



Reading data directly into your analysis script: Introduction to APIs



Erica Krimmel

iDigBio, Florida State University
ekrimmel@fsu.edu // @ekrimmel

Ecological Society of America Virtual Meeting
Career Central, [CC 6 - Data Help Desk: Using Data](#)
August 5, 2020

#DataHelpDesk

TALK ABSTRACT: Do you write scripts to analyze your data? An Application Programming Interface (API) can provide direct access to data and metadata in online repositories, saving you time and increasing the reproducibility of your analyses. This talk will provide an introduction in R to using APIs from several repositories of ecological data.

RECORDING SCRIPT: Hello, my name is Erica Krimmel and this career central talk will provide an introduction to APIs, including how they can be used for reading data directly into an analysis script.

Data Help Desk



<https://bit.ly/datahelpesa2020>

#DataHelpDesk

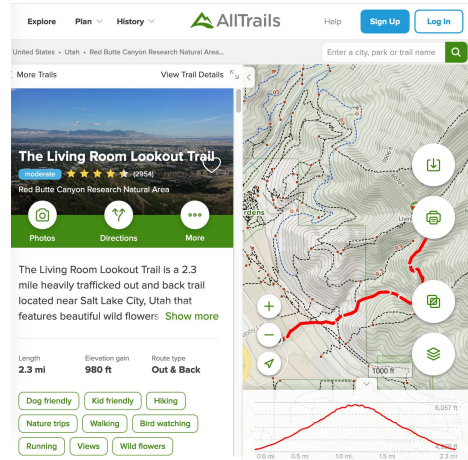
2

RECORDING SCRIPT: This talk is brought to you by the ESA Data Help Desk, a collaboration between The Arctic Data Center, the Consortium of Universities for the Advancement of Hydrologic Science (CUAHSI), DataONE, the Environmental Data Initiative (EDI), the Global Biodiversity Information Facility (GBIF), iDigBio, NEON, and Neotoma. You can find out more about the Data Help Desk and see a full list of our activities at the 2020 ESA Career Central by following the bit.ly link on this slide. You can also find us on Twitter this week using the hashtag #DataHelpDesk.



What is an API?

Application Programming Interface



An organization

has a system

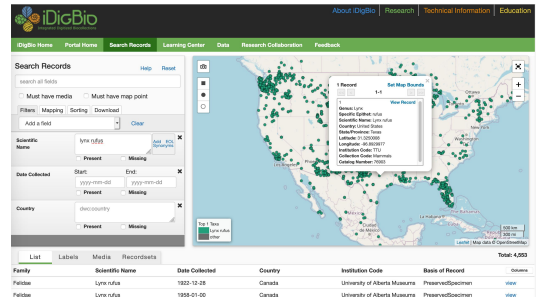
that external users can interact with

RECORDING SCRIPT: Let's begin with the basics. An API is an application programming interface, which is essentially a way for an organization to allow external users to interact with their systems. A really common scenario is when an organization, like Google, has a system, like Google Maps, that lots of external users, like this AllTrails app, want to use. It would be risky to provide external users with direct access to the system and so the organization makes access available via an API. Some APIs require users to be authenticated, e.g. by registering with the organization.



What is an API?

Application Programming Interface



An organization

has a system

that external users can interact with

RECORDING SCRIPT: You may be more familiar with APIs than you realize, as many websites use their own APIs to allow web users to search for content on the site. For example, the iDigBio data portal provides a single place to discover specimens held by natural history collections. When users search in our portal, they are using the iDigBio search API, just through a visual interface rather than in a programming environment like R or Python.



What is an API useful for?

An API facilitates programmatic data access, which enables...

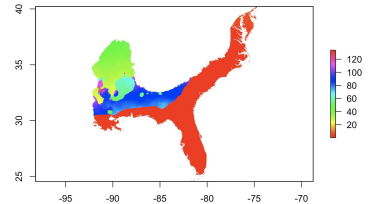
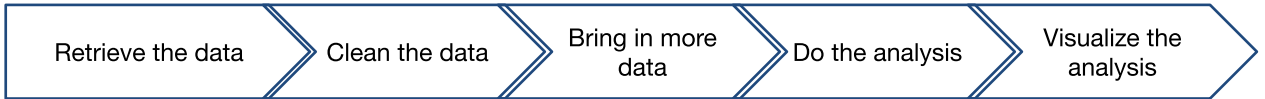
- Automating data retrieval
- Documenting procedures within your code
- Running analyses in a way that is reproducible
- Bringing data from different sources together, e.g. via:
 - ◆ multiple APIs
 - ◆ multiple downloaded datasets
 - ◆ your own data
 - ◆ any combination of the above

And, you can bring your own programming language!

