Histlabapi

Histlabapi is a collection of R functions used to interface with the History Lab API. The functions can be used to extract different types of data from the History Lab collections.

History Lab has a steadily growing collection of declassified government documents from a variety of sources. Presently, there are 9 different collections that can be searched: Foreign Relations of the United States (frus), the State Department Central Foreign Policy Files (statedeptcables), Kissinger telephone conversations from the time he was Secretary of State (kissinger), the Clinton e-mails from the time when she was Secretary of State (clinton), the President’s Daily Briefs (pdb), the Robert S. McNamara World Bank papers (worldbank), ddrs, cpdoc, and the British Cabinet papers (cabinet).

The names of **collections** that can be searched are: cpdoc, clinton, kissinger, statedeptcables, frus, ddrs, cabinet, pdb, worldbank

The valid **fields** that can be specified are: body, body\_html, body\_summary, chapt\_title, countries, collection, date, date\_year, date\_month, from\_field, id, location, nuclear, persons, topics, classification, refs, cable\_references, source, source\_path, cable\_type, subject, title, to\_field, tags, description, category, pdf, title\_docview, orighand, concepts, type, office,readability

*[Note: Right now, I hardcode the collections and fields in functions in the histlabapi\_utils.R file. I would like to eventually put the valid options for both in a yaml file.]*

The valid **entity.types** are: countries, persons, topics.

For options allowing multiple values, such as coll.names (in most cases), fields, IDs, etc., the list of values can be given either in quotation marks with a space, comma, or comma and space as separators or with the c() function.

Dates have to be specified in numeric format, but month, days, and years can be separated with a “/”, ”-“ or “.”. The dates are converted to Y-M-D values, but MDY or DMY or even DYM are acceptable.

Below we describe the 6 functions that make up the package.

*[Note: Each function has a run option which is used to tell R whether to actually retrieve the query from the API or to only return the url. Not running the query is the default so that I can make sure that the API endpoint looks correct and not overload the API. This option will be removed for the final version. It is also inconsistently coded. Sometimes I coded it as NULL by default and other times as FALSE.]*

**hlapi\_random**() **–** This function returns a list of random IDs from across the History Lab collections. The only allowable option is limit which controls the number of IDs returned.

**hlapi\_id(ids=NULL, fields=NULL, topics=FALSE, run = NULL, …)** – This function returns all of the fields for an ID or a list of IDs. It can also be used to return the topics associated with an ID or list of IDs. *[Note: I have already put an issue in GitHub to expand the query to return the persons associated with IDs and the countries associated with IDs, but neither is possible now.]*

The fields and the topics options cannot be used at the same time.

**hlapi\_overview (start.date=NULL,end.date=NULL, limit = 25,fields=FALSE,collections=FALSE, coll.name=NULL,entity.type=NULL, entity=FALSE,run=FALSE,...)**  - This function presents several different ways to get an overview of the History Lab collections.

Fields and collections cannot be used with any of the other options. Setting fields to true will return a list of all the fields available across the History Lab collections. Setting collections to true will return the list of collections that can be queried. *[Note: I would like an option to return the fields available for each collection because some of the fields are collection-specific. It has been raised as an issue on github.]*

If entity is true and start.date and end.date are not specified, the function returns a list of the entities available in a collection. If an entity.type is also listed, the function will return a count of each of the entities of that type for a collection. So specifying entity.type “countries” and collection “frus” will return a list of the countries available in FRUS with the number of times the country appears across all FRUS documents. *[Note: There is an issue with this API endpoint. For some reason, the page\_size option tells the API where to start the list in addition to the number of results to return. The issue has been raised on github.]* coll.name is required for either entity option.

Finally, if either a start.date or end.date is specified, the function assumes a top entity search. Coll.name and entity.type are required options. The function will not return the top 100 entities from a collection for the specific date range along with a count of the number of documents the entity appears in.

**hlapi\_search(s.text, limit = 25,coll.name=NULL,start.date=NULL,end.date=NULL,run=NULL,...)** - This function runs a full-text search across different collections for a given date range. Both the coll.name and the start.date/end.date options are required. The fields option is not allowed. Presently the fields returned by the full-text search are hard-coded in the Python code and cannot be adjusted. The s.text option works with a single word or with a multiple word phrase.

**hlapi\_date (date=NULL,start.date=NULL,end.date=NULL, limit = 25,fields=NULL,coll.name=NULL,run=FALSE,...)** – This function searches the API for documents on a given date or across a date range given by the start.date and end.date options.

**hlapi\_entity (entity.type, value, limit = 25,fields=NULL,coll.name=NULL,date=NULL,start.date=NULL,end.date=NULL,run=NULL,...)** – This function will return the documents that fit the criteria given by entity.type and value. The valid entity.types are countries, persons, and topics. Countries must be a 2 or 3-digit number that reflects a countries ISO-3166 code.

The results from the function could be limited by the coll.name, the date/start.date&end.date, or the fields options. Date and start.date/end.date cannot be specified together but all the other options can be.