ORCID US Community Call - Using R to Find ORCID Holders Currently Affiliated with your Organization

- Tues. Feb. 16 @ 2-3:30 Eastern/ 1-2:30 Central/ 12-1:30 Mountain/ 11-12:30 Pacific
- **Recording:** https://vimeo.com/575385098 (PW: ORCID**)**
- **Notes** are at the bottom of this doc
- **To prepare (optional):**
    - If you would like to follow along during the demo, you will need to have R Studio installed on your computer (download the Free version of R Studio here: https://rstudio.com/products/rstudio/download/) and you will need to have completed the steps listed in the "Setting up rorcid" section of this tutorial: https://ciakovx.github.io/rorcid.html#Setting_up_rorcid
    - To download the R script
        - As an R file: https://drive.google.com/file/d/1i56Ueo0fmL_53eoENItMp0yyELXFhQ9V/view?usp=sharing
        - As a text file: https://drive.google.com/file/d/1uJ985wWSKgas9ew9juKl_vouNIeL0Esa/view?usp=sharing
        - Note: If your organization has more than one name that is commonly used, or if you want to also search for a FundRef ID in addition to Ringgold and GRID, contact sheila.rabun@lyrasis.org
    - To get the GRID ID and Ringgold ID for your organization:
        - GRID – go to https://www.grid.ac/institutes and search for your organization.
        - Ringgold – because Ringgold is proprietary, you would need to register as a guest on their site to use their free ID lookup: https://www.ringgold.com/identify-online-guests/ A workaround will be demonstrated on the call. You can also email sheila.rabun@lyrasis.org and I can look up your Ringgold ID for you.
    - You might want to take a look at the additional possibilities outlined in Clarke Iakovakis' larger Rorcid tutorial: https://ciakovx.github.io/rorcid.html - note that on this call the focus will be a demo on using an R script (below) to find current employees with ORCID iDs at your organization. We may cover other topics if time allows, but will not be going through the entire tutorial.

Present:

- Sheila Rabun (ORCID US Community Specialist at LYRASIS) sheila.rabun@lyrasis.org
- Clarke Iakovakis (Scholarly Services Librarian, Oklahoma State University)
- + 46 attendees

Agenda:

- Welcome (Sheila)
    - Logistics of searching in ORCID

        - Public vs. Private vs. Trusted Parties visibility
        - Public API vs. Member API
    - Looking up your organization's Ringgold ID and GRID ID
- Rorcid intro and demo (Clarke)
    - Intro to R
    - Intro to Rorcid
    - Finding current employees

