
datacatalogtordf

Stig B. Dørmænen

Sep 22, 2022

CONTENTS

1	License	1
2	Reference	5
3	Installation	51
4	Usage	53
	Python Module Index	55
	Index	57

LICENSE

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION**1. Definitions.**

“License” shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

“Licensor” shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

“Legal Entity” shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

“You” (or “Your”) shall mean an individual or Legal Entity exercising permissions granted by this License.

“Source” form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

“Object” form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

“Work” shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

“Derivative Works” shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

“Contribution” shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to the Licensor or its

representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as “Not a Contribution.”

“Contributor” shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.
3. **Grant of Patent License.** Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.
4. **Redistribution.** You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:
 - (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and
 - (b) You must cause any modified files to carry prominent notices stating that You changed the files; and
 - (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and
 - (d) If the Work includes a “NOTICE” text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions.** Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.
7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.
8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets “[]” replaced with your own identifying information. (Don’t include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format.

We also recommend that a file or class name and description of purpose be included on the same “printed page” as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

REFERENCE

- *datacatalogtordf.resource*
- *datacatalogtordf.relationship*
- *datacatalogtordf.catalog*
- *datacatalogtordf.catalogrecord*
- *datacatalogtordf.dataservice*
- *datacatalogtordf.dataset*
- *datacatalogtordf.dataset_series*
- *datacatalogtordf.distribution*
- *datacatalogtordf.uri*
- *datacatalogtordf.periodoftime*
- *datacatalogtordf.location*
- *datacatalogtordf.agent*
- *datacatalogtordf.exceptions*

2.1 datacatalogtordf.resource

Resource module for mapping a sub-classes to rdf.

This module contains methods for mapping a sub-class objects to rdf according to the [dcat-ap-no v.2 standard](#)

Refer to sub-class for typical usage examples.

class datacatalogtordf.resource.Resource

Bases: [ABC](#)

An abstract class representing a dcat:Resource.

Ref: [dcat:Resource](#).

property access_rights: [str](#)

A link to information about who can access the resource or an indication of its security status.

Type
URI

Return type

`str`

property conforms_to: `List[str]`

A list of links to established standards to which the described resource conforms.

Type

`List[URI]`

Return type

`List[str]`

property contactpoint: `Contact`

Relevant contact information for the cataloged resource.

Type

`Contact`

Return type

`Contact`

property creator: `str`

Link to the entity responsible for producing the resource.

Type

`URI`

Return type

`str`

property description: `Dict[str, str]`

A free-text account of the item. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

classmethod from_json(json)

Convert a JSON (dict).

Parameters

json (`Dict`) – A dict representing this class.

Returns

The object.

Return type

`Resource`

property has_policy: `str`

A link to an ODRL conformant policy expressing the rights associated with the resource.

Type

`URI`

Return type

`str`

property identifier: `str`

A URI uniquely identifying the resource.

Type

URI

Return type

`str`

property is_referenced_by: `List[Resource]`

A list of related resources, such as a publication, that references, cites, or otherwise points to the cataloged resource.

Type

`List[Resource]`

Return type

`List[Resource]`

property keyword: `Dict[str, str]`

A keyword or tag describing the resource. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

property landing_page: `List[str]`

A list of links to web pages that can be navigated to in a Web browser to gain access to the catalog, a dataset, its distributions and/or additional information.

Type

`List[URI]`

Return type

`List[str]`

property language: `List[str]`

A list of links to languages of the item.

Type

`List[str]`

Return type

`List[str]`

property license: `str`

A link to a legal document under which the resource is made available.

Type

URI

Return type

`str`

property modification_date: `str`

Most recent date on which the item was changed, updated or modified.

Type

Date

Return type*str***property prev:** *Resource*

The previous resource in an ordered collection or series of resources.

Type*Resource***Return type***Resource***property publisher:** *Union[Agent, str]*

A URI uniquely identifying the publisher of the resource.

Type*Union[Agent, str]***Return type***Union[Agent, str]***property qualified_attributions:** *List[Dict]*

List of links to an Agent having some form of responsibility for the resource.

Type*List[Dict]***Return type***List[Dict]***property qualified_relation:** *List[Relationship]*

A list of links to a description of a relationship with another resource.

Type*List[Relationship]***Return type***List[Relationship]***property release_date:** *str*

Date of formal issuance (e.g., publication) of the item.

Type*Date***Return type***str***property resource_relation:** *List[str]*

A list of links to resources with an unspecified relationship to the cataloged item.

Type*List[URI]***Return type***List[str]***property rights:** *str*

license or dct:accessRights, such as copyright statements.

Returns

The link

Return type`str`**Type**`URI`**Type**

A link to a statement that concerns all rights not addressed with dct

property theme: `List[str]`

A list of links to categories of the resource.

Type`List[URI]`**Return type**`List[str]`**property title:** `Dict[str, str]`

A name given to the item. key is language code.

Type`Dict[str, str]`**Return type**`Dict[str, str]`**to_json()**

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type`Dict`**to_rdf**(*format*='turtle', *encoding*='utf-8')

Maps the distribution to rdf.

Available formats:

- turtle (default)
- xml
- json-ld

Parameters

- **format** (`str`) – a valid format.
- **encoding** (`Optional[str]`) – the encoding to serialize into

Return type`Union[bytes, str]`**Returns**

a rdf serialization as a bytes literal according to format.

Example

```
>>> from datacatalogtordf import Catalog
>>>
>>> catalog = Catalog()
>>> catalog.identifier = "http://example.com/catalogs/1"
>>> catalog.title = {'en': 'Title of catalog'}
>>> bool(catalog.to_rdf())
True
```

property `type_genre:` `str`

A link to the nature or genre of the resource.

Type

URI

Return type

`str`

2.2 datacatalogtordf.relationship

Relationship module for mapping a relationship to rdf.

This module contains methods for mapping a relationship object to rdf according to the [dcat-ap-no v.2 standard](#)

Example

```
>>> from datacatalogtordf import Relationship
>>>
>>> relationship = Relationship()
>>> relationship.identifier = "http://example.com/relations/1"
>>> relationship.title = {"en": "Title of relationship"}
>>>
>>> bool(relationship.to_rdf())
True
```

class `datacatalogtordf.relationship.Relationship(identifier=None)`

Bases: `object`

A class representing a dcat:Relationship.

Ref: [dcat:Relationship](#)

classmethod `from_json(json)`

Convert a JSON (dict).

Parameters

`json` (`Dict`) – A dict representing this class.

Returns

The object.

Return type

Relationship

property had_role: `str`

A URI identifying the role.

Type

`URI`

Return type

`str`

property identifier: `str`

a URI uniquely identifying the resource.

Type

`URI`

Return type

`str`

property relation: `Resource`

A URI uniquely identifying related resource.

Type

`Resource`

Return type

`Resource`

to_json()

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type

`Dict`

to_rdf(*format='turtle', encoding='utf-8'*)

Maps the relationship to rdf.

Parameters

- **format** (`str`) – a valid format. Default: turtle
- **encoding** (`Optional[str]`) – the encoding to serialize into

Return type

`Union[bytes, str]`

Returns

a rdf serialization as a bytes literal according to format.

2.3 datacatalogtordf.catalog

Catalog module for mapping a catalog to rdf.

This module contains methods for mapping a catalog object to rdf according to the [dcat-ap-no v.2 standard](#)

Example

```
>>> from datacatalogtordf import Catalog, Dataset
>>>
>>> catalog = Catalog()
>>> catalog.identifier = "http://example.com/catalogs/1"
>>> catalog.title = {"en": "Title of catalog"}
>>>
>>> a_dataset = Dataset()
>>> a_dataset.identifier = "http://example.com/datasets/1"
>>> catalog.datasets.append(a_dataset)
>>>
>>> bool(catalog.to_rdf())
True
```

class datacatalogtordf.catalog.Catalog(identifier=None)

Bases: [Dataset](#)

A class representing a dcat:Catalog.

Ref: [dcat:Catalog](#).

property access_rights: [str](#)

A link to information about who can access the resource or an indication of its security status.

Type

[URI](#)

Return type

[str](#)

property access_rights_comments: [List\[str\]](#)

Referanse til hjemmel (kilde for påstand) i offentlighetsloven, sikkerhetsloven, beskyttelsesinstruksen eller annet lovverk som ligger til grunn for vurdering av tilgangsnivå.

