Librair Documentation

Release 2022.11.27

Donatus Herre

Nov 27, 2022

Contents

1 Services 1

2 Interfaces 7

3 Protocols 11

4 Schemas 13

5 Parsers 15

6 Indices and tables 17

Python Module Index 19

Index 21

CHAPTER 1

Services

1.1 cerl module

librair.services.cerl.**address**(*idn*, *schema*, *style*)
get url of entity specified by idn in given schema and style

librair.services.cerl.**request**(*idn*, *schema='rdfxml'*, *style=None*)
request data of entity specified by idn in given schema and (optional) style

SCHEMA	STYLE	RETURN TYPE
rdfxml	None	lxml.etree.Element
json	None	dict
json	jsonld	dict
txt	ttl	str
txt	internal	str

librair.services.cerl.**store**(*idn*, *schema='json'*, *style=None*, *path='.'*)

request data of entity specified by idn in given schema and (optional) style
afterwards save it to directory at path

1.2 dnb module

librair.services.dnb.**address**(*idn*, *schema*)
get url of entity specified by idn in given schema

librair.services.dnb.**request**(*idn*, *schema='lds'*)
request data of entity specified by idn in given schema

SCHEMA	RETURN TYPE
bibframe	str
lds	str
marcxml	lxml.etree.Element

to do:

- handle lds (RDF Turtle)
- handle bibframe

librair.services.dnb.**store**(*idn*, *schema='lds'*, *path='.'*)

request data of entity specified by idn in given schema
afterwards save it to directory at path

1.3 entityfacts module

librair.services.entityfacts.**address**(*idn*)
get url of entity specified by idn

librair.services.entityfacts.**request**(*idn*)
request data of entity specified by idn

librair.services.entityfacts.**store**(*idn*, *path='.'*)

request data of entity specified by idn
afterwards save it to directory at path

1.4 gnd module

librair.services.gnd.**address**(*idn*, *schema*)
get url of entity specified by idn in given schema

librair.services.gnd.**request**(*idn*, *schema='lds'*)
request data of entity specified by idn in given schema

SCHEMA	RETURN TYPE
lds	str
marcxml	lxml.etree.Element

to do:

- handle lds (RDF Turtle)

librair.services.gnd.**store**(*idn*, *schema='lds'*, *path='.'*)

request data of entity specified by idn in given schema
afterwards save it to directory at path

1.5 kalliope module

librair.services.kalliope.**address**(*idn*)
get url of entity specified by idn

librair.services.kalliope.**request**(*idn*)
request data of entity specified by idn

librair.services.kalliope.**store**(*idn*, *path='.'*)

request data of entity specified by idn
afterwards save it to file at path

1.6 lobid module

librair.services.lobid.**gnd_address**(*idn*)
get url of entity specified by idn

librair.services.lobid.**gnd_request**(*idn*)
request data of entity specified by idn

librair.services.lobid.**gnd_store**(*idn*, *path='.'*)

request data of entity specified by idn
afterwards save it to directory at path

1.7 oclc module

librair.services.oclc.**address**(*idn*, *schema*)
get url of entity specified by idn in given schema

librair.services.oclc.**request**(*idn*, *schema='jsonld'*)
request data of entity specified by idn in given schema

SCHEMA	RETURN TYPE
nt	str
rdf	lxml.etree.Element
ttl	str
jsonld	dict

librair.services.oclc.**store**(*idn*, *schema='rdf'*, *path='.'*)

request data of entity specified by idn in given schema
afterwards save it to file at path

1.8 seealso module

librair.services.seealso.**address**(*idn*, *schema*)
get url of entity specified by idn in given schema

librair.services.seealso.**request**(*idn*, *schema='seealso'*)
request data of entity specified by idn in given schema

SCHEMA	RETURN TYPE
seealso	json
sources	str (html)
redirect	str (html)
opensearchdescription	lxml.etree.Element

librair.services.seealso.**store**(*idn*, *schema='rdf'*, *path='.'*)

request data of entity specified by idn in given schema
afterwards save it to directory at path

1.9 viaf module

librair.services.viaf.**address**(*idn*, *schema*)
get url of entity specified by idn in given schema

librair.services.viaf.**request**(*idn*, *schema='rdf'*)
request data of entity specified by idn in given schema

