What Can You Do with a TaxonWorks API?

Yoder M[1], Pereira H[2], Pereira JL[3], Dmitriev DA[1], Ower G[1], Flood JR[1]

[1] University of Illinois at Urbana-Champaign, Champaign, Illinois, United States of America; [2] División Entomología, Museo de La Plata, CEPAVE-CONICET, La Plata, Argentinal;
[3] División Entomología, Museo de La Plata, Universidad Nacional de La Plata, La Plata, Argentina

Application Programming Interface

API is short for "Application Programming Interface". It describes how to interact with software or applications.

This poster: **https://doi.org/10.3897/biss.4.59170** @ TDWG 2020

What?

TaxonWorks (**http://taxonworks.org**) is a web-application workbench focused on enabling the data curation process that underlies biodiversity science. It is presently used by natural history collection curators, taxonomists, evolutionary biologists, and their collaborators. A majority of the web-application is built on an internal API that serves JSON. We're exposing that internal API outside of TaxonWorks' web-based interfaces so that users can query and share the efforts of their curation via programmatic access by themselves or others . This poster is meant to be an introduction to API use in general and to TaxonWorks' API specifically, then it illustrates just a few possible uses. This poster will evolve to become part of the growing help documentation for TaxonWorks.

How?

- Companion code to this poster is in the corresponding presentation at **https://github.com/mjy/presentations**
- All figures were derived from data exclusively drawn from existing projects in TaxonWorks via external API calls
- Data from API calls was fed to graphing libraries or HTML wrappers of those libraries to produce SVG graphs

Current scope of the API

Biodiversity informatics covers and immense variety of different types of data. A TaxonWorks API currently exposes a subset of the data types that it curates internally. Those types are:

TaxonNames	BiologicalAssociations	Identifiers
OTUs	CollectionObjects	AssertedDistributions
TaxonDeterminations	CollectingEvents	Citations
People	Downloads	Sources

Plurality

TaxonWorks considers Identifiers to be annotations that "localize" one to the target of interest. You can stack as many as you want, of different types, on (almost) any data instance.

Cite anything

Similar to identifiers you can add citations to (nearly) any type of data. In doing so you assert that this instance of data was described fully, or in part, in this Source.

Share your zips

A downloads endpoint lists URLs to data you've filtered internally and packaged for sharing by URL.

A request to a TaxonWorks API

API requests often follow a pattern like that below. You can find a full description of the TaxonWorks API, its resources and parameters at **https://api.taxonworks.org.**

Personal tokens

A person with a TaxonWorks account can generate their own user **`token`**. This lets them use the API without exposing their project to use by others.

Finding authentication tokens

Find open **`project_tokens`** by visiting the root of the API instance, e.g. **`https://sfg.taxonworks.org/api/v1`**.

```
https://sandbox.taxonworks.org  /api/v1      /people    ?first_name=Thomas&last_name=McElrath     &token=abc&project_token=123
```

A base URL — The API version — A resource, or "endpoint" — Some parameters — Authentication (one or both tokens, depending on the endpoint).

v1 is evolving

We apologize in advance for tweaks that might break your use of it.*

A simple start

To demonstrate how you might use the API, and what you might see when you do, we can execute a simple plan find out what we know about Tommy McElrath, curator of the Illinois Natural History Survey Insect Collection (Illinois, USA), and a major contributor to the TaxonWorks community. We start with the simplest of queries, to the people endpoint (below left). This returns a list of results, in this case a single one (zero beers!).

Format your JSON!