Type

[List\[URI\]](#)

Return type

[List\[str\]](#)

property catalogrecords: [List\[CatalogRecord\]](#)

A list of records describing the registration of a single dataset or data service that is part of the catalog.

Type

[List\[CatalogRecord\]](#)

Return type

[List\[CatalogRecord\]](#)

property catalogs: `List[Catalog]`

A list of catalogs that are of interest in the context of this catalog.

Type

`List[Catalog]`

Return type

`List[Catalog]`

property conforms_to: `List[str]`

A list of links to established standards to which the described resource conforms.

Type

`List[URI]`

Return type

`List[str]`

property contactpoint: `Contact`

Relevant contact information for the cataloged resource.

Type

`Contact`

Return type

`Contact`

property creator: `str`

Link to the entity responsible for producing the resource.

Type

`URI`

Return type

`str`

property datasets: `List[Dataset]`

A list of datasets that is listed in the catalog.

Type

`List[Dataset]`

Return type

`List[Dataset]`

property dct_identifier: `str`

the identifier for the catalog.

Type

`str`

Return type

`str`

property description: `Dict[str, str]`

A free-text account of the item. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

property distributions: `List[Distribution]`

A list of distributions of the dataset.

Type

`List[Distribution]`

Return type

`List[Distribution]`

property frequency: `str`

A link to resource describing the frequency at which dataset is published.

Type

`[URI]`

Return type

`str`

classmethod from_json(*json*)

Convert a JSON (dict).

Parameters

json (`Dict`) – A dict representing this class.

Returns

The object.

Return type

`Resource`

property has_parts: `List[Catalog]`

A list of resources that is listed in the catalog.

Type

`List[Catalog]`

Return type

`List[Catalog]`

property has_policy: `str`

A link to an ODRL conformant policy expressing the rights associated with the resource.

Type

`URI`

Return type

`str`

property homepage: `str`

A link to a homepage for the catalog.

Type

`URI`

Return type

`str`

property identifier: `str`

A URI uniquely identifying the resource.

Type

`URI`

Return type`str`**property in_series:** `DatasetSeries`

A dataset series of which the dataset is part.

Type`DatasetSeries`**Return type**`DatasetSeries`**property is_referenced_by:** `List[Resource]`

A list of related resources, such as a publication, that references, cites, or otherwise points to the cataloged resource.

Type`List[Resource]`**Return type**`List[Resource]`**property keyword:** `Dict[str, str]`

A keyword or tag describing the resource. key is language code.

Type`Dict[str, str]`**Return type**`Dict[str, str]`**property landing_page:** `List[str]`

A list of links to web pages that can be navigated to in a Web browser to gain access to the catalog, a dataset, its distributions and/or additional information.

Type`List[URI]`**Return type**`List[str]`**property language:** `List[str]`

A list of links to languages of the item.

Type`List[str]`**Return type**`List[str]`**property license:** `str`

A link to a legal document under which the resource is made available.

Type`URI`**Return type**`str`**property modification_date:** `str`

Most recent date on which the item was changed, updated or modified.

Type*Date***Return type***str***property prev:** *Resource*

The previous resource in an ordered collection or series of resources.

Type*Resource***Return type***Resource***property publisher:** *Union[Agent, str]*

A URI uniquely identifying the publisher of the resource.

Type*Union[Agent, str]***Return type***Union[Agent, str]***property qualified_attributions:** *List[Dict]*

List of links to an Agent having some form of responsibility for the resource.

Type*List[Dict]***Return type***List[Dict]***property qualified_relation:** *List[Relationship]*

A list of links to a description of a relationship with another resource.

Type*List[Relationship]***Return type***List[Relationship]***property release_date:** *str*

Date of formal issuance (e.g., publication) of the item.

Type*Date***Return type***str***property resource_relation:** *List[str]*

A list of links to resources with an unspecified relationship to the cataloged item.

Type*List[URI]***Return type***List[str]*

property rights: `str`

license or dct:accessRights, such as copyright statements.

Returns

The link

Return type

`str`

Type

URI

Type

A link to a statement that concerns all rights not addressed with dct

property services: `List[DataService]`

A list of dataservices of sites or end-points that is listed in the catalog.

Type

`List[DataService]`

Return type

`List[DataService]`

property spatial: `List[Union[Location, str]]`

A list of geographical areas covered by the dataset.

Type

`List[Location]`

Return type

`List[Union[Location, str]]`

property spatial_resolution_in_meters: `List[Decimal]`

A list of minimum spatial separation resolvables in a dataset, measured in meters.

Type

`List[Decimal]`

Return type

`List[Decimal]`

property temporal: `List[PeriodOfTime]`

A list of temporal periods that the dataset covers.

Type

`List[PeriodOfTime]`

Return type

`List[PeriodOfTime]`

property temporal_resolution: `List[str]`

A list of minimum time period resolvables in the dataset.

Type

`List[str]`

Return type

`List[str]`

property theme: `List[str]`

A list of links to categories of the resource.

Type

`List[URI]`

Return type

`List[str]`

property themes: `List[str]`

A list of links to knowledge organization system (KOS) used to classify catalog.

Type

`List[URI]`

Return type

`List[str]`

property title: `Dict[str, str]`

A name given to the item. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

to_json()

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type

`Dict`

to_rdf(*format='turtle', encoding='utf-8', include_datasets=True, include_services=True*)

Maps the catalog to rdf.

Available formats:

- turtle (default)
- xml
- json-ld

Parameters

- **format** (*str*) – a valid format.
- **encoding** (*str*) – the encoding to serialize into
- **include_datasets** (*bool*) – includes the dataset graphs in the catalog
- **include_services** (*bool*) – includes the services in the catalog

Return type

`Union[bytes, str]`

Returns

a rdf serialization as a bytes literal according to format.

property type_genre: `str`

A link to the nature or genre of the resource.

Type

`URI`

Return type

`str`

property was_generated_by: `str`

A link to an activity that generated, or provides the business context for, the creation of the dataset.

Type

`URI`

Return type

`str`

2.4 datacatalogtordf.catalogrecord

CatalogRecord module for mapping a catalogrecord to rdf.

This module contains methods for mapping a catalogrecord object to rdf according to the [dcat-ap-no v.2 standard](#)

Example

```
>>> from datacatalogtordf import CatalogRecord
>>>
>>> catalogrecord = CatalogRecord()
>>> catalogrecord.identifier = "http://example.com/catalogrecords/1"
>>> catalogrecord.title = {"en": "Title of catalogrecord"}
>>>
>>> bool(catalogrecord.to_rdf())
True
```

class datacatalogtordf.catalogrecord.CatalogRecord(*identifier=None*)

Bases: `object`

A class representing a dcat:CatalogRecord.

Ref: https://www.w3.org/TR/vocab-dcat-2/#Class:Catalog_Record

Parameters

identifier (`URI`) – the identifier of the dataset-series.

property conforms_to: `List[str]`

An established standard to which the described resource conforms.

Type

`List[URI]`

Return type

`List[str]`

property description: `Dict[str, str]`

A free-text account of the record. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

classmethod `from_json(json)`

Convert a JSON (dict).

Parameters

json (`Dict`) – A dict representing this class.

Returns

The object.

Return type

CatalogRecord

property identifier: `str`

a URI uniquely identifying the catalog record.

Type

URI

Return type

`str`

property listing_date: `str`

The date of listing.

Type

Date

Return type

`str`

property modification_date: `str`

Most recent date on which the catalog entry was changed.

Type

Date

Return type

`str`

property primary_topic: *Resource*

Resource (dataset or service) described in the record.

Type

Resource

Type

The dcat

Return type

Resource

property title: `Dict[str, str]`

A name given to the record. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

to_json()

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type

`Dict`

to_rdf(*format='turtle', encoding='utf-8'*)

Maps the catalogrecord to rdf.

Return type

`Union[bytes, str]`

2.5 datacatalogtordf.dataservice

DataService module for mapping a dataService to rdf.

This module contains methods for mapping a dataservice object to rdf according to the [dcat-ap-no v.2 standard](#)

Example

```
>>> from datacatalogtordf import DataService
>>>
>>> dataservice = DataService()
>>> dataservice.identifier = "http://example.com/dataservices/1"
>>> dataservice.title = {"en": "Title of dataservice"}
>>>
>>> bool(dataservice.to_rdf())
True
```

class datacatalogtordf.dataservice.**DataService**(*identifier=None*)

Bases: [Resource](#)

A class representing a dcat:DataService.

