

# Package: OpenMindat (via r-universe)

January 15, 2025

**Type** Package

**Title** Quickly Retrieve Datasets from the 'Mindat' API

**Version** 1.0.0

**Imports** httr (>= 1.4.4), jsonlite (>= 1.8.4), readxl (>= 1.4.3),  
utils, stringi, stringr, usethis

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Maintainer** Xiang Que <xiangq@uidaho.edu>

**Description** The goal of OpenMindat R package is to provide functions  
for users or machines to quickly and easily retrieve datasets  
from the mindat.org API  
(<<https://api.mindat.org/schema/redoc/>>).

**VignetteBuilder** knitr

**Encoding** UTF-8

**LazyData** true

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.3

**URL** <https://github.com/quexiang/OpenMindat>,  
<https://quexiang.github.io/OpenMindat/>

**BugReports** <https://github.com/quexiang/OpenMindat/issues>

**License** MIT + file LICENSE

**Config/pak/sysreqs** git make libgit2-dev libicu-dev libssl-dev  
libx11-dev

**Repository** <https://quexiang.r-universe.dev>

**RemoteUrl** <https://github.com/quexiang/openmindat>

**RemoteRef** HEAD

**RemoteSha** 1358a5b6c9df2b1546b929b0887eaad998c5515b

## Contents

ConvertDF2JsonLD . . . . .	4
ConvertDF2TTL . . . . .	4
geomaterials_bi_greater_than . . . . .	5
geomaterials_bi_less_than . . . . .	6
geomaterials_bi_range . . . . .	6
geomaterials_by_groupid . . . . .	7
geomaterials_cleavagetype . . . . .	8
geomaterials_colour . . . . .	9
geomaterials_contain_all_but_not_elems . . . . .	9
geomaterials_contain_all_elems . . . . .	10
geomaterials_contain_any_but_not_elems . . . . .	11
geomaterials_contain_any_elems . . . . .	12
geomaterials_contain_only_elems . . . . .	12
geomaterials_crystal_system . . . . .	13
geomaterials_dens_greater_than . . . . .	14
geomaterials_dens_less_than . . . . .	15
geomaterials_dens_range . . . . .	16
geomaterials_diapheny . . . . .	16
geomaterials_entrytype . . . . .	17
geomaterials_expand . . . . .	18
geomaterials_field_exists . . . . .	19
geomaterials_fracturetype . . . . .	19
geomaterials_hardness_gt . . . . .	20
geomaterials_hardness_lt . . . . .	21
geomaterials_hardness_range . . . . .	22
geomaterials_ima . . . . .	22
geomaterials_ima_notes . . . . .	23
geomaterials_ima_status . . . . .	24
geomaterials_lustretype . . . . .	25
geomaterials_meteoritical_code . . . . .	25
geomaterials_name . . . . .	26
geomaterials_not_contain_elems . . . . .	27
geomaterials_optical2v_max . . . . .	28
geomaterials_optical2v_min . . . . .	28
geomaterials_optical2v_range . . . . .	29
geomaterials_opticalsign . . . . .	30
geomaterials_opticaltype . . . . .	31
geomaterials_polytypeof . . . . .	31
geomaterials_ri_gt . . . . .	32
geomaterials_ri_lt . . . . .	33
geomaterials_ri_range . . . . .	34
geomaterials_search_name . . . . .	34
geomaterials_streak . . . . .	35
geomaterials_synid . . . . .	36
geomaterials_updated_at . . . . .	37
geomaterials_varietyof . . . . .	37

geomeaterials_non_utf . . . . .	38
geomeaterials_ordering . . . . .	39
getExtension . . . . .	40
localities_list_all . . . . .	40
localities_list_country . . . . .	41
localities_list_description . . . . .	42
localities_list_elems_exc . . . . .	43
localities_list_elems_inc . . . . .	43
localities_list_elems_inc_exc . . . . .	44
localities_list_expand . . . . .	45
localities_list_txt . . . . .	46
localities_list_updated_at . . . . .	46
localities_retrieve_id . . . . .	47
localities_status_list . . . . .	48
localities_status_retrieve . . . . .	49
locality_type_retrieve . . . . .	49
locality_age . . . . .	50
locality_age_list . . . . .	51
locality_type_list . . . . .	51
mindat_build_querystring . . . . .	52
mindat_cache_delete . . . . .	53
mindat_cache_empty . . . . .	53
mindat_cache_get . . . . .	54
mindat_cache_has . . . . .	54
mindat_cache_return_or_setup . . . . .	55
mindat_cache_set . . . . .	55
mindat_connection . . . . .	56
mindat_countries . . . . .	56
mindat_country . . . . .	57
mindat_extract_response_body . . . . .	57
mindat_geomaterial . . . . .	58
mindat_geomaterial_list . . . . .	59
mindat_geomaterial_search . . . . .	59
mindat_geomaterial_varieties . . . . .	60
mindat_get_data_from_uri . . . . .	60
mindat_localities_list . . . . .	61
mindat_locality . . . . .	62
mindat_locality_status . . . . .	62
mindat_locality_status_list . . . . .	63
mindat_locality_type . . . . .	64
mindat_locality_type_list . . . . .	64
mindat_make_data_frame . . . . .	65
mindat_mineral_ima . . . . .	66
mindat_mineral_ima_list . . . . .	66
mindat_parse_raw_data . . . . .	67
mindat_query . . . . .	67
mindat_setup . . . . .	68
minerals_ima_list . . . . .	69

minerals_ima_list_expand . . . . .	69
minerals_ima_list_ima . . . . .	70
minerals_ima_retrieve . . . . .	71
minerals_ima_updated_at . . . . .	71
params_to_string . . . . .	72
saveMindatDataAs . . . . .	73
set_api_base . . . . .	73
set_api_token . . . . .	74
set_page_size . . . . .	74

<b>Index</b>	<b>75</b>
--------------	-----------

---

ConvertDF2JsonLD	<i>Output file as a given format</i>
------------------	--------------------------------------

---

### Description

Convert the mindat R dataframe to JSON-LD string

### Usage

```
ConvertDF2JsonLD(inputdata, template = NULL)
```

### Arguments

inputdata	R dataframe of retrieved data from Mindat database.
template	filepath to the template

### Examples

```
## Not run:
df <- geomaterials_search_name("Quartz")
df_out <- ConvertDF2JsonLD(df)

## End(Not run)
```

---

ConvertDF2TTL	<i>Convert a dataframe to a string of TTL format</i>
---------------	--

---

### Description

Convert the mindat R dataframe to TTL string

### Usage

```
ConvertDF2TTL (inputdata, template = NULL)
```

**Arguments**

inputdata        R dataframe of retrieved data from Mindat database.  
 template        filepath to the template

**Examples**

```
## Not run:
df <-geomaterials_search_name("Quartz")
df_out <-ConvertDF2TTL(df)

## End(Not run)
```

---

```
geomaterials_bi_greater_than
      retrieve the geomaterials whose birifrigence are higher than the given
      value.
```

---

**Description**

: Queries the list of geomaterials that minmum value of the given birifrigence value.

**Usage**

```
geomaterials_bi_greater_than(gt, ...)
```

**Arguments**

gt                float value.Birifrigence is calculated from refractive index as (rimax-rimin).Range:  
                   bi\_min - bi\_max.  
 ...              Further named parameters.Other optional arguments.

**Details**

This function related to the field "bi\_min" of geomaterials. retrieve all the geomaterials that has higher birifrigence than the given value(gt).

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_bi_greater_than(0.2)

## End(Not run)
```

---

geomaterials\_bi\_less\_than

*retrieve the geomaterials whose birifrigence are lower density than the given value.*

---

### Description

: Queries the list of geomaterials that have lower birifrigence than lt.

### Usage

```
geomaterials_bi_less_than(lt, ...)
```

### Arguments

lt                    float value. Birifrigence is calculated from refractive index as (rimax-rimin). Range: bi\_min - bi\_max.  
 ...                   Further named parameters. Other optional arguments.

### Details

This function related to the field "bi\_max" of geomaterials. retrieve all the geomaterials that has higher birifrigence than the given value(lt).

### Value

df, a data frame of geomaterials

### Examples

```
## Not run:
df <- geomaterials_bi_less_than(0.3)

## End(Not run)
```

---

geomaterials\_bi\_range *retrieve the geomaterials whose birifrigence are higher and lower than the given value.*

---

### Description

: Queries the list of geomaterials that have lower birifrigence than lt.

### Usage

```
geomaterials_bi_range(gt,lt, ...)
```

**Arguments**

gt float value. Birifrigence is calculated from refractive index as (rimax-rimin). Range: bi\_min - bi\_max.

lt float value. Birifrigence is calculated from refractive index as (rimax-rimin). Range: bi\_min - bi\_max.

... Further named parameters. Other optional arguments.

**Details**

This function related to the fields "bi\_min" and "bi\_max" of geomaterials. retrieve all the geomaterials that has the birifrigence within the given range of (gt,lt).

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <- geomaterials_bi_range(0.2, 0.3)

## End(Not run)
```

---

geomaterials\_by\_groupid

*retrieve the geomaterials by an given value of groupid.*

---

**Description**

: Queries the list of geomaterials that match an given groupid.

**Usage**

```
geomaterials_by_groupid(gid, ...)
```

**Arguments**

gid integer value. The id of the group to which this mineral belongs

... Further named parameters. Other optional arguments.