RECORDING SCRIPT: So now that we're familiar with the essence of what an API is, why or when would we want to use one? APIs facilitate programmatic access to data, which, for research, is often an important part of creating a reproducible data workflow. If you have steps in your research pipeline that involve searching for and downloading data online, there is a good chance you might be able to do the same searches via an API. Sometimes this benefit can save you a lot of time, for instance, if you need to download specimen images from iDigBio you can only do so one at a time in the online portal, but via the API and a few lines of code you can automate this task. Instead of clicking through a thousand images, you can hit "go" and come back later to find them all downloaded and waiting for you. In addition to enabling automation, because you can use APIs in a programming environment, you have all of the benefits associated with that, including documenting procedures within your code, running analyses in a way that is reproducible, and integrating data from multiple sources. The best part is that APIs are language agnostic, so you can use them in whatever programming language you like, or even in other data wrangling user interfaces such as OpenRefine.



For example...



See a full example at <https://github.com/mgaynor1/CURE-FL-Plants>

RECORDING SCRIPT: Let's look at an example of how you might integrate APIs into your research pipeline. This example is brought to us by Shelly Gaynor, a grad student at the University of Florida studying, among other things, how Florida plants might respond to climate change. Shelly first needs to get occurrence records for the species she is interested in, and to do so she queries the APIs from GBIF, iDigBio, and USGS BISON. Her data come directly into R, which sets her up for the data cleaning phase. She next needs to get climate data from WorldClim, which she can also read into R, and then she can move along to her analysis and visualization. Because all of these steps are written in code, Shelly can provide an additional layer of documentation. Have you ever searched for data in an online portal, downloaded it, used it, and then later realized you don't exactly remember what search terms you used? Using APIs to retrieve data helps avoid this situation because your search terms are written into the code. If you already write code to run your analyses, this is an easy step to add that significantly increases the reproducibility of your research.



What is an endpoint?

An endpoint is an address, often a URL, where you can find a particular API

```
https://maps.googleapis.com/maps/api/directions/
```

↳ Let me ask Google Maps for directions

```
https://maps.googleapis.com/maps/api/geocode/
```

↳ Let me ask Google Maps to find a place on a map

```
https://search.idigbio.org/v2/search/
```

↳ Let me ask iDigBio to look for certain specimen records

```
https://search.idigbio.org/v2/download/
```

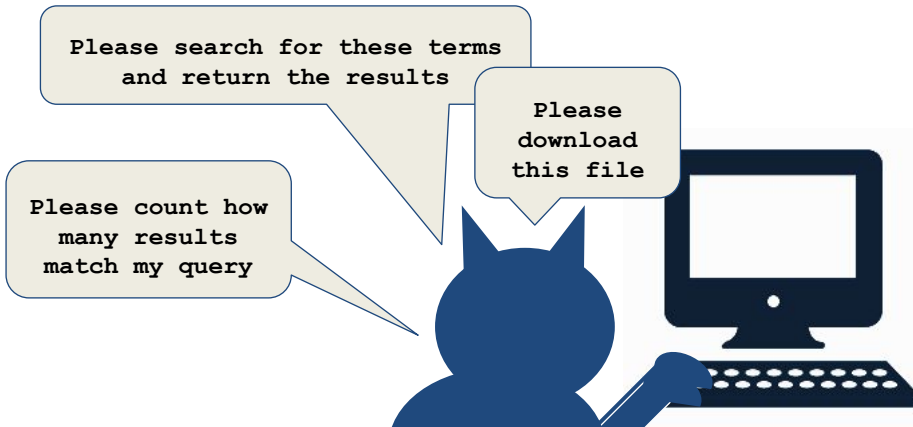
↳ Let me ask iDigBio to download certain specimen records

RECORDING SCRIPT: Now that we're sold on why APIs are useful, I want to clarify a few additional terms that you'll hear in relation to APIs: "endpoint" and "request." An endpoint is simply an address where you can find a particular API. Quite often, an API endpoint is a URL like the examples on this slide. It's a good practice for organizations to make API endpoints interpretable by humans, for example when you look at the Google API endpoints here it's pretty easy to notice that we're looking at two different APIs, one for "directions" and one for "geocode."



How do I talk to an API?

You talk to an API by making a request, either through a user interface or programmatically, e.g. via R or Python.