Parameters

identifier ([URI](#)) – the identifier of the datasetservice.

Ref: [dcat:DataService](#).

property access_rights: `str`

A link to information about who can access the resource or an indication of its security status.

Type

[URI](#)

Return type`str`**property conforms_to:** `List[str]`

A list of links to established standards to which the described resource conforms.

Type`List[URI]`**Return type**`List[str]`**property contactpoint:** `Contact`

Relevant contact information for the cataloged resource.

Type`Contact`**Return type**`Contact`**property creator:** `str`

Link to the entity responsible for producing the resource.

Type`URI`**Return type**`str`**property description:** `Dict[str, str]`

A free-text account of the item. key is language code.

Type`Dict[str, str]`**Return type**`Dict[str, str]`**property endpointDescription:** `str`

A description of the services available via the end-points, including their operations, parameters etc.

Type`URI`**Return type**`str`**property endpointURL:** `str`

The root location or primary endpoint of the service (a Web-resolvable IRI).

Type`URI`**Return type**`str`**classmethod from_json(*json*)**

Convert a JSON (dict).

Parameters

json (`Dict`) – A dict representing this class.

Returns

The object.

Return type

Resource

property has_policy: `str`

A link to an ODRL conformant policy expressing the rights associated with the resource.

Type

URI

Return type

`str`

property identifier: `str`

A URI uniquely identifying the resource.

Type

URI

Return type

`str`

property is_referenced_by: `List[Resource]`

A list of related resources, such as a publication, that references, cites, or otherwise points to the cataloged resource.

Type

`List[Resource]`

Return type

`List[Resource]`

property keyword: `Dict[str, str]`

A keyword or tag describing the resource. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

property landing_page: `List[str]`

A list of links to web pages that can be navigated to in a Web browser to gain access to the catalog, a dataset, its distributions and/or additional information.

Type

`List[URI]`

Return type

`List[str]`

property language: `List[str]`

A list of links to languages of the item.

Type

`List[str]`

Return type

`List[str]`

property license: `str`

A link to a legal document under which the resource is made available.

Type

URI

Return type

`str`

property media_types: `List[str]`

A list of media types that is offered in the responses.

Type

`List[src]`

Return type

`List[str]`

property modification_date: `str`

Most recent date on which the item was changed, updated or modified.

Type

Date

Return type

`str`

property prev: *Resource*

The previous resource in an ordered collection or series of resources.

Type

Resource

Return type

Resource

property publisher: `Union[Agent, str]`

A URI uniquely identifying the publisher of the resource.

Type

`Union[Agent, str]`

Return type

`Union[Agent, str]`

property qualified_attributions: `List[Dict]`

List of links to an Agent having some form of responsibility for the resource.

Type

`List[Dict]`

Return type

`List[Dict]`

property qualified_relation: `List[Relationship]`

A list of links to a description of a relationship with another resource.

Type

`List[Relationship]`

Return type

`List[Relationship]`

property release_date: str

Date of formal issuance (e.g., publication) of the item.

Type

Date

Return type

str

property resource_relation: List[str]

A list of links to resources with an unspecified relationship to the cataloged item.

Type

List[*URI*]

Return type

List[str]

property rights: str

license or dct:accessRights, such as copyright statements.

Returns

The link

Return type

str

Type

URI

Type

A link to a statement that concerns all rights not addressed with dct

property servesdatasets: List[Dataset]

A list of datasets that this service serves.

Type

List[*Dataset*]

Return type

List[*Dataset*]

property theme: List[str]

A list of links to categories of the resource.

Type

List[*URI*]

Return type

List[str]

property title: Dict[str, str]

A name given to the item. key is language code.

Type

Dict[str, str]

Return type

Dict[str, str]

`to_json()`

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type

Dict

`to_rdf(format='turtle', encoding='utf-8')`

Maps the distribution to rdf.

Available formats:

- turtle (default)
- xml
- json-ld

Parameters

- **format** (`str`) – a valid format.
- **encoding** (`Optional[str]`) – the encoding to serialize into

Return type

`Union[bytes, str]`

Returns

a rdf serialization as a bytes literal according to format.

Example

```
>>> from datacatalogtordf import Catalog
>>>
>>> catalog = Catalog()
>>> catalog.identifier = "http://example.com/catalogs/1"
>>> catalog.title = {'en': 'Title of catalog'}
>>> bool(catalog.to_rdf())
True
```

property `type_genre: str`

A link to the nature or genre of the resource.

Type

URI

Return type

`str`

2.6 datacatalogtordf.dataset

Dataset module for mapping a dataset to rdf.

This module contains methods for mapping a dataset object to rdf according to the [dcat-ap-no v.2 standard](#)

Example

```
>>> from datacatalogtordf import Dataset, Distribution
>>>
>>> dataset = Dataset()
>>> dataset.identifier = "http://example.com/datasets/1"
>>> dataset.title = {"en": "Title of dataset"}
>>>
>>> a_distribution = Distribution()
>>> a_distribution.identifier = "http://example.com/dataservices/1"
>>> a_distribution.title = {"en": "Title of distribution"}
>>> dataset.distributions.append(a_distribution)
>>>
>>> bool(dataset.to_rdf())
True
```

class datacatalogtordf.dataset.Dataset(identifier=None)

Bases: [Resource](#)

A class representing a dcat:Dataset.

Parameters

identifier ([URI](#)) – the identifier of the dataset.

Ref: [dcat:Dataset](#).

property access_rights: [str](#)

A link to information about who can access the resource or an indication of its security status.

Type

[URI](#)

Return type

[str](#)

property access_rights_comments: [List\[str\]](#)

Referanse til hjemmel (kilde for påstand) i offentlighetsloven, sikkerhetsloven, beskyttelsesinstruksen eller annet lovverk som ligger til grunn for vurdering av tilgangsnivå.

Type

[List\[URI\]](#)

Return type

[List\[str\]](#)

property conforms_to: [List\[str\]](#)

A list of links to established standards to which the described resource conforms.

Type

[List\[URI\]](#)

Return type`List[str]`**property contactpoint: Contact**

Relevant contact information for the cataloged resource.

Type`Contact`**Return type**`Contact`**property creator: str**

Link to the entity responsible for producing the resource.

Type`URI`**Return type**`str`**property dct_identifier: str**

the identifier for the dataset.

Type`str`**Return type**`str`**property description: Dict[str, str]**

A free-text account of the item. key is language code.

Type`Dict[str, str]`**Return type**`Dict[str, str]`**property distributions: List[Distribution]**

A list of distributions of the dataset.

Type`List[Distribution]`**Return type**`List[Distribution]`**property frequency: str**

A link to resource describing the frequency at which dataset is published.

Type`[URI]`**Return type**`str`**classmethod from_json(json)**

Convert a JSON (dict).

Parameters

json (`Dict`) – A dict representing this class.

Returns

The object.

Return type

Resource

property has_policy: `str`

A link to an ODRL conformant policy expressing the rights associated with the resource.

Type

URI

Return type

`str`

property identifier: `str`

A URI uniquely identifying the resource.

Type

URI

Return type

`str`

property in_series: *DatasetSeries*

A dataset series of which the dataset is part.

Type

DatasetSeries

Return type

DatasetSeries

property is_referenced_by: `List[Resource]`

A list of related resources, such as a publication, that references, cites, or otherwise points to the cataloged resource.

Type

`List[Resource]`

Return type

`List[Resource]`

property keyword: `Dict[str, str]`

A keyword or tag describing the resource. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

property landing_page: `List[str]`

A list of links to web pages that can be navigated to in a Web browser to gain access to the catalog, a dataset, its distributions and/or additional information.

Type

`List[URI]`

Return type

`List[str]`

property language: `List[str]`

A list of links to languages of the item.

Type

`List[str]`

Return type

`List[str]`

property license: `str`

A link to a legal document under which the resource is made available.

Type

URI

Return type

`str`

property modification_date: `str`

Most recent date on which the item was changed, updated or modified.

Type

Date

Return type

`str`

property prev: *Resource*

The previous resource in an ordered collection or series of resources.

Type

Resource

Return type

Resource

property publisher: `Union[Agent, str]`

A URI uniquely identifying the publisher of the resource.

Type

`Union[Agent, str]`

Return type

`Union[Agent, str]`

property qualified_attributions: `List[Dict]`

List of links to an Agent having some form of responsibility for the resource.