We recommend use of the command-line tool 'jq' to format (and query!) JSON responses when testing API endpoints from the command line.

```
[
    [0] {
                      "id" => 55281,
                    "type" => "Person::Vetted",
               "last_name" => "McElrath",
              "first_name" => "Thomas Christian",
                  "suffix" => nil,
                  "prefix" => nil,
                  "cached" => "McElrath, Thomas Christian",
               "year_born" => 1989,
               "year_died" => nil,
       "year_active_start" => 2010,
         "year_active_end" => 2020,
           "created_by_id" => 44,
           "updated_by_id" => 44,
              "created_at" => "2019-04-03T17:22:25.646Z",
              "updated_at" => "2020-02-04T17:31:01.525Z",
               "global_id" => "gid://taxon-works/Person::Vetted/55281"
    }
]
```

`/people?first_name=Thomas&last_name=McElrath`

global_id in TaxonWorks

At the moment, a `global_id` is an internal id that let's us navigate through the data. In TaxonWorks you can stack as many Identifiers as you want on to the concepts you've curated in TaxonWorks.

But how can we know this is the *real* Tommy? A second request (below right), to the identifiers endpoint returns all identifiers associated with the Person instance. Since we know Tommy's ORCiD we can confirm it's him. Now we also have the TaxonWorks internal identifier that we can use more directly in our future queries.

```
... "type" => "Identifier::Global::Orcid",
    "cached" => "https://orcid.org/0000-0003-0390-4227",

... "type" => "Identifier::Global::Wikidata",
     "cached" => "Q26494089",
```

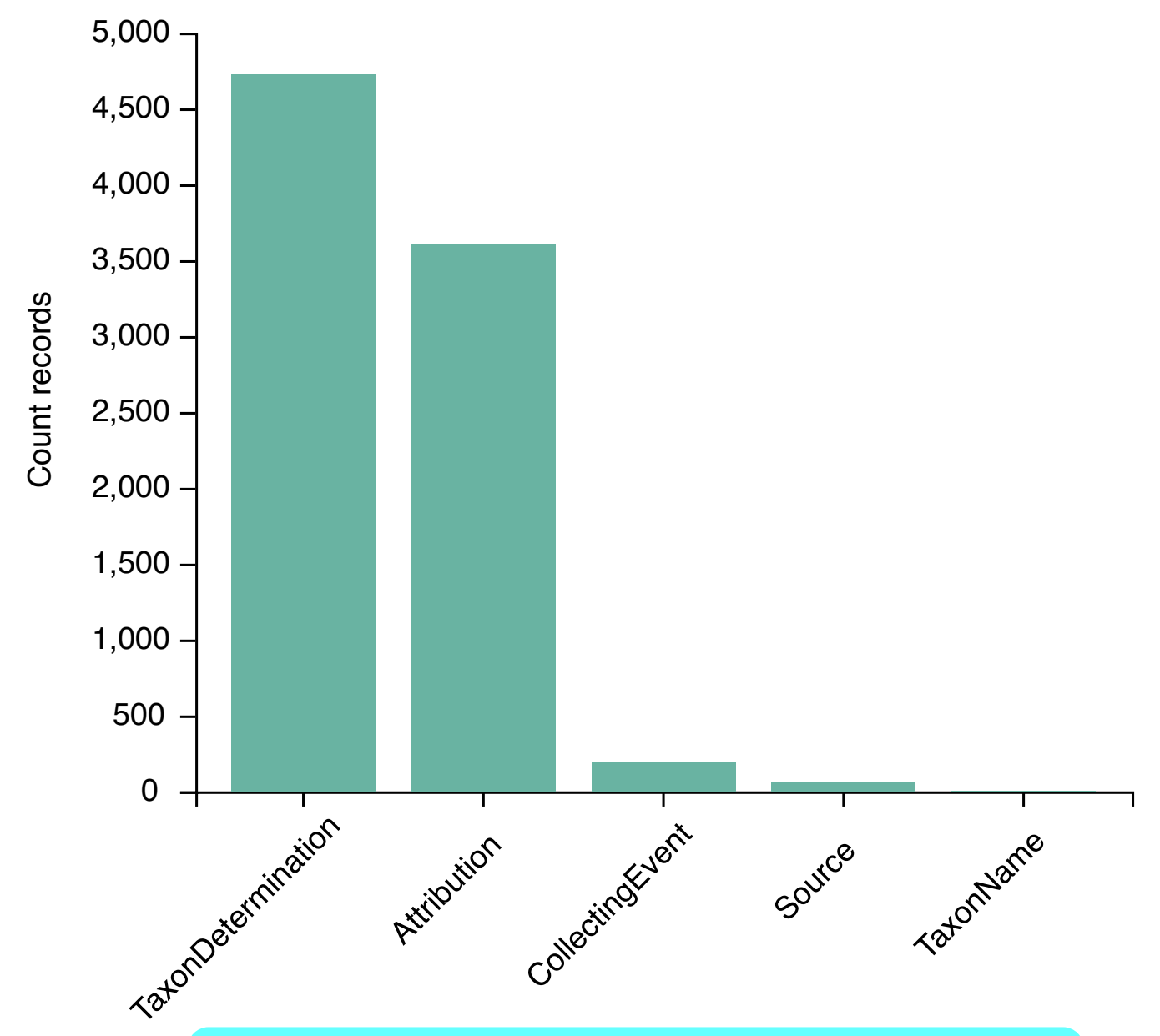
`/identifiers?object_global_id=gid://taxon-works/Person::Vetted/55281&project_id=1`