**Details**

This function related to the field "groupid" of geomaterials. retrieve all the geomaterials that match an given groupid.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_by_groupid(0)  
  
## End(Not run)
```

---

geomaterials\_cleavagetype  
*geomaterials that match an given cleavagetype*

---

**Description**

: Queries the list of geomaterials that match an given cleavagetype

**Usage**

```
geomaterials_cleavagetype(types, ...)
```

**Arguments**

types	vector of given cleavagetype (array of strings or null). The field "cleavage" is used to describe the crystallographic orientation of cleavage directions or planes and quality.
...	Further named parameters.Other optional arguments-Additional arguments.

**Details**

This function related to the field "cleavagetype" of geomaterials. Items Enum: "Distinct/Good" "Imperfect/Fair" "None Observed" "Perfect" "Poor/Indistinct" "Very Good"

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_cleavagetype(c("Poor/Indistinct"))  
  
## End(Not run)
```



---

geomaterials\_colour     *geomaterials that have the given colors*

---

**Description**

: Queries the list of geomaterials that match a given colors.

**Usage**

```
geomaterials_colour(colors, ...)
```

**Arguments**

colors                vector of given colors. colors of the mineral or rock - individual minerals at localities can also have color information.

...                    Further named parameters. Other optional arguments-Additional arguments.

**Details**

This function related to the field "colour" of geomaterials. For example: "Brown", "Yellow", "green", "Pink", "White", "Orange", "Blue", "Gold", "Dark brown", "Purple".

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <- geomaterials_colour(c("bright blue"))

## End(Not run)
```

---

geomaterials\_contain\_all\_but\_not\_elems  
*geomaterials that contain all of some given elements but without any  
of some other given elements.*

---

**Description**

Queries the list of geomaterials that contain all the given elements listed in icl\_elm\_vector, but do not contain the given elements listed in ecl\_elms\_vector.

**Usage**

```
geomaterials_contain_all_but_not_elems(icl_elm_vector, ecl_elms_vector, ...)
```

**Arguments**

`icl_elm_vector` vector of elements.  
`ecl_elms_vector`  
 vector of elements.  
 ... Further named parameters. Other optional arguments-Additional arguments.

**Details**

This function related to the field "elements\_inc" and "elements\_exc" of geomaterials. This function queries the list of geological materials that contain an given list of elements (`icl_elm_vector`), but not contain the other list of elements (`ecl_elms_vector`). It performs the query operation by calling the `mindat_geomaterial_list` function.

**Value**

`df`, a data frame of geomaterials.

**Examples**

```
## Not run:
df<-geomaterials_contain_all_but_not_elems (c('Fe','S'), c('O'),fields ="id,name,mindat_formula,elements")
## End(Not run)
```

---

```
geomaterials_contain_all_elems
  geomaterials_contain_all_elems
```

---

**Description**

retrieve the geomaterials that contain all of the elements. This function queries the list of geomaterials that contain all the given elements. It performs the query operation by calling the `mindat_geomaterial_list` function

**Usage**

```
geomaterials_contain_all_elems(icl_elms_vector,...)
```

**Arguments**

`icl_elms_vector`  
 vector of elements.  
 ... Further named parameters. Other optional arguments-Additional arguments that can be passed to the `mindat_geomaterial_list` function.

**Details**

This function related to the field "elements\_inc" of geomaterials.

**Value**

df, a data frame of geomaterials list.

**Examples**

```
## Not run:
df <-geomaterials_contain_all_elems (c('Fe', 'S'), fields ="id,name,mindat_formula,elements")

## End(Not run)
```

---

geomaterials\_contain\_any\_but\_not\_elems

*geomaterials that contain any of some given elements but with out any of some other given elements*

---

**Description**

: Queries the list of geological materials that contain any one of the given elements.

**Usage**

```
geomaterials_contain_any_but_not_elems(any_elems_vector, ecl_elms_vector, ...)
```

**Arguments**

any\_elems\_vector  
vector of elements. vector of any elements contained.

ecl\_elms\_vector  
vector of elements. vector of any elements excluded.

...  
Further named parameters.Other optional arguments-Additional arguments.

**Details**

This function related to the field "elements\_inc" of geomaterials. This function queries the list of geological materials that contain any element of an given list (any\_elems). It performs the query operation by looping through each given element and calling the mindat\_geomaterial\_list function.

**Value**

df, a data frame of geomaterials.

**Examples**

```
## Not run:
df <-geomaterials_contain_any_but_not_elems(c('Fe', 'S'), c('O'))

## End(Not run)
```

---

 geomaterials\_contain\_any\_elems

*geomaterials that contain any one of the given elements*


---

### Description

: Queries the list of geological materials that contain any one of the given elements.

### Usage

```
geomaterials_contain_any_elems(any_elems, ...)
```

### Arguments

any\_elems        vector of elements.  
 ...              Further named parameters. Other optional arguments-Additional arguments.

### Details

This function related to the field "elements\_inc" of geomaterials. This function queries the list of geological materials that contain any element of an given list (any\_elems). It performs the query operation by looping through each given element and calling the mindat\_geomaterial\_list function.

### Value

df, a data frame of geomaterials.

### Examples

```
## Not run:
df <-geomaterials_contain_any_elems (c('Fe', 'S'), fields ="id,name,mindat_formula,elements")

## End(Not run)
```

---

 geomaterials\_contain\_only\_elems

*geomaterials\_contain\_only\_elems*


---

### Description

retrieve the geomaterials that only contain elements in an given list (icl\_only\_elms\_vector).

### Usage

```
geomaterials_contain_only_elems (icl_only_elms_vector, ...)
```

**Arguments**

`icl_only_elms_vector`  
vector of elements.

... Further named parameters. Other optional arguments-Additional arguments that can be passed to the `mindat_geomaterial_list` function.

**Details**

This function related to the fields "elements\_inc" and "elements\_exc" of geomaterials. Here is a list of all elements that can make up geomaterials: 'H', 'Li', 'Be', 'B', 'C', 'N', 'O', 'F', 'Na', 'Mg', 'Al', 'Si', 'P', 'S', 'Cl', 'K', 'Ca', 'Sc', 'Ti', 'V', 'Cr', 'Mn', 'Fe', 'Co', 'Ni', 'Cu', 'Zn', 'Ga', 'Ge', 'As', 'Se', 'Br', 'Rb', 'Sr', 'Y', 'Zr', 'Nb', 'Mo', 'Ru', 'Rh', 'Pd', 'Ag', 'Cd', 'In', 'Sn', 'Sb', 'Te', 'I', 'Cs', 'Ba', 'La', 'Ce', 'Nd', 'Sm', 'Gd', 'Dy', 'Er', 'Yb', 'Hf', 'Ta', 'W', 'Re', 'Os', 'Ir', 'Pt', 'Au', 'Hg', 'Tl', 'Pb', 'Bi', 'Th', 'U' It performs the query operation by calling the `mindat_geomaterial_list` function

**Value**

df, a data frame of geomaterials.

**Examples**

```
## Not run:
df <-geomaterials_contain_only_elms(c('Fe', 'S'), fields ="id,name,mindat_formula,elements")

## End(Not run)
```

---

`geomaterials_crystal_system`

*geomaterials that have the given crystal*

---

**Description**

: Queries the list of geomaterials that have the given crystal system.

**Usage**

```
geomaterials_crystal_system(crystals, ...)
```

**Arguments**

`crystals` vector of given crystals. "crystal system of the mineral; "Amorphous", "Hexagonal", "Icosahedral", "Isomet  
... Further named parameters. Other optional arguments.

**Details**

This function related to the field "crystal\_system" of geomaterials. Items Enum: "Amorphous" "Hexagonal" "Icosahedral" "Isometric" "Monoclinic" "Orthorhombic" "Tetragonal" "Triclinic" "Trigonal"

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_crystal_system(c("Icosahedral"))

## End(Not run)
```

---

geomaterials\_dens\_greater\_than

*retrieve the geomaterials whose density are higher than a given value.*

---

**Description**

: Queries the list of geomaterials that have higher density than gt.

**Usage**

```
geomaterials_dens_greater_than(gt, ...)
```

**Arguments**

gt	float value. dmeas: measured density of the mineral. This is either the lower limit (if there is a dmeas2) or average (if there is no dmeas2). dmeas2:measured maximum density of mineral
...	Further named parameters.Other optional arguments.

**Details**

This function related to the field "density\_min" of geomaterials. retrieve all the geomaterials that has higher density than the given density(gt).

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_dens_greater_than(2)  
  
## End(Not run)
```

---

```
geomaterials_dens_less_than  
    retrieve the geomaterials whose density are lower density than a given  
    value.
```

---

**Description**

: Queries the list of geomaterials that have lower density than lt.

**Usage**

```
geomaterials_dens_less_than(lt, ...)
```

**Arguments**

lt	float value. dmeas: measured density of the mineral. This is either the lower limit (if there is a dmeas2) or average (if there is no dmeas2). dmeas2:measured maximum density of mineral
...	Further named parameters.Other optional arguments.

**Details**

This function related to the field "density\_max" of geomaterials. retrieve all the geomaterials that has higher density than the given density(lt).

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_dens_less_than(3)  
  
## End(Not run)
```

---

geomaterials\_dens\_range

*retrieve the geomaterials whose density are within an given value.*

---

### Description

: Queries the list of geomaterials that match an given range.

### Usage

```
geomaterials_dens_range(gt,lt, ...)
```

### Arguments

gt	float value
lt	float value dmeas: measured density of the mineral. This is either the lower limit (if there is a dmeas2) or average (if there is no dmeas2). dmeas2:measured maximum density of mineral
...	Further named parameters.Other optional arguments.

