Mr. DLib User Guide

Contents

[1 Web Services offered by Mr. DLib 2](#_Toc317787251)

[2 API Overview 3](#_Toc317787252)

[3 Specification of Resource Types 5](#_Toc317787253)

[3.1 import 5](#_Toc317787254)

[3.2 authors 5](#_Toc317787255)

[3.3 documents 5](#_Toc317787256)

[3.4 fulltext 6](#_Toc317787257)

[3.5 login 6](#_Toc317787258)

[3.6 organization 6](#_Toc317787259)

[3.7 service 7](#_Toc317787260)

[3.8 Tools 7](#_Toc317787261)

[3.9 xref 7](#_Toc317787262)

[4 Current Development Status and Future Plans 9](#_Toc317787263)

[5 Appendix 10](#_Toc317787264)

This document describes the application programming interface (API) of Mr. DLib, which is an acronym for machine-readable digital library. The first section of the document gives an overview of Mr. DLib’s RESTful API. The second section describes the interface, its parameters and the used output format. The third section lists the available resources and the information they offer. The fourth section summarizes the development status of individual API components and gives an outlook on planned features. The appendix contains examples for common queries.

# Web Services offered by Mr. DLib

This section introduces the web services offered by Mr. DLib via a RESTful API. The REST architecture closely resembles the design of the HTTP standard. This similarity of basic concepts in REST and HTTP simplifies the usage of the API for web developers.

REST is resource-focused. The main resources in Mr. DLib are *documents*[[1]](#footnote-2), *persons*[[2]](#footnote-3)*, conferences*, *journals*, and *organizations*[[3]](#footnote-4)*.* Mr. DLib provides links between these resources, e.g. between authors and articles, and between articles and conference proceedings wherever possible.

Retrieving data from Mr. DLib requires a simple HTTP GET command on an URI using a HTTP client (e.g. a web browser). Analogously, adding, modifying and deleting resources in Mr. DLib will be possible (not yet part of the public API) by using the POST, PUT and DELETE methods defined in the HTTP standard. Likewise, the Mr. DLib API uses status messages that are adopted from the HTTP standard. For instance, if a resource does not exist, Mr. DLib returns a 404 NOT FOUND error. If a request is successful, the API returns a 200 OK status.

# API Overview

Mr. DLib is accessible via the HTTP protocol. The base URL is:

<http://api.mr-dlib.org/>

The complete URL structure is:

[http://api.mr‑dlib.org/<resource\_type>[/<resource\_sel>[/<resource\_element>]][/][<parameters](http://api.mrdlib.org/%3cresource_type%3e%5b/%3cresource_sel%3e%5b/%3cresource_element%3e%5d%5d%5b/%5d%5b%3cparameters)>]

All arguments in parentheses (so all except resource\_type) are optional. The individual parameters are:

*resource\_type*: The argument specifies the type of the resource to be requested. Details of the resource types available in Mr. DLib are listed in section 3. For instance, a request of the URL http://api.mr-dlib.org/documents returns a list of all documents in Mr. DLib. The output includes basic metadata for each document and a URI pointing to the specific document resource as shown in Figure 1.

