



Go.Data SOP

Retrieving your data collections from API for advanced analysis

Goal of this document:

This SOP will briefly outline the instructions for **obtaining your data directly from your Go.Data instance application programming interface (API)**. This process will provide you with cleaned, flattened excel files for advanced analysis in a program of your choice. Although there are multiple ways to retrieve the data collections including installing and connecting directly to the MongoDB database on your machine, this SOP outlines how to do this simply using the **open-source software R** – where advanced R skills are not required.

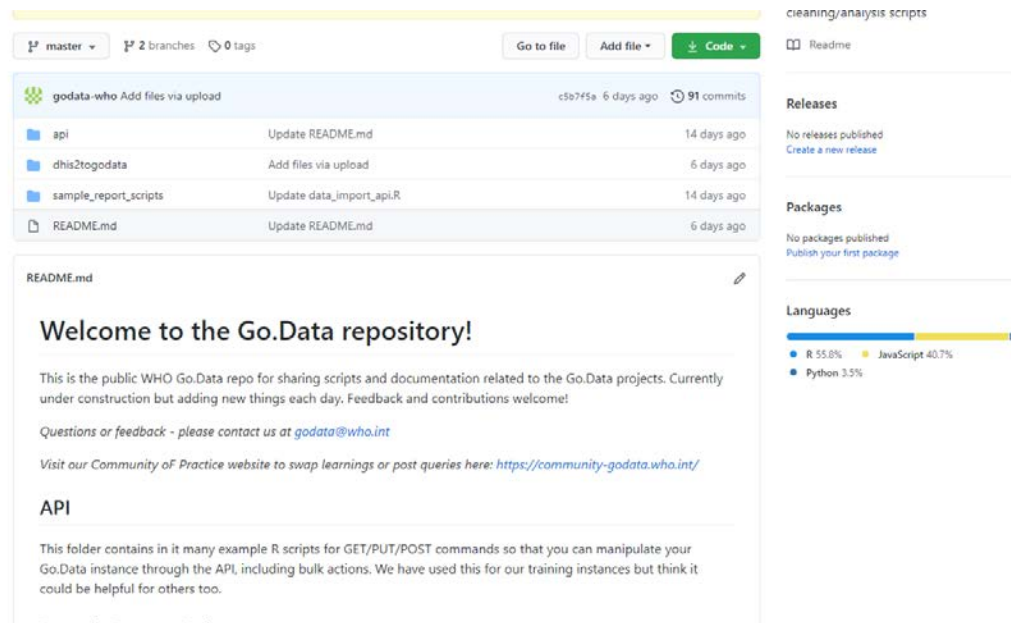
Requirements:

- R (version 3.6.1 or higher) – download here: <https://www.r-project.org/>
- Valid Go.Data login credentials for the instance you are accessing
- Folder directory with scripts downloaded onto your computer, accessed from Go.Data Github repository: https://github.com/godata-who/godata/tree/master/sample_report_scripts

For questions please email godata@who.int

Step 1. Replicate folder directory to your local machine

Navigate to godata [Github repository](#). Copy the [Go.Data analytics folder hierarchy](#) (labeled **sample_report_scripts**) and all its contents onto your local `://C` drive to ensure that there are no issues with sourcing scripts and saving outputs to the proper locations when you are in your R session.

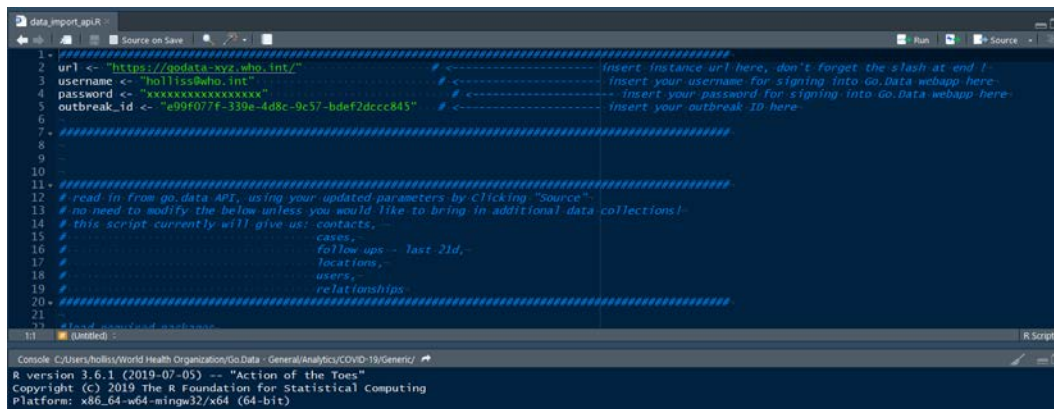


The folder hierarchy contains a **godata.Rproj** file from where you can double-click to start your R session, **report_sources** folder that contains R scripts necessary to execute, **report_outputs** folder where your HTML reports will be saved to, **data** folder where your cleaned data will be saved to, and **scripts** folder that has additional functions or scripts that the **report_sources** are sourcing from but that you do not need to run separately.