### Details

This function related to the fields "density\_min" and "density\_max" of geomaterials. retrieve all the geomaterials records that has the density within an given range of (gt,lt).

### Value

df, a data frame of geomaterials

### Examples

```
## Not run:
df <-geomaterials_dens_range(2,3)

## End(Not run)
```

---

geomaterials\_diapheny *retrieve the geomaterials that have an given diapheny.*

---

### Description

: Queries the list of geomaterials that have an given diapheny.

### Usage

```
geomaterials_diapheny(diapheny, ...)
```



**Arguments**

diapheny            string. The diaphany of the mineral - transparent; translucent; opaque  
 ...                Further named parameters.Other optional arguments.

**Details**

This function related to the field "diapheny" of geomaterials. The diaphany of the mineral(Items Enum): "Opaque" "Translucent" "Transparent"

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_diapheny("Transparent")

## End(Not run)
```

---

geomaterials\_entrytype

*retrieve the geomaterials that have the given entrytype*

---

**Description**

: Queries the list of geomaterials that have the given entrytype

**Usage**

```
geomaterials_entrytype(types,...)
```

**Arguments**

types              list of entry types.  
 ...                Further named parameters.Other optional arguments.

**Details**

This function related to the field "entrytype" of geomaterials. Items Enum: 0 1 2 3 4 5 6 7 8  
 Multiple choice: 0- mineral; 1-synonym; 2-variety; 3-mixture; 4-series; 5-grouplist; 6-polytype;  
 7-rock; 8-commodity Related field: entrytype\_text (description of the entrytype).

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_entrytype(c('1'))  
  
## End(Not run)
```

---

geomaterials\_expand    *retrieve the geomaterials that have the given expand.*

---

**Description**

: Queries the list of geomaterials that have the given expand.

**Usage**

```
geomaterials_expand(expanded_fields,...)
```

**Arguments**

expanded\_fields    list of expand (Array of strings (Expanded fields)).Select fields to expand.  
...    Further named parameters.Other optional arguments.

**Details**

This function related to the field "expand" of geomaterials. The field expand(Items Enum): "description" "type\_localities" "localities" "relations" "~all" "\*"

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_expand("~all")  
  
## End(Not run)
```

---

```
geomaterials_field_exists
```

*retrieve the geomaterials records of empty or not empty of a given field.*

---

### Description

: Queries the list of geomaterials with an empty or not empty of a given field.

### Usage

```
geomaterials_field_exists(fieldname,bexists,...)
```

### Arguments

fieldname	string
bexists	bool
...	Further named parameters.Other optional arguments.

### Details

This function related to all the fields of geomaterials. e.g. meteoritical\_code\_exists.Meteoritical code exists. Include non-empty (true) / include empty only (false) retrieve the geomaterial list with an empty or not empty of a given field.

### Value

df, a list of geomaterials

### Examples

```
## Not run:
df <-geomaterials_field_exists("meteoritical_code")

## End(Not run)
```

---

```
geomaterials_fracturetype
```

*retrieve the geomaterials that have the given fracturetype.*

---

### Description

: Queries the list of geomaterials that have the given fracturetype.

**Usage**

```
geomaterials_fracturetype(types, ...)
```

**Arguments**

types	list of types.fracturetype(Array of strings or null): How the mineral breaks-"Conchoidal" "Fibrous" "Hackly" "Irregular/Uneven" "Micaceous" "None observed" "Splintery" "Step-Like" "Sub-Conchoidal".
...	Further named parameters.Other optional arguments.

**Details**

This function related to the field "fracturetype" of geomaterials. fracturetype(Items Enum): "Conchoidal" "Fibrous" "Hackly" "Irregular/Uneven" "Micaceous" "None observed" "Splintery" "Step-Like" "Sub-Conchoidal"

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_fracturetype(c("Step-Like"))

## End(Not run)
```

---

```
geomaterials_hardness_gt
```

*retrieve the geomaterials whose hardness are higher than an given value.*

---

**Description**

: Queries the list of geomaterials that have higher hardness than an given value(hmin).

**Usage**

```
geomaterials_hardness_gt(hmin, ...)
```

**Arguments**

hmin	float value of the Mohs scale of mineral hardness, which ranging from 0 to 10.
...	Further named parameters.Other optional arguments.

**Details**

This function related to the field "hardness\_min" of geomaterials. retrieve all the geomaterials that has higher hardness than the given value(hmin). hmin:the given value of minimum Moh's hardness

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_hardness_gt(8)

## End(Not run)
```

---

geomaterials\_hardness\_lt

*retrieve the geomaterials whose hardness are lower than an given value.*

---

**Description**

: Queries the list of geomaterials that have lower hardness than an given vlaue(hmax).

**Usage**

```
geomaterials_hardness_lt(hmax, ...)
```

**Arguments**

hmax	float value of the Mohs scale of mineral hardness, which ranging from 0 to 10.
...	Further named parameters.Other optional arguments.

**Details**

This function related to the field "hardness\_max" of geomaterials. retrieve all the geomaterials that has lower hardness than an given value(hmax). hamx: maximum Moh's hardness

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_hardness_lt(2)

## End(Not run)
```

---

geomaterials\_hardness\_range

*retrieve the geomaterials whose hardness is within the given range.*

---

### Description

: Queries the list of geomaterials that have hardness within the given range.

### Usage

```
geomaterials_hardness_range(hmin,hmax, ...)
```

### Arguments

hmin	float value of the Mohs scale of mineral hardness, which ranging from 0 to 10.
hmax	float value of the Mohs scale of mineral hardness, which ranging from 0 to 10.
...	Further named parameters.Other optional arguments.

### Details

This function related to the fields "hardness\_min" and "hardness\_max" of geomaterials. retrieve all the geomaterials that has the hardness within an given range(hmin,hmax). hmin:the given value of minimum Moh's hardness hamx: maximum Moh's hardness

### Value

df, a data frame of geomaterials

### Examples

```
## Not run:
df <-geomaterials_hardness_range(2,3)

## End(Not run)
```

---

geomaterials\_ima

*retrieve the geomaterials approved by IMA or not.*

---

### Description

: Queries the geomaterials within or without the ima.

### Usage

```
geomaterials_ima(btrue,...)
```

**Arguments**

btrue           boolean value.TRUE IMA approved, otherwise not approved.  
 ...            Further named parameters.Other optional arguments.

**Details**

This function related to the field "ima" of geomaterials. retrieve all the geomaterials that are approved by the IMA or not.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_ima(TRUE)

## End(Not run)
```

---

geomaterials\_ima\_notes

*retrieve the geomaterials match given notes.*

---

**Description**

: Queries the geomaterials with an given .

**Usage**

```
geomaterials_ima_notes(enum_item,...)
```

**Arguments**

enum\_item       Array of integers or null. Ima notes: multiple choice (OR) : "GROUP" "INTERMEDIATE" "NAMED\_AMPHIBOLE" "PENDING\_APPROVAL" "PUBLISHED\_WITHOUT\_APPROVAL" "REDEFINED" "REJECTED" "RENAMED" "UNNAMED\_INVALID" "UNNAMED\_VALID"  
 ...            Further named parameters.Other optional arguments.

**Details**

This function related to the field "ima\_notes" of geomaterials. Rejected by the IMA; Pending approval; IMA Approved Group Name; Redefined by the IMA; Renamed by the IMA; Intermediate member of a solid-solution series; Published without approval; Unnamed (probably valid); Unnamed (probably invalid); Named Amphibole

retrieve all the geomaterials that match the input IMA notes.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_ima_notes(c("PENDING_APPROVAL"))  
  
## End(Not run)
```

---

geomaterials\_ima\_status

*retrieve the geomaterials matched given IMA status.*

---

**Description**

: Queries the geomaterials with an given ima status.

**Usage**

```
geomaterials_ima_status(enum_status,...)
```

**Arguments**

enum_status	Ima status: multiple choice (OR):"APPROVED" "DISCREDITED" "GRAND-FATHERED" "PENDING_PUBLICATION" "QUESTIONABLE"
...	Further named parameters.Other optional arguments.

**Details**

This function related to the field "ima\_status" of geomaterials. retrieve all the geomaterials that match the input IMA notes.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_ima_status(c("APPROVED"))  
  
## End(Not run)
```



---

```
geomaterials_lustretype
```

*retrieve the geomaterials that match an given lustretype.*

---

### Description

: Queries the geomaterials that match an given lustretype.

### Usage

```
geomaterials_lustretype(types, ...)
```

### Arguments

types	string of the type name (Array of strings or null). adamantine, subadamantine, vitreous, subvitreous, resinous, waxy, greasy, silky, pearly, metallic, submetallic, dull, earthy
...	Further named parameters. Other optional arguments.

### Details

This function related to the field "lustretype" of geomaterials. lustretype(Items Enum): "Adamantine" "Dull" "Earthy" "Greasy" "Metallic" "Pearly" "Resinous" "Silky" "Sub-Adamantine" "Sub-Metallic" "Sub-Vitreous" "Vitreous" "Waxy" multiple choice (AND)

### Value

df, a data frame of geomaterials

### Examples

```
## Not run:
df <-geomaterials_lustretype(c("Adamantine"))

## End(Not run)
```

---

```
geomaterials_meteoritical_code
```

*retrieve the geomaterials matched a given string in its meteoritical code.*

---

### Description

: Queries the geomaterials with a given string matched its given meteoritical\_code.

**Usage**

```
geomaterials_meteoritical_code(str_meteoritical_code,...)
```

**Arguments**

```
str_meteoritical_code      boolean, meteoritical code exists. Include non-empty (TRUE) / include empty
                           only (FALSE).
...                        Further named parameters.Other optional arguments.
```

**Details**

This function related to the field "meteoritical\_code\_exists" of geomaterials. Meteoritical code exists. Include non-empty (true) / include empty only (false). retrieve all the geomaterials that match the input str\_meteoritical\_code.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_meteoritical_code(TRUE)

## End(Not run)
```

---

geomaterials_name	<i>retrieve the geomaterials matched a given string in its name.</i>
-------------------	--

---

**Description**

: Queries the geomaterials with a given name.