Basic R script for finding current employees (students script is included down below):

```
# See https://ciakovx.github.io/rorcid.html for a full walkthrough, including explanatory text

# See https://mybinder.org/v2/gh/ciakovx/ciakovx.github.io/master?filepath=rorcid.ipynb for a
# Binder link to a Jupyter Notebook to try this out in your browser

# Install and load packages ----------------------------------------------

# you will need to install these packages first, using the following
# if you've already installed them, skip this step
install.packages('rorcid')
install.packages('tidyverse')
install.packages('usethis')
install.packages('anytime')
install.packages('janitor')
install.packages('glue')

# load the packages
library(rorcid)
library(usethis)
library(tidyverse)
library(anytime)
library(lubridate)
library(janitor)
library(glue)

# authorize ORCID API (should prompt a window to open in your browser to login to your
ORCID account)
orcid_auth()

# build the query  -------------------------------------------------------

ringgold_id <- "enter your institution's ringgold"
```

```
grid_id <- "enter your institution's grid ID"
email_domain <- "enter your institution's email domain"
organization_name <- "enter your organization's name"

# example
# ringgold_id <- "7618"
# grid_id <- "grid.65519.3e"
# email_domain <- "@okstate.edu"
# organization_name <- "Oklahoma State University"

# create the query
my_query <- glue('ringgold-org-id:',
                 ringgold_id,
                 ' OR grid-org-id:',
                 grid_id,
                 ' OR email:*',
                 email_domain,
                 ' OR affiliation-org-name:"',
                 organization_name,
                 '"')

# get the counts
orcid_count <- base::attr(rorcid::orcid(query = my_query),
                          "found")

# create the page vector
my_pages <- seq(from = 0, to = orcid_count, by = 200)

# get the ORCID iDs
my_orcids <- purrr::map(
  my_pages,
  function(page) {
    print(page)
    my_orcids <- rorcid::orcid(query = my_query,
                               rows = 200,
                               start = page)
    return(my_orcids)
  })

# put the ORCID iDs into a single tibble
my_orcids_data <- my_orcids %>%
  map_dfr(., as_tibble) %>%
  janitor::clean_names()
```

```
# get employment data ----------------------------------------------------

# get the employments from the orcid_identifier_path column
# be patient, this may take a while
my_employment <- rorcid::orcid_employments(my_orcids_data$orcid_identifier_path)

# extract the employment data from the JSON file and mutate the dates
my_employment_data <- my_employment %>%
  purrr::map(., purrr::pluck, "affiliation-group", "summaries") %>%
  purrr::flatten_dfr() %>%
  janitor::clean_names() %>%
  dplyr::mutate(employment_summary_end_date =
anytime::anydate(employment_summary_end_date/1000),
                employment_summary_created_date_value =
anytime::anydate(employment_summary_created_date_value/1000),
                employment_summary_last_modified_date_value =
anytime::anydate(employment_summary_last_modified_date_value/1000))

# clean up the column names
names(my_employment_data) <- names(my_employment_data) %>%
  stringr::str_replace(., "employment_summary_", "") %>%
  stringr::str_replace(., "source_source_", "") %>%
  stringr::str_replace(., "organization_disambiguated_", "")

# view the unique institutions in the organization names columns
# keep in mind this will include all institutions a person has in their employments section
my_organizations <- my_employment_data %>%
  group_by(organization_name) %>%
  count() %>%
  arrange(desc(n))

# you can also filter it with a keyword:
my_organizations_filtered <- my_organizations %>%
  filter(str_detect(organization_name, "Oklahoma"))

# filter the dataset to include only the institutions you want.
# As you can see in the below example, there may be messiness in the hand-entered ones
# See example:
my_employment_data_filtered <- my_employment_data %>%
  dplyr::filter(organization_name == "Oklahoma State University Stillwater"
                | organization_name == "Oklahoma State University Tulsa"
                | organization_name == "Oklahoma State University"
```

```
            | organization_name == "Oklahoma State University "
            | organization_name == "Oklahoma State University System"
            | organization_name == "Oklahoma State University Oklahoma Agricultural
Experiment Station"
            | organization_name == "Oklahoma State University Center for Veterinary Sciences"
            | organization_name == "Oklahoma State University, Stillwater"
            | organization_name == "College of Veterinary Medicine, Oklahoma State University"
            | organization_name == "Interim Dean, College of Education, Health & Aviation,
Oklahoma State University"
            | organization_name == "Oklahoma state university")

# finally, filter to include only people who have NA as the end date
my_employment_data_filtered_current <- my_employment_data_filtered %>%
  dplyr::filter(is.na(end_date_year_value))

#write to CSV
write_csv(my_employment_data_filtered_current, "C:/Users/rabun/Desktop/Employment.csv")

# Education ----------------------------------------------------------

# you can also get data on people whose degree information includes your university
# then filter that to get current students
my_education <- rorcid::orcid_educations(my_orcids_data$orcid_identifier_path)

# then generally follow the steps above, making modifications to variable names as necessary.
my_education_data <- my_education %>%
  purrr::map(., purrr::pluck, "affiliation-group", "summaries") %>%
  purrr::flatten_dfr() %>%
  janitor::clean_names() %>%
  dplyr::mutate(education_summary_end_date =
anytime::anydate(education_summary_end_date/1000),
            education_summary_created_date_value =
anytime::anydate(education_summary_created_date_value/1000),
            education_summary_last_modified_date_value =
anytime::anydate(education_summary_last_modified_date_value/1000))

names(my_education_data) <- names(my_education_data) %>%
  stringr::str_replace(., "education_summary_", "") %>%
  stringr::str_replace(., "source_source_", "") %>%
  stringr::str_replace(., "organization_disambiguated_", "")

my_education_organizations <- my_education_data %>%
  group_by(organization_name) %>%
```

```
  count() %>%
  arrange(desc(n))

my_education_data_filtered <- my_education_data %>%
  dplyr::filter(organization_name == "Oklahoma State University Stillwater"
                | organization_name == "Oklahoma State University Tulsa"
                | organization_name == "Oklahoma State University"
                | organization_name == "Oklahoma State University "
                | organization_name == "Oklahoma State University System")

my_education_data_filtered_current <- my_education_data_filtered %>%
  dplyr::filter(is.na(end_date_year_value))

#write to CSV
write_csv(my_education_data_filtered_current, "C:/Users/rabun/Desktop/Education.csv")
```

