

---

# **iatikit Documentation**

**Andy Lulham**

**Jul 31, 2020**



---

## Contents

---

<b>1</b>	<b>Contents:</b>	<b>3</b>
1.1	Getting started . . . . .	3
1.2	Usage . . . . .	4
1.3	Examples . . . . .	5
1.4	Reference . . . . .	7
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>





iatikit is a toolkit for using [IATI data](#). It includes a query language wrapper around [XPath](#), to make dealing with disparate IATI versions easier.

The name was inspired by [Open Contracting's ocdskit](#).



## 1.1 Getting started

### 1.1.1 Installation

iatikit is tested for pythons 2.7, 3.4, 3.5, 3.6 and 3.7.

You can install iatikit using pip:

```
pip install iatikit
```

If you're on Windows, we recommend using [Jupyter Notebook](#), which you can get by installing [Anaconda](#).

Once Jupyter is installed, you can run the following inside a Notebook to install iatikit:

```
import sys
!{sys.executable} -m pip install --upgrade iatikit
```

### 1.1.2 Setup

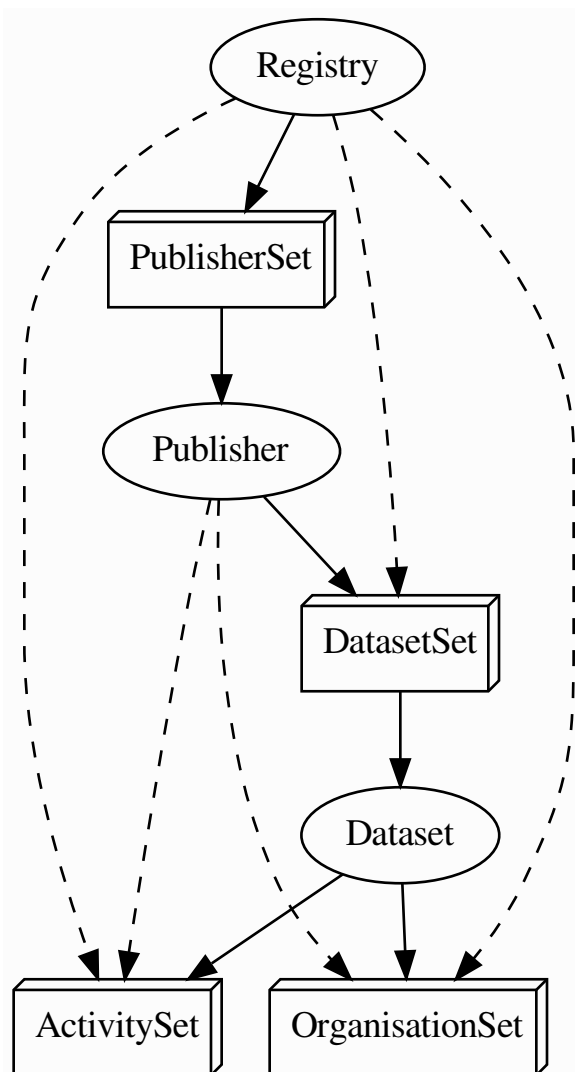
Once iatikit is installed, you'll need to fetch a recent version of all IATI data from [the registry](#), as well as [the latest codelists and schemas](#).

```
import iatikit
# download all schemas and codelists
iatikit.download.standard()
# download all XML in the registry
iatikit.download.data()
```

## 1.2 Usage

### 1.2.1 Data structure

iatikit uses a model that reflects IATI architecture.



The solid arrows show the main links between data types. The dotted arrows show additional links that iatikit provides.

The *registry* contains a list of *publishers*. Each *publisher* has zero or more *datasets*. Each *dataset* contains zero or more *activities*, or zero or more *organisations*.



## 1.2.2 Data operations

To construct a new *Registry* object, use:

```
import iatikit

registry = iatikit.data()
```

If no data can be found, a `NoDataError` is raised. If data is found to be “stale” (i.e. more than 7 days old) a warning is shown.

