MarcExtraction Documentation

Release Beta

Tyler Danstrom

Jun 05, 2018

Contents:

CHAPTER 1

Introduction

This is a Python library that allows a consumer to extract MARC records from

- a file exported to disk
- a VuFind API

CHAPTER 2

Quick start

CHAPTER 3

Authored by

verbalhanglider (tdanstrom@uchicago.edu)

3.1 Interface Classes

the interface classes to allow for building a list of records and/or searching for relevant records

class marcextraction.interfaces.**OLERecordFinder**(*bibnumber*, *ole_domain*, *ole_scheme*, *ole_path*)

a class to use for finding a particular MARC record from the OLE API

Useage: finder = OLERecordFinder("1003495521", "https://example.com/oledocstore") is_it_there, data = finder.get_record() if is_it_there:

return data

get_record()

a public method to get the matching record (if one was found for the inputted bibnumber)

Returns: tuple. first element is boolean result

class marcextraction.interfaces.**OnDiskSearcher**(*writeable_object=None*, *location=None*)

a class to use for building up a list of exported MARC files at a particular location on-disk

Useage: searcher = OnDiskSeacher(location='/path/to/marc/records') searcher.search('Cartographic Mathematical Data', 'Spatial coordinates')

count()

a method to return the total number of records extracted

Returns: int. total records found on-disk

classmethod from_flo(*flo*)

a method to instantiate an instance of OnDiskExtractor from a file-like object

Args: flo (File Object): a file-like object with read, write methods

Returns: OnDiskSearcher

search(*query_term*, *field*, *subfields*)
a method to search for records matching query term and field lookup

Args: query_term (str): the string to be searched. This string will be stemmed in Solr searches. field (str): a MARC field number as a string subfields (list): a list of subfield codes related to the field that you want to search

Returns:

list. an iterable containing dicitonaries

:rtype list

class marcextraction.interfaces.**SolrIndexSearcher**(*index_url*, *index_type*)
a class to be used to search a Solr index for a query

search(*query_term*, *field*, *subfields*, *rows=1000*, *phrase_search=False*)
a method to run a search on the index for a particular value in a particular field

Args: query_term (str): the string to be searched. This string will be stemmed in Solr searches.. field (str): a MARC field number as a string. subfields (list): a list of subfield codes related to the field that you want to search.

KWArgs: rows (int): the number of records that you want to retrieve from the Solr index. default is 1000. phrase_search (bool): a flag indicating whether you want to perform a full phrase search. Default

is False which will perform a word search.

Returns:

list. An iterable containing dictionaries for each matching record in the Solr index for the query_term, query_field, and query_subfield.

3.2 Utilities for Building Index Field Names and Query Strings

utility functions for working with ole index data

marcextraction.utils.**create_ole_index_field**(*field_name*)
a method to return the marc field name as entered in the OLE index

Args: field_name (str): a MARC field number with a subfield code as a single string. Ex '245a'

marcextraction.utils.**create_ole_query**(*field_name*, *query_term*, *phrase_term=False*)
a method to return the query string for searching for making a field query in OLE

Args: field_name (str): a MARC field combined with a subfield code with prefix '**mdf_**. Ex. 'mdf_245a'. query_term (str): a word or phrase

Returns: str. A full query string to be entered into a Solr index for searching on a particular field. Ex. 'mdf_245a:banana'

marcextraction.utils.**find_ole_bib_numbers**(*ole_data_list*)
a method to find bib numbers from a set of OLE results

Args: ole_data_list (list): a list of dictionaries containing output from a Solr search of an OLE index.

Returns: list. an iterable containing strings that should represent bib numbers. Ex. ['1000435999', '10045334500']

CHAPTER 4

Indices and tables

- genindex
- modindex
- search

Python Module Index

m

Index