Community data

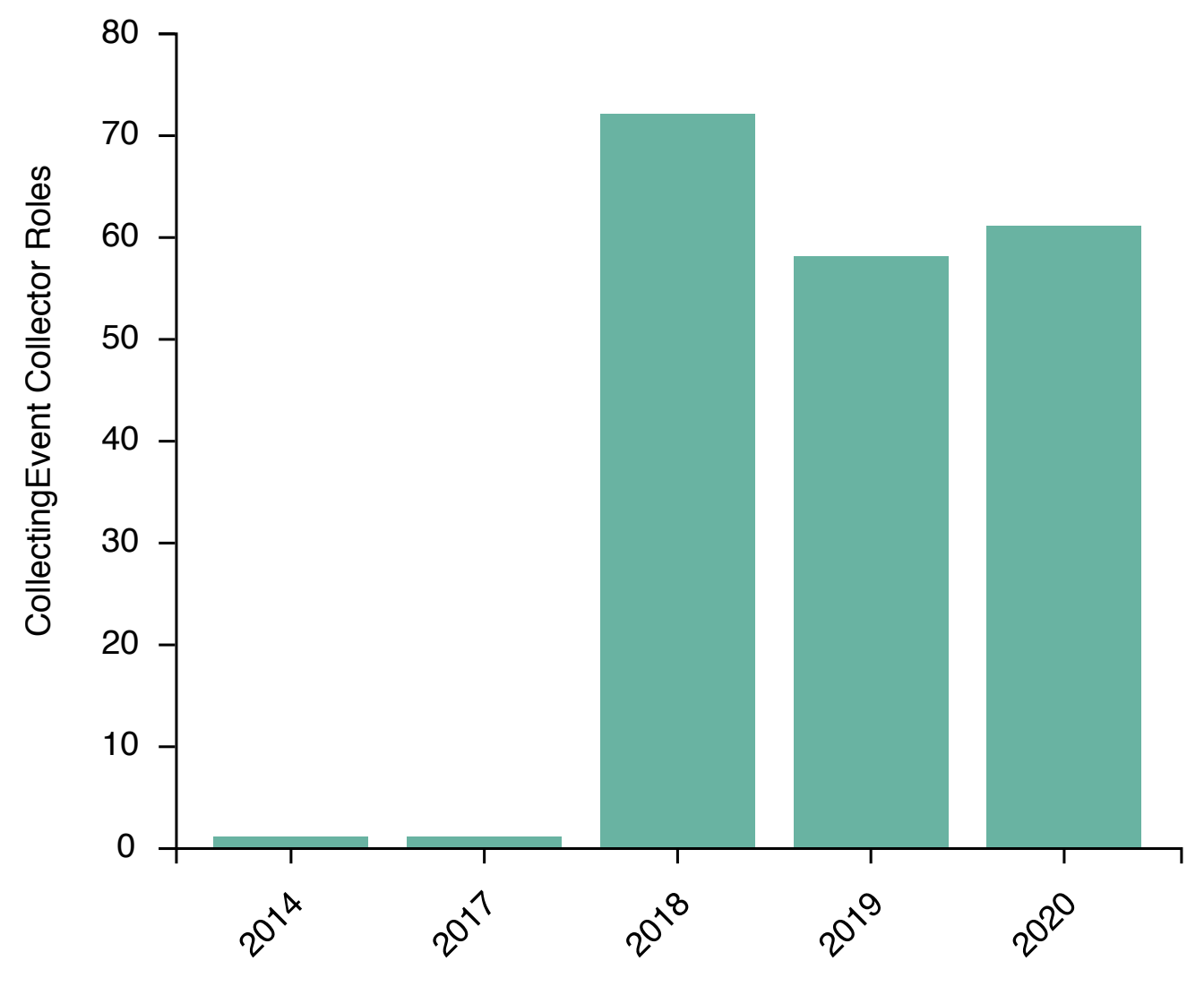
Why do we need `project_id`? We shouldn't! In TaxonWorks a Person is Community data, their records are shared across projects so you should be able to ask for identifiers without referencing a project. We're working on it.