## 1.3 Examples

### 1.3.1 Count datasets and publishers on the registry

```
import iatikit

registry = iatikit.data()

publishers = registry.publishers
total_publishers = len(publishers)
total_datasets = sum([len(pub.datasets) for pub in publishers])
print('There are {:,} publishers and {:,} datasets on the registry'.format(
    total_publishers, total_datasets))

# There are 855 publishers and 6,682 datasets on the registry
```

### 1.3.2 Count datasets for a publisher

```
import iatikit

registry = iatikit.data()

usaid = registry.publishers.find(name='usaid')
print('USAID has {:,} datasets.'.format(len(usaid.datasets)))

# USAID has 177 datasets.
```

### 1.3.3 Find an activity by its identifier

```
import iatikit

registry = iatikit.data()
iati_identifier = 'GB-1-201724-151'

dfid = registry.publishers.find(name='dfid')
act = dfid.activities.where(
    iati_identifier=iati_identifier
).first()
```

(continues on next page)

(continued from previous page)

```
print(act)

# <Activity (GB-1-201724-151)>
```

### 1.3.4 Find activities that include an element

```
import iatikit

registry = iatikit.data()

mcc = registry.publishers.find(name='millenniumchallenge')
total_with_locations = len(mcc.activities.where(location__exists=True))
total_activities = len(mcc.activities)
print('{:,} of {:,} MCC activities have location data.'.format(
    total_with_locations, total_activities))

# 279 of 3,038 MCC activities have location data.
```

### 1.3.5 List all publishers by date of first publication

```
from datetime import datetime
import iatikit

registry = iatikit.data()

publishers = sorted(
    [(min([d.metadata.get('metadata_created')
          for d in p.datasets]
         ), p.metadata.get('title'))
     for p in registry.publishers])

for idx, tup in enumerate(publishers):
    print('{order}: {name} ({date})'.format(
        order=(idx + 1),
        name=tup[1],
        date=datetime.strptime(tup[0], '%Y-%m-%dT%H:%M:%S.%f').date())
    )

# 1: UK - Department for International Development (DFID) (2011-01-29)
# 2: The William and Flora Hewlett Foundation (2011-03-31)
# 3: The World Bank (2011-05-14)
# ...
```

### 1.3.6 More complicated activity filters

```
import iatikit

registry = iatikit.data()

dfid = registry.publishers.find(name='dfid')
```

(continues on next page)

(continued from previous page)

```
sector_category = iatikit.sector(311, 2) # Agriculture

ag_acts = dfid.activities.where(
    actual_start__lte='2017-12-31', # started before 2018
    actual_end__gte='2017-01-01', # ended after 2016
    sector__in=sector_category,
)
print('DFID had {:,} agricultural activities running during 2017.'.format(
    len(ag_acts)))

# DFID had 180 agricultural activities running during 2017.
```

## 1.4 Reference

### 1.4.1 iatikit

`iatikit.data` (*path=None*)  
 Helper function for constructing a Registry object.

### 1.4.2 Registry

**class** `iatikit.data.registry.Registry` (*path=None*)  
 Class representing the IATI registry.

**activities**  
 Return an iterator of all IATI activities on the registry.

**datasets**  
 Return an iterator of all IATI datasets on the registry.

**last\_updated**  
 Return the datetime when the local cache was last updated.

**organisations**  
 Return an iterator of all IATI organisations on the registry.

**publishers**  
 Return an iterator of all publishers on the registry.

### 1.4.3 PublisherSet

**class** `iatikit.data.publisher.PublisherSet` (*data\_path, metadata\_path, \*\*kwargs*)  
 Class representing a grouping of Publisher objects.

Objects in this grouping can be filtered and iterated over. Queries are only constructed and run when needed, so they can be efficient.

**all** ()  
 Return a list of all items in this set.

**count** ()  
 The number of items in this set.  
 Equivalent to `len(self)`.

**filter** (*\*\*kwargs*)  
Return a new set, with the filters provided in *\*\*kwargs*.  
Alias of `where (**kwargs)`.

**find** (*\*\*kwargs*)  
Return the first matching item from the set, according to the filters provided in *kwargs*.  
If no matching item is found, an `IndexError` is raised.

**first** ()  
Return the first item in this set.  
Raises an `IndexError` if the set contains zero items.  
Equivalent to `self[0]`.

**get** (*item, default=None*)  
Return an item from the set, according to the primary key.  
If no matching item is found, *default* is returned.

