
provserver-api Documentation

Release 0.1.3

Sam Millar

July 21, 2014

1	Installation	3
2	Contribute	5
3	API Documentation	7
3.1	provstore package	7
4	Indices and tables	13
	Python Module Index	15

provstore-api is a python API client for [ProvStore](#).

Installation

Install by running:

```
pip install provstore-api
```

Contribute

- Issue Tracker: [GitHub issues](#).
- Source Code: [GitHub](#).

API Documentation

3.1 provstore package

3.1.1 provstore.api module

class `provstore.api.Api` (*username=None, api_key=None, base_url=None*)

Bases: `object`

Main ProvStore API client object

Most functions are not used directly but are instead accessed by functions of the Document, BundleManager and Bundle objects.

To create a new Api object:

```
>>> from provstore.api import Api
>>> api = Api(username="provstore username", api_key="api key")
```

Note: The username and api_key parameters can also be omitted in which case the client will look for **PROVSTORE_USERNAME** and **PROVSTORE_API_KEY** environment variables.

FORMAT_MAP = {'json': 'application/json'}

add_bundle (*document_id, prov_bundle, identifier*)

delete_document (*document_id*)

document

get_bundle (*document_id, bundle_id, prov_format=<class 'prov.model.ProvDocument'>*)

get_bundles (*document_id*)

get_document_meta (*document_id*)

get_document_prov (*document_id, prov_format=<class 'prov.model.ProvDocument'>*)

headers

post_document (*prov_document, prov_format, name, public=False*)

exception `provstore.api.NotFoundException`

Bases: `exceptions.Exception`

3.1.2 provstore.bundle module

class `provstore.bundle.Bundle` (*api, document, bundle*)

Bases: `object`

created_at

Returns When the bundle was added

Return type `datetime.datetime`

identifier

Returns Identifier of the document, used as index on `provstore.bundle_manager.BundleManager`

Return type `str`

prov

Returns This bundle's provenance

Return type `prov.model.ProvDocument`

3.1.3 provstore.bundle_manager module

class `provstore.bundle_manager.BundleManager` (*api, document*)

Bases: `object`

A document's bundle manager.

This is an iterable and will iterate through all of a document's bundles.

Example getting and adding bundles:

```
>>> api = Api()
>>> api.document.create(prov_document, name="name")
>>> api.bundles
A BundleManager object for this document
>>> api.bundles['ex:bundle']
A Bundle with the identifier given (if exists)
>>> api.bundles['ex:new_bundle'] = prov_bundle
Saves a new bundle with the identifier specified
```

refresh()

Reload list of bundles from the store

Returns `self`

3.1.4 provstore.document module

exception `provstore.document.AbstractDocumentException`

Bases: `provstore.document.DocumentException`

class `provstore.document.Document` (*api*)

Bases: `object`

ProvStore Document model.

Note: This class should not be instantiated manually but should be accessed via `provstore.api.Api.document()` like so:

```
>>> from provstore.api import Api
>>> api = Api()
>>> api.document
<provstore.document.Document at ...>
```

abstract

True if this document doesn't reference a ProvStore document yet

add_bundle (*prov_bundle*, *identifier*)

Verbose method of adding a bundle.

Can also be done as:

```
>>> api = Api()
>>> document = api.document.get(148)
>>> document.bundles['identifier'] = prov_bundle
```

Parameters

- **prov_bundle** (*prov.model.ProvDocument* or *str*) – The bundle to be added
- **identifier** (*str*) – URI or QName for this bundle

bundles

Returns This document's bundle manager

Return type *provstore.bundle_manager.BundleManager*

create (*prov_document*, *prov_format=None*, *refresh=False*, ***props*)

Create a document on ProvStore.

Parameters

- **prov_document** (*prov.model.ProvDocument* or *str*) – The document to be stored
- **prov_format** (*str* or *None*) – The format of the document provided
- **refresh** (*bool*) – Whether or not to load back the document after saving
- **props** (*dict*) – Properties for this document [**name** (required), **public** = *False*]

Returns This document itself but with a reference to the newly stored document

Return type *provstore.document.Document*

Raises ImmutableDocumentException If this instance already refers to another document

created_at

Returns When the document was created

Return type *datetime.datetime*

delete ()

Remove the document and all of its bundles from ProvStore.

Warning: Cannot be undone.

get (*document_id*)

Associate this model with a document on ProvStore.

Example::

```
>>> api = Api()
>>> api.document.get(148)
https://provenance.ecs.soton.ac.uk/store/api/v0/documents/148
>>> api.id
148
>>> api.name
ex:bundles1-sep
```

Parameters `document_id` – The document ID on ProvStore

Returns `self`

id

Unique ID of the document as defined by ProvStore. Used in `get()` and `set()`

name

Name of document as seen on ProvStore

owner

Username of document creator

prov

Provenance stored for this document as `prov.model.ProvDocument`

public

Is this document visible to anyone?

read (*document_id=None*)

Load the document contents and metadata from the server.

The following are equivalent::

```
>>> api = Api()
>>> api.set(148).read()
>>> api.get(148)
```

Parameters `document_id` – (optional) Set the document ID if this is an abstract document.

Returns `self`

read_meta (*document_id=None*)

Load metadata associated with the document

Note: This method is called automatically if needed when a property is first accessed. You will not normally have to use this method manually.

Parameters `document_id` – (optional) set the document id if this is an `abstract()` document

Returns `self`

read_prov (*document_id=None*)

Load the provenance of this document

Note: This method is called automatically if needed when the `prov()` property is accessed. Manual use of this method is unusual.

Parameters `document_id` – (optional) set the document id if this is an `abstract()` document

Returns :py:class:`prov.model.ProvDocument`

refresh()

Update information about the document from ProvStore.

Returns `self`

save (*prov_document*, *prov_format=None*, *refresh=False*, ***props*)

Create a document on ProvStore.

Parameters

- **prov_document** (`prov.model.ProvDocument` or `str`) – The document to be stored
- **prov_format** (`str` or `None`) – The format of the document provided
- **refresh** (`bool`) – Whether or not to load back the document after saving
- **props** (*dict*) – Properties for this document [**name** (required), **public** = `False`]

Returns This document itself but with a reference to the newly stored document

Return type `provstore.document.Document`

Raises `ImmutableDocumentException` If this instance already refers to another document

set (*document_id*)

Associate this document with a ProvStore document without making any calls to the API. :param int document_id: ID of the document on ProvStore :return: self

url

URL of document on ProvStore

Example

```
>>> stored_document.url
'https://provenance.ecs.soton.ac.uk/store/documents/148'
```

views

Number of views this document has received on ProvStore

exception `provstore.document.DocumentException`

Bases: `exceptions.Exception`

exception `provstore.document.EmptyDocumentException`

Bases: `provstore.document.DocumentException`

exception `provstore.document.ImmutableDocumentException`

Bases: `provstore.document.DocumentException`

Indices and tables

- *genindex*
- *modindex*
- *search*

p

- `provstore.api`, [7](#)
- `provstore.bundle`, [8](#)
- `provstore.bundle_manager`, [8](#)
- `provstore.document`, [8](#)