SCHEMA	RETURN TYPE
links	dict
marc21	lxml.etree.Element
rdf	lxml.etree.Element
viaf	lxml.etree.Element

librair.services.viaf.**store**(*idn*, *schema='rdf'*, *path='.'*)

request data of entity specified by idn in given schema
afterwards save it to directory at path

1.10 zdb module

librair.services.zdb.**address**(*idn*, *schema*)
get url of entity specified by idn in given schema

librair.services.zdb.**request**(*idn*, *schema='jsonld'*)
request data of entity specified by idn in given schema

SCHEMA	RETURN TYPE
html	str
rdf	lxml.etree.Element
ttl	str
jsonld	dict

librair.services.zdb.**store**(*idn*, *schema='jsonld'*, *path='.'*)

request data of entity specified by idn in given schema
afterwards save it to directory at path

CHAPTER 2

Interfaces

2.1 hydra module

class librair.interfaces.hydra.**Client**(*URL*)
Bases: object

Hydra client for database given by URL

address(*query*, *size*, *page*)
get url for given query, size and page

request(*idn*)
request data specified by idn

retrieve(*query*, *size=10*, *page=1*)
retrieve data for given query in size from page

scroll(*query*, *size=100*, *page=1*)

scroll items matching given query, in steps specified by size, starting from given page

total(*query*)
determine total number of results for given query

class librair.interfaces.hydra.**Response**(*data*)
Bases: object

wrapper for json response from hydra interface

members()
get record data from response

total()
get total number of records from response

view_next()
get url of next items from result set

2.2 sru module

class librair.interfaces.sru.**Client**(*URL*, *token=None*)
Bases: object

sru client for database given by url

address(*query*, *schema*, *records=10*, *operation='searchRetrieve'*)
generate request url for given query and schema

query(*index*, *value*)
get query for given index and value

scroll(*query*, *schema*, *size=100*)
scroll items matching given query, request given schema and in steps given by size, return all collected records

search(*query*, *schema*, *records=1*)
search items matching given query, request given schema and return number of items given by records

secret(*token*)
set access token for sru endpoint

total(*query*, *schema*)
determine total number of results for given query

class librair.interfaces.sru.**Explain**(*URL*, *load=False*)
Bases: object

fetch and process explain response from given URL

request is sent if load=True otherwise via self.load()

load()
general routine to load data

load_config()
config_info = self.root.find("configInfo", relative=False)

load_database()
load info about schemas given in sru explain response

load_indexes()
load info about indexes given in sru explain response

load_raw()
load sru explain response

load_root()
load explain element from response

load_schemas()
load info about schemas given in sru explain response

load_server()
server_info = self.root.find("serverInfo")

load_version()
load version of interface given in sru explain response

store(*db*, *path=''*)
write explain xml to file at path

class librair.interfaces.sru.**Response**(*element*, *ns=None*)
Bases: *librair.schemas.xml.Element*

wrapper for xml response from sru interface

diagnostics()
get diagnostics for response

items()
get record data from response

total()
get total number of records from response

2.3 unapi module

class librair.interfaces.unapi.**Client**(*URL*, *DB*, *VAR*)
Bases: object

unAPI client for database given by url and id and items specified by given variable

address(*idn*, *schema*)
get url of entity specified by idn in given schema

request(*idn*, *schema*)
request data of entity specified by idn in given schema

CHAPTER 3

Protocols

3.1 http module

librair.protocols.http.**get_request**(*url*, *headers={}*)
send http get request to given url with (optional) headers

librair.protocols.http.**response_json**(*response*)
get json data from http repsonse

librair.protocols.http.**response_ok**(*response*)

librair.protocols.http.**response_text**(*response*)
get text data from http repsonse

librair.protocols.http.**response_xml**(*response*)
get xml data from http repsonse

3.2 sru module

librair.protocols.sru.**address**(*base*, *query*, *schema='mods'*, *records=10*, *version='1.2'*, *operation='searchRetrieve'*)
get http address for given parameters

librair.protocols.sru.**explain**(*base*, *version*)
return explanation page for given base and version

librair.protocols.sru.**retrieve**(*url*)
send request to given url and return xml response