RECORDING SCRIPT: A request is what allows you to talk to an API. You can make a request either through a user interface, as we saw in the earlier example of the iDigBio web portal, or programmatically via a language like R or Python. If you are new to APIs, or new to programming in general, it can be helpful to start by thinking of your request as a regular sentence, keeping in mind that you need to be clear about what you want. You can then take that sentence and translate it into code so that the API can understand.



What does a request look like?

You can talk to an API by making a request, often including parameters, to an endpoint

directly in your browser

request

results

[https://search.idigbio.org/v2/search/records?rq={\"genus\":\"acer\"}](https://search.idigbio.org/v2/search/records?rq={\)

RECORDING SCRIPT: When API endpoints are URLs, you can make a request directly via your web browser rather than from a programming environment like R or Python. You can test this out for the iDigBio search API by going to the URL on this slide. Notice that we've already seen the first part of this URL (in bold) on the earlier slide when we learned what an endpoint is. The second part of the URL is new, and it is the part where we are making our request, in this case, that we want to retrieve specimen records where the genus is "acer." If you go to this URL, your browser will show you the results of this request, which are contained in JSON. Some browsers, like Firefox and Chrome, will automatically format the JSON so that it is easy to read, as shown in the screenshot here.



What does a request look like?

You can talk to an API by making a request, often including parameters, to an endpoint

in a programming environment, like R

```

RStudio
Source
Console ~/Documents/GitHub/nsf-rapid-grant-2033973/
> idig_search_records(rq = list(genus = "acer"))
      uid occurrenceid
1 000095c8-592c-497d-83e8-c5a1fd26cd http://n2t.net/ark:/65665/3b3822d1b-4355-4059-9767-d501064643b1
2 0000f63b-42ea-4c06-9c55-d42e911841e4      8c58d112-67d2-4a88-a6f9-a4c152a99f82
3 00017511-04f9-4d1c-9ce4-2b3ef70a38b5 http://coldb.mnhn.fr/catalognumber/mnhn/p/p05188416
4 0002e0d5-d731-4f81-9385-bac039a21113 urn:uuid:8a20ea2b-cf62-4f58-8e95-b90436f4aa0d
5 00034fd0-bc75-46df-94ca-09bc9ebee2e 25da76e1-5264-4487-bb6c-a39da693b1fd
6 00037cb2-5f3b-43f6-a274-fda6dab6f372 ac6f5df3-e107-4b04-8e98-837abe298288
7 00037f84-2063-44e5-a5c9-5809a7be4b3e http://ucjeps.berkeley.edu/cgi-bin/new_detail.pl?rsa819912&related=yes
8 00041678-5df1-4a23-ba78-8c12f60af369 b275f928-5c0d-4832-ae82-fde363d8fde1
9 00047545-1ac4-457f-8a99-5c90b77c7ebc b28d301a-8db6-4c19-9a04-488ef2d7826b
10 00060e1b-9ca1-4e55-bb8a-1852bf6e85c0 4daf22b-d451-40bd-b847-50b4e10ef110
11 00072caf-0f24-447f-b68e-a20299f6afc7 40428b90-27a5-11e3-8d47-005056be0003
12 00087574-9941-49b2-a3db-92a48ceed3aa urn:catalog:ucmp:p:9859
  
```

RECORDING SCRIPT: Now we have the same request but coming from RStudio. You'll notice that the syntax we are using to call the API is totally different, but elements of it are the same, like we still see "genus = acer." And now instead of seeing results in our browser we are seeing them in our programming environment, which means we can store them in an object and use them however we like in our research data pipeline.



What is an API useful for?

An API facilitates programmatic data access, e.g. using the iDigBio API in R

*Document procedures
within your code*

Automate data retrieval

*Run analyses in a way
that is reproducible*

```
# Load package to access iDigBio API
library(ridigbio)

# Retrieve records for specimens identified as being in the genus Acer and
# collected in Utah
data <- idig_search_records(rq = list(genus = "acer",
                                     stateprovince = "utah"),
                           fields = c("uuid",
                                     "institutioncode",
                                     "collectioncode",
                                     "catalognumber",
                                     "stateprovince",
                                     "county",
                                     "locality",
                                     "geopoint"),
                           limit = 1000)

# Begin data cleaning and analysis...
```

RECORDING SCRIPT: So to recap, APIs facilitate programmatic data access, which can be an essential part of creating a reproducible research workflow. This slide shows an example of a script in R where we are documenting our procedures directly within the code, automating our data retrieval, and then moving on to running our analyses.