Type

`List[Dict]`

Return type

`List[Dict]`

property qualified_relation: `List[Relationship]`

A list of links to a description of a relationship with another resource.

Type

`List[Relationship]`

Return type

`List[Relationship]`

property release_date: str

Date of formal issuance (e.g., publication) of the item.

Type

Date

Return type

str

property resource_relation: List[str]

A list of links to resources with an unspecified relationship to the cataloged item.

Type

List[*URI*]

Return type

List[str]

property rights: str

license or dct:accessRights, such as copyright statements.

Returns

The link

Return type

str

Type

URI

Type

A link to a statement that concerns all rights not addressed with dct

property spatial: List[Union[Location, str]]

A list of geographical areas covered by the dataset.

Type

List[*Location*]

Return type

List[Union[*Location*, str]]

property spatial_resolution_in_meters: List[Decimal]

A list of minimum spatial separation resolvables in a dataset, measured in meters.

Type

List[Decimal]

Return type

List[Decimal]

property temporal: List[PeriodOfTime]

A list of temporal periods that the dataset covers.

Type

List[*PeriodOfTime*]

Return type

List[*PeriodOfTime*]

property temporal_resolution: `List[str]`

A list of minimum time period resolvables in the dataset.

Type

`List[str]`

Return type

`List[str]`

property theme: `List[str]`

A list of links to categories of the resource.

Type

`List[URI]`

Return type

`List[str]`

property title: `Dict[str, str]`

A name given to the item. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

to_json()

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type

`Dict`

to_rdf(*format='turtle', encoding='utf-8', include_distributions=True*)

Maps the catalog to rdf.

Available formats:

- turtle (default)
- xml
- json-ld

Parameters

- **format** (*str*) – a valid format.
- **encoding** (*str*) – the encoding to serialize into
- **include_distributions** (*bool*) – includes the distributions in the graph

Return type

`Union[bytes, str]`

Returns

a rdf serialization as a bytes literal according to format.

property type_genre: `str`

A link to the nature or genre of the resource.

Type

`URI`

Return type

`str`

property was_generated_by: `str`

A link to an activity that generated, or provides the business context for, the creation of the dataset.

Type

`URI`

Return type

`str`

2.7 datacatalogtordf.dataset_series

DatasetSeries module for mapping a dataset_series to rdf.

This module contains methods for mapping a dataset_series object to rdf according to the `dcap-ap-no v.2` standard

Example

```
>>> from datacatalogtordf import Catalog, DatasetSeries, Dataset
>>>
>>> dataset_series = DatasetSeries("http://example.com/dataset_series/1")
>>> dataset_series.title = {"en": "Title of dataset_series"}
>>>
>>> catalog = Catalog("http://example.com/catalog/1")
>>> catalog.datasets.append(dataset_series)
>>>
>>> first_dataset = Dataset()
>>> first_dataset.identifier = "http://example.com/datasets/1"
>>> first_dataset.in_series = dataset_series
>>>
>>> second_dataset = Dataset()
>>> second_dataset.identifier = "http://example.com/datasets/2"
>>> second_dataset.in_series = dataset_series
>>> second_dataset.prev = first_dataset
>>>
>>> dataset_series.first = first_dataset
>>> dataset_series.last = second_dataset
>>>
>>> bool(catalog.to_rdf())
True
```

class datacatalogtordf.dataset_series.DatasetSeries(*identifier=None*)

Bases: `Dataset`

A class representing a dcat:DatasetSeries.

Ref: `dcap:DatasetSeries`.

Parameters

identifier (*URI*) – the identifier of the dataset-series.

property access_rights: *str*

A link to information about who can access the resource or an indication of its security status.

Type

URI

Return type

str

property access_rights_comments: *List[str]*

Referanse til hjemmel (kilde for påstand) i offentlighetsloven, sikkerhetsloven, beskyttelsesinstruksen eller annet lovverk som ligger til grunn for vurdering av tilgangsnivå.

Type

List[URI]

Return type

List[str]

property conforms_to: *List[str]*

A list of links to established standards to which the described resource conforms.

Type

List[URI]

Return type

List[str]

property contactpoint: *Contact*

Relevant contact information for the cataloged resource.

Type

Contact

Return type

Contact

property creator: *str*

Link to the entity responsible for producing the resource.

Type

URI

Return type

str

property dct_identifier: *str*

the identifier for the dataset.

Type

str

Return type

str

property description: *Dict[str, str]*

A free-text account of the item. key is language code.

Type`Dict[str, str]`**Return type**`Dict[str, str]`**property distributions:** `List[Distribution]`

A list of distributions of the dataset.

Type`List[Distribution]`**Return type**`List[Distribution]`**property first:** `Dataset`

The first resource in an ordered collection or series of resources, to which the current resource belongs.

Type`Dataset`**Return type**`Dataset`**property frequency:** `str`

A link to resource describing the frequency at which dataset is published.

Type`[URI]`**Return type**`str`**classmethod from_json(json)**

Convert a JSON (dict).

Parameters**json** (`Dict`) – A dict representing this class.**Returns**

The object.

Return type`Resource`**property has_policy:** `str`

A link to an ODRL conformant policy expressing the rights associated with the resource.

Type`URI`**Return type**`str`**property identifier:** `str`

A URI uniquely identifying the resource.

Type`URI`**Return type**`str`

property in_series: *DatasetSeries*

A dataset series of which the dataset is part.

Type

DatasetSeries

Return type

DatasetSeries

property is_referenced_by: *List[Resource]*

A list of related resources, such as a publication, that references, cites, or otherwise points to the cataloged resource.

Type

List[Resource]

Return type

List[Resource]

property keyword: *Dict[str, str]*

A keyword or tag describing the resource. key is language code.

Type

Dict[str, str]

Return type

Dict[str, str]

property landing_page: *List[str]*

A list of links to web pages that can be navigated to in a Web browser to gain access to the catalog, a dataset, its distributions and/or additional information.

Type

List[URI]

Return type

List[str]

property language: *List[str]*

A list of links to languages of the item.

Type

List[str]

Return type

List[str]

property last: *Dataset*

The last resource in an ordered collection or series of resources, to which the current resource belongs.

Type

Dataset

Return type

Dataset

property license: *str*

A link to a legal document under which the resource is made available.

Type

URI

Return type*str***property modification_date:** *str*

Most recent date on which the item was changed, updated or modified.

Type*Date***Return type***str***property prev:** *Resource*

The previous resource in an ordered collection or series of resources.

Type*Resource***Return type***Resource***property publisher:** *Union[Agent, str]*

A URI uniquely identifying the publisher of the resource.

Type*Union[Agent, str]***Return type***Union[Agent, str]***property qualified_attributions:** *List[Dict]*

List of links to an Agent having some form of responsibility for the resource.

Type*List[Dict]***Return type***List[Dict]***property qualified_relation:** *List[Relationship]*

A list of links to a description of a relationship with another resource.

Type*List[Relationship]***Return type***List[Relationship]***property release_date:** *str*

Date of formal issuance (e.g., publication) of the item.

Type*Date***Return type***str***property resource_relation:** *List[str]*

A list of links to resources with an unspecified relationship to the cataloged item.

Type*List[URI]*

Return type`List[str]`**property rights:** `str`

license or dct:accessRights, such as copyright statements.

Returns

The link

Return type`str`**Type**`URI`**Type**

A link to a statement that concerns all rights not addressed with dct

property spatial: `List[Union[Location, str]]`

A list of geographical areas covered by the dataset.

Type`List[Location]`**Return type**`List[Union[Location, str]]`**property spatial_resolution_in_meters:** `List[Decimal]`

A list of minimum spatial separation resolvables in a dataset, measured in meters.

Type`List[Decimal]`**Return type**`List[Decimal]`**property temporal:** `List[PeriodOfTime]`

A list of temporal periods that the dataset covers.

Type`List[PeriodOfTime]`**Return type**`List[PeriodOfTime]`**property temporal_resolution:** `List[str]`

A list of minimum time period resolvables in the dataset.

Type`List[str]`**Return type**`List[str]`**property theme:** `List[str]`

A list of links to categories of the resource.

Type`List[URI]`**Return type**`List[str]`

property title: `Dict[str, str]`

A name given to the item. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

to_json()

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type

`Dict`

to_rdf(*format='turtle', encoding='utf-8', include_datasets=True, include_services=True, include_models=True, include_contains_services=True*)

Maps the catalog to rdf.