Name	Status	Date modified	Type	Size
godata.Rproj	✓	18-Jun-20 19:20	R Project	1 KB
report_outputs	✓	08-Jun-20 15:27	File folder	
data	✓	08-Jun-20 14:28	File folder	
report_sources	✓	08-Jun-20 14:28	File folder	
scripts	✓	08-Jun-20 14:28	File folder	

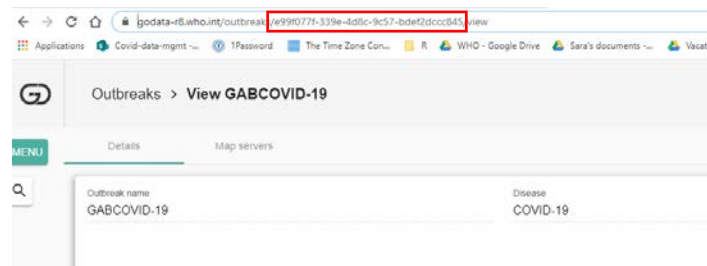
Step 2. Run data_import_api script with your Go.Data credentials

Open up R project by double-clicking on **godata.Rproj**. Navigate to **01_data_import_api.R** in the **report_sources** folder and open it in your R console. At the top of the script, fill in the appropriate **URL**, your Go.Data **username** and **password**, and **outbreak_id** of interest.

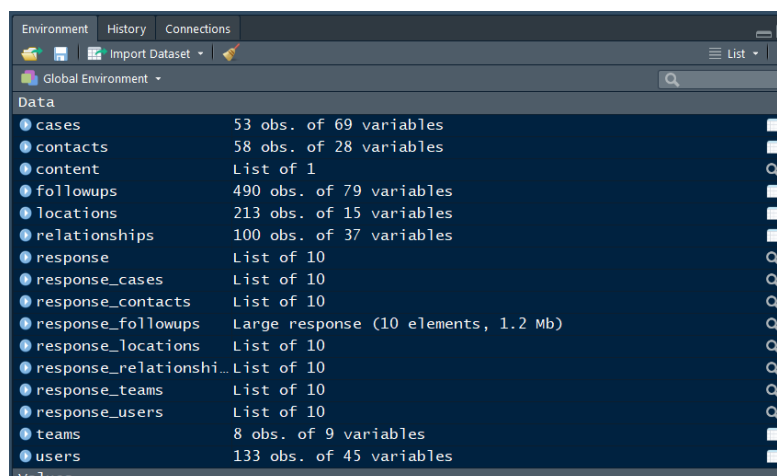


```
1. #####
2. url <- "https://godata-wva.who.int/" # ----- insert instance url here, don't forget the slash at end !
3. username <- "hol1ss@who.int" # ----- insert your username for signing into Go.Data webapp here
4. password <- "xxxxxxxxxxxx" # ----- insert your password for signing into Go.Data webapp here
5. outbreak_id <- "e99f077f-339e-4d8c-9c57-bdef2dccc845" # ----- insert your outbreak ID here
6. #####
7. #####
8. #####
9. #####
10. #####
11. #####
12. # read in from go.data API, using your updated parameters by clicking "source"
13. # no need to modify the below unless you would like to bring in additional data collections!
14. # this script currently will give us: contacts,
15. # cases,
16. # followups -- last 21d,
17. # locations,
18. # users,
19. # relationships
20. #####
21. #####
22. # read downloaded packages
23. #####
```

In order to obtain your outbreak ID, navigate to View Outbreak in Go.Data and you can find it in the URL.



At the end of this script, you should have created the following data frames in your R global environment that will be used in subsequent cleaning scripts: **cases**; **contacts**; **followups**; **locations**; **outbreaks**; **relationships**; **teams**; **users**



NOTE: please switch your language to English in your Go.Data instance before running this API script, to ensure core data elements are all brought back in a consistent form

Step 3. Run cleaning script to clean and export dataframes

The dataframes as retrieved straight from API contain many nested arrays in lists. In order to properly unnest relevant fields and do some basic data manipulation to these data frames before exporting to .CSV (or prepping for additional analysis in R):

Navigate to **02_clean_data_api.R** (also in **report_sources** folder) and run script



```
1 ## This script is for cleaning the Go data data for COVID - core variables
2
3
4 # check to see which of these are actually needed, take out those that are not
5 path_to_functions <- here::here("scripts")
6 scripts_files <- dir(path_to_functions, pattern = ".R$", full.names=TRUE)
7 for (file in scripts_files) source(file, local = TRUE)
8
9 ## Load data from API R tibbles
10
11 ## We use rio package (r input output) for reading in the data
12
13 database_date <- Sys.Date()
14
15 followups <- followups %>%
16   as_tibble()
17
18 cases <- cases %>%
19   as_tibble()
```

