ELA Public API reference

This API reference is designed to interact with the public search API of the Eurasian Latin Archive (ELA) hosted at ela.unisi.it

All APIs responds with 200 and a structure like this:

```json
{
    "responseType": "OK",
    "messageCode": "OK",
    "value": []
}
```

Value can be a string, a JSON array or a JSON object.

Terminology used:

- `index` refers to the Elasticsearch index which holds all the informations and tags
- `core` is the backend communicating with Elasticsearch
- `field` an indexed entity
- `value` value of an indexed entity
- `CLTK` is the language processing engine used (Classical Languate ToolKit)

GET Get all searchable fields

http://ela.unisi.it/DasMemo/api/search/fields

Returns all the fields searchable in the index.

Every field has this format:

```json
{
      "name": "author",
      "description": "Document Author",
      "searchType": "header"
}
```

RESPONSE:

- `name` is the name of the tag
- `description` is a label describing what the tag is

- `searchType` can have value `header` or `body` and means that the relevant tag is in the TEI header or in the body of the document.

Ignore the tag with `name` equals to `---` which is a placeholder.

HEADERS

Content-Type	application/json

GET Get all tags from index

http://ela.unisi.it/DasMemo/api/list/tags

Returns a list of all tags in the index.

Every tag has this format:

json

```
{
      "externalName": "persName",
      "attributes": [
        {
          "externalName": "key",
          "value": "Bartholameus Redondo"
        }
      ],
      "cleanedValue": "Bartholameus Redondo"
    }
```

RESPONSE

- `externalName` is the tag name
- `attributes` is an object containing the various attributes of the tag
  - `externalName` is the attribute name
  - `value` is the attribute value
- `cleanedValue` is the formatted value of the tag without noise (tabs, appended or prepended spaces, ...)

HEADERS

Content-Type	application/json

POST Search

http://ela.unisi.it/DasMemo/api/search/search

Search the index for documents.

REQUEST

- `plainText` is the text to be searched on the index
- `lemmasText` is a space separated list of lemmas to be searched on the index
- `tags` is list of tags to be searched. In detail:
  - `searchType` is the type of search (included or not included). Can assume only `MUST` or `MUST_NOT`
  - `tagName` is the name of the tag to be searched (refer to tags list API)
  - `bodyHeader` which should assume `header` or `body` tells the core to search the relevant tag in the TEI header or in the TEI body
  - `value` is the tag value to be searched

All the information in the request (text, lemmas and tags) are combined using **AND** operator.

RESPONSE:

The results are store in the `results` array.

- `documentId` is the ID of the document
- `shortUrl` is the short code of the document
- `title` is the title of the document
- `author` is the author of the document
- `date` is the date of the document (don't expect an ISO formatted date here9

Other than the `results` array there is the `size` key that indicates the size of the array

HEADERS

Connection	keep-alive

Body raw (json)

json

```
{
    "plainText": "",
    "lemmasText": "",
    "tags": [
        {
            "searchType": "MUST",
            "bodyHeader": "header",
            "tagName": "author",
            "value": "Andreas de Perusio O.F.M."
        }
    ]
}
```

GET Get a document

http://ela.unisi.it/DasMemo/document/79bea2f1d1fc36bd05edb292ddf7897b

Return the XML of a document. Uses message code 200 in case of document found or 500 in other cases.

GET Download document

http://ela.unisi.it/DasMemo/api/document/download/79bea2f1d1fc36bd05edb292ddf7897b?format=XML

Downloads a document. Supported values for `format` are:

- `TXT`
- `PDF`
- `XML`

PARAMS

format	XML

GET Get document id by short code

http://ela.unisi.it/DasMemo/api/document/short/8cf12

Returns for the given short url the document id

GET Get CLTK informations

http://ela.unisi.it/DasMemo/api/cltk/document/79bea2f1d1fc36bd05edb292ddf7897b

Returns the CLTK information for the document.

RESPONSE

- `collocations` holds the collocations informations
  - `collocations_windowsize2` collocations of size 2
  - `collocations_windowsize3` collocations of size 3
  - `collocations_windowsize4` collocations of size 4

    - `collocations_windowsize5` collocations of size 5
- `fulltext_statistics` holds various statistics about the text
    - `words_number` is the number of words
    - `type_mean_lenght` is the mean lenght of the types
    - `types_number` is the number of types
    - `types_min_lenght` is the minimum lenght of the types
    - `type_list` is an array holding all the types
    - `word_frequencies` holds a key value map where the key is the word and the value the absolute frequency
    - `word_frequencies_case_insensitive` the same as above, but case insensitive
    - `word_list` is an array holding all the words
    - `word_list_lowercase` the same as above, but lowercase
    - `text` holds the plain text of the document
- `lemma collocations` is the same as `collocations` but for lemmas
- `tei_lists` contains information extracted during the analysis
    - `xmltei_places` contains a `list` object with all the tags of the places and a `frquequencies` object wich holds the occurences of the places
    - `xmltei_dates` holds the dates extracted, in a list of objects which have `value` (the value from the text) and `iso` (the normalized ISO value)
    - `xmltei_persons` is exactly like the `xmltei_places`
- `tei_attributes.xml_tei_attributes` holds the TEI information of the document (the information in the TEI header)
- `ngrams` contains the informations about the N-grams in the text
    - `ngrams2` N-grams of size 2
    - `ngrams3` N-grams of size 3
    - `ngrams4` N-grams of size 4
    - `ngrams5` N-grams of size 5
- `concordance` holds the text concordance
- `statistics` holds the same information of `full_textstatistic` but splitted for `paragraphs`, plus the `stop_frequencies` (stop words frequencies) and `lemma_frequencies (lemma frequencies)`

HEADERS

Content-Type	application/json

POST Compare CLTK informations

http://ela.unisi.it/DasMemo/api/cltk/compare

Compare the CLTK informations using buckets of documents.

REQUEST:

- `left` contains the documents IDs (comma separated) of the left bucket
- `right` contains the documents IDs (comma separated) of the left right bucket

RESPONSE:

- `left` contains the union of CLTK informations for the left bucket
- `right` contains the union of CLTK informations for the left bucket
- `leftMinusRight` contains the union of informations in the left bucket minus the union of the informations in the right bucket
- `rightMinusLeft` contains the union of informations in the right bucket minus the union of the informations in the left bucket
- `leftIntersectedRight` contains the intersection of the union of the information between the two buckets
- `leftJoinedRight` is the union of the unions of the two buckets

Every key here as the same format of the response of the CLTK information get.

Intersection and union are refer to insiemistic operations.

HEADERS

Accept	*/*
Accept-Language	it-IT,it;q=0.9,en-US;q=0.8,en;q=0.7
Connection	keep-alive
Content-Type	application/x-www-form-urlencoded
DNT	1
Origin	http://212.110.20.141:86
Referer	http://212.110.20.141:86/ela/
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36
X-Requested-With	XMLHttpRequest

Body urlencoded

left	de89da2bc2d921195cddd4d15a2325ac
right	e421b17a4c555bb0bf0f5807e57ad1f1