**Usage**

```
geomaterials_name(str_name,...)
```

**Arguments**

```
str_name      Text search supporting: _ as wildcards, e.g. "qu_rtz", "bario*"
...          Further named parameters.Other optional arguments.
```

**Details**

This function related to the field "name" of geomaterials. retrieve all the geomaterials that match the input IMA notes.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_name("qu_rtz")  
  
## End(Not run)
```

---

geomaterials\_not\_contain\_elems  
*geomaterials that do not contain the elements*

---

**Description**

retrieve the geomaterials that do not contain any of the given elements.

**Usage**

```
geomaterials_not_contain_elems (ecl_elms_vector, ...)
```

**Arguments**

ecl\_elms\_vector  
vector of elements.  
... Further named parameters.Other optional arguments-Additional arguments.

**Details**

This function related to the field "elements\_exc" of geomaterials.

**Value**

df, a data frame of geomaterials list.

**Examples**

```
## Not run:  
df <-geomaterials_not_contain_elems (c('Fe', 'S', 'O'), fields ="id,name,mindat_formula,elements")  
  
## End(Not run)
```

---

```
geomaterials_optical2v_max
    retrieve the geomaterials that less than the given optical 2v.
```

---

**Description**

: Queries the geomaterials have the lower optical 2v value than the given lt.

**Usage**

```
geomaterials_optical2v_max(lt, ...)
```

**Arguments**

lt                    list of the signs.Please refer to the details.  
 ...                   Further named parameters.Other optional arguments.

**Details**

This function related to the field "optical2v\_max" of geomaterials. optical2vcalc:The calculated 2V angle minimum or average of biaxial minerals optical2vcalc2:The calculated 2V angle maximum of biaxial minerals optical2vmeasured:The measured 2V angle minimum or average of biaxial minerals optical2vmeasured2:The measured 2V angle maximum of biaxial minerals

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_optical2v_max(0.2)

## End(Not run)
```

---

```
geomaterials_optical2v_min
    retrieve the geomaterials that has higher value than the given optical
    2v.
```

---

**Description**

: Queries the geomaterials have the higher optical 2v value than the given gt.

**Usage**

```
geomaterials_optical2v_min(gt, ...)
```

**Arguments**

gt                    given value of optical 2v of mineral.Please refer to the details.  
 ...                    Further named parameters.Other optional arguments.

**Details**

This function related to the field "optical2v\_mix" of geomaterials. optical2vcalc:The calculated 2V angle minimum or average of biaxial minerals optical2vcalc2:The calculated 2V angle maximum of biaxial minerals optical2vmeasured:The measured 2V angle minimum or average of biaxial minerals optical2vmeasured2:The measured 2V angle maximum of biaxial minerals

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_optical2v_min(0.1)

## End(Not run)
```

---

geomaterials\_optical2v\_range

*retrieve the geomaterials that has the given range of optical 2v.*

---

**Description**

: Queries the geomaterials have the higher optical 2v value than the given lt.

**Usage**

```
geomaterials_optical2v_range(gt,lt,...)
```

**Arguments**

gt                    given value of minimum of optical 2v of mineral.Please refer to the details.  
 lt                    an given value of maximum of optical 2v of mineral.Please refer to the details.  
 ...                    Further named parameters.Other optional arguments.

**Details**

This function related to the field "optical2v\_min" and "optical2v\_max" of geomaterials. optical2vcalc:The calculated 2V angle minimum or average of biaxial minerals optical2vcalc2:The calculated 2V angle maximum of biaxial minerals optical2vmeasured:The measured 2V angle minimum or average of biaxial minerals optical2vmeasured2:The measured 2V angle maximum of biaxial minerals

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_optical2v_range(0.1,0.2)

## End(Not run)
```

---

geomaterials\_opticalsign

*retrieve the geomaterials that match an given optical signs.*

---

**Description**

: Queries the geomaterials match an given optical signs.

**Usage**

```
geomaterials_opticalsign(signs, ...)
```

**Arguments**

signs	list of the signs(string or null). sign for uniaxial and biaxial minerals: +;-;+/- .Please refer to the details.
...	Further named parameters.Other optional arguments.

**Details**

This function related to the field "opticalsign" of geomaterials. Optical sign: single choice (Enum): "+", "+/-", "-"

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_opticalsign("-")

## End(Not run)
```

---

 geomaterials\_opticaltype

*retrieve the geomaterials that match an given optical type.*


---

### Description

: Queries the geomaterials match an given optical type.

### Usage

```
geomaterials_opticaltype(types, ...)
```

### Arguments

types            list of the types for the field of opticaltype. Please refer to the details.  
 ...              Further named parameters.Other optional arguments.

### Details

This function related to the field "opticaltype" of geomaterials. transparent mineral. optical-type(Enum): "Biaxial" "Isotropic" "Uniaxial"

### Value

df, a data frame of geomaterials

### Examples

```
## Not run:
df <-geomaterials_opticaltype("Isotropic")

## End(Not run)
```

---

 geomaterials\_polytypeof

*retrieve the geomaterials by an given id of polytype of (the id of the mineral that this record is the polytype of. )*


---

### Description

: Queries the geomaterials by an given id for its polytype. A mineral that differs from another only in the stacking of similar structural units in its atomic structure

### Usage

```
geomaterials_polytypeof(ptype, ...)
```

**Arguments**

ptype            integer. an mindat id of the mineral that this record is the polytype of  
 ...              Further named parameters. Other optional arguments.

**Details**

This function related to the field "polytypeof" of geomaterials. retrieve the geomaterials with an given id of polytypeof.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_polytypeof(0,fields = "id,name,polytypeof")
## End(Not run)
```

---

geomaterials\_ri\_gt     *retrieve the geomaterials that refractive index higher than an given value(gt).*

---

**Description**

: Queries the geomaterials have the higher refractive index than an given value(gt).

**Usage**

```
geomaterials_ri_gt(gt, ...)
```

**Arguments**

gt                 float value. Refractive index, from (rimax>=).  
 ...                Further named parameters. Other optional arguments.

**Details**

This function related to the field "ri\_min" of geomaterials. retrieve the geomaterials with the refractive index higher than an given value(gt).

**Value**

df, a data frame of geomaterials



**Examples**

```
## Not run:  
df <-geomaterials_ri_gt(0.3)  
  
## End(Not run)
```

---

geomaterials\_ri\_lt     *retrieve the geomaterials that refractive index lower than an given value(lt).*

---

**Description**

: Queries the geomaterials have the lower refractive index than an given value(lt).

**Usage**

```
geomaterials_ri_lt(lt,...)
```

**Arguments**

lt                    float value. Refractive index, to (rimin<=)  
...                   Further named parameters.Other optional arguments.

**Details**

This function related to the field "ri\_max" of geomaterials. retrieve the geomaterials with the refractive index lower than an given value(lt).

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_ri_lt(0.5)  
  
## End(Not run)
```

---

geomaterials\_ri\_range *retrieve the geomaterials whose refractive index is within an given range(gt,lt).*

---

### Description

: Queries the list of geomaterials that have refractive index within an given range(gt,lt).

### Usage

```
geomaterials_ri_range(gt,lt, ...)
```

### Arguments

gt	float value. Refractive index, from (rimax>=).
lt	float value. Refractive index, to (rimin<=)
...	Further named parameters.Other optional arguments.

### Details

This function related to the fields "ri\_min" and "ri\_max" of geomaterials. retrieve all the geomaterials that has the refractive index within the range of (gt,lt).

### Value

df, a data frame of geomaterials

### Examples

```
## Not run:  
df <-geomaterials_ri_range(0.2,0.5)  
  
## End(Not run)
```

---

geomaterials\_search\_name  
*retrieve the geomaterials by a given name.*

---

### Description

: Queries the list of geomaterials by a given name.

### Usage

```
geomaterials_search_name(name,...)
```

**Arguments**

name                    string. Text search supporting wildcards, e.g. qu\_rtz, bario\*"  
 ...                    Further named parameters. Other optional arguments.

**Details**

This function related to the fields "name" of geomaterials. retrieve the geomaterial list that match the given name.

**Value**

df, a list of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_search_name("Quartz")

## End(Not run)
```

---

geomaterials\_streak    *retrieve the geomaterials that match an given streak.*

---

**Description**

: Queries the list of geomaterials that match an given steak.

**Usage**

```
geomaterials_streak(str,...)
```

**Arguments**

str                    string. The color of the streak (color of powdered mineral)  
 ...                    Further named parameters. Other optional arguments.

