

---

# **collection-json Documentation**

***Release***

**Ricardo Kirkner**

March 03, 2015



<b>1</b>	<b>Guide</b>	<b>3</b>
1.1	License . . . . .	3
1.2	Installation . . . . .	3
1.3	Example usage . . . . .	4
1.4	API Reference . . . . .	6
<b>2</b>	<b>Project info</b>	<b>9</b>
2.1	Changelog . . . . .	9
2.2	Authors . . . . .	9
	<b>Python Module Index</b>	<b>11</b>



Release 0.1.1. ([Changelog](#))

*collection-json* is a Python (2 and 3) library for working with Collection+JSON documents.



## 1.1 License

Copyright (c) 2015, Ricardo Kirkner All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.2 Installation

### 1.2.1 From PyPI

```
$ pip install collection-json
```

### 1.2.2 From source

*collection-json* is being hosted on [Github](#).

You can clone the master repo:

```
$ git clone https://github.com/ricardokirkner/collection-json.python.git
```

Or download one of the following:

- [tarball](#)
- [zipfile](#)

Once you have the source, you can install it locally by running

```
$ python setup.py install
```

### 1.2.3 Get the bleeding edge version

To get the latest development version of collection-json, run

```
$ pip install -U git+https://github.com/ricardokirkner/collection-json.python.git@dev
```

## Python

*collection-json* is compatible with Python 2.7, 3.3 and 3.4.

## 1.3 Example usage

Parse a json document into a Collection object

```
>>> from collection_json import Collection
>>> data = '{"collection": {"version": "1.0", "href": "..."}}'
>>> collection = Collection.from_json(data)
>>> collection
<Collection: version='1.0' href='... '>
```

Serialize a Collection object into a dictionary

```
>>> from collection_json import Collection
>>> data = '{"collection": {"version": "1.0", "href": "..."}}'
>>> collection = Collection.from_json(data)
>>> expected = {'collection': {'version': '1.0', 'href': '...'}}
>>> collection.to_dict() == expected
True
```

Serialize a Collection object into a json string

```
>>> import json
>>> from collection_json import Collection
>>> data = '{"collection": {"version": "1.0", "href": "..."}}'
>>> collection = Collection.from_json(data)
>>> expected = json.dumps(collection.to_dict())
>>> str(collection) == expected
True
```

Inspect collection properties

```
>>> from collection_json import Collection
>>> data = '{"collection": {"version": "1.0", "href": "..."}}'
>>> collection = Collection.from_json(data)
>>> collection.version
'1.0'
```



```
>>> collection.href
'...'
>>> collection.error is None
True
>>> collection.template is None
True
>>> collection.items
[]
>>> collection.links
[]
>>> collection.queries
[]
```

### Inspecting an error

```
>>> from collection_json import Collection
>>> error = {'code': '1234', 'message': 'Something bad happened.', 'title': 'Error 1234'}
>>> collection = Collection('...', error=error)
>>> collection.error
<Error: code='1234' message='Something bad happened.' title='Error 1234'>
>>> collection.error.code
'1234'
```

### Inspecting a template

```
>>> from collection_json import Collection
>>> template = {'data': [
...     {'name': 'first_name', 'value': '', 'prompt': 'First name'},
...     {'name': 'last_name', 'value': '', 'prompt': 'Last name'}]}
>>> collection = Collection('...', template=template)
>>> collection.template
<Template: data=['first_name', 'last_name']>
>>> collection.template.first_name
<Data: name='first_name' prompt='First name'>
```