CHAPTER 4

Schemas

4.1 json module

librair.schemas.json.**filepath**(*term*, *base*, *schema*, *service='api'*)
generate file path for given parameters

librair.schemas.json.**reader**(*path*)
read json file from given path

librair.schemas.json.**writer**(*data*, *file*, *path='.'*)
write given dict data to json file at path

4.2 plain module

librair.schemas.plain.**reader**(*path*, *encoding='utf-8'*)
read txt file from given path

librair.schemas.plain.**writer**(*data=None*, *file='out.txt'*, *path='.'*)
write given data (str or list of str) to txt file at path

4.3 xml module

class librair.schemas.xml.**Element**(*tree*, *ns=None*)
Bases: object

wrapper for xml element

find(*tag*, *relative=True*)
find tag in raw xml from namespace of root

findall(*tag*, *relative=True*)
find all tags in raw xml from namespace of root

namespace()
extract full namespace (URL) from root element

xpath(*tag*, *relative=True*)
create xpath query for given tag

librair.schemas.xml.**filepath**(*term*, *base*, *schema*, *service='sru'*)
generate file path for given parameters

librair.schemas.xml.**pretty**(*elements*)
pretty print elements

librair.schemas.xml.**reader**(*path*)
read xml file at given path

librair.schemas.xml.**unlist**(*elements*)
create tree from list of elements

librair.schemas.xml.**writer**(*tree*, *file*, *path='.'*)
write given element tree to file at path

CHAPTER 5

Parsers

class librair.parsers.**Beacon**(*path=''*, *url=''*)

Bases: object

Beacon parser for files exposed by given path or url

save(*file='beacon.txt'*, *path='.'*)

save beacon data to given path and file

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

l

librair.interfaces.hydra, 7
librair.interfaces.sru, 8
librair.interfaces.unapi, 9
librair.parsers, 15
librair.protocols.http, 11
librair.protocols.sru, 11
librair.schemas.json, 13
librair.schemas.plain, 13
librair.schemas.xml, 13
librair.services.cerl, 1
librair.services.dnb, 1
librair.services.entityfacts, 2
librair.services.gnd, 2
librair.services.kalliope, 3
librair.services.lobid, 3
librair.services.oclc, 3
librair.services.seealso, 4
librair.services.viaf, 4
librair.services.zdb, 4

Index

A

address() (*in module librair.protocols.sru*), 11
address() (*in module librair.services.cerl*), 1
address() (*in module librair.services.dnb*), 1
address() (*in module librair.services.entityfacts*), 2
address() (*in module librair.services.gnd*), 2
address() (*in module librair.services.kalliope*), 3
address() (*in module librair.services.oclc*), 3
address() (*in module librair.services.seealso*), 4
address() (*in module librair.services.viaf*), 4
address() (*in module librair.services.zdb*), 4
address() (*librair.interfaces.hydra.Client method*), 7
address() (*librair.interfaces.sru.Client method*), 8
address() (*librair.interfaces.unapi.Client method*), 9

B

Beacon (*class in librair.parsers*), 15

C

Client (*class in librair.interfaces.hydra*), 7
Client (*class in librair.interfaces.sru*), 8
Client (*class in librair.interfaces.unapi*), 9

D

diagnostics() (*librair.interfaces.sru.Response method*), 9

E

Element (*class in librair.schemas.xml*), 13
Explain (*class in librair.interfaces.sru*), 8
explain() (*in module librair.protocols.sru*), 11

F

filepath() (*in module librair.schemas.json*), 13
filepath() (*in module librair.schemas.xml*), 14
find() (*librair.schemas.xml.Element method*), 13
findall() (*librair.schemas.xml.Element method*), 13

G

get_request() (*in module librair.protocols.http*), 11
gnd_address() (*in module librair.services.lobid*), 3
gnd_request() (*in module librair.services.lobid*), 3
gnd_store() (*in module librair.services.lobid*), 3