**Details**

This function related to the fields "steak" of geomaterials. The color of the streak (color of powdered mineral). retrieve the geomaterials that has the given steak.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_streak("orange")  
  
## End(Not run)
```

---

geomaterials\_synid     *retrieve the geomaterials by an given synid.*

---

**Description**

: Queries the list of geomaterials that match an given synid.

**Usage**

```
geomaterials_synid(idnum,...)
```

**Arguments**

idnum            integer,an given synonym id.  
...              Further named parameters.Other optional arguments.

**Details**

This function related to the fields "synid" of geomaterials. The id of the geomaterial that is the synonym of this record (this geomaterial cannot be added to a locality). retrieve the geomaterials that has an given synid.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_synid(3777)  
  
## End(Not run)
```

---

`geomaterials_updated_at`*retrieve the geomaterials updated at an given time.*

---

**Description**

: Queries the list of geomaterials that were updated at an given time

**Usage**

```
geomaterials_updated_at(strDate,...)
```

**Arguments**

<code>strDate</code>	string, Last updated datetime in format %Y-%m-%d %H:%M:%S
<code>...</code>	Further named parameters. Other optional arguments.

**Details**

This function related to the fields "updated\_at" of geomaterials. Last updated datetime in format %Y-%m-%d %H:%M:%S retrieve the geomaterials that have the latest updated at the given time.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:  
df <-geomaterials_updated_at("2020-2-12 10:15:12")  
  
## End(Not run)
```

---

`geomaterials_varietyof`*retrieve the geomaterials that are varieties of an given id of geomaterials.*

---

**Description**

: Queries the list of geomaterials that match the given varietyof.

**Usage**

```
geomaterials_varietyof(intvalue,...)
```

**Arguments**

intvalue            integer, id of mineral that has this variety.  
 ...                Further named parameters. Other optional arguments.

**Details**

This function related to the fields "varietyof" of geomaterials. Varieties are geomaterials that have a special distinction from the main geomaterial ie. amethyst var. quartz retrieve the geomaterials that are varieties of an given id of geomaterials.

**Value**

df, a data frame of geomaterials

**Examples**

```
## Not run:
df <-geomaterials_varietyof(3337)

## End(Not run)
```

---

`geomeaterials_non_utf` *retrieve the geomaterials that include non-utf mineral names or not.*

---

**Description**

: Queries the geomaterials include non-utf mineral names or not.

**Usage**

```
geomeaterials_non_utf(btrue =TRUE,...)
```

**Arguments**

btrue              boolean. Include non-UTF mineral names?.Default is TRUE.  
 ...                Further named parameters. Other optional arguments.

**Details**

This function related to the field "non\_utf" of geomaterials. retrieve the geomaterials that contain (or not contain) the non-utf name.

**Value**

df, a data frame of geomaterials

## Examples

```
## Not run:  
df <-geomeaterials_non_utf(TRUE,fields = "id,name,non_utf")  
  
## End(Not run)
```

---

geomeaterials\_ordering

*retrieve the geomaterials by an given ordering.*

---

## Description

: Queries the geomaterials by an given ordering.

## Usage

```
geomeaterials_ordering(ord, ...)
```

## Arguments

ord                    string of field. Prepend "-" to the field name for descending order. Enum: "approval\_year" "id" "minstats\_\_ms\_locentries" "minstats\_\_ms\_photos" "name" "updttime" "weighting".

...                    Further named parameters.Other optional arguments.

## Details

This function related to the field "ordering" of geomaterials. ordering=-id - sort by id descending. Prepend "-" to the field name for descending order. fields:"approval\_year" "id" "minstats\_\_ms\_locentries" "minstats\_\_ms\_photos" "name" "updttime" "weighting". retrieve the geomaterials by an given ordering.

## Value

df, a data frame of geomaterials

## Examples

```
## Not run:  
df <-geomeaterials_ordering(-id)  
  
## End(Not run)
```

---

getExtension	<i>Output the file extension of a filename</i>
--------------	--

---

**Description**

Convert the mindat R dataframe to JSON-LD string

**Usage**

```
getExtension (filename)
```

**Arguments**

filename            R dataframe of retrieved data from Mindat database.

**Examples**

```
filename<- "fname.txt"  
fname_extension<- getExtension(filename)
```

---

localities_list_all	<i>retrieve the localities list.</i>
---------------------	--------------------------------------

---

**Description**

: Queries the list of localities.

**Usage**

```
localities_list_all(...)
```

**Arguments**

...                Further named parameters. Other optional arguments.

**Details**

This function related to the fields "ids" of localities. retrieve all the localities.

**Value**

df, a data frame of localities

**Examples**

```
## Not run:  
df <-localities_list_all(fields = "id,name,latitude,longitude")  
  
## End(Not run)
```



---

 localities\_list\_country

*retrieve the localities list that are belong to a given country.*


---

### Description

: Queries the list of localities that are within a given country.

### Usage

```
localities_list_country(country,...)
```

### Arguments

country	name of country,
...	Further named parameters.Other optional arguments.

### Details

This function related to the field "country" of localities. Enum: "Afghanistan" "Albania" "Algeria" "American Samoa" "Andorra" "Angola" "Anguilla" "Antigua and Barbuda" "Argentina" "Armenia" "Aruba" "Ashmore and Cartier Islands" "Australia" "Austria" "Azerbaijan" "Bahamas" "Bahrain" "Bangladesh" "Barbados" "Belarus" "Belgium" "Belize" "Benin" "Bermuda" "Bhutan" "Bolivia" "Bosnia And Herzegovina" "Botswana" "Bouvet Island" "Brazil" "British Indian Ocean Territories" "British Solomon Islands" "British Virgin Islands" "Brunei" "Bulgaria" "Burkina Faso" "Burundi" "Cambodia" "Cameroon" "Canada" "Cape Verde" "Cayman Islands" "Central African Republic" "Chad" "Chile" "China" "Christmas Island" "Cocos Islands" "Colombia" "Comoro Islands" "Cook Islands" "Costa Rica" "Croatia" "Cuba" "Cyprus" "Czech Republic" "Democratic Republic of the Congo" "Denmark" "Djibouti" "Dominica" "Dominican Republic" "East Timor" "Ecuador" "Egypt" "El Salvador" "Equatorial Guinea" "Estonia" "Ethiopia" "Faeroe Islands" "Falkland Islands" "Federated States of Micronesia" "Fiji" "Finland" "France" "French Guiana" "French Polynesia" "Gabon" "Gambia" "Georgia" "Germany" "Ghana" "Gibraltar" "Greece" "Greenland" "Grenada" "Guadeloupe" "Guam" "Guatemala" "Guernsey" "Guinea" "Guinea-Bissau" "Guyana" "Haiti" "Honduras" "Hong Kong" "Hungary" "Iceland" "India" "Indonesia" "Iran" "Iraq" "Ireland" "Isle of Man" "Israel" "Italy" "Ivory Coast (Côte d'Ivoire)" "Jamaica" "Japan" "Jersey" "Jordan" "Kazakhstan" "Kenya" "Kiribati" "Kosovo" "Kuwait" "Kyrgyzstan" "Laos" "Latvia" "Lebanon" "Lesotho" "Liberia" "Libya" "Liechtenstein" "Lithuania" "Luxembourg" "Macao" "Madagascar" "Malawi" "Malaysia" "Maldives" "Mali" "Malta" "Martinique" "Mauritania" "Mauritius" "Mexico" "Moldova" "Monaco" "Mongolia" "Montenegro" "Montserrat" "Morocco" "Mozambique" "Myanmar" "Namibia" "Nauru" "Nepal" "Netherlands" "Netherlands Antilles" "New Caledonia" "New Zealand" "Nicaragua" "Niger" "Nigeria" "North Korea" "Norway" "Oman" "Pakistan" "Panama" "Papua New Guinea" "Paraguay" "Peru" "Philippines" "Poland" "Portugal" "Puerto Rico" "Qatar" "Republic of Congo (Brazzaville)" "Republic of Macedonia" "Reunion Island" "Romania" "Russia" "Rwanda" "Saint Helena" "Saint Lucia" "Saint Vincent and the Grenadines" "San Marino" "Sao Tome And Principe" "Saudi Arabia" "Senegal" "Serbia" "Seychelles" "Sierra Leone" "Singapore" "Slovakia" "Slovenia" "Solomon Islands" "Somalia" "South Africa" "South Korea" "Spain"

"Sri Lanka" "St Christopher-Nevis Islands" "Sudan" "Suriname" "Swaziland" "Sweden" "Switzerland" "Syria" "Taiwan" "Tajikistan" "Tanzania" "Thailand" "Togo" "Tonga" "Trinidad And Tobago" "Tunisia" "Turkey" "Turkmenistan" "Turks And Caicos Islands" "Tuvalu" "U.S. Virgin Islands" "Uganda" "Ukraine" "United Arab Emirates" "United Kingdom" "United States" "Uruguay" "Uzbekistan" "Vanuatu (Republic of Vanuatu; New Hebrides)" "Venezuela" "Vietnam" "Western Sahara" "Western Samoa" "Yemen" "Zambia" "Zimbabwe"

### Value

df, a data frame of localities

### Examples

```
## Not run:
df <-localities_list_country ("Norway")

## End(Not run)
```

---

localities\_list\_description

*retrieve the localities that contain the given description*

---

### Description

: Queries the list of localities that contain the given description.

### Usage

```
localities_list_description(desc,...)
```

### Arguments

desc                    string,  
...                      Further named parameters.Other optional arguments.

### Details

This function related to all the field "description" of localities. retrieve the localities that contain the given description

### Value

df, a data frame of localities

### Examples

```
## Not run:
df <-localities_list_description("volcano")