### Inspecting items in a collection

```
>>> from collection_json import Collection
>>> data = '{"collection": {"version": "1.0", "href": "...", "items": [
...     {"href": "...", "data": [
...         {"prompt": "First name", "name": "first_name", "value": "John"},
...         {"prompt": "Last name", "name": "last_name", "value": "Doe"}]},
...     {"href": "...", "data": [
...         {"prompt": "First name", "name": "first_name", "value": "Jane"},
...         {"prompt": "Last name", "name": "last_name", "value": "Doe"}]}}]}'
>>> collection = Collection.from_json(data)
>>> collection
<Collection: version='1.0' href='...'>
>>> collection.items
[<Item: href='...'>, <Item: href='...'>]
>>> collection.items[0].properties
['first_name', 'last_name']
>>> collection.items[0].first_name
<Data: name='first_name' prompt='First name'>
>>> collection.items[0].first_name.value
'John'
>>> collection.items[0].data
[<Data: name='first_name' prompt='First name'>, <Data: name='last_name' prompt='Last name'>]
>>> collection.items[0].data[0].name
```

```
'first_name'
>>> collection.items[0].data[0].value
'John'
```

Serializing a template from json

```
>>> from collection_json import Template
>>> data = '{"template": {"data": [ {"name": "first_name", "value": "John"}, {"name": "last_name", "value": "Doe"}]}}'
>>> template = Template.from_json(data)
>>> template.first_name.value
'John'
>>> template.last_name.value
'Doe'
```

## 1.4 API Reference

Classes for representing a Collection+JSON document.

**class** `collection_json.Array` (*item\_class*, *collection\_name*, *items*)

Bases: `collection_json.ComparableObject`, `list`

Object representing a Collection+JSON array.

**find** (*name=None*, *rel=None*)

Return a list of items in the array matching name and/or rel.

If both name and rel parameters are provided, returned items must match both properties.

**get** (*name=None*, *rel=None*)

Return the first item in the array matching name and/or rel.

If both name and rel parameters are provided, the returned item must match both properties.

If no item is found, raises `ValueError`.

**to\_dict** ()

Return a dictionary representing an Array object.

**class** `collection_json.ArrayProperty` (*cls*, *name*)

Bases: `object`

A descriptor that converts from any enumerable to a typed Array.

**class** `collection_json.Collection` (*href*, *links=None*, *items=None*, *queries=None*, *template=None*, *error=None*, *version='1.0'*)

Bases: `collection_json.ComparableObject`

Object representing a Collection+JSON document.

**error**

A descriptor for assigning only a specific type of instance.

Additionally supports assigning a dictionary convertible to the type.

**static from\_json** (*data*)

Return a Collection instance.

This method parses a json string into a Collection object.

Raises `ValueError` when no valid document is provided.

**items**

A descriptor that converts from any enumerable to a typed Array.

**links**

A descriptor that converts from any enumerable to a typed Array.

**queries**

A descriptor that converts from any enumerable to a typed Array.

**template**

A descriptor for assigning only a specific type of instance.

Additionally supports assigning a dictionary convertible to the type.

**to\_dict()**

Return a dictionary representing a Collection object.

**class** `collection_json.ComparableObject`

Bases: `object`

Abstract base class for objects implementing equality comparison.

This class provides default `__eq__` and `__ne__` implementations.

**class** `collection_json.Data(name, value=None, prompt=None)`

Bases: `collection_json.ComparableObject`

Object representing a Collection+JSON data object.

**to\_dict()**

Return a dictionary representing a Data object.

**class** `collection_json.Error(code=None, message=None, title=None)`

Bases: `collection_json.ComparableObject`

Object representing a Collection+JSON error object.

**to\_dict()**

Return a dictionary representing the Error instance.

**class** `collection_json.Item(href=None, data=None, links=None)`

Bases: `collection_json.ComparableObject`

Object representing a Collection+JSON item object.

**data**

A descriptor that converts from any enumerable to a typed Array.

**links**

A descriptor that converts from any enumerable to a typed Array.

**properties**

Return a list of names that can be looked up on the item.

**to\_dict()**

Return a dictionary representing an Item object.

**class** `collection_json.Link(href, rel, name=None, render=None, prompt=None)`

Bases: `collection_json.ComparableObject`

Object representing a Collection+JSON link object.

**to\_dict()**

Return a dictionary representing a Link object.