Basics

People are data in TaxonWorks. They're linked to other data via roles. We can query Tommy's roles (below left). He's been identifying specimens (TaxonDetermination), creating Media and assigning Attribution (knowing Tommy it's CC 0), collecting (CollectingEvent), publishing (Source), and describing new taxa (TaxonName). We can learn many similar things by iterating through single API requests, for example when Tommy has been collecting for the INHS Insect Collection (below right).



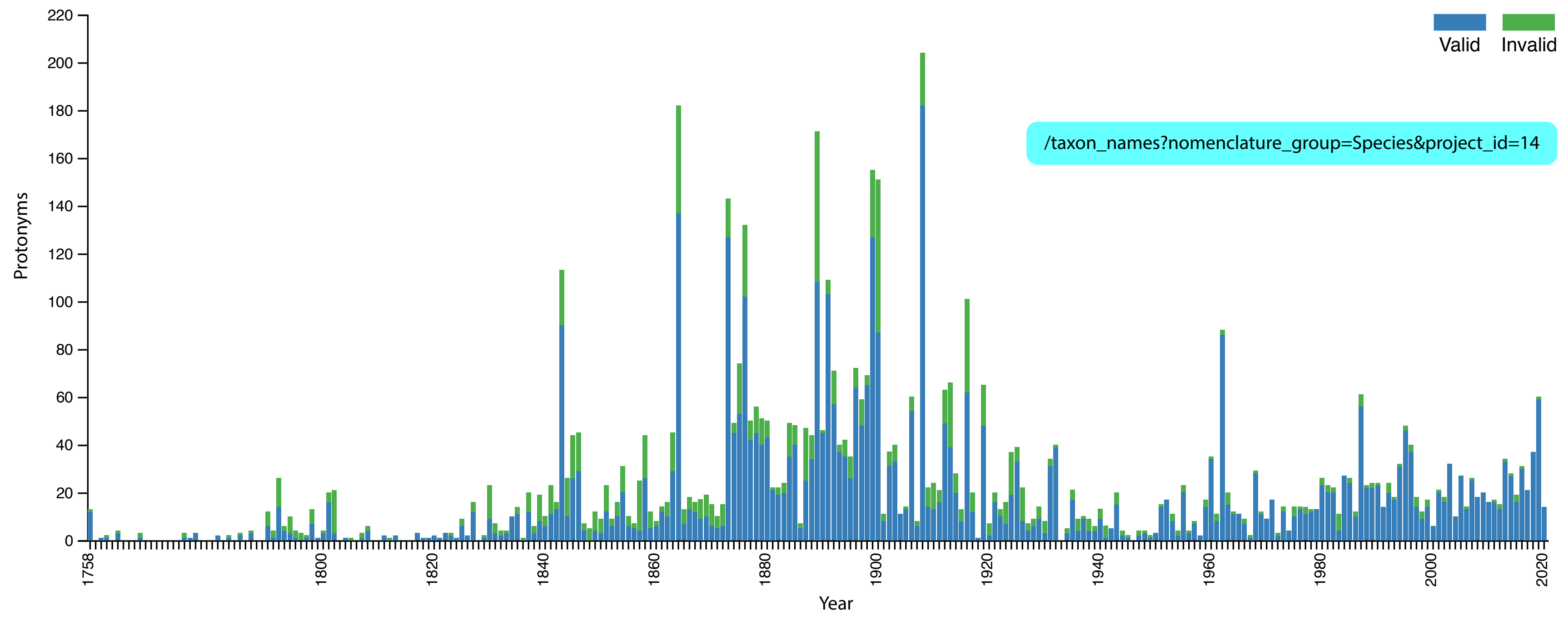
`/people/55281?include_roles=true`



`/collecting_events?collector_ids[]=55281&project_id=1`

Nomenclature catalogs

Tommy and his collaborators are editors of the World Catalog of Cucujoidea (aka some of many other little brown beetles) in TaxonWorks. Basic data like year published, and name validity are straightforward queries and data summaries. Below, a plot of species-group protonyms published per-year for the Cucujoidea. It's remarkable how few years contain no species-descriptions or synonymy.



`/taxon_names?nomenclature_group=Species&project_id=14`

Who collects with who?