Available formats:

- turtle (default)
- xml
- json-ld

Parameters

- **format** (*str*) – a valid format.
- **encoding** (*str*) – the encoding to serialize into
- **include_datasets** (*bool*) – includes the dataset graphs in the catalog
- **include_services** (*bool*) – includes the services in the catalog
- **include_models** (*bool*) – includes the models in the catalog
- **include_contains_services** (*bool*) – includes the services (cpsvno) in the catalog

Return type

`Union[bytes, str]`

Returns

a rdf serialization as a bytes literal according to format.

property type_genre: `str`

A link to the nature or genre of the resource.

Type

URI

Return type

`str`

property was_generated_by: `str`

A link to an activity that generated, or provides the business context for, the creation of the dataset.

Type

URI

Return type`str`

2.8 datacatalogtordf.distribution

Distribution module for mapping a distribution to rdf.

This module contains methods for mapping a distribution object to rdf according to the `dcap-ap-no v.2` standard

Example

```
>>> from datacatalogtordf import Distribution
>>>
>>> distribution = Distribution()
>>> distribution.identifier = "http://example.com/dataservices/1"
>>> distribution.title = {"en": "Title of distribution"}
>>>
>>> bool(distribution.to_rdf())
True
```

class `datacatalogtordf.distribution.Distribution(identifier=None)`

Bases: `object`

A class representing a `dcap:Distribution`.

Ref: `dcap:Distribution`

Parameters

identifier (`URI`) – the identifier of the dataset-series.

property access_URL: `str`

A URL of the resource that gives access to a distribution of the dataset. E.g. landing page, feed, SPARQL endpoint.

Type

`URI`

Return type

`str`

property access_rights: `str`

A link to rights statement that concerns how the distribution is accessed.

Type

`URI`

Return type

`str`

property access_service: `DataService`

A data service that gives access to the distribution of the dataset.

Type

`DataService`

Return type

`DataService`

property byte_size: `Decimal`

The size of a distribution in bytes.

Type

`Decimal`

Return type

`Decimal`

property compression_format: `str`

Link to the compression format of the distribution in which the data is contained in a compressed form, e.g. to reduce the size of the downloadable file.

Type

`URI`

Return type

`str`

property conforms_to: `List[str]`

A list of links to established standards to which the distribution conforms.

Type

`List[URI]`

Return type

`List[str]`

property description: `Dict[str, str]`

A free-text account of the distribution.

Type

`Dict[str]`

Return type

`Dict[str, str]`

property download_URL: `str`

format and/or dcat:mediaType.

Type

`URI`

Type

The URL of the downloadable file in a given format. E.g. CSV file or RDF file. The format is indicated by the distribution's dct

Return type

`str`

property formats: `List[str]`

A list of file formats of the distribution.

Type

`List[URI]`

Return type

`List[str]`

classmethod from_json(json)

Convert a JSON (dict).

Parameters

json (*Dict*) – A dict representing this class.

Returns

The object.

Return type

Distribution

property has_policy: *str*

A link to an ODRL conformant policy expressing the rights associated with the distribution.

Type

URI

Return type

str

property identifier: *str*

A URI uniquely identifying the resource.

Type

URI

Return type

str

property license: *str*

A link to legal document under which the distribution is made available.

Type

URI

Return type

str

property media_types: *List[str]*

A list of media types of the distribution as defined by IANA.

Type

List[URI]

Return type

List[str]

property modification_date: *str*

Most recent date on which the distribution was changed, updated or modified.

Type

Date

Return type

str

property package_format: *str*

Link to the package format of the distribution in which one or more data files are grouped together, e.g. to enable a set of related files to be downloaded together.

Type

URI

Return type

str

property release_date: `str`

Date of formal issuance (e.g., publication) of the distribution.

Type

`Date`

Return type

`str`

property rights: `str`

A link to information about rights held in and over the distribution.

Type

`URI`

Return type

`str`

property spatial_resolution_in_meters: `List[Decimal]`

A list of minimum spatial separation resolvables in a dataset distribution, measured in meters.

Type

`List[Decimal]`

Return type

`List[Decimal]`

property temporal_resolution: `List[str]`

A list of minimum time period resolvables in the dataset distribution.

Type

`List[str]`

Return type

`List[str]`

property title: `Dict[str, str]`

A dict with title in multiple languages.

Type

`Dict[str]`

Return type

`Dict[str, str]`

to_json()

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type

`Dict`

to_rdf(*format='turtle', encoding='utf-8'*)

Maps the distribution to rdf.

Return type

`Union[bytes, str]`

2.9 datacatalogtordf.uri

URI helper module for very basic validation of a uri.

class datacatalogtordf.uri.**URI**(link)

Bases: `str`

A helper class to validate a URI.

If the string is serializable as an `rdflib.URIRef`, it is valid. Otherwise not.

Essentially it is serializable if none of the following chars occurs:

```
_invalid_uri_chars = '<>" {}|\\^`'
```

Parameters

link (`str`) – The string to validate as URI.

Raises

InvalidURIError – If the string is not a valid URI.

Ref: <https://github.com/RDFLib/rdflib/blob/master/rdflib/term.py#L75>

Example

```
>>> from datacatalogtordf import Dataset, URI
>>>
>>> dataset = Dataset()
>>> dataset.identifier = URI("http://example.com/datasets/1")
>>> dataset.identifier
'http://example.com/datasets/1'
```

2.10 datacatalogtordf.periodoftime

PeriodOfTime module for mapping a `period_of_time` to rdf.

This module contains methods for mapping a `period_of_time` object to rdf according to the [dcat-ap-no v.2 standard](#)

Example

```
>>> from datacatalogtordf import PeriodOfTime
>>>
>>> period_of_time = PeriodOfTime()
>>> period_of_time.start_date = "2019-12-31"
>>> period_of_time.end_date = "2020-12-31"
>>>
>>> bool(period_of_time.to_rdf())
True
```

class datacatalogtordf.periodoftime.Date(*link*)

Bases: `str`

A helper class to validate a Date.

If the string is a valid of the format “%Y-%m-%d”, it is valid. Otherwise not.

Raises

InvalidDateError – If the str does not represent a valid date

class datacatalogtordf.periodoftime.PeriodOfTime

Bases: `object`

A class representing a dcat:PeriodOfTime.

Ref: `dcat:PeriodOfTime`

Raises

InvalidDateIntervalError – If the interval is “negative”, i.e. start date is after the end date

property end_date: `str`

date signfying the end of the period.

Type

`str`

Return type

`str`

classmethod from_json(*json*)

Convert a JSON (dict).

Parameters

json (`Dict`) – A dict representing this class.

Returns

The object.

Return type

PeriodOfTime

property start_date: `str`

date signfying the start of the period.

Type

`str`

Return type

`str`

to_json()

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type

`Dict`

to_rdf(*format='turtle', encoding='utf-8'*)

Maps the period_of_time to rdf.

Return type

`Union[bytes, str]`

2.11 datacatalogtordf.location

Location module for mapping a location to rdf.

This module contains methods for mapping a location object to rdf according to the [dcat-ap-no v.2 standard](#)

Example

```
>>> from datacatalogtordf import Location
>>>
>>> location = Location()
>>> location.identifier = "http://example.com/relations/1"
>>> location.centroid = "POINT(4.88412 52.37509)"
>>>
>>> bool(location.to_rdf())
True
```

class datacatalogtordf.location.Location(*identifier=None*)

Bases: `object`

A class representing a dcat:Location.

Ref: `dcat:Location`

property bounding_box: `str`

The geographic bounding box of a resource.

Type

`str`

Return type

`str`

property centroid: `str`

The geographic center (centroid) of a resource.

Type

`str`

Return type

`str`

classmethod from_json(*json*)

Convert a JSON (dict).

Parameters

json (`Dict`) – A dict representing this class.

Returns

The object.

Return type

`Location`

property geometry: `str`

Associates any resource with the corresponding geometry.

Type

`str`

Return type`str`**property identifier:** `str`

an URI uniquely identifying the resource.

Type`URI`**Return type**`str`**to_json()**

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type`Dict`**to_rdf**(*format='turtle', encoding='utf-8'*)

Maps the location to rdf.

Parameters

- **format** (`str`) – a valid format. Default: turtle
- **encoding** (`Optional[str]`) – the encoding to serialize into

Return type`Union[bytes, str]`**Returns**

a rdf serialization as a bytes literal according to format.

2.12 datacatalogtordf.agent

Agent module for mapping a agent to rdf.