## End(Not run)
```

---

`localities_list_elems_exc`*localities that do not contain the given elements*

---

**Description**

Queries the list of localities that do not contain the given elements.

**Usage**

```
localities_list_elems_exc(exc_elems_list, ...)
```

**Arguments**

`exc_elems_list` vector of elements.

`...` Further named parameters. Other optional arguments-Additional arguments.

**Details**

This function related to the field "elements\_exc" of localities. This function queries the list of localities that do contain the specified elements.

**Value**

`df`, a data frame of localities

**Examples**

```
## Not run:  
df <-localities_list_elems_exc(c("H", "O", "Si", "Al", "Fe", "Ca", "Na", "K", "P", "C", "Mn", "F", "Mg", "S"))  
  
## End(Not run)
```

---

`localities_list_elems_inc`*localities that contain the given elements*

---

**Description**

Queries the list of localities that contain the given elements.

**Usage**

```
localities_list_elems_inc(inc_elems_list, ...)
```

**Arguments**

inc\_elems\_list vector of elements.  
 ... Further named parameters.Other optional arguments-Additional arguments.

**Details**

This function related to the field "elements\_inc" of localities. This function queries the list of localities that contain the given elements.

**Value**

df, a data frame of localities

**Examples**

```
## Not run:
df <-localities_list_elems_inc(c("Dy"))

## End(Not run)
```

---

localities\_list\_elems\_inc\_exc

*localities that contain the given elements but not contain some other given elements.*

---

**Description**

Queries the list of localities that contain the given elements,but not contain some other given elements.

**Usage**

```
localities_list_elems_inc_exc(inc_elems_list,exc_elems_list, ...)
```

**Arguments**

inc\_elems\_list vector of elements.  
 exc\_elems\_list vector of elements.  
 ... Further named parameters.Other optional arguments-Additional arguments.

**Details**

This function related to the fields "elements\_inc" and "elements\_exc" of localities. This function queries the list of localities that contain the given elements,but not contain some other given elements.

**Value**

df, a data frame of localities

**Examples**

```
## Not run:  
df <-localities_list_elems_inc_exc(c("Dy"), c("Li"))  
  
## End(Not run)
```

---

localities\_list\_expand

*localities that contain the given expands.*

---

**Description**

Queries the list of localities that contain the given expands.

**Usage**

```
localities_list_expand(expands,...)
```

**Arguments**

expands            vector of expands.  
...                Further named parameters.Other optional arguments-Additional arguments.

**Details**

This function related to the fields "expand" of localities. Items Enum: "geomaterials" "~all" "\*" This function queries the list of localities that contain the given expands.

**Value**

df, a data frame of localities

**Examples**

```
## Not run:  
df <-localities_list_expand("~all")  
  
## End(Not run)
```

---

localities\_list\_txt    *localities that contain the given txt name.*

---

### Description

Queries the list of localities that contain the given txt name.

### Usage

```
localities_list_txt(txt,...)
```

### Arguments

txt	string.
...	Further named parameters.Other optional arguments.

### Details

This function related to the fields "txt" of localities. This function queries the list of localities that contain the given txt name.

### Value

df, a data frame of localities

### Examples

```
## Not run:
df <-localities_list_txt("lava")

## End(Not run)
```

---

localities\_list\_updated\_at  
*retrieve the localities list updated at the given time.*

---

### Description

: Queries the list of localities that have the given time

### Usage

```
localities_list_updated_at(updateDate,...)
```

**Arguments**

updateDate      string, Last updated datetime in format %Y-%m-%d %H:%M:%S  
...              Further named parameters.Other optional arguments.

**Details**

This function related to all the fields "updated\_at" of localities. retrieve the localities that have the latest updated at the given time.

**Value**

df, a data frame of localities

**Examples**

```
## Not run:  
df <-localities_list_updated_at("lava")  
  
## End(Not run)
```

---

localities\_retrieve\_id

*retrieve the localities by a given mindat id.*

---

**Description**

: Queries the localitiy by given id.

**Usage**

```
localities_retrieve_id(id,...)
```

**Arguments**

id              integer. the mindat localitiy id.  
...              Further named parameters.Other optional arguments.

**Details**

This function related to all the fields "id" of localities. retrieve the localities by a given id.

**Value**

df, a data frame of localities

**Examples**

```
## Not run:  
df <-localities_retrieve_id(3337)  
  
## End(Not run)
```

---

```
localities_status_list  
                          localities_status_list
```

---

**Description**

retrieve all locality status list.

**Usage**

```
localities_status_list (...)
```

**Arguments**

...                   Further named parameters.

**Details**

This function is to retrieve all the locality\_status list.

**Value**

df, data frame of locality status

**Examples**

```
## Not run:  
df <-localities_status_list()  
  
## End(Not run)
```



---

localities\_status\_retrieve  
*localities\_status\_retrieve*

---

**Description**

retrieve locality status by its id.

**Usage**

```
localities_status_retrieve (id,...)
```

**Arguments**

id	the mindat locality status id
...	Further named parameters.

**Details**

This function is to retrieve the locality\_status by an given id of locality.

**Value**

df, data frame of locality status.

**Examples**

```
## Not run:  
df <-localities_status_retrieve(10)  
  
## End(Not run)
```

---

localitiy\_type\_retrieve  
*localitiy\_type\_retrieve*

---

**Description**

retrieve locality type by an given id of locality.

**Usage**

```
localitiy_type_retrieve (id,...)
```

**Arguments**

id                    the mindat localitiy id  
...                    Further named parameters.

**Details**

This function is to retrieve the locality types by an given id of locality. @export

**Value**

df, data frame of locality status.

**Examples**

```
## Not run:  
df <-localitiy_type_retrieve(50)  
  
## End(Not run)
```

---

locality_age	<i>locality_age</i>
--------------	---------------------

---

**Description**

retrieve locality age by its id

**Usage**

```
locality_age (id,...)
```

**Arguments**

id                    the mindat localitiy age id.  
...                    Further named parameters.

**Details**

This function related to the fields "id" of locality\_age and locality.

**Value**

df, data frame of locality age.

**Examples**

```
## Not run:  
df <-locality_age(3337)  
  
## End(Not run)
```

---

locality\_age\_list      *locality\_age\_list*

---

**Description**

retrieve all locality age list or by its conditions

**Usage**

```
locality_age_list (...)
```

**Arguments**

...                  Further named parameters.

**Details**

This function is to retrieve all the locality\_age list.

**Value**

df, data frame of locality age.

**Examples**

```
## Not run:  
df <-locality_age_list()  
  
## End(Not run)
```

---

locality\_type\_list      *locality\_type\_list*

---

**Description**

retrieve all locality type list.

**Usage**

```
locality_type_list (...)
```

**Arguments**

...                  Further named parameters.

**Details**

This function is to retrieve the locality types list.

**Value**

df, data frame of locality type.

**Examples**

```
## Not run:  
df <-locality_type_list()  
  
## End(Not run)
```

---

```
mindat_build_querystring  
                          mindat_build_querystring
```

---

**Description**

Build query string based on the query conditions.

**Usage**

```
mindat_build_querystring (args)
```

**Arguments**

args                    query args.

**Value**

qs. generated query string.

**Examples**

```
## Not run:  
mindat_cache_set('page_size', 800)  
ids<-c("")  
hardness_min<- 9.3  
fields<- c("name,hardness")  
args<- list(ids,hardness_min,fields)  
querystring<-mindat_build_querystring(args)  
  
## End(Not run)
```

---

mindat\_cache\_delete     *Delete a cached value by the users input varname*

---

**Description**

Remove (clear) the cache named varname in current environment.

**Usage**

```
mindat_cache_delete(varname)
```

**Arguments**

varname                string input a cached name. Set a cached value empty by the given varname. A string, list or other objects.

**Examples**

```
mindat_cache_delete('api_token')
```

---

mindat\_cache\_empty     *Remove all cached values*

---

**Description**

Clear all current cached values. Set current environment cache empty.

**Usage**

```
mindat_cache_empty()
```

**Examples**

```
mindat_cache_empty()
```

mindat\_cache\_get      *Get cache value*

---

**Description**

Retrieve the value of the cache named varname in current environment.

**Usage**

```
mindat_cache_get(varname)
```

**Arguments**

varname              string

**Value**

cached value. A string, list or other objects.

**Examples**

```
token<- mindat_cache_get('api_token')
```

---

mindat\_cache\_has      *Check if the current environment has the cached value of varname.*

---

**Description**

Check whether or not the current environment has the cache named varname.

**Usage**

```
mindat_cache_has(varname)
```

**Arguments**

varname              string.

**Value**

Boolean value. if the varname is found in current environment cache, return True otherwise return False.

**Examples**

```
b_has <- mindat_cache_has('api_token')
```

---

`mindat_cache_return_or_setup`

*Check if the current environment has the cached function named varname.*

---

**Description**

Check whether the current environment has the cached function named varname, if has, return it. if not, setup up a new cache function named varname.

**Usage**

```
mindat_cache_return_or_setup(varname, setupfun)
```

**Arguments**

varname	string.
setupfun	boolean, if the cached is a setup function.

**Value**

If the varname is found in current environment cache, return cached function. If not, eval the function and return cached function.

**Examples**

```
aep<- api_end_points<-mindat_cache_return_or_setup('api_end_points', function(){return (list()) })
```

---

`mindat_cache_set`      *Set cache name and value*

---

**Description**

Assigns the value to the cache named varname in current environment.

**Usage**

```
mindat_cache_set(varname, value)
```

**Arguments**

varname	string. The cached varname.
value	string.

**Examples**