Who has APIs for me to use?

- **Consortium of Universities for the Advancement of Hydrologic Science (CUAHSI)**
 - <https://www.cuahsi.org/data-models/for-developers/>
 - Various clients
- **DataONE**
 - <https://dataone-architecture-documentation.readthedocs.io/en/latest/>
 - Clients for Java, Python, R, MATLAB
- **Environmental Data Initiative**
 - <https://environmentaldatainitiative.org/>
- **Global Biodiversity Information Facility (GBIF)**
 - <https://www.gbif.org/developer/summary>
 - Clients for Python, R
- **iDigBio**
 - https://www.idigbio.org/wiki/index.php/IDigBio_API
 - Clients for Python, R
- **NEON**
 - <https://data.neonscience.org/data-api/>
 - Client for R
- **Neotoma**
 - <https://api.neotomadb.org/doc/use>
 - Client for R

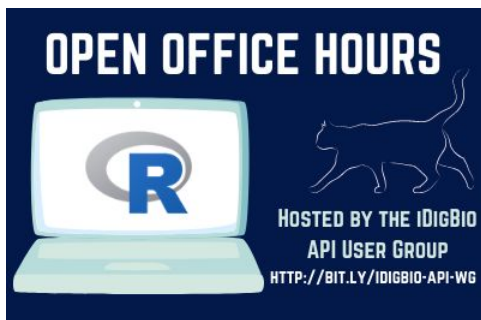
RECORDING SCRIPT: There are so many APIs. On this slide, I'm highlighting APIs provided by the Data Help Desk collaborators, and am also noting which APIs have clients for what programming languages. In general, if you find yourself looking for data from an online source, it's worth checking to see if that source has an API. They are becoming increasingly common and so the answer is likely yes. Even if an organization does not have a client for your preferred programming language, you can still use their API. The clients often just provide streamlined functionality.



Resources to learn more

iDigBio office hours: an informal drop-in session where anyone is welcome to bring their questions or ideas about using tools such as the iDigBio API to work with biodiversity occurrence data.

Every 2nd and 4th Wednesday of the month at 3:30pm Eastern.



- Focus on using the R language
- Often will do code demos but experience is not required or expected
- More details at <https://bit.ly/2Z5iYul>
- Example API code at <https://bit.ly/bio-spm-data>

RECORDING SCRIPT: If you want to learn more about the iDigBio APIs, we host an informal office hour twice monthly, and we also have a GitHub repository of example code snippets for people to reuse. You can find more details about our office hours and this code snippet repo via the links on this slide.



Resources to learn more

Data Help Desk

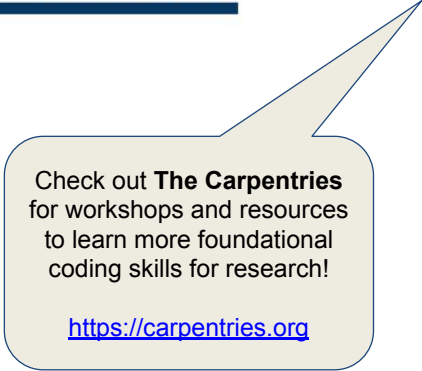


ESA Career Central Live Q&A
on Wednesday, August 5th, 12:30 - 1:00 Eastern

Data Help Desk Wiki

<https://bit.ly/datahelpesa2020>

#DataHelpDesk on Twitter



Check out **The Carpentries**
for workshops and resources
to learn more foundational
coding skills for research!

<https://carpentries.org>

RECORDING SCRIPT: More broadly, we encourage you to come to our Data Help Desk live Q&A on Wednesday, August 5th from 12:30 to 1pm Eastern. You can also find links to more resources on our Data Help Desk wiki, and ask more questions by calling us out on Twitter using the #DataHelpDesk hashtag. If you are looking to gain foundational coding skills, check out The Carpentries, which organize workshops and other training resources.



Thank you

To all of our Data Help Desk Partners for their collaboration in preparation for ESA 2020.

-

To Ron Canepa & Michelle Gaynor for their leadership in the iDigBio API working group.

-

To Deb Paul & Cat Chapman for their support in representing iDigBio at ESA 2020.

Data Help Desk



iDigBio is funded by grants from the National Science Foundation's Advancing Digitization of Biodiversity Collections Program [DBI-1115210 (2011-2018) and DBI-1547229 (2016-2021)]. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

RECORDING SCRIPT: Thank you for joining me today in this brief intro to APIs, and we hope to see you in other ESA Data Help Desk activities this week!