Notes:

- Whether you are using the Public or Member API, searching in ORCID will only return “Public” data. If someone does not have data in their ORCID record set to public visibility, they will not show up in your search results.
- Data returned in searches should not be used to load ORCID iDs into systems. It should only be used for assessment and planning purposes. If you want to get your researchers’ ORCID iDs to display in your systems or read/write data, you need to use the ORCID Member API and OAuth process, described here: https://youtu.be/bQGN4KrNrLY
- You do not need a computer science/programming background to use R - R has been around since the mid-90s
- This web page explains in detail about how to use R with ORCID: https://ciakovx.github.io/rorcid.html
- Shout out to rOpenSci
- 4 panes in R Studio:
    - Upper left = script pane - where commands are entered
    - Bottom left = console pane - results of the script will appear here
    - Upper right = environment pane - starts blank, more info will appear as you run scripts
    - Bottom right = navigation pane - help info, packages, visualizations
- R is basically a calculator, interface to APIs to pull in data from external systems, stats analysis, write programs in R, clean data, visualize data, etc.
- Today’s focus is R as a command line - typing out expressions and then evaluating expressions
- Packages extend the functionality of R - the first step with the R script is to install and load several packages. Packages only need to be installed once, but they need to be loaded each time you open R.

- You do need to authorize the ORCID API - get a token to authorize R to access the ORCID API. Once you authorize once, you won't have to authorize again. Instructions for authorizing with the ORCID API: https://ciakovx.github.io/rorcid.html#Setting_up_rorcid
- Set your values for Ringgold, GRID, email domain, and organization name - the character values will be stored in your R session. Basically we are "gluing" these values together to run a query.
- Hint - Ctrl+L will clear out the console for a cleaner view.
- Query is in Solr, a query language based on java
- Query will look in all sections of the ORCID record first - casting a wide net of matching results - at first you will only get the number of results, then based on those results, we will filter to find only people that have your organization in their "employment" section for example and get the actual data.
- We can only get 200 results at once, so we have to cycle through a series of 200 results at a time.
- To get the employments, we are getting the employment information from every ORCID record returned in the wide search. We then need to filter to only include employments that have the name of the institution in question.
- Dates in ORCID API appear in Unix time (number of seconds that have elapsed since the 1970s?) so we need to clean the dates - the script is set up to convert the dates to something we can understand.
- Organization names in the employment section need to be filtered to only include your organization's name(s) - make sure to include all of the different name variations that people might have on their ORCID record. There will probably be mis-spellings and lots of variations, so be aware of that.
- The final step is to filter the results to only include records where there is no end date in their employment affiliation. We can assume that no end date means the person is still employed at your organization.
- Contact sheila.rabun@lyrasis.org if you have any questions or run into errors, etc.