
cmif Documentation

Release 2022.11.27

Donatus Herre

Nov 27, 2022

Contents

1	Reference	1
1.1	build module	1
1.2	extract module	3
1.3	demo module	5
1.4	local module	5
1.5	retrieve module	5
1.6	service module	6
1.7	authority module	6
1.8	model module	6
2	Indices and tables	7
	Python Module Index	9
	Index	11

1.1 build module

build XML data in CMI format

`cmif.build.add_attrib(element, name, value)`
add attribute to element if value != ""

`cmif.build.add_child(parent, element)`
add element to parent if not None

`cmif.build.add_children(parent, elements)`
add elements to parent if not None

`cmif.build.add_pi(tree)`
add processing instruction <?xml-model?> to given element tree

`cmif.build.ns_cs(attrib)`
add correspSearch namespace to given attribute

`cmif.build.ns_xml(attrib)`
add xml namespace to given attribute

`cmif.build.pi_rng()`
create processing instruction <?xml-model?>

`cmif.build.pretty(element)`
pretty print given elements

`cmif.build.tei_availability(child_license=None)`
create TEI element <availability> with (optional) child element <licence>

`cmif.build.tei_bibl(elem_text, attrib_type, attrib_xml_id=None, domain=None)`

create TEI element <bibl> with given text, @type and (optional) @xml:id
if @xml:id is None a uuid is being generated

if domain is None the uuid will be random
if domain is not None the uuid will be static

`cmif.build.tei_bibl_id(domain=None)`

generate uuid for @xml:id of TEI element <bibl> by given domain
if domain is None a random uuid is being generated

`cmif.build.tei_body()`
create TEI element <body>

`cmif.build.tei_corresp_action(attrib_type, children=None)`
create TEI element <correspAction> with @type and (optional) children

`cmif.build.tei_corresp_desc(attrib_key=", attrib_ref=", attrib_source=", children=None)`
create TEI element <correspDesc> with @ref, @source and (optional) children

`cmif.build.tei_date(attrib_when=", attrib_from=", attrib_to=", attrib_not_before=", attrib_not_after=", attrib_evidence=None, attrib_cert=None)`

create TEI element <date> with @when, @from, @to, @notBefore or @notAfter
and (optional) @evidence and @cert

`cmif.build.tei_editor(elem_text)`
create TEI element <editor> with given element text

`cmif.build.tei_email(elem_text)`
create TEI element <email> with given element text

`cmif.build.tei_file_desc(children=None)`
create TEI element <fileDesc> with (optional) children

`cmif.build.tei_header(children=None)`
create TEI element <teiHeader> with (optional) children

`cmif.build.tei_idno(elem_text, attrib_type='url')`
create TEI element <idno> with @type

`cmif.build.tei_license(elem_text=", attrib_target=")`
create TEI element <licence> with (optional) text and @target

`cmif.build.tei_org_name(elem_text, attrib_ref=", attrib_evidence=None, attrib_cert=None)`

create TEI element <orgName> with given element text, @ref
and (optional) @evidence and @cert

`cmif.build.tei_p()`
create TEI element <p>

`cmif.build.tei_pers_name(elem_text, attrib_ref=", attrib_evidence=None, attrib_cert=None)`

create TEI element <persName> with given element text, @ref
and (optional) @evidence and @cert

`cmif.build.tei_place_name(elem_text, attrib_ref=", attrib_evidence=None, attrib_cert=None)`

create TEI element <placeName> with given element text, @ref
and (optional) @evidence and @cert

`cmif.build.tei_profile_desc` (*children=None*)
create TEI element <profileDesc> with (optional) children

`cmif.build.tei_publication_stmt` (*children=None*)
create TEI element <publicationStmt> with (optional) children

`cmif.build.tei_publisher` (*child_ref=None*)
create TEI element <publisher> with (optional) child element <ref>

`cmif.build.tei_ref` (*elem_text, attrib_target*)
create TEI element <ref> with @target

`cmif.build.tei_root` (*children=None*)
create TEI root element <TEI> with (optional) children

`cmif.build.tei_source_desc` (*children=None*)
create TEI element <sourceDesc> with (optional) children

`cmif.build.tei_text` ()
create TEI element <text>

`cmif.build.tei_text_empty` ()
create TEI element <text> with child elements <body> and <p> (empty)

`cmif.build.tei_title` (*elem_text*)
create TEI element <title> with given element text

`cmif.build.tei_title_stmt` (*children=None*)
create TEI element <titleStmt> with (optional) children

