api_workshop

August 14, 2018

```
In [1]: # Preamble slide - put stuff not for consumption by ppl
        # run prior to setting up pres
        def take(i, it):
            it = iter(it)
            out = []
            for _ in range(i):
                try:
                     out.append(next(it))
                except StopIteration:
                    pass
            return out
```

1 ArchivesSnake - Abstraction Layer

Being an attempt to convey some of the functions of ASnake meant to remove one from the drudgery of everyday API manipulation, and thus enable one to experience the transports and delights of having easy access to one's data.

1.1 Initialization and Logging

There comes a time in the affairs of hummankind where one must do things, and also write down that those things have been done and roughly when.

```
In [2]: from asnake.aspace import ASpace
        import asnake.logging as logging

        logging.setup_logging(logging.INFO_TO_STDOUT)
        aspace = ASpace() # using default configuration values
        log = logging.get_logger('my_script')
```

1.2 everything = ASpace()

The ASpace class produces objects that represent the entirety of the API. It's intended to make it easy to get records out of the system, and to particularly make convenient access to individual fields in records.

For example, to print the titles of all resources in the system, you can do this:

```
In [3]: [res.title for res in aspace.resources]

Out[3]: ['Gérard Defaux papers', 'Records of the Best University Library']
```

Or, say you wanted to do something with the titles of all archival objects in all of resources?

```
In [4]: # take(limit, thing) gets the first `limit` objects in `thing`
        for res in aspace.resources:
            for ao in take(5, res.tree.walk):
                if ao.jsonmodel_type == 'archival_object':
                    print(ao.title)
```

```
Research Materials
Research Notes
Rabelais-Villon
Research Notes
Subgroup 1: Office of the Director
Subgroup 2: Library Committee
Subgroup 3: Assistant Librarians
Subgroup 4: Library Personnel Office
```

The ASpace object and objects returned from it largely have methods that match either routes in the API or keys in the JSON returned by the API.

For example,

1.2.1 API Route

```
http://localhost:4567/repositories/2/resources
```

1.2.2 ASnakeClient

```
aspace.client.get('repositories/2/resources')
```

1.2.3 ASpace

```
ASpace.repositories(2).resources
```

1.3 That third one though

A lot's happening in that third one, so let's examine it more closely.

```
In [5]: aspace.repositories

Out[5]: #<JSONModelRelation:/repositories:{'all_ids': True}>
```

What's a JSONModelRelation? Well, it's an object that represents an API route in ArchivesSpace.

```
In [6]: aspace.repositories(2)
```

```
Out[6]: #<repository:/repositories/2>
```

```
In [7]: type(aspace.repositories(2))
```

```
Out[7]: asnake.jsonmodel.JSONModelObject
```

What's a JSONModelObject? It's an object that represents a singular record returned from the API (a single resource, archival object, digital object, etc.)

```
In [8]: aspace.repositories(2).json()
```

```
Out[8]: {'lock_version': 0,
          'repo_code': 'api_wrkshp',
          'name': 'API Workshop Repository',
          'created_by': 'admin',
          'last_modified_by': 'admin',
          'create_time': '2017-05-24T17:20:25Z',
          'system_mtime': '2018-03-16T16:07:32Z',
          'user_mtime': '2017-05-24T17:20:25Z',
          'publish': True,
          'jsonmodel_type': 'repository',
          'uri': '/repositories/2',
          'display_string': 'API Workshop Repository (api_wrkshp)',
          'agent_representation': {'ref': '/agents/corporate_entities/1'}}
```

So, you can get the JSON out of it pretty easily... but often, you don't really care about the whole shebang.

```
In [9]: aspace.repositories(2).display_string
```

```
Out[9]: 'API Workshop Repository (api_wrkshp)'
```