**where** (*\*\*kwargs*)  
Return a new set, with the filters provided in *\*\*kwargs*.

#### 1.4.4 Publisher

**class** `iatikit.data.publisher.Publisher` (*data\_path, metadata\_path, metadata\_filepath*)  
Class representing an IATI publisher.

**activities**  
Return an iterator of all activities for this publisher.

**datasets**  
Return an iterator of all datasets for this publisher.

**metadata**  
Return a dictionary of registry metadata for this publisher.

**name**  
Return the “registry name” or “shortname” of this publisher, derived from the filepath.

**organisations**  
Return an iterator of all organisations for this publisher.

**show** ()  
Open a new browser tab to the `iatiregistry.org` page for this publisher.

#### 1.4.5 DatasetSet

**class** `iatikit.data.dataset.DatasetSet` (*data\_path, metadata\_path, \*\*kwargs*)  
Class representing a grouping of `Dataset` objects.

Objects in this grouping can be filtered and iterated over. Queries are only constructed and run when needed, so they can be efficient.

**all** ()  
Return a list of all items in this set.

**count** ()  
The number of items in this set.  
Equivalent to `len(self)`.

**filter** (\*\*kwargs)  
Return a new set, with the filters provided in `**kwargs`.  
Alias of `where(**kwargs)`.

**find** (\*\*kwargs)  
Return the first matching item from the set, according to the filters provided in `kwargs`.  
If no matching item is found, an `IndexError` is raised.

**first** ()  
Return the first item in this set.  
Raises an `IndexError` if the set contains zero items.  
Equivalent to `self[0]`.

**get** (item, default=None)  
Return an item from the set, according to the primary key.  
If no matching item is found, `default` is returned.

**where** (\*\*kwargs)  
Return a new set, with the filters provided in `**kwargs`.

### 1.4.6 Dataset

**class** `iatikit.data.dataset.Dataset` (*data\_path*, *metadata\_path=None*)  
Class representing an IATI dataset.

**activities**  
Return an iterator of all activities in this dataset.

**etree**  
Return the XML of this dataset, as an `lxml` element tree.

**filetype**  
Return the filetype according to the metadata (i.e. “activity” or “organisation”).  
If it can’t be found in the metadata, revert to using the XML root node.  
Returns `None` if the filetype can’t be determined.

**metadata**  
Return a dictionary of registry metadata for this dataset.

**name**  
Return the name of this dataset, derived from the filename.

**organisations**  
Return an iterator of all organisations in this dataset.

**root**  
Return the name of the XML root node.

**schema**  
Get the XSD Schema for this dataset.

**show ()**  
Open a new browser tab to the [iatiregistry.org](http://iatiregistry.org) page for this dataset.

**validate\_codelists ()**  
Validate dataset against the relevant IATI codelists.

**validate\_iati ()**  
Validate dataset against the relevant IATI schema.

**validate\_xml ()**  
Check whether the XML in this dataset can be parsed.

**version**  
Return the IATI version according to the XML root node.  
Return “1.01” if the version can’t be determined.

**xml**  
Return the raw XML of this dataset, as a byte-string.

### 1.4.7 ActivitySet

**class** `iatikit.data.activity.ActivitySet` (*datasets*, *\*\*kwargs*)  
Class representing a grouping of `Activity` objects.

Objects in this grouping can be filtered and iterated over. Queries are only constructed and run when needed, so they can be efficient.

**all ()**  
Return a list of all items in this set.

**count ()**  
The number of items in this set.  
Equivalent to `len (self)`.

**filter (\*\*kwargs)**  
Return a new set, with the filters provided in *\*\*kwargs*.  
Alias of `where (**kwargs)`.

**find (\*\*kwargs)**  
Return the first matching item from the set, according to the filters provided in *kwargs*.  
If no matching item is found, an `IndexError` is raised.

**first ()**  
Return the first item in this set.  
Raises an `IndexError` if the set contains zero items.  
Equivalent to `self [0]`.

**get (item, default=None)**  
Return an item from the set, according to the primary key.  
If no matching item is found, `default` is returned.

**where (\*\*kwargs)**  
Return a new set, with the filters provided in *\*\*kwargs*.

### 1.4.8 Activity

**class** `iatikit.data.activity.Activity` (*etree, dataset=None, schema=None*)

Class representing an IATI activity.