This module contains methods for mapping a agent object to rdf according to the `dcap-ap-no v.2` standard

Example

```
>>> from datacatalogtordf import Agent, Dataset
>>> dataset = Dataset()
>>> dataset.identifier = "http://example.com/datasets/1"
>>> # Create an agent:
>>> agent = Agent()
>>> agent.identifier = "http://example.com/agents/1"
>>> agent.name = {"en": "James Bond", "nb": "Djeims Bånd"}
>>> # Assign the agent to the publisher property:
>>> dataset.publisher = agent
>>> bool(dataset.to_rdf())
True
```

class datacatalogtordf.agent.**Agent**(*identifier=None*)

Bases: `object`

A class representing a foaf:Agent.

Parameters

identifier (`URI`) – the identifier of the dataset.

classmethod **from_json**(*json*)

Convert a JSON (dict).

Parameters

json (`Dict`) – A dict representing this class

Returns

The object

Return type

`Agent`

property **identifier**: `str`

A URI uniquely identifying the agent.

Type

`URI`

Return type

`str`

property **name**: `Dict[str, str]`

A name given to the agent. key is language code.

Type

`Dict[str, str]`

Return type

`Dict[str, str]`

property **organization_id**: `str`

The organization's identifier.

Type

`str`

Return type

`str`

property **organization_type**: `str`

Link to a concept designating the type of the agent.

Type

`URI`

Return type

`str`

property **same_as**: `str`

Link to another resource that is the same as this one.

Type

`URI`

Return type`str`**to_json()**

Convert the Resource to a json / dict. It will omit the non-initialized fields.

Returns

The json representation of this instance.

Return type`Dict`**to_rdf(*format='turtle', encoding='utf-8'*)**

Maps the agent to rdf.

Parameters

- **format** (`str`) – a valid format. Default: turtle
- **encoding** (`Optional[str]`) – the encoding to serialize into

Return type`Union[bytes, str]`**Returns**

a rdf serialization as a bytes literal according to format.

2.13 datacatalogtordf.exceptions

Exceptions module for datacatalogtordf.

exception datacatalogtordf.exceptions.**Error**(*msg=None*)

Bases: `Exception`

Base class for exceptions.

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception datacatalogtordf.exceptions.**InvalidDateError**(*date, msg*)

Bases: `Error`

Exception raised for errors in the input.

str -- input str in which the error occurred

message -- explanation of the error

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

exception datacatalogtordf.exceptions.**InvalidDateIntervalError**(*start_date, end_date, msg*)

Bases: `Error`

Exception raised for errors in the input.

str -- input str in which the error occurred

message -- explanation of the error

with_traceback()

Exception.with_traceback(tb) – set self.__traceback__ to tb and return self.

A small Python library for mapping a data catalog to rdf

The library contains helper classes for the following dcat classes:

- [Catalog](#)
- [Dataset](#)
- [Distribution](#)
- [Data Service](#)

Other relevant classes are also supported, such as:

- [Contact](#) ([vcard:Kind](#))

The library will map to the [Norwegian Application Profile of the DCAT standard](#).

INSTALLATION

To install the `datacatalogtordf` package, run this command in your terminal:

```
$ pip install datacatalogtordf
```


USAGE

This package can be used like this:

```
from datacatalogtordf import Catalog, Dataset

# Create catalog object
catalog = Catalog()
catalog.identifier = "http://example.com/catalogs/1"
catalog.title = {"en": "A dataset catalog"}
catalog.publisher = "https://example.com/publishers/1"

# Create a dataset:
dataset = Dataset()
dataset.identifier = "http://example.com/datasets/1"
dataset.title = {"nb": "inntektsAPI", "en": "incomeAPI"}
#
# Add concept to catalog:
catalog.datasets.append(dataset)

# get rdf representation in turtle (default)
rdf = catalog.to_rdf(format="turtle")
print(rdf.decode())
```

Will produce the following output:

```
@prefix dcat: <http://www.w3.org/ns/dcat#> .
@prefix dct: <http://purl.org/dc/terms/> .

<http://example.com/catalogs/1> a dcat:Catalog ;
    dct:publisher <https://example.com/publishers/1> ;
    dct:title "A dataset catalog"@en ;
    dcat:dataset <http://example.com/datasets/1> .

<http://example.com/datasets/1> a dcat:Dataset ;
    dct:title "incomeAPI"@en,
        "inntekstAPI"@nb .
```


PYTHON MODULE INDEX

d

- `datacatalogtordf.agent`, [47](#)
- `datacatalogtordf.catalog`, [12](#)
- `datacatalogtordf.catalogrecord`, [19](#)
- `datacatalogtordf.dataservice`, [21](#)
- `datacatalogtordf.dataset`, [27](#)
- `datacatalogtordf.dataset_series`, [33](#)
- `datacatalogtordf.distribution`, [40](#)
- `datacatalogtordf.exceptions`, [49](#)
- `datacatalogtordf.location`, [46](#)
- `datacatalogtordf.periodoftime`, [44](#)
- `datacatalogtordf.relationship`, [10](#)
- `datacatalogtordf.resource`, [5](#)
- `datacatalogtordf.uri`, [44](#)

A

`access_rights` (`datacatalogtordf.catalog.Catalog` property), 12

`access_rights` (`datacatalogtordf.dataservice.DataService` property), 21

`access_rights` (`datacatalogtordf.dataset.Dataset` property), 27

`access_rights` (`datacatalogtordf.dataset_series.DatasetSeries` property), 34

`access_rights` (`datacatalogtordf.distribution.Distribution` property), 40

`access_rights` (`datacatalogtordf.resource.Resource` property), 5

`access_rights_comments` (`datacatalogtordf.catalog.Catalog` property), 12

`access_rights_comments` (`datacatalogtordf.dataset.Dataset` property), 27

`access_rights_comments` (`datacatalogtordf.dataset_series.DatasetSeries` property), 34

`access_service` (`datacatalogtordf.distribution.Distribution` property), 40

`access_URL` (`datacatalogtordf.distribution.Distribution` property), 40

`Agent` (class in `datacatalogtordf.agent`), 47

B

`bounding_box` (`datacatalogtordf.location.Location` property), 46

`byte_size` (`datacatalogtordf.distribution.Distribution` property), 40

C

`Catalog` (class in `datacatalogtordf.catalog`), 12

`CatalogRecord` (class in `datacatalogtordf.catalogrecord`), 19

`catalogrecords` (`datacatalogtordf.catalog.Catalog` property), 12

`catalogs` (`datacatalogtordf.catalog.Catalog` property), 12

`centroid` (`datacatalogtordf.location.Location` property), 46

`compression_format` (`datacatalogtordf.distribution.Distribution` property), 41

`conforms_to` (`datacatalogtordf.catalog.Catalog` property), 13

`conforms_to` (`datacatalogtordf.catalogrecord.CatalogRecord` property), 19

`conforms_to` (`datacatalogtordf.dataservice.DataService` property), 22

`conforms_to` (`datacatalogtordf.dataset.Dataset` property), 27

`conforms_to` (`datacatalogtordf.dataset_series.DatasetSeries` property), 34

`conforms_to` (`datacatalogtordf.distribution.Distribution` property), 41

`conforms_to` (`datacatalogtordf.resource.Resource` property), 6

`contactpoint` (`datacatalogtordf.catalog.Catalog` property), 13

`contactpoint` (`datacatalogtordf.dataservice.DataService` property), 22

`contactpoint` (`datacatalogtordf.dataset.Dataset` property), 28

`contactpoint` (`datacatalogtordf.dataset_series.DatasetSeries` property), 34

`contactpoint` (`datacatalogtordf.resource.Resource` property), 6

`creator` (`datacatalogtordf.catalog.Catalog` property), 13

`creator` (`datacatalogtordf.dataservice.DataService` property), 22

`creator` (`datacatalogtordf.dataset.Dataset` property), 28

`creator` (`datacatalogtordf.dataset_series.DatasetSeries` property), 34

property), 34
 creator (datacatalogtordf.resource.Resource property),
 6

D

datacatalogtordf.agent
 module, 47
 datacatalogtordf.catalog
 module, 12
 datacatalogtordf.catalogrecord
 module, 19
 datacatalogtordf.dataservice
 module, 21
 datacatalogtordf.dataset
 module, 27
 datacatalogtordf.dataset_series
 module, 33
 datacatalogtordf.distribution
 module, 40
 datacatalogtordf.exceptions
 module, 49
 datacatalogtordf.location
 module, 46
 datacatalogtordf.periodoftime
 module, 44
 datacatalogtordf.relationship
 module, 10
 datacatalogtordf.resource
 module, 5
 datacatalogtordf.uri
 module, 44
 DataService (class in datacatalogtordf.dataservice), 21
 Dataset (class in datacatalogtordf.dataset), 27
 datasets (datacatalogtordf.catalog.Catalog property),
 13
 DatasetSeries (class in datacatalog-
 tordf.dataset_series), 33
 Date (class in datacatalogtordf.periodoftime), 44
 dct_identifier (datacatalogtordf.catalog.Catalog
 property), 13
 dct_identifier (datacatalogtordf.dataset.Dataset
 property), 28
 dct_identifier (datacatalog-
 tordf.dataset_series.DatasetSeries property),
 34
 description (datacatalogtordf.catalog.Catalog prop-
 erty), 13
 description (datacatalog-
 tordf.catalogrecord.CatalogRecord property),
 19
 description (datacatalog-
 tordf.dataservice.DataService property),
 22