Or, sometimes, you want to get something that's logically descendant (i.e. exists in the API in a subsidiary route) but not present in the JSON.

```
In [10]: aspace.repositories(2).archival_objects
```

```
Out[10]: #<JSONModelRelation:/repositories/2/archival_objects:{}>
```

```
In [11]: take(10, aspace.repositories(2).archival_objects)
```

```
Out[11]: [#<archival_object:/repositories/2/archival_objects/1>,
           #<archival_object:/repositories/2/archival_objects/2>,
           #<archival_object:/repositories/2/archival_objects/3>,
           #<archival_object:/repositories/2/archival_objects/4>,
           #<archival_object:/repositories/2/archival_objects/5>,
           #<archival_object:/repositories/2/archival_objects/6>,
           #<archival_object:/repositories/2/archival_objects/7>,
           #<archival_object:/repositories/2/archival_objects/8>,
           #<archival_object:/repositories/2/archival_objects/9>,
           #<archival_object:/repositories/2/archival_objects/10>]
```

1.4 Short Version

The ASpace object is meant to make it easy to get things out of ArchivesSpace… but you need to know:

- what API routes do my records live at?
- what information do I need to provide them to get my records?
- what properties do the records returned by the API have?

1.5 Where's the Map?

1.6 What API routes do my records live at?

Largely a solved problem, enumerated accurately in docs.

- https://archivesspace.github.io/archivesspace/api/#get-repositories-repo_id-archival_objects

- https://pobocks.github.io/aspace_api_cheatsheet/

1.7 What information do I need to provide them to get my records?

NOT a solved problem - for various reasons, the API docs are often not explicit enough, all examples are raw curl only, and many examples are outright incorrect.

1.8 What information do I need to provide them to get my records?

- Example objects
  - https://github.com/archivesspace-labs/ArchivesSnake/wiki/Commonly-Used-Objects
  -
- Schemas - https://github.com/archivesspace/archivesspace/tree/master/common/schemas

In [12]:
```
# ALL SCHEMAS, output too big to display
aspace.client.get('schemas')

# Any particular schema!
aspace.client.get('schemas/term').json()
```

Out[12]:
```
{'$schema': 'http://www.archivesspace.org/archivesspace.json',
 'version': 1,
 'type': 'object',
 'uri': '/terms',
 'properties': {'uri': {'type': 'string', 'required': False},
  'term': {'type': 'string',
   'maxLength': 255,
   'minLength': 1,
   'ifmissing': 'error'},
  'term_type': {'type': 'string',
   'minLength': 1,
   'ifmissing': 'error',
   'dynamic_enum': 'subject_term_type'},
  'vocabulary': {'type': 'JSONModel(:vocabulary) uri', 'ifmissing': 'error'},
  'lock_version': {'type': ['integer', 'string'], 'required': False},
  'jsonmodel_type': {'type': 'string', 'ifmissing': 'error'},
  'created_by': {'type': 'string', 'readonly': True},
  'last_modified_by': {'type': 'string', 'readonly': True},
  'user_mtime': {'type': 'date-time', 'readonly': True},
  'system_mtime': {'type': 'date-time', 'readonly': True},
  'create_time': {'type': 'date-time', 'readonly': True},
  'repository': {'type': 'object',
   'subtype': 'ref',
   'readonly': 'true',
   'properties': {'ref': {'type': 'JSONModel(:repository) uri',
     'ifmissing': 'error',
     'readonly': 'true'},
    '_resolved': {'type': 'object', 'readonly': 'true'}}}}}
```

1.9 What information do I need to provide them to get my records?

- Example objects - https://github.com/archivesspace-labs/ArchivesSnake/wiki/Commonly-Used-Objects
- Schemas - https://github.com/archivesspace/archivesspace/tree/master/common/schemas
- The Actual Code - https://github.com/archivesspace/archivesspace/tree/master/backend/app/controlle

1.10 Thank you!

ArchivesSnake * https://github.com/archivesspace-labs/ArchivesSnake
Dave Mayo * Twitter: @pobocks * Email: dave_mayo@harvard.edu