This will result in the following cleaned .csv files saved in the **data > clean** folder:

- contacts_clean_[today's date].csv
- cases_clean_[today's date].csv
- followups_clean_[today's date].csv
- relationships_clean_[today's date].csv
- locations_clean_[today's date].csv

NOTE: these cleaning scripts focus on the CORE data variables and not custom questionnaire variables, as questionnaires are configurable for each country or institution deploying Go.Data. No core data elements (those living outside of questionnaires) should need updating in terms of coding; however, if you would like to pull in additional questionnaire data elements you may need to slightly modify this script to accommodate these extra fields. Additionally, it is possible that your location hierarchy or team structure may vary in your deployment setting (i.e. supervisor registered at a different admin level) so changes may need to be made to the location cleaning scripts. If you encounter any issues here please reach out to your Go.Data focal point and they can assist to help you troubleshoot.

The cleaned datasets will now be much easier to do additional analysis whether inside or outside of R. We have created some sample scripts to get you started in some basic dashboard analyses (see, for example, **03_daily_dashboard_today_current.Rmd** for a ready-made HTML dashboard that will give you stats on a range of operational metrics to be monitored by supervisor and contact tracer on a daily basis). Screenshots below show some of these graphics, such as contact follow-up status by a given admin level and a follow-up table by supervisor. Additionally, you can build on these reports to make more customized analyses.



Location	Date du rapport	Cas ID	% de variables complets	Fardeau syndromique	Isolée	Non
Beni	2020-04-27	rdcmgn072978ben	80%	6	oui	
Beni	2020-04-17	rdcmgn072402ben	70%	7	oui	
Beni	2020-04-16	rdcmgn072756ben	75%	4	non	
Beni	2020-04-15	rdcmgn072482ben	80%	7	oui	
Kasanga	2020-04-13	rdcmgn072123ben	70%	2	oui	
Kasanga	2020-04-12	rdcmgn070239ben	70%	12	non	
NA	2020-04-10	rdcmgn072075ben	60%	-	non	

Go Data Rapport d'Opérations Quotidiennes - Tableau de bord - Aujourd'hui - Sunday 03 May 2020 - Contacts - Cases - Administration - Plus +											
<div>739</div> <div>Total Contacts Anés</div>	<div>Total de contacts par superviseur - tableau</div> <div>Active Contacts by Area</div> <div>Contacts by Supervisor</div> <div>Contacts by RUCCO</div>										
	Statut de la visite d'aujourd'hui							De contacts actifs			
	Superviseur	Alira de Santi	Contacts actifs	Vu avec signes	Vu sans signes	Non vu	Plus d'action	Sans première semaine	Sans deuxième semaine	Perte vu	Annulé vu
<div>0</div> <div>Contacts Perte Vu</div>	Jean Bosco Karera Mutsakumbi	Bolere	1	0	1	0	0	1	0	0	0
	<i>accus correctement attribué</i>	Kumfi	61	0	60	0	1	0	0	0	36
	<i>accus correctement attribué</i>	Futuli	116	1	118	5	4	0	0	0	36
<div>267</div> <div>Contacts Annulé Vu</div>	<i>accus correctement attribué</i>	Kanazimuli	5	0	0	0	5	5	0	0	5
	Omer Kananzi Muthoko	Kanzimbuli	29	0	24	0	5	5	0	0	13
	<i>accus correctement attribué</i>	Kasibigolia	30	0	10	0	0	0	0	0	0
<div>0</div> <div>Contacts Perte Vu</div>	<i>accus correctement attribué</i>	Kwanga	245	1	245	2	136	0	0	0	43
	Jackson Karera Mutyeyi	Kazanga	71	0	62	2	0	71	0	0	6
	<i>accus correctement attribué</i>	Mababanga	5	0	5	0	0	0	0	0	6
<div>0</div> <div>Contacts Perte Perte Annulé</div>	<i>accus correctement attribué</i>	Machuridie	2	0	2	0	0	0	0	0	2
	<i>accus correctement attribué</i>	Maipe	36	0	36	0	0	0	0	0	35
	Rukia Kabalo Mutho	Maipe	30	0	20	1	1	2	0	0	6
<div>81</div> <div>Contacts Vu Avec Signes</div>	Milusi Muhanda Mutsomanda	Ngeridie	6	0	6	0	0	0	0	0	0
	Chico Karera Kikanda	Nwagerima	6	0	5	0	1	1	0	0	5
	Jean Wera Palulu	Nwagerima	20	0	20	0	2	0	0	0	20
<div>0</div> <div>Contacts Vu Avec Signes</div>	<i>accus correctement attribué</i>	Isamanda	7	0	7	0	0	0	0	0	7

Please check back frequently at our [Community of Practice](#) site where we will be maintaining active discussions across the user base on different analytics learnings and methodologies that can be applied across settings.