description (datacatalogtordf.dataset.Dataset prop-
 erty), 28
 description (datacatalog-
 tordf.dataset_series.DatasetSeries property),
 34
 description (datacatalog-
 tordf.distribution.Distribution property),
 41
 description (datacatalogtordf.resource.Resource prop-
 erty), 6
 Distribution (class in datacatalogtordf.distribution),
 40
 distributions (datacatalogtordf.catalog.Catalog
 property), 13
 distributions (datacatalogtordf.dataset.Dataset prop-
 erty), 28
 distributions (datacatalog-
 tordf.dataset_series.DatasetSeries property),
 35
 download_URL (datacatalog-
 tordf.distribution.Distribution property),
 41

E

end_date (datacatalogtordf.periodoftime.PeriodOfTime
 property), 45
 endpointDescription (datacatalog-
 tordf.dataservice.DataService property),
 22
 endpointURL (datacatalog-
 tordf.dataservice.DataService property),
 22
 Error, 49

F

first (datacatalogtordf.dataset_series.DatasetSeries
 property), 35
 formats (datacatalogtordf.distribution.Distribution
 property), 41
 frequency (datacatalogtordf.catalog.Catalog property),
 14
 frequency (datacatalogtordf.dataset.Dataset property),
 28
 frequency (datacatalog-
 tordf.dataset_series.DatasetSeries property),
 35
 from_json() (datacatalogtordf.agent.Agent class
 method), 48
 from_json() (datacatalogtordf.catalog.Catalog class
 method), 14
 from_json() (datacatalog-
 tordf.catalogrecord.CatalogRecord class
 method), 20

- `from_json()` (*datacatalogtordf.dataservice.DataService* class method), 22
- `from_json()` (*datacatalogtordf.dataset.Dataset* class method), 28
- `from_json()` (*datacatalogtordf.dataset_series.DatasetSeries* class method), 35
- `from_json()` (*datacatalogtordf.distribution.Distribution* class method), 41
- `from_json()` (*datacatalogtordf.location.Location* class method), 46
- `from_json()` (*datacatalogtordf.periodoftime.PeriodOfTime* class method), 45
- `from_json()` (*datacatalogtordf.relationship.Relationship* class method), 10
- `from_json()` (*datacatalogtordf.resource.Resource* class method), 6
- ## G
- `geometry` (*datacatalogtordf.location.Location* property), 46

H

`had_role` (*datacatalogtordf.relationship.Relationship* property), 10

`has_parts` (*datacatalogtordf.catalog.Catalog* property), 14

`has_policy` (*datacatalogtordf.catalog.Catalog* property), 14

`has_policy` (*datacatalogtordf.dataservice.DataService* property), 23

`has_policy` (*datacatalogtordf.dataset.Dataset* property), 29

`has_policy` (*datacatalogtordf.dataset_series.DatasetSeries* property), 35

`has_policy` (*datacatalogtordf.distribution.Distribution* property), 42

`has_policy` (*datacatalogtordf.resource.Resource* property), 6

`homepage` (*datacatalogtordf.catalog.Catalog* property), 14

I

`identifier` (*datacatalogtordf.agent.Agent* property), 48

`identifier` (*datacatalogtordf.catalog.Catalog* property), 14

`identifier` (*datacatalogtordf.catalogrecord.CatalogRecord* property), 20

`identifier` (*datacatalogtordf.dataservice.DataService* property), 23

`identifier` (*datacatalogtordf.dataset.Dataset* property), 29

`identifier` (*datacatalogtordf.dataset_series.DatasetSeries* property), 35

`identifier` (*datacatalogtordf.distribution.Distribution* property), 42

`identifier` (*datacatalogtordf.location.Location* property), 47

`identifier` (*datacatalogtordf.relationship.Relationship* property), 11

`identifier` (*datacatalogtordf.resource.Resource* property), 6

`in_series` (*datacatalogtordf.catalog.Catalog* property), 15

`in_series` (*datacatalogtordf.dataset.Dataset* property), 29

`in_series` (*datacatalogtordf.dataset_series.DatasetSeries* property), 35

`InvalidDateError`, 49

`InvalidDateIntervalError`, 49

`is_referenced_by` (*datacatalogtordf.catalog.Catalog* property), 15

`is_referenced_by` (*datacatalogtordf.dataservice.DataService* property), 23

`is_referenced_by` (*datacatalogtordf.dataset.Dataset* property), 29

`is_referenced_by` (*datacatalogtordf.dataset_series.DatasetSeries* property), 36

`is_referenced_by` (*datacatalogtordf.resource.Resource* property), 7

K

`keyword` (*datacatalogtordf.catalog.Catalog* property), 15

`keyword` (*datacatalogtordf.dataservice.DataService* property), 23

`keyword` (*datacatalogtordf.dataset.Dataset* property), 29

`keyword` (*datacatalogtordf.dataset_series.DatasetSeries* property), 36

`keyword` (*datacatalogtordf.resource.Resource* property), 7

L

`landing_page` (*datacatalogtordf.catalog.Catalog* property), 15

`landing_page` (*datacatalogtordf.dataservice.DataService* property), 23

landing_page (*datacatalogtordf.dataset.Dataset* property), 29

landing_page (*datacatalogtordf.dataset_series.DatasetSeries* property), 36

landing_page (*datacatalogtordf.resource.Resource* property), 7

language (*datacatalogtordf.catalog.Catalog* property), 15

language (*datacatalogtordf.dataservice.DataService* property), 23

language (*datacatalogtordf.dataset.Dataset* property), 29

language (*datacatalogtordf.dataset_series.DatasetSeries* property), 36

language (*datacatalogtordf.resource.Resource* property), 7

last (*datacatalogtordf.dataset_series.DatasetSeries* property), 36

license (*datacatalogtordf.catalog.Catalog* property), 15

license (*datacatalogtordf.dataservice.DataService* property), 23

license (*datacatalogtordf.dataset.Dataset* property), 30

license (*datacatalogtordf.dataset_series.DatasetSeries* property), 36

license (*datacatalogtordf.distribution.Distribution* property), 42

license (*datacatalogtordf.resource.Resource* property), 7

listing_date (*datacatalogtordf.catalogrecord.CatalogRecord* property), 20

Location (class in *datacatalogtordf.location*), 46

M

media_types (*datacatalogtordf.dataservice.DataService* property), 24

media_types (*datacatalogtordf.distribution.Distribution* property), 42

modification_date (*datacatalogtordf.catalog.Catalog* property), 15

modification_date (*datacatalogtordf.catalogrecord.CatalogRecord* property), 20

modification_date (*datacatalogtordf.dataservice.DataService* property), 24

modification_date (*datacatalogtordf.dataset.Dataset* property), 30

modification_date (*datacatalogtordf.dataset_series.DatasetSeries* property),

37

modification_date (*datacatalogtordf.distribution.Distribution* property), 42

modification_date (*datacatalogtordf.resource.Resource* property), 7

module

- datacatalogtordf.agent*, 47
- datacatalogtordf.catalog*, 12
- datacatalogtordf.catalogrecord*, 19
- datacatalogtordf.dataservice*, 21
- datacatalogtordf.dataset*, 27
- datacatalogtordf.dataset_series*, 33
- datacatalogtordf.distribution*, 40
- datacatalogtordf.exceptions*, 49
- datacatalogtordf.location*, 46
- datacatalogtordf.periodoftime*, 44
- datacatalogtordf.relationship*, 10
- datacatalogtordf.resource*, 5
- datacatalogtordf.uri*, 44

