2)
- Use schemas to specify data structure (3)
- Document data changes (3)
- Use a machine-readable format (4)
- Use a non-proprietary format (4)
- Consider open standards (4)
- Consider linked data principles (4)

GETTING IN TOUCH WITH THE EU

In person
All over the European Union there are hundreds of Europe Direct information centres. You can find the address of the centre nearest you at: https://europa.eu/european-union/contact_en

On the phone or by email
Europe Direct is a service that answers your questions about the European Union. You can contact this service:

— by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),
— at the following standard number: +32 22999696, or
— by email via: https://europa.eu/european-union/contact_en

FINDING INFORMATION ABOUT THE EU

Online
Information about the European Union in all the official languages of the EU is available on the Europa website at: https://europa.eu/european-union/index_en

EU publications
You can download or order free and priced EU publications from: https://op.europa.eu/en/publications. Multiple copies of free publications may be obtained by contacting Europe Direct or your local information centre (see https://europa.eu/european-union/contact_en).

EU law and related documents
For access to legal information from the EU, including all EU law since 1951 in all the official language versions, go to EUR-Lex at: http://eur-lex.europa.eu

Open data from the EU
The official portal for European data (https://data.europa.eu/en) provides access to datasets from the EU. Data can be downloaded and reused for free, for both commercial and non-commercial purposes.

OA-06-22-121-EN-N

Publications Office
of the European Union

ISBN 978-92-78-42763-4