**actual\_end**

Return the actual end date for this activity, as a python `date`.

**actual\_start**

Return the actual start date for this activity, as a python `date`.

**description**

Return a list of descriptions for this activity.

**end**

Return the actual end date for this activity, if present. Otherwise, return the planned end.

**humanitarian**

Return True if the humanitarian flag is set for this activity.

**iati\_identifier**

Return the iati-identifier for this activity, or None if it isn't provided.

**id**

Alias of `iati_identifier`.

**location**

Return a list of locations for this activity.

**planned\_end**

Return the planned end date for this activity, as a python `date`.

**planned\_start**

Return the planned start date for this activity, as a python `date`.

**sector**

Return a list of sectors for this activity.

**show()**

Open a new browser tab to the `d-portal.org` page for this dataset.

**start**

Return the actual start date for this activity, if present. Otherwise, return the planned start.

**title**

Return a list of titles for this activity.

**xml**

Return the raw XML of this activity, as a byte-string.

### 1.4.9 OrganisationSet

**class** `iatikit.data.organisation.OrganisationSet` (*datasets, \*\*kwargs*)

Class representing a grouping of `Organisation` objects.

Objects in this grouping can be filtered and iterated over. Queries are only constructed and run when needed, so they can be efficient.

**all()**

Return a list of all items in this set.

**count** ()  
The number of items in this set.  
Equivalent to `len(self)`.

**filter** (\*\*kwargs)  
Return a new set, with the filters provided in `**kwargs`.  
Alias of `where(**kwargs)`.

**find** (\*\*kwargs)  
Return the first matching item from the set, according to the filters provided in `kwargs`.  
If no matching item is found, an `IndexError` is raised.

**first** ()  
Return the first item in this set.  
Raises an `IndexError` if the set contains zero items.  
Equivalent to `self[0]`.

**get** (item, default=None)  
Return an item from the set, according to the primary key.  
If no matching item is found, `default` is returned.

**where** (\*\*kwargs)  
Return a new set, with the filters provided in `**kwargs`.

### 1.4.10 Organisation

**class** `iatikit.data.organisation.Organisation` (*etree, dataset=None, schema=None*)  
Class representing an IATI organisation.

**id**  
Alias of `org_identifier`.

**org\_identifier**  
Return the org-identifier for this organisation, or `None` if it isn't provided.

**show** ()  
Open a new browser tab to the `d-portal.org` page for this organisation.

**xml**  
Return the raw XML of this organisation, as a byte-string.



i

iatikit, 7



## A

activities (*iatikit.data.dataset.Dataset* attribute), 9  
 activities (*iatikit.data.publisher.Publisher* attribute), 8  
 activities (*iatikit.data.registry.Registry* attribute), 7  
 Activity (*class in iatikit.data.activity*), 11  
 ActivitySet (*class in iatikit.data.activity*), 10  
 actual\_end (*iatikit.data.activity.Activity* attribute), 11  
 actual\_start (*iatikit.data.activity.Activity* attribute), 11  
 all() (*iatikit.data.activity.ActivitySet* method), 10  
 all() (*iatikit.data.dataset.DatasetSet* method), 8  
 all() (*iatikit.data.organisation.OrganisationSet* method), 11  
 all() (*iatikit.data.publisher.PublisherSet* method), 7

## C

count() (*iatikit.data.activity.ActivitySet* method), 10  
 count() (*iatikit.data.dataset.DatasetSet* method), 8  
 count() (*iatikit.data.organisation.OrganisationSet* method), 11  
 count() (*iatikit.data.publisher.PublisherSet* method), 7

## D

data() (*in module iatikit*), 7  
 Dataset (*class in iatikit.data.dataset*), 9  
 datasets (*iatikit.data.publisher.Publisher* attribute), 8  
 datasets (*iatikit.data.registry.Registry* attribute), 7  
 DatasetSet (*class in iatikit.data.dataset*), 8  
 description (*iatikit.data.activity.Activity* attribute), 11

## E

end (*iatikit.data.activity.Activity* attribute), 11  
 etree (*iatikit.data.dataset.Dataset* attribute), 9