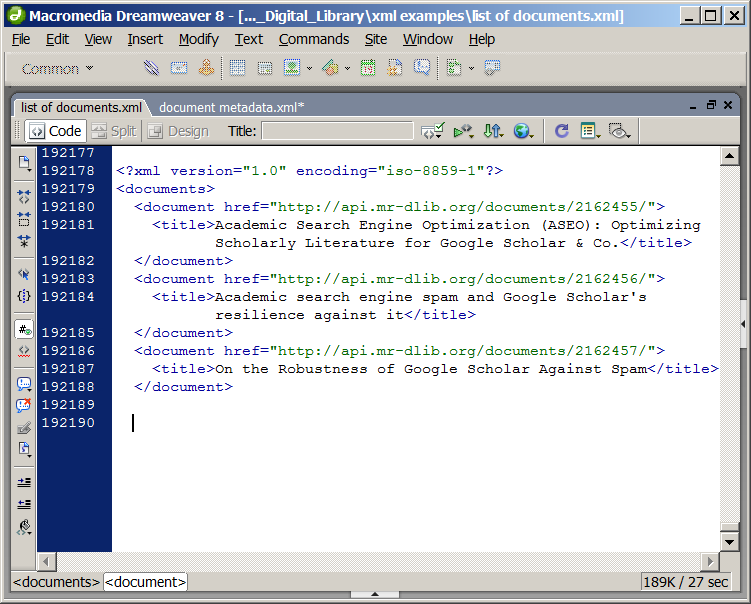
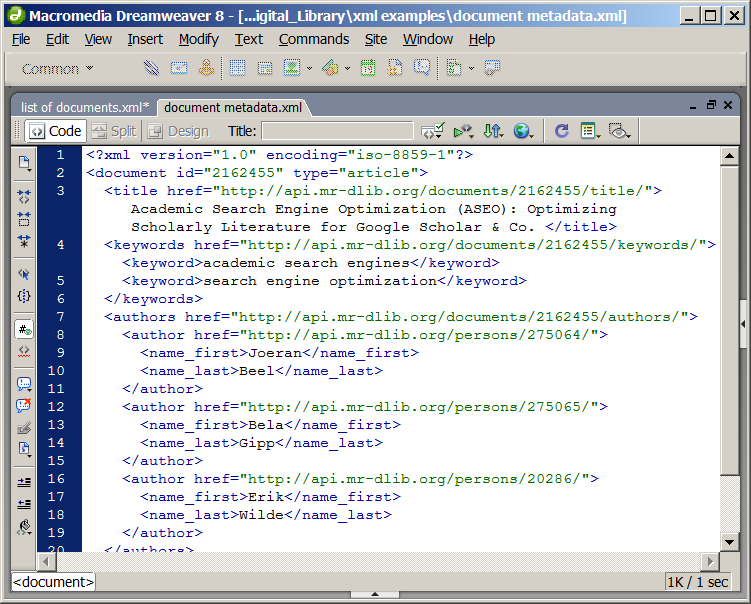


Figure 1: List of documents in XML format

*resource\_sel (Resource Selector)*: Resource selectors can be specified for limiting the returned result set to specific resources. Depending on the resource type, different selectors can be used, which are explained in section 3. Each record in Mr. DLib has a unique numeric identifier (ID), which is a valid selector for all resource types. Details of a specific resource can be retrieved by requesting a URL of the form: [http://api.mr‑dlib.org/<resource\_type>/<resource\_id](http://api.mrdlib.org/%3cresource_type%3e/%3cresource_id)>*.* For instance, requesting the URL http://api.mr-dlib.org/documents/1234 returns all data for the document having the ID 1234 as shown in Figure 2.



**Figure 2: Metadata for a document (excerpt)**

*resource\_element (Resource Element)*: The argument can be used to limit the output to certain elements of the requested resource. The web service returns different data elements depending on the resource type and the availability of the specific data element for a certain entity. For instance, a journal article consists of elements including *title*, *authors*, *keywords*, *publishing date*, *page numbers*, and *URIs to full-text*s.

*parameters*: Different parameters can be specified for controlling the behavior of Mr. DLib. These are:

* ‘*format’* specifies the data format in which results are to be returned. Mr. DLib supports the following formats:
  + ‘xml’ (default): XML
  + ‘json’: JSON
* ‘*filter*’ (unstable API): is a special parameter used for limiting the original query to a subset of the data contained in Mr. DLib. For instance, issuing the request http://api.mr*‑*dlib.org/documents/?filter=arxiv would display all documents in Mr. DLib`s database originating from arXiv.org
* ‘*dlo*’ (date low) and ‘*dhi’* (date high): can be used to filter for publication date of documents. For instance, issuing the request http://api.mr‑dlib.org/documents/?dlo=2011-01-01&dhi=2011-02-28returns documents published in January and February 2011
* ‘*number*’ (default=1000): adjusts the number of results per page
* ‘*page*’ (default=1): sets the page number of the result list to be returned

# Specification of Resource Types

This section describes the resource types available in Mr. DLib, the respective data elements they consist of and the selectors available for retrieving individual resources.

## import

POST /import/resource adds a resource from XML file (internal).

## authors

The authors resource type is used to store author information in Mr. DLib.