I

items() (*librair.interfaces.sru.Response method*), 9

L

librair.interfaces.hydra (*module*), 7
librair.interfaces.sru (*module*), 8
librair.interfaces.unapi (*module*), 9
librair.parsers (*module*), 15
librair.protocols.http (*module*), 11
librair.protocols.sru (*module*), 11
librair.schemas.json (*module*), 13
librair.schemas.plain (*module*), 13
librair.schemas.xml (*module*), 13
librair.services.cerl (*module*), 1
librair.services.dnb (*module*), 1
librair.services.entityfacts (*module*), 2
librair.services.gnd (*module*), 2
librair.services.kalliope (*module*), 3
librair.services.lobid (*module*), 3
librair.services.oclc (*module*), 3
librair.services.seealso (*module*), 4
librair.services.viaf (*module*), 4
librair.services.zdb (*module*), 4
load() (*librair.interfaces.sru.Explain method*), 8
load_config() (*librair.interfaces.sru.Explain method*), 8
load_database() (*librair.interfaces.sru.Explain method*), 8
load_indexes() (*librair.interfaces.sru.Explain method*), 8
load_raw() (*librair.interfaces.sru.Explain method*), 8
load_root() (*librair.interfaces.sru.Explain method*), 8

load_schemas() (*librair.interfaces.sru.Explain method*), 8
load_server() (*librair.interfaces.sru.Explain method*), 8
load_version() (*librair.interfaces.sru.Explain method*), 8

M

members() (*librair.interfaces.hydra.Response method*), 7

N

namespace() (*librair.schemas.xml.Element method*), 13

P

pretty() (*in module librair.schemas.xml*), 14

Q

query() (*librair.interfaces.sru.Client method*), 8

R

reader() (*in module librair.schemas.json*), 13
reader() (*in module librair.schemas.plain*), 13
reader() (*in module librair.schemas.xml*), 14
request() (*in module librair.services.cerl*), 1
request() (*in module librair.services.dnb*), 1
request() (*in module librair.services.entityfacts*), 2
request() (*in module librair.services.gnd*), 2
request() (*in module librair.services.kalliope*), 3
request() (*in module librair.services.oclc*), 3
request() (*in module librair.services.seealso*), 4
request() (*in module librair.services.viaf*), 4
request() (*in module librair.services.zdb*), 4
request() (*librair.interfaces.hydra.Client method*), 7
request() (*librair.interfaces.unapi.Client method*), 9
Response (*class in librair.interfaces.hydra*), 7
Response (*class in librair.interfaces.sru*), 9
response_json() (*in module librair.protocols.http*), 11
response_ok() (*in module librair.protocols.http*), 11
response_text() (*in module librair.protocols.http*), 11
response_xml() (*in module librair.protocols.http*), 11
retrieve() (*in module librair.protocols.sru*), 11
retrieve() (*librair.interfaces.hydra.Client method*), 7

S

save() (*librair.parsers.Beacon method*), 15
scroll() (*librair.interfaces.hydra.Client method*), 7
scroll() (*librair.interfaces.sru.Client method*), 8
search() (*librair.interfaces.sru.Client method*), 8
secret() (*librair.interfaces.sru.Client method*), 8
store() (*in module librair.services.cerl*), 1
store() (*in module librair.services.dnb*), 2
store() (*in module librair.services.entityfacts*), 2
store() (*in module librair.services.gnd*), 2
store() (*in module librair.services.kalliope*), 3
store() (*in module librair.services.oclc*), 3
store() (*in module librair.services.seealso*), 4
store() (*in module librair.services.viaf*), 4
store() (*in module librair.services.zdb*), 5
store() (*librair.interfaces.sru.Explain method*), 8

T

total() (*librair.interfaces.hydra.Client method*), 7
total() (*librair.interfaces.hydra.Response method*), 7
total() (*librair.interfaces.sru.Client method*), 8
total() (*librair.interfaces.sru.Response method*), 9

U

unlist() (*in module librair.schemas.xml*), 14

V

view_next() (*librair.interfaces.hydra.Response method*), 7

W

writer() (*in module librair.schemas.json*), 13
writer() (*in module librair.schemas.plain*), 13
writer() (*in module librair.schemas.xml*), 14

X

xpath() (*librair.schemas.xml.Element method*), 14