N

name (*datacatalogtordf.agent.Agent* property), 48

O

organization_id (*datacatalogtordf.agent.Agent* property), 48

organization_type (*datacatalogtordf.agent.Agent* property), 48

P

package_format (*datacatalogtordf.distribution.Distribution* property), 42

PeriodOfTime (class in *datacatalogtordf.periodoftime*), 45

prev (*datacatalogtordf.catalog.Catalog* property), 16

prev (*datacatalogtordf.dataservice.DataService* property), 24

prev (*datacatalogtordf.dataset.Dataset* property), 30

prev (*datacatalogtordf.dataset_series.DatasetSeries* property), 37

prev (*datacatalogtordf.resource.Resource* property), 8

primary_topic (*datacatalogtordf.catalogrecord.CatalogRecord* property), 20

publisher (*datacatalogtordf.catalog.Catalog* property), 16

publisher (*datacatalogtordf.dataservice.DataService* property), 24

publisher (*datacatalogtordf.dataset.Dataset* property), 30

`publisher` (*datacatalogtordf.dataset_series.DatasetSeries* property), 37
`publisher` (*datacatalogtordf.resource.Resource* property), 8

Q

`qualified_attributions` (*datacatalogtordf.catalog.Catalog* property), 16
`qualified_attributions` (*datacatalogtordf.dataservice.DataService* property), 24
`qualified_attributions` (*datacatalogtordf.dataset.Dataset* property), 30
`qualified_attributions` (*datacatalogtordf.dataset_series.DatasetSeries* property), 37
`qualified_attributions` (*datacatalogtordf.resource.Resource* property), 8
`qualified_relation` (*datacatalogtordf.catalog.Catalog* property), 16
`qualified_relation` (*datacatalogtordf.dataservice.DataService* property), 24
`qualified_relation` (*datacatalogtordf.dataset.Dataset* property), 30
`qualified_relation` (*datacatalogtordf.dataset_series.DatasetSeries* property), 37
`qualified_relation` (*datacatalogtordf.resource.Resource* property), 8

R

`relation` (*datacatalogtordf.relationship.Relationship* property), 11
`Relationship` (class in *datacatalogtordf.relationship*), 10
`release_date` (*datacatalogtordf.catalog.Catalog* property), 16
`release_date` (*datacatalogtordf.dataservice.DataService* property), 24
`release_date` (*datacatalogtordf.dataset.Dataset* property), 30
`release_date` (*datacatalogtordf.dataset_series.DatasetSeries* property), 37
`release_date` (*datacatalogtordf.distribution.Distribution* property), 42
`release_date` (*datacatalogtordf.resource.Resource* property), 8
`Resource` (class in *datacatalogtordf.resource*), 5

`resource_relation` (*datacatalogtordf.catalog.Catalog* property), 16
`resource_relation` (*datacatalogtordf.dataservice.DataService* property), 25
`resource_relation` (*datacatalogtordf.dataset.Dataset* property), 31
`resource_relation` (*datacatalogtordf.dataset_series.DatasetSeries* property), 37
`resource_relation` (*datacatalogtordf.resource.Resource* property), 8
`rights` (*datacatalogtordf.catalog.Catalog* property), 16
`rights` (*datacatalogtordf.dataservice.DataService* property), 25
`rights` (*datacatalogtordf.dataset.Dataset* property), 31
`rights` (*datacatalogtordf.dataset_series.DatasetSeries* property), 38
`rights` (*datacatalogtordf.distribution.Distribution* property), 43
`rights` (*datacatalogtordf.resource.Resource* property), 8

S

`same_as` (*datacatalogtordf.agent.Agent* property), 48
`servesdatasets` (*datacatalogtordf.dataservice.DataService* property), 25
`services` (*datacatalogtordf.catalog.Catalog* property), 17
`spatial` (*datacatalogtordf.catalog.Catalog* property), 17
`spatial` (*datacatalogtordf.dataset.Dataset* property), 31
`spatial` (*datacatalogtordf.dataset_series.DatasetSeries* property), 38
`spatial_resolution_in_meters` (*datacatalogtordf.catalog.Catalog* property), 17
`spatial_resolution_in_meters` (*datacatalogtordf.dataset.Dataset* property), 31
`spatial_resolution_in_meters` (*datacatalogtordf.dataset_series.DatasetSeries* property), 38
`spatial_resolution_in_meters` (*datacatalogtordf.distribution.Distribution* property), 43
`start_date` (*datacatalogtordf.periodoftime.PeriodOfTime* property), 45

T

`temporal` (*datacatalogtordf.catalog.Catalog* property), 17
`temporal` (*datacatalogtordf.dataset.Dataset* property), 31
`temporal` (*datacatalogtordf.dataset_series.DatasetSeries* property), 38

temporal_resolution (datacatalogtordf.catalog.Catalog property), 17

temporal_resolution (datacatalogtordf.dataset.Dataset property), 31

temporal_resolution (datacatalogtordf.dataset_series.DatasetSeries property), 38

temporal_resolution (datacatalogtordf.distribution.Distribution property), 43

theme (datacatalogtordf.catalog.Catalog property), 17

theme (datacatalogtordf.dataservice.DataService property), 25

theme (datacatalogtordf.dataset.Dataset property), 32

theme (datacatalogtordf.dataset_series.DatasetSeries property), 38

theme (datacatalogtordf.resource.Resource property), 9

themes (datacatalogtordf.catalog.Catalog property), 18

title (datacatalogtordf.catalog.Catalog property), 18

title (datacatalogtordf.catalogrecord.CatalogRecord property), 20

title (datacatalogtordf.dataservice.DataService property), 25

title (datacatalogtordf.dataset.Dataset property), 32

title (datacatalogtordf.dataset_series.DatasetSeries property), 38

title (datacatalogtordf.distribution.Distribution property), 43

title (datacatalogtordf.resource.Resource property), 9

to_json() (datacatalogtordf.agent.Agent method), 49

to_json() (datacatalogtordf.catalog.Catalog method), 18

to_json() (datacatalogtordf.catalogrecord.CatalogRecord method), 21

to_json() (datacatalogtordf.dataservice.DataService method), 25

to_json() (datacatalogtordf.dataset.Dataset method), 32

to_json() (datacatalogtordf.dataset_series.DatasetSeries method), 39

to_json() (datacatalogtordf.distribution.Distribution method), 43

to_json() (datacatalogtordf.location.Location method), 47

to_json() (datacatalogtordf.periodoftime.PeriodOfTime method), 45

to_json() (datacatalogtordf.relationship.Relationship method), 11

to_json() (datacatalogtordf.resource.Resource method), 9

to_rdf() (datacatalogtordf.agent.Agent method), 49

to_rdf() (datacatalogtordf.catalog.Catalog method), 18

to_rdf() (datacatalogtordf.catalogrecord.CatalogRecord method), 21

to_rdf() (datacatalogtordf.dataservice.DataService method), 26

to_rdf() (datacatalogtordf.dataset.Dataset method), 32

to_rdf() (datacatalogtordf.dataset_series.DatasetSeries method), 39

to_rdf() (datacatalogtordf.distribution.Distribution method), 43

to_rdf() (datacatalogtordf.location.Location method), 47

to_rdf() (datacatalogtordf.periodoftime.PeriodOfTime method), 45

to_rdf() (datacatalogtordf.relationship.Relationship method), 11

to_rdf() (datacatalogtordf.resource.Resource method), 9

type_genre (datacatalogtordf.catalog.Catalog property), 18

type_genre (datacatalogtordf.dataservice.DataService property), 26

type_genre (datacatalogtordf.dataset.Dataset property), 32

type_genre (datacatalogtordf.dataset_series.DatasetSeries property), 39

type_genre (datacatalogtordf.resource.Resource property), 10

U

URI (class in datacatalogtordf.uri), 44

W

was_generated_by (datacatalogtordf.catalog.Catalog property), 19

was_generated_by (datacatalogtordf.dataset.Dataset property), 33

was_generated_by (datacatalogtordf.dataset_series.DatasetSeries property), 39

with_traceback() (datacatalogtordf.exceptions.Error method), 49

with_traceback() (datacatalogtordf.exceptions.InvalidDateError method), 49

with_traceback() (datacatalogtordf.exceptions.InvalidDateIntervalError method), 49