```
mindat_cache_set('api_token', "9ce67655d74bcd981e937be80dcea9cb")
```

---

mindat\_connection      *Initializing Mindat API*

---

### Description

Initializing API Call. Setup the base\_url, token and format.

### Usage

```
mindat_connection(token, base_url = "https://api.mindat.org", fmt = "json", page_size = 800)
```

### Arguments

token	string. You can apply a token from Mindat.org.
base_url	string. The base url of mindat API, default is "https://api.mindat.org".
fmt	string. The format of the request and response, default is json.
page_size	interger, setting the page size of responded data from the API server.

### Examples

```
mindat_connection("9ce67655d74bcd981e937be80dcea9cb", page_size = 1500)
```

---

mindat\_countries      *mindat\_countries*

---

### Description

retrieve all countries list or the contries by given conditions.

### Usage

```
mindat_countries (...)
```

### Arguments

...	Further named parameters.
-----	---------------------------

### Value

df, data frame of countries list

### Examples

```
## Not run:
df<- mindat_countries()

## End(Not run)
```



---

mindat_country	<i>mindat_country</i>
----------------	-----------------------

---

**Description**

retrieve the country by given its id.

**Usage**

```
mindat_country (id,...)
```

**Arguments**

id	country id in mindat.
...	Further named parameters.

**Value**

df, a data frame of country

**Examples**

```
## Not run:  
df<- mindat_country(1)  
  
## End(Not run)
```

---

mindat_extract_response_body	<i>mindat_extract_response_body</i>
------------------------------	-------------------------------------

---

**Description**

.

**Usage**

```
mindat_extract_response_body (response)
```

**Arguments**

response	response json
----------	---------------

**Value**

if status of the response is success (200),return the all\_data\_text(the content of response). Otherwise,report the errors.

## Examples

```
## Not run:
library(httr)
uri<- "https://api.mindat.org/geomaterials/?id__in=&hardness_min=9.3&fields=name,+
hardness&page_size=1500"
api_token<- "9ce67655d74bcd981e937be80dcea9cb"
response <- GET(uri,add_headers('Authorization'= paste('Token ',api_token,sep = "")))
raw_data <- mindat_extract_response_body(response)

## End(Not run)
```

---

mindat\_geomaterial      *mindat\_geomaterial*

---

## Description

retrieve geomaterial by its id

## Usage

```
mindat_geomaterial (id,...)
```

## Arguments

id	geomaterial id
...	Further named parameters.

## Value

df, data frame of locality type list

## Examples

```
## Not run:
df<- mindat_geomaterial(3337)

## End(Not run)
```

---

```
mindat_geomaterial_list
      mindat_geomaterial_list
```

---

**Description**

retrieve all the geomaterial list or the geomaterial by given conditions.

**Usage**

```
mindat_geomaterial_list(...)
```

**Arguments**

```
...          Further named parameters.
```

**Value**

df, data frame of locality type list

**Examples**

```
## Not run:
df<- mindat_geomaterial_list()

## End(Not run)
```

---

```
mindat_geomaterial_search
      mindat_geomaterial_search
```

---

**Description**

retrieve all the geomaterial list or the geomaterial by given conditions.

**Usage**

```
mindat_geomaterial_search (...)
```

**Arguments**

```
...          Further named parameters.
```

**Value**

df, data frame of geomaterials match the search

**Examples**

```
## Not run:  
df<- mindat_geomaterial_search(q="Quartz")  
  
## End(Not run)
```

---

```
mindat_geomaterial_varieties  
  mindat_geomaterial_varieties
```

---

**Description**

retrieve the geomaterial varieties by the id of geomaterial.

**Usage**

```
mindat_geomaterial_varieties (id,...)
```

**Arguments**

```
id          geomaterial id  
...        Further named parameters.
```

**Value**

df, data frame of locality type list

**Examples**

```
## Not run:  
df<- mindat_geomaterial_varieties(3337)  
  
## End(Not run)
```

---

```
mindat_get_data_from_uri  
  mindat_get_data_from_uri
```

---

**Description**

retrieve data from the uri.

**Usage**

```
mindat_get_data_from_uri (uri)
```

**Arguments**

`uri` request uri

**Value**

df. R data frame of the request uri.

**Examples**

```
## Not run:
library(httr)
uri <- "https://api.mindat.org/geomaterials/?id__in=&hardness_min=9.3&fields=name,+
hardness&page_size=1500"
mindat_cache_set('api_token', "9ce67655d74bcd981e937be80dcea9cb")
df <- mindat_get_data_from_uri(uri)

## End(Not run)
```

---

`mindat_localities_list`  
*mindat\_localities\_list*

---

**Description**

retrieve localities list

**Usage**

```
mindat_localities_list (...)
```

**Arguments**

... Further named parameters.

**Value**

df. data frame of localities list.

**Examples**

```
## Not run:
df<- mindat_localities_list()

## End(Not run)
```

---

mindat_locality	<i>mindat_locality</i>
-----------------	------------------------

---

**Description**

retrieve locality by its id

**Usage**

```
mindat_locality (id,...)
```

**Arguments**

id	the mindat locality id
...	Further named parameters.

**Value**

df, data frame of locality

**Examples**

```
## Not run:  
df<- mindat_locality(3337)  
  
## End(Not run)
```

---

mindat_locality_status	<i>mindat_locality_status</i>
------------------------	-------------------------------

---

**Description**

retrieve all locality status by its id

**Usage**

```
mindat_locality_status (id,...)
```

**Arguments**

id	the mindat locality status id.
...	Further named parameters.

**Value**

df, data frame of locality status

**Examples**

```
## Not run:  
df<- mindat_locality_status(10)  
  
## End(Not run)
```

---

*mindat\_locality\_status\_list*  
*mindat\_locality\_status\_list*

---

**Description**

retrieve all locality status list

**Usage**

```
mindat_locality_status_list (...)
```

**Arguments**

... Further named parameters.

**Value**

df, data frame of locality status list

**Examples**

```
## Not run:  
df<- mindat_locality_status_list()  
  
## End(Not run)
```

---

mindat\_locality\_type    *mindat\_locality\_type*

---

**Description**

retrieve locality type by its id

**Usage**

```
mindat_locality_type (id,...)
```

**Arguments**

id	locality type id
...	Further named parameters.

**Value**

df, data frame of locality type list

**Examples**

```
## Not run:  
df<- mindat_locality_type(50)  
  
## End(Not run)
```

---

mindat\_locality\_type\_list  
                          *mindat\_locality\_type\_list*

---

**Description**

retrieve all locality type list

**Usage**

```
mindat_locality_type_list (...)
```

**Arguments**

...	Further named parameters.
-----	---------------------------

**Value**

df, data frame of locality type list



**Examples**

```
## Not run:  
df<- mindat_locality_type_list()  
  
## End(Not run)
```

---

```
mindat_make_data_frame  
                          mindat_make_data_frame
```

---

**Description**

convert the response json to dataframe of R

**Usage**

```
mindat_make_data_frame (reg_list)
```

**Arguments**

reg\_list            response json data to list format obj.

**Value**

df\_out, R data frame

**Examples**

```
## Not run:  
id<- c('42155','9300','11282','48322')  
name<- c('Cuarzo opalescente', 'Cupromagnesite', 'Cuprozippeite', 'Quartz-anorthosite')  
ima_status <- c(0,0,0,0)  
synid <- c(42133, 9281, 0, 0)  
list_cvt <- list(id=id, name=name, ima_status=ima_status, synid=synid)  
df<- mindat_make_data_frame(list_cvt)  
  
## End(Not run)
```

---

```
mindat_mineral_ima  mindat_mineral_ima
```

---

**Description**

retrieve ima mineral by its id.

**Usage**

```
mindat_mineral_ima (id,...)
```

**Arguments**

```
id          mindat id
...         Further parameters.
```

**Value**

df. query results in data frame format.

**Examples**

```
## Not run:
df<- mindat_mineral_ima(3337)

## End(Not run)
```

---

```
mindat_mineral_ima_list
      mindat_mineral_ima_list
```

---

**Description**

retrieve ima mineral list

**Usage**

```
mindat_mineral_ima_list (...)
```

**Arguments**

```
...          , Further named parameters.
```

**Value**

df, data frame of mineral list.

**Examples**

```
## Not run:
df<- mindat_mineral_ima_list()

## End(Not run)
```

---

mindat\_parse\_raw\_data    *mindat\_parse\_raw\_data*

---

**Description**

parse the raw response of json to dataframe of R. If the raw\_data obtained from the response is paged, request all the pages and then add them into the df\_out data frame.

**Usage**

```
mindat_parse_raw_data (raw_data)
```

**Arguments**

raw\_data            content of the response body

**Value**

df\_out, R data frame

**Examples**

```
## Not run:
rd<-"{"count":5,"next":null,"previous":null,+
"results":[{"name":"Diamond"},{"name":"Khamrabaevite"},+
{"name":"Moissanite"},{"name":"Qingsongite"},{"name":"Uakitite"}]}"
df<- mindat_parse_raw_data(rd)

## End(Not run)
```

---

mindat\_query            *mindat\_query*

---

**Description**

Basic function for query dataset at a specified endpoint.

**Usage**

```
mindat_query (endpoint, query = list())
```

**Arguments**

endpoint            query endpoint, e.g.'minerals\_ima'.  
query                list for query conditions.

**Value**

df query results in data frame format.

**Examples**

```
## Not run:  
df <-mindat_query("geomaterials_list",list(ids="",hardness_min=9))  
  