## F

filetype (*iatikit.data.dataset.Dataset* attribute), 9  
 filter() (*iatikit.data.activity.ActivitySet* method), 10

filter() (*iatikit.data.dataset.DatasetSet* method), 9  
 filter() (*iatikit.data.organisation.OrganisationSet* method), 12  
 filter() (*iatikit.data.publisher.PublisherSet* method), 7  
 find() (*iatikit.data.activity.ActivitySet* method), 10  
 find() (*iatikit.data.dataset.DatasetSet* method), 9  
 find() (*iatikit.data.organisation.OrganisationSet* method), 12  
 find() (*iatikit.data.publisher.PublisherSet* method), 8  
 first() (*iatikit.data.activity.ActivitySet* method), 10  
 first() (*iatikit.data.dataset.DatasetSet* method), 9  
 first() (*iatikit.data.organisation.OrganisationSet* method), 12  
 first() (*iatikit.data.publisher.PublisherSet* method), 8

## G

get() (*iatikit.data.activity.ActivitySet* method), 10  
 get() (*iatikit.data.dataset.DatasetSet* method), 9  
 get() (*iatikit.data.organisation.OrganisationSet* method), 12  
 get() (*iatikit.data.publisher.PublisherSet* method), 8

## H

humanitarian (*iatikit.data.activity.Activity* attribute), 11

## I

iatikit\_identifier (*iatikit.data.activity.Activity* attribute), 11  
 iatikit (*module*), 7  
 id (*iatikit.data.activity.Activity* attribute), 11  
 id (*iatikit.data.organisation.Organisation* attribute), 12

## L

last\_updated (*iatikit.data.registry.Registry* attribute), 7  
 location (*iatikit.data.activity.Activity* attribute), 11

**M**

metadata (*iatikit.data.dataset.Dataset* attribute), 9  
metadata (*iatikit.data.publisher.Publisher* attribute), 8

**N**

name (*iatikit.data.dataset.Dataset* attribute), 9  
name (*iatikit.data.publisher.Publisher* attribute), 8

**O**

org\_identifier (*iatikit.data.organisation.Organisation* attribute), 12  
Organisation (class in *iatikit.data.organisation*), 12  
organisations (*iatikit.data.dataset.Dataset* attribute), 9  
organisations (*iatikit.data.publisher.Publisher* attribute), 8  
organisations (*iatikit.data.registry.Registry* attribute), 7  
OrganisationSet (class in *iatikit.data.organisation*), 11

**P**

planned\_end (*iatikit.data.activity.Activity* attribute), 11  
planned\_start (*iatikit.data.activity.Activity* attribute), 11  
Publisher (class in *iatikit.data.publisher*), 8  
publishers (*iatikit.data.registry.Registry* attribute), 7  
PublisherSet (class in *iatikit.data.publisher*), 7

**R**

Registry (class in *iatikit.data.registry*), 7  
root (*iatikit.data.dataset.Dataset* attribute), 9

**S**

schema (*iatikit.data.dataset.Dataset* attribute), 9  
sector (*iatikit.data.activity.Activity* attribute), 11  
show () (*iatikit.data.activity.Activity* method), 11  
show () (*iatikit.data.dataset.Dataset* method), 9  
show () (*iatikit.data.organisation.Organisation* method), 12  
show () (*iatikit.data.publisher.Publisher* method), 8  
start (*iatikit.data.activity.Activity* attribute), 11

**T**

title (*iatikit.data.activity.Activity* attribute), 11

**V**

validate\_codelists () (*iatikit.data.dataset.Dataset* method), 10  
validate\_iati () (*iatikit.data.dataset.Dataset* method), 10

validate\_xml () (*iatikit.data.dataset.Dataset* method), 10  
version (*iatikit.data.dataset.Dataset* attribute), 10

**W**

where () (*iatikit.data.activity.ActivitySet* method), 10  
where () (*iatikit.data.dataset.DatasetSet* method), 9  
where () (*iatikit.data.organisation.OrganisationSet* method), 12  
where () (*iatikit.data.publisher.PublisherSet* method), 8

**X**

xml (*iatikit.data.activity.Activity* attribute), 11  
xml (*iatikit.data.dataset.Dataset* attribute), 10  
xml (*iatikit.data.organisation.Organisation* attribute), 12