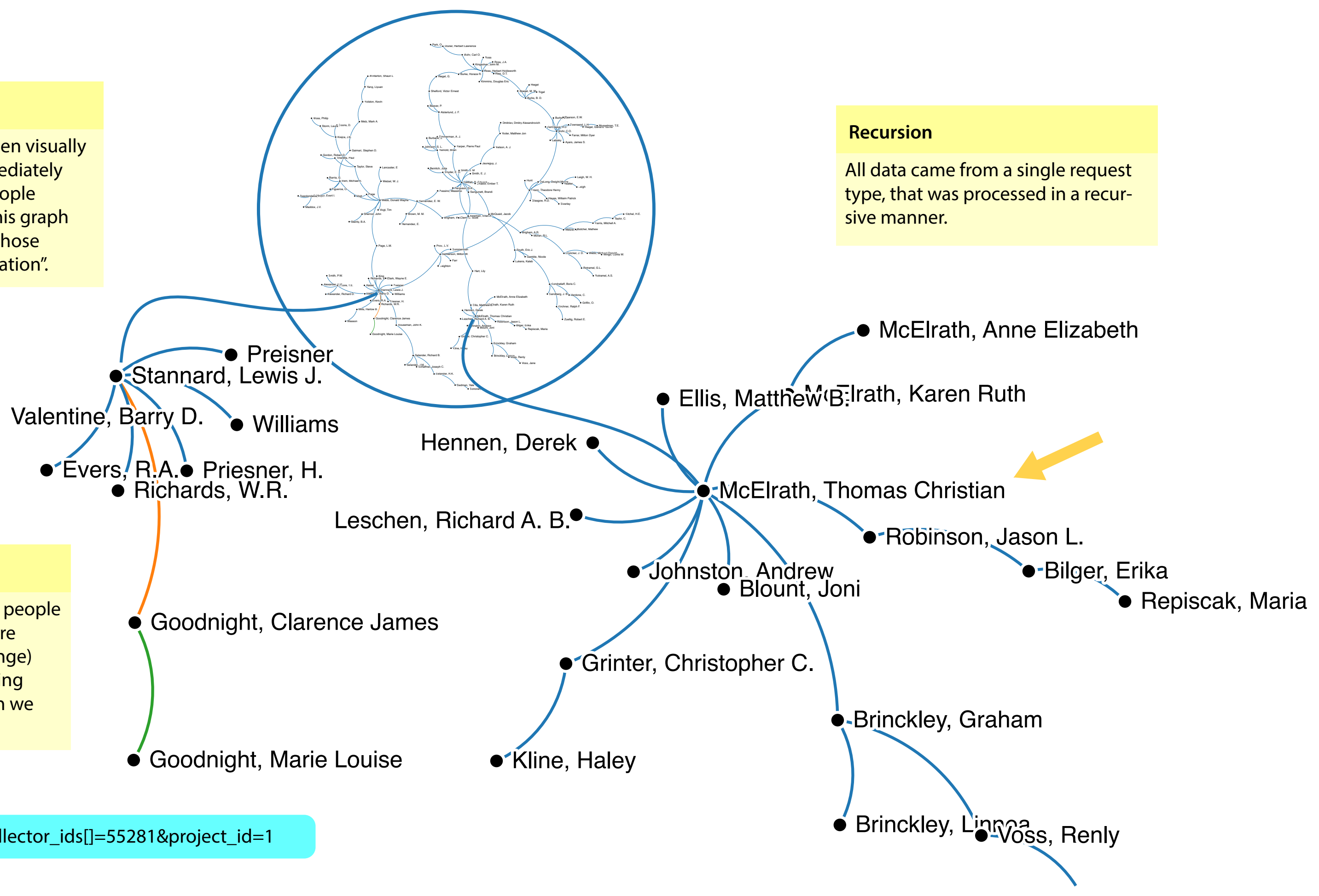
Starting with Tommy, we can recursively query CollectingEvent data with a simple script to see who he collected with, and who the collectors he collected with collected with, and so on. The result is remarkable, demonstrating the close connectedness of, in this case, the entomological community. Tommy is connected to a minimum of 160 other collectors.

Validation

While not visible here, when visually inspected the graph immediately made visible duplicate People records. In TaxonWorks this graph could be made to link to those people for their "uniquification".

Recursion

All data came from a single request type, that was processed in a recursive manner.



Born on

We can map whether the two people collecting with each other were younger (green) or older (orange) than one another. We're missing some data, but in the long run we can infer a lot with this.

`/collecting_events?collector_ids[]=55281&project_id=1`

What's next?

We've only touched the surface of what you can query from a TaxonWorks API. With well over 100 parameters now available many opportunities exist for exploring data curated in TaxonWorks in novel ways. We'll be exposing more data classes, including Images, Markdown marked-up text-content, and Observations (data matrices) in the weeks to come. We'll also be further curating code examples , and ways to integrate the API, that demonstrate how to map, graph, and visualize data from a TaxonWorks API. Ultimately our goal is to provide external tools to build full blown taxon pages via API calls.

- Instances of TaxonWorks have their own APIs. We're aggregating pointers to them at **https://api.taxonworks.org**
- Follow us at Twitter at **@TaxonWorks**
- Contact us, always live, at **https://gitter.im/SpeciesFileGroup/taxonworks**
- TaxonWorks is open source. Hack with us: **http://github.com/SpeciesFileGroup/taxonworks**
- We meet in user-groups online, a lot. Contact Matt if you're interested.
- Watch for our remote "hack" week activities coming Fall 2020. Details coming on Twitter and at **https://speciesfilegroup.org**.
- The Species File Group has a new Community Liason! Contact Deb Paul and welcome her to our group!

* No API endpoints were harmed in the making of this poster, just refined. Well, actually, some were destroyed when we realized they didn't do quite what we wanted.