

DataCite REST API

Overview

Each Collection TCIA publishes is issued a Digital Object Identifier (DOI) through [DataCite](#). The [DataCite Commons](#) is a web search interface for the [PID Graph](#), the graph formed by the collection of scholarly resources such as publications, datasets, people and research organizations, and their connections.

The [DataCite REST API](#) can be used to programmatically access Collection metadata such as their DOIs, titles and abstracts. *Please note that this API was not developed by TCIA*. See <https://support.datacite.org/> for any technical questions. The TCIA Helpdesk may be able to assist if your inquiry is related to the content of the data itself.

Official Datacite Documentation

- DataCite REST API - <https://support.datacite.org/reference/introduction>
- DataCite REST API Guide - <https://support.datacite.org/docs/api>
- DataCite Schema - <http://schema.datacite.org>

TCIA Metadata in DataCite

TCIA utilizes the following Properties of the DataCite schema.

Property ID	Property	Property Description
1	Identifier	DOI of the Dataset
2	Creator	Authors of the Dataset, preferably with ORCIDID
3	Title	Published Title of the Dataset
4	Publisher	The Cancer Imaging Archive
5	PublicationYear	The Year the Dataset was published in TCIA
10	ResourceType	Dataset; Equivalent to a TCIA Collection
11 *	AlternateIdentifier	TCIA "Short Name" for the Dataset. These short names appear in various places such as https://www.cancerimagingarchive.net/collections/ and https://www.cancerimagingarchive.net/tcia-analysis-results/
15 *	Version	The Current Version of the Dataset
16 *	Rights	Licensing Information
17 *	Description	Dataset Abstract

* indicates properties that are "Recommended and Optional" per the Datacite Schema whereas the others are required to create a DOI.

TCIA-Utills

The [tcia_utils](#) package contains functions to simplify common tasks one might perform when interacting with The Cancer Imaging Archive (TCIA) via Python. Issues with this package should be submitted at https://github.com/kirbyju/tcia_utils/issues. Installation can be achieved with this Pip command:

```
pip install tcia_utils
```

To import functions related to Datacite:

```
from tcia_utils import datacite
```

An example notebook demonstrating tcia_utils functionality with DataCite's API can be found at https://github.com/kirbyju/TCIA_Notebooks/blob/main/TCIA_DataCite_Queries.ipynb.

Example Queries

Retrieve a single DataCite record in JSON format.

For this example we are using a Published Collection called "Pseudo-PHI-DICOM-Data":

<https://api.datacite.org/doi/10.7937/s17z-r072>

Return a list of DOIs using the TCIA provider id (tcia)

<https://api.datacite.org/doi/provider-id=tcia>

By default, only 25 records are returned. You can control the number of records returned using pagination options. For example, to return only 5 records

[https://api.datacite.org/doi/provider-id=tcia&page\[size\]=5](https://api.datacite.org/doi/provider-id=tcia&page[size]=5)

or

[https://api.datacite.org/providers/tcia/doi/page\[size\]=5](https://api.datacite.org/providers/tcia/doi/page[size]=5)

Query on specific information populated in the DataCite schema

For instance, return the records published by The Cancer Imaging Archive that were published in 2016:

<https://api.datacite.org/doi?query=publisher:%22The%20Cancer%20Imaging%20Archive%22+publicationYear:2016>

or

<https://api.datacite.org/providers/tcia/doi/created=2016>

Use the "activities" endpoint to see metadata updates in JSON format for a specified DataCite record.

For this example we are using a Published Collection called "Pseudo-PHI-DICOM-Data":

<https://api.datacite.org/doi/10.7937/s17z-r072/activities>