**class** `collection_json.Query` (*href, rel, name=None, prompt=None, data=None*)

Bases: `collection_json.ComparableObject`

Object representing a Collection+JSON query object.

**data**

A descriptor that converts from any enumerable to a typed Array.

**to\_dict** ()

Return a dictionary representing a Query object.

**class** `collection_json.Template` (*data=None*)

Bases: `collection_json.ComparableObject`

Object representing a Collection+JSON template object.

**data**

A descriptor that converts from any enumerable to a typed Array.

**static from\_json** (*data*)

Return a template instance.

Convenience method for parsing ‘write’ responses, which should only contain a template object.

This method parses a json string into a Template object.

Raises *ValueError* when no valid document is provided.

**properties**

Return a list of names that can be looked up on the template.

**to\_dict** ()

Return a dictionary representing a Template object.

**class** `collection_json.TypedProperty` (*cls, name*)

Bases: `object`

A descriptor for assigning only a specific type of instance.

Additionally supports assigning a dictionary convertible to the type.

---

**Project info**

---

## **2.1 Changelog**

### **2.1.1 0.1.1 (2015-03-03): Usability**

- cast value to the right type when setting arrays (items, links, queries, data) and typed values (error, template)
- added support for Template to load from json
- support serializing Collection directly as string
- added helper for returning the first match in an array

### **2.1.2 0.1.0 (2014-02-24): Basics**

- added basic classes for representing Collection+JSON documents
- allow querying array items by rel and name attributes
- attribute lookup on array searches items by name
- ensure href is a required parameter for collection
- expose data item names as 'properties' attribute in Item and Template
- access data items via attribute lookup on item name

### **2.1.3 0.0.1dev (2014-02-21)**

- Initial release.

## **2.2 Authors**

collection-json is written and maintained by Ricardo Kirkner and various contributors.

### **2.2.1 Development Lead**

- Ricardo Kirkner <[ricardo@kirkner.com.ar](mailto:ricardo@kirkner.com.ar)> (@[ricardokirkner](https://twitter.com/ricardokirkner))

## 2.2.2 Contributors

- Chris Marinos (@ChrisMarinos)
- Daniel DeSousa (@dandesousa)

**C**

`collection_json`, 6





## A

Array (class in collection\_json), 6  
ArrayProperty (class in collection\_json), 6

## C

Collection (class in collection\_json), 6  
collection\_json (module), 6  
ComparableObject (class in collection\_json), 7

## D

Data (class in collection\_json), 7  
data (collection\_json.Item attribute), 7  
data (collection\_json.Query attribute), 8  
data (collection\_json.Template attribute), 8

## E

Error (class in collection\_json), 7  
error (collection\_json.Collection attribute), 6

## F

find() (collection\_json.Array method), 6  
from\_json() (collection\_json.Collection static method), 6  
from\_json() (collection\_json.Template static method), 8

## G

get() (collection\_json.Array method), 6

## I

Item (class in collection\_json), 7  
items (collection\_json.Collection attribute), 6

## L

Link (class in collection\_json), 7  
links (collection\_json.Collection attribute), 7  
links (collection\_json.Item attribute), 7

## P

properties (collection\_json.Item attribute), 7  
properties (collection\_json.Template attribute), 8

## Q

queries (collection\_json.Collection attribute), 7  
Query (class in collection\_json), 7

## T

Template (class in collection\_json), 8  
template (collection\_json.Collection attribute), 7  
to\_dict() (collection\_json.Array method), 6  
to\_dict() (collection\_json.Collection method), 7  
to\_dict() (collection\_json.Data method), 7  
to\_dict() (collection\_json.Error method), 7  
to\_dict() (collection\_json.Item method), 7  
to\_dict() (collection\_json.Link method), 7  
to\_dict() (collection\_json.Query method), 8  
to\_dict() (collection\_json.Template method), 8  
TypedProperty (class in collection\_json), 8