### Resource Selectors

* none: Lists all authors
* {id}: Returns solely the author resource with the specified numeric ID
* {name}: Returns last name(s) starting with the specified string

### Resource Elements

none

## documents

The documents resource type is used to store document metadata in Mr. DLib.

### Resource selectors

* none (POST): Adds a document given as XML or PDF file to Mr. DLib and returns the newly created resource
* none (GET): Lists all documents
* {id}: Returns the document resource having the given id
* {q}: Document matching a Lucene search query. Lucene’s query format is described in the Lucene documentation <<http://lucene.apache.org/core/old_versioned_docs/versions/3_5_0/queryparsersyntax.html>>. The following fields are available in Mr. DLib:
  + id: ID
  + type: article type
  + title: article title
  + abstract: articles abstract
  + author: documents authors
  + affiliation: documents authors affiliation
  + keyword: keyword
  + publishedin: the venue (e.g. the journal or conference proceedings)
  + pubdate: publication date
  + location: place of publication
  + publisher: the publisher
  + edition: journal edition
  + number: journal number
  + volume: journal volume
  + pages: pages of the document in the journal
  + series: series of the journal
  + doi: DOI
  + issn: ISSN
  + isbn: ISBN
  + lang: language
  + oid: id at other sources. Format: <source>/<sources\_id>
  + reflist: search reference list of the article

Examples:

* http://api.mr-dlib.org/documents/author:John returns all documents that have an author whose name contains “John”
* http://api.mr-dlib.org/documents/author:John%20AND%20abstract:study returns all documents that have an author whose name contains “John” and the abstract contains “study”
* totalamount: Returns the count of documents in Mr. DLib
* latestpublicationdate: Returns latest publication date of all documents

### Resource Elements

* title: Returns document title(s)
* abstract: Returns document abstract(s)
* authors: Returns author(s)
* fulltexts: Returns document fulltext(s)
* xrefs: Returns document cross-reference(s)

## fulltext

The fulltext resource type is used to store the full text of documents in Mr. DLib.

### Resource selectors

* {id}: Returns the document fulltext having the given ID

### Resource Elements

none

## login

## organizations

The organization resource is used to store information about organizations such as universities and publishers in Mr. DLib.

### Resource Selectors

* {id}: Returns the organization having the given ID

### Resource Elements

none

## service

### Resource Selectors

* versioncheck/{appID}/current
* stats/usage/{appID}/{event}/
* metadata/available
* metadata (POST)
* metadata (POST) ?????

### Resource Elements

## Tools

The tools resource type provides data processing utilities that do not necessarily depend on the data contained in Mr. DLib.

### Resource Selectors

* pdfextract (available via POST and GET): Extracts the metadata from a PDF file and returns it in the format used for document resources in Mr. DLib.
  + The selector requires the specification of ONE of the following arguments
    - url: a URL to a PDF file
    - file: a PDF file

The *file* parameter works only via POST and with multipart/form-data encoding. If a PDF is supplied via an URL, *pdfextract* works using the "normal" POST and GET method. Sample requests for each variant using HTML (via web browser) and Python can be found in appendix XXX.

### Resource Elements

none

## xref

The xref resource contains information about cross-references from individual resources to other services. For example, a document may contain a cross-reference to another third-party service that is also offering data on the document.

### Resource Selectors

* {id}: The xref resource with the given ID

### Resource Elements

none

# Current Development Status and Future Plans

Mr. DLib is currently in beta: it is functional and in use by some third parties such as *Docear*[[4]](#footnote-5). We consider the available resources as stable. However, authentication methods, URIs, filters, query parameters, and other details may change in future versions.

We are working on improving the interface by providing more flexible methods for selecting elements and filtering results. In addition, we plan to provide data import methods that include document de-duplication, author name disambiguation and access restrictions. Another planned extension is the integration of a recommender system for related articles.

# Appendix

1. Such as journal and conference articles, conference proceedings, books, and theses [↑](#footnote-ref-2)
2. Such as authors, editors, and conference chairs [↑](#footnote-ref-3)
3. Such as universities and publishers [↑](#footnote-ref-4)
4. http://www.docear.org [↑](#footnote-ref-5)