Short report: Epigraphic Database Heidelberg using R

Antonio Rivero Ostoic
28-11-2019

This post is about accessing the "Epigraphic Database Heidelberg" (EDH), which is one of the longest running database projects in digital Latin epigraphy. The Epigraphic Database Heidelberg (EDH) database started as early as year 1986, and in 1997 the Epigraphic Database Heidelberg website was launched at `https:/edh-www.adw.uni-heidelberg.de` where inscriptions, images, bibliographic and geographic records can be searched and browsed online.

Despite the possibility of accessing the EDH database through a Web browser, it is many times convenient to get the Open Data Repository by the EDH through its public Application Programming Interface (API).

For inscriptions, the generic search pattern Uniform Resource Identifier (URI) is:

```
https://edh-www.adw.uni-heidelberg.de/data/api/inscriptions/search?par_1=value&par_2=value&par_n
    =value
```

with parameters `par` $1, 2, ...n$.

The response from a query is in a Java Script Object Notation (JSON) format such as:

```
{
  "total" : 61,
  "limit" : "20",
  "items" : [ ... ]
}
```

Accessing the EDH database using R

Accessing the EDH database Application Programming Interface (API) using R is possible with a convenient function that produces the generic search pattern Uniform Resource Identifier (URI). Hence, the function `get.edh()` allows having access to the data with the available parameters that are recorded as arguments. Then the returned Java Script Object Notation (JSON) file is converted into a list data object with function `fromJSON()` from the rjson package.

Basically, the function `get.edh()` allows getting data with the `search` parameter either from `"inscriptions"` (the default option) or else from `"geography"`. The other two options from the EDH database API, which are `"photos"` and `"bibliography"` may be implemented in the future.

The following parameter description is from `https://edh-www.adw.uni-heidelberg.de/data/api`:

Search parameters for inscriptions and geography

`province` get list of valid values at `https://edh-www.adw.uni-heidelberg.de/data/api/terms/province`, case insensitive

`country` get list of valid values at `https://edh-www.adw.uni-heidelberg.de/data/api/terms/country`, case insensitive

`findspot_modern` add leading and/or trailing truncation by asterisk *, e.g. findspot_modern=köln*, case insensitive

`findspot_ancient` add leading and/or trailing truncation by asterisk *, e.g. findspot_ancient=aquae*, case insensitive

`bbox` bounding box in the format bbox=minLong, minLat, maxLong, maxLat, example: `https://edh-www.adw.uni-heidelberg.de/data/api/inscriptions/search?bbox=11,47,12,48`

Just make sure to quote the arguments in `get.edh()` for the different parameters that are not integers. This means for example that the query for the last parameter with the two search options is written as

```
R> get.edh(search="inscriptions", bbox="11,47,12,48")
R> get.edh(search="geography", bbox="11,47,12,48")
```

Search parameters for inscriptions

`hd_nr` HD-No of inscription

`year_not_before` integer, BC years are negative integers

`year_not_after` integer, BC years are negative integers

`tm_nr` integer value (?)

`transcription` automatic leading and trailing truncation, brackets are ignored

`type` of inscription, get list of values at `https://edh-www.adw.uni-heidelberg.de/data/api/terms/type`, case insensitive

Search parameters for geography

`findspot` level of village, street etc.; add leading and/or trailing truncation by asterisk *, e.g. findspot_modern=koln*, case insensitive

`pleiades_id` Pleiades identifier of a place; integer value

`geonames_id` Geonames identifier of a place; integer value

Since with the `"inscriptions"` option the `id` "component" of the output list is not with a numeric format, then function `get.edh()` adds an `ID` at the beginning of the list with the identifier with a numerical format.

Hence, the query

```
R> get.edh(findspot_modern="madrid")
```

returns this truncated output:

```
$ID
[1] "041220"

$commentary
[1] " Verschollen. Mogliche Datierung: 99-100."

$country
[1] "Spain"

$diplomatic_text
[1] "[ ] / [ ] / [ ] / GER PO[ ]TIF / [ ] / [ ] / [ ] / ["

...
```

Having a numerical identifier is useful for plotting the results for example. However, it is possible to prevent this addition by disabling argument `addID` with `FALSE`.

```
R> get.edh(findspot_modern="madrid", addID=FALSE)
```

Finally, it is worth to mention that further extensions that the EDH database API may add in the future can be handled with similar arguments in function `get.edh()`.

Acronyms

API Application Programming Interface.

EDH Epigraphic Database Heidelberg.

JSON Java Script Object Notation.

URI Uniform Resource Identifier.