## End(Not run)
```

---

mindat\_setup

*mindat\_setup*

---

**Description**

set up of the mindat basic uri, endpoints, and cache

**Usage**

```
mindat_setup(base_uri = 'https://api.mindat.org',page_size = 800)
```

**Arguments**

base\_uri            base uri of mindat API.  
page\_size           interger,setting the page size of responded data from the API server.

**Examples**

```
## Not run:  
mindat_setup()  
  
## End(Not run)
```

---

minerals_ima_list	<i>minerals_ima_list</i>
-------------------	--------------------------

---

**Description**

retrieve all mineral ima list.

**Usage**

```
minerals_ima_list (...)
```

**Arguments**

... Further named parameters.

**Details**

This function is to retrieve the IMA minerals list.

**Value**

df, data frame of minerals.

**Examples**

```
## Not run:  
df <-minerals_ima_list()  
  
## End(Not run)
```

---

minerals_ima_list_expand	<i>minerals_ima_list_expand</i>
--------------------------	---------------------------------

---

**Description**

retrieve mineral ima list with the given expand.

**Usage**

```
minerals_ima_list_expand (expand,...)
```

**Arguments**

expand description  
... Further named parameters.

**Details**

This function is related to the filed "expand" of ima mineral. Items Enum: "~all" "\*"

**Value**

df, data frame of ima minerals with expanded fields.

**Examples**

```
## Not run:  
df <-minerals_ima_list_expand("~all")  
  
## End(Not run)
```

---

minerals\_ima\_list\_ima *minerals\_ima\_list\_ima*

---

**Description**

retrieve mineral ima list with the given intValue.

**Usage**

```
minerals_ima_list_ima (intValue,...)
```

**Arguments**

intValue	Integer
...	Further named parameters.

**Details**

This function is related to the filed "ima" of ima minerals. Integer. 0: "PENDING\_PUBLICATION"  
1: "APPROVED"

**Value**

df, data frame of locality type.

**Examples**

```
## Not run:  
df <-minerals_ima_list_ima(1)  
  
## End(Not run)
```

---

```
minerals_ima_retrieve minerals_ima_retrieve
```

---

**Description**

retrieve mineral ima by its id.

**Usage**

```
minerals_ima_retrieve (id,...)
```

**Arguments**

id	the mindat ima id
...	Further named parameters.

**Details**

This function is related to the filed "id" of ima minerals.

**Value**

df, data frame of ima mineral by a given id.

**Examples**

```
## Not run:
df <-minerals_ima_retrieve(3337)

## End(Not run)
```

---

```
minerals_ima_updated_at
      retrieve the mineral_ima list updated at the given time.
```

---

**Description**

: Queries the list of mineral\_ima that have the given time

**Usage**

```
minerals_ima_updated_at(updateDate,...)
```

**Arguments**

updateDate	string, Last updated datetime in format %Y-%m-%d %H:%M:%S
...	Further named parameters.Other optional arguments.

**Details**

This function is related to the filed "updated\_at" of ima minerals. retrieve the localities that have the latest updated at the given time.

**Value**

df, a data frame of localities

**Examples**

```
## Not run:  
df <-minerals_ima_updated_at("2020-11-10 10:12:20")  
  
## End(Not run)
```

---

params_to_string	<i>params_to_string</i>
------------------	-------------------------

---

**Description**

Prase params to string,so that the query function can deal with the other exteranl condition set by the users.

**Usage**

```
params_to_string (params)
```

**Arguments**

params            convert params to string,which is used by the mindat query function.

**Value**

str .

**Examples**

```
## Not run:  
params_to_string("")  
  
## End(Not run)
```



---

saveMindatDataAs	<i>Output file as a given format</i>
------------------	--------------------------------------

---

**Description**

Save the mindat R dataframe to a specify format

**Usage**

```
saveMindatDataAs (inputdata,outputfname)
```

**Arguments**

inputdata	R dataframe of retrieved data from Mindat database.
outputfname	string. the output file name.

**Examples**

```
## Not run:  
df <-geomaterials_search_name("Quartz")  
saveMindatDataAs(df,"test.jsonld")  
  
## End(Not run)
```

---

set_api_base	<i>set_api_base</i>
--------------	---------------------

---

**Description**

set base uri of current environment

**Usage**

```
set_api_base (api_base)
```

**Arguments**

api_base	string. The base uri of mindat api.
----------	-------------------------------------

**Examples**

```
set_api_base("9ce67655d74bcd981e937be80dcea9cb")
```

---

set_api_token	<i>set_api_token</i>
---------------	----------------------

---

**Description**

set the token of current environment

**Usage**

```
set_api_token (api_token)
```

**Arguments**

api\_token      string. The token of mindat api.

**Examples**

```
set_api_token("9ce67655d74bcd981e937be80dcea9cb")
```

---

set_page_size	<i>set_page_size</i>
---------------	----------------------

---

**Description**

set the page\_size of response records.

**Usage**

```
set_page_size (page_size)
```

**Arguments**

page\_size      string. The token of mindat api.

**Examples**

```
set_page_size(800)
```

# Index

- ConvertDF2JsonLD, 4
- ConvertDF2TTL, 4
- geomaterials\_bi\_greater\_than, 5
- geomaterials\_bi\_less\_than, 6
- geomaterials\_bi\_range, 6
- geomaterials\_by\_groupid, 7
- geomaterials\_cleavagetype, 8
- geomaterials\_colour, 9
- geomaterials\_contain\_all\_but\_not\_elems, 9
- geomaterials\_contain\_all\_elems, 10
- geomaterials\_contain\_any\_but\_not\_elems, 11
- geomaterials\_contain\_any\_elems, 12
- geomaterials\_contain\_only\_elems, 12
- geomaterials\_crystal\_system, 13
- geomaterials\_dens\_greater\_than, 14
- geomaterials\_dens\_less\_than, 15
- geomaterials\_dens\_range, 16
- geomaterials\_diapheny, 16
- geomaterials\_entrytype, 17
- geomaterials\_expand, 18
- geomaterials\_field\_exists, 19
- geomaterials\_fracturetype, 19
- geomaterials\_hardness\_gt, 20
- geomaterials\_hardness\_lt, 21
- geomaterials\_hardness\_range, 22
- geomaterials\_ima, 22
- geomaterials\_ima\_notes, 23
- geomaterials\_ima\_status, 24
- geomaterials\_lustretype, 25
- geomaterials\_meteoritical\_code, 25
- geomaterials\_name, 26
- geomaterials\_not\_contain\_elems, 27
- geomaterials\_optical2v\_max, 28
- geomaterials\_optical2v\_min, 28
- geomaterials\_optical2v\_range, 29
- geomaterials\_opticalsign, 30
- geomaterials\_opticaltype, 31
- geomaterials\_polytypeof, 31
- geomaterials\_ri\_gt, 32
- geomaterials\_ri\_lt, 33
- geomaterials\_ri\_range, 34
- geomaterials\_search\_name, 34
- geomaterials\_streak, 35
- geomaterials\_synid, 36
- geomaterials\_updated\_at, 37
- geomaterials\_varietyof, 37
- geomeaterials\_non\_utf, 38
- geomeaterials\_ordering, 39
- getExtension, 40
- localities\_list\_all, 40
- localities\_list\_country, 41
- localities\_list\_description, 42
- localities\_list\_elems\_exc, 43
- localities\_list\_elems\_inc, 43
- localities\_list\_elems\_inc\_exc, 44
- localities\_list\_expand, 45
- localities\_list\_txt, 46
- localities\_list\_updated\_at, 46
- localities\_retrieve\_id, 47
- localities\_status\_list, 48
- localities\_status\_retrieve, 49
- locality\_type\_retrieve, 49
- locality\_age, 50
- locality\_age\_list, 51
- locality\_type\_list, 51
- mindat\_build\_querystring, 52
- mindat\_cache\_delete, 53
- mindat\_cache\_empty, 53
- mindat\_cache\_get, 54
- mindat\_cache\_has, 54
- mindat\_cache\_return\_or\_setup, 55
- mindat\_cache\_set, 55
- mindat\_connection, 56
- mindat\_countries, 56
- mindat\_country, 57

mindat\_extract\_response\_body, [57](#)  
mindat\_geomaterial, [58](#)  
mindat\_geomaterial\_list, [59](#)  
mindat\_geomaterial\_search, [59](#)  
mindat\_geomaterial\_varieties, [60](#)  
mindat\_get\_data\_from\_uri, [60](#)  
mindat\_localities\_list, [61](#)  
mindat\_locality, [62](#)  
mindat\_locality\_status, [62](#)  
mindat\_locality\_status\_list, [63](#)  
mindat\_locality\_type, [64](#)  
mindat\_locality\_type\_list, [64](#)  
mindat\_make\_data\_frame, [65](#)  
mindat\_mineral\_ima, [66](#)  
mindat\_mineral\_ima\_list, [66](#)  
mindat\_parse\_raw\_data, [67](#)  
mindat\_query, [67](#)  
mindat\_setup, [68](#)  
minerals\_ima\_list, [69](#)  
minerals\_ima\_list\_expand, [69](#)  
minerals\_ima\_list\_ima, [70](#)  
minerals\_ima\_retrieve, [71](#)  
minerals\_ima\_updated\_at, [71](#)

params\_to\_string, [72](#)

saveMindatDataAs, [73](#)  
set\_api\_base, [73](#)  
set\_api\_token, [74](#)  
set\_page\_size, [74](#)