`cmif.build.tostr` (*element*)
convert given element to str

1.2 extract module

extract XML data in CMI format

`cmif.extract.bibl` (*data, multi=False*)

extract TEI element <bibl>
set multi to True if multiple references exist

`cmif.extract.bibl_id` (*data, multi=False*)

extract @xml:id from TEI element <bibl>
set multi to True if multiple references exist

`cmif.extract.bibl_text` (*data, multi=False*)

extract text of TEI element <bibl>
set multi to True if multiple references exist

`cmif.extract.bibl_type` (*data, multi=False*)

extract @type from TEI element <bibl>
set multi to True if multiple references exist

```
cmif.extract.correspaction(data)
    extract TEI elements <correspAction>

cmif.extract.correspaction_type(data)
    extract @type from TEI elements <correspAction>

cmif.extract.correspdesc(data)
    extract TEI elements <correspDesc>

cmif.extract.correspdesc_key(data)
    extract @source from TEI elements <correspDesc>

cmif.extract.correspdesc_source(data)
    extract @source from TEI elements <correspDesc>

cmif.extract.date_attrib(data)
    extract @ from TEI element <date>

cmif.extract.date_from(data)
    extract @when from TEI element <date>

cmif.extract.date_not_after(data)
    extract @when from TEI element <date>

cmif.extract.date_not_before(data)
    extract @when from TEI element <date>

cmif.extract.date_to(data)
    extract @when from TEI element <date>

cmif.extract.date_when(data)
    extract @when from TEI element <date>

cmif.extract.editor(data, multi=False)

    extract TEI element <editor>
    set multi to True if multiple editors exist

cmif.extract.editor_email(data, multi=False)

    extract text of TEI element <email> from parent <editor>
    set multi to True if multiple editors exist

cmif.extract.editor_name(data, multi=False)

    extract text of TEI element <editor>
    set multi to True if multiple editors exist

cmif.extract.idno(data)
    extract text from TEI element <idno>

cmif.extract.license(data)
    extract text of TEI element <licence>

cmif.extract.license_target(data)
    extract @target from TEI element <licence>

cmif.extract.org_name(data)
    extract text from TEI element <orgName>
```



```
cmif.extract.org_name_ref(data)
    extract @ref from TEI element <orgName>

cmif.extract.pers_name(data)
    extract text from TEI element <persName>

cmif.extract.pers_name_ref(data)
    extract @ref from TEI element <persName>

cmif.extract.place_name(data)
    extract text from TEI element <placeName>

cmif.extract.place_name_ref(data)
    extract @ref from TEI element <placeName>

cmif.extract.publisher(data)
    extract text from child <ref> of TEI element <publisher>

cmif.extract.publisher_target(data)
    extract @target from child <ref> of TEI element <publisher>

cmif.extract.title(data)
    extract text of TEI element <title>
```

1.3 demo module

demo XML data in CMI format

```
cmif.demo.gottsched()
    get data of Gottsched's Briefwechsel as demo for CMI-format

cmif.demo.wieland()
    get data of Wieland's Briefe (1750-1760) as demo for CMI-format
```

1.4 local module

read and write local XML data in CMI format

```
cmif.local.reader(filepath)
    read xml data from given file path

cmif.local.writer(root, file='cmif.xml', path='.')
    write given root element to file at path
```

1.5 retrieve module

retrieve remote XML data in CMI format

```
cmif.retrieve.remote_file(url)
    send http get request to given url
```

1.6 service module

access data in CMI format through correspSearch webservises

```
cmif.service.correspsearch (correspondent=None,      sender=None,      addressee=None,
                             startdate=None,         enddate=None,      place=None,      place-
                             Sender=None,            placeAddressee=None, cmiFile=None,
                             publication=None,        available=None,    strictDate=False,
                             endpoint='https://correspsearch.net/api/v1.1/tei-xml.xml')
```

query data in CMI format via correspSearch API by given parameters

available endpoints: TEI_XML (default) / TEI_JSON

see <https://correspsearch.net/index.xml?id=api> for details

1.7 authority module

search or retrieve GeoNames

you need to set the environment variable GEO_USER like this:

```
$ export GEO_USER=[my-geonames-account-username]
```

or pass the variable username as parameter to the functions!

```
cmif.authority.geonames.request (geoname_id, style='full', lang=None, username=None)
    request GeoNames entity by given id (requires GeoNames username)
```

```
cmif.authority.geonames.search (place, bias='DE', username=None)
    search GeoNames entity for given place (requires GeoNames username)
```

1.8 model module

schema models of CMI format

```
cmif.model.odd()
    get ODD specification of CMI format
```

```
cmif.model.rng()
    get RNG schema of CMI format
```

```
cmif.model.validate (root)
    validate XML data of given root against RNG schema
```

CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`

C

- `cmif.authority.geonames`, 6
- `cmif.build`, 1
- `cmif.demo`, 5
- `cmif.extract`, 3
- `cmif.local`, 5
- `cmif.model`, 6
- `cmif.retrieve`, 5
- `cmif.service`, 6

A

add_attrib() (in module *cmif.build*), 1
 add_child() (in module *cmif.build*), 1
 add_children() (in module *cmif.build*), 1
 add_pi() (in module *cmif.build*), 1

B

bibl() (in module *cmif.extract*), 3
 bibl_id() (in module *cmif.extract*), 3
 bibl_text() (in module *cmif.extract*), 3
 bibl_type() (in module *cmif.extract*), 3

C

cmif.authority.geonames (module), 6
 cmif.build (module), 1
 cmif.demo (module), 5
 cmif.extract (module), 3
 cmif.local (module), 5
 cmif.model (module), 6
 cmif.retrieve (module), 5
 cmif.service (module), 6
 correspaaction() (in module *cmif.extract*), 3
 correspaaction_type() (in module *cmif.extract*), 4
 correspdesc() (in module *cmif.extract*), 4
 correspdesc_key() (in module *cmif.extract*), 4
 correspdesc_source() (in module *cmif.extract*), 4
 correspsearch() (in module *cmif.service*), 6

D

date_attrib() (in module *cmif.extract*), 4
 date_from() (in module *cmif.extract*), 4
 date_not_after() (in module *cmif.extract*), 4
 date_not_before() (in module *cmif.extract*), 4
 date_to() (in module *cmif.extract*), 4
 date_when() (in module *cmif.extract*), 4

E

editor() (in module *cmif.extract*), 4
 editor_email() (in module *cmif.extract*), 4

editor_name() (in module *cmif.extract*), 4

G

gottsched() (in module *cmif.demo*), 5

I

idno() (in module *cmif.extract*), 4

L

license() (in module *cmif.extract*), 4
 license_target() (in module *cmif.extract*), 4

N

ns_cs() (in module *cmif.build*), 1
 ns_xml() (in module *cmif.build*), 1

O

odd() (in module *cmif.model*), 6
 org_name() (in module *cmif.extract*), 4
 org_name_ref() (in module *cmif.extract*), 4

P

pers_name() (in module *cmif.extract*), 5
 pers_name_ref() (in module *cmif.extract*), 5
 pi_rng() (in module *cmif.build*), 1
 place_name() (in module *cmif.extract*), 5
 place_name_ref() (in module *cmif.extract*), 5
 pretty() (in module *cmif.build*), 1
 publisher() (in module *cmif.extract*), 5
 publisher_target() (in module *cmif.extract*), 5

R

reader() (in module *cmif.local*), 5
 remote_file() (in module *cmif.retrieve*), 5
 request() (in module *cmif.authority.geonames*), 6
 rng() (in module *cmif.model*), 6

S

search() (in module *cmif.authority.geonames*), 6

T

`tei_availability()` (in module *cmif.build*), 1
`tei_bibl()` (in module *cmif.build*), 1
`tei_bibl_id()` (in module *cmif.build*), 2
`tei_body()` (in module *cmif.build*), 2
`tei_corresp_action()` (in module *cmif.build*), 2
`tei_corresp_desc()` (in module *cmif.build*), 2
`tei_date()` (in module *cmif.build*), 2
`tei_editor()` (in module *cmif.build*), 2
`tei_email()` (in module *cmif.build*), 2
`tei_file_desc()` (in module *cmif.build*), 2
`tei_header()` (in module *cmif.build*), 2
`tei_idno()` (in module *cmif.build*), 2
`tei_license()` (in module *cmif.build*), 2
`tei_org_name()` (in module *cmif.build*), 2
`tei_p()` (in module *cmif.build*), 2
`tei_pers_name()` (in module *cmif.build*), 2
`tei_place_name()` (in module *cmif.build*), 2
`tei_profile_desc()` (in module *cmif.build*), 2
`tei_publication_stmt()` (in module *cmif.build*),
3
`tei_publisher()` (in module *cmif.build*), 3
`tei_ref()` (in module *cmif.build*), 3
`tei_root()` (in module *cmif.build*), 3
`tei_source_desc()` (in module *cmif.build*), 3
`tei_text()` (in module *cmif.build*), 3
`tei_text_empty()` (in module *cmif.build*), 3
`tei_title()` (in module *cmif.build*), 3
`tei_title_stmt()` (in module *cmif.build*), 3
`title()` (in module *cmif.extract*), 5
`tostr()` (in module *cmif.build*), 3

V

`validate()` (in module *cmif.model*), 6

W

`wieland()` (in module *cmif.demo*), 5
`writer()` (in module *cmif.local*), 5