Table of Contents

INDIVIDUAL DOCUMENTS 1

SEARCH RESULTS 3

QUERY NOTES 5

Industry Documents Library Solr API

The Industry Documents Library (IDL) uses Apache Solr to index the document corpus. Users who are interested in accessing the data programmatically can query the Industry Documents Library (IDL) Solr server directly. This allows the user to easily export documents to another system, execute search queries and process search results by program. Data can be exported in these formats: xml, json, python, ruby, php, and csv.

Individual Documents

Each document is uniquely identified by an ID. The ID is an alphanumeric string. It consists of 8 characters with four letters followed by four digits, e.g. kylw0221. The ID is not case sensitive: KYLW0221, kYLw0221, kylw0221 all refer to the same document.

To access a document's metadata in xml format, please query the Industry Documents Library (IDL) Solr server with the ID.

For example, to extract the information of document with ID kylw0221:
https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=id:kylw0221

The response looks like this:

```
<response>
<lst name="responseHeader">
<int name="status">0</int>
<int name="QTime">10</int>
<lst name="params">
<str name="q">id:kylw0221</str>
</lst>
</lst>
<result name="response" numFound="1" start="0">
<doc>
<str name="id">kylw0221</str>
<str name="tid">ctg37j00</str>
<arr name="collection">
<str>Depositions and Trial Testimony (DATTA)</str>
</arr>
<arr name="availability">
```

```
<str>no restrictions</str>
<str>public</str>
</arr>
<arr name="case">
<str>
Engle Progeny; Andy R. Allen, Sr. and Patricia L. Allen, Case No. 16-
2007-CA-008311-BXXX-MA, Case No. 2008-CA-15000
</str>
</arr>
<str name="titie">
In Re: Engle Progeny Cases Tobacco Litigation. Pertains to Andy R.
Allen, Sr., as Personal Representative for the Estate of Patricia L.
Allen. Jury Trial
</str>
<str name="documentdate">2014 November 25</str>
<arr name="type">
<str>trial transcript</str>
</arr>
<int name="pages">155</int>
<str name="bates">figlarj20141125</str>
<str name="witness">Figlar, James, Ph.D.</str>
<str name="dateaddeducsf">2015 March 05</str>
</doc>
</result>
</response>
```

The parameters of interest are:

Parameter	**Description**	**Comment**
q	query	id:[ID]
wt	writer type	xml (default) json python ruby php csv

If one is interested in extracting the same data in json format, simply attach &wt=json to the url:
https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=id:kylw0221&wt=json

The response looks like this:

```
{
  "responseHeader":{
    "status":0,
    "QTime":2,
    "params":{
      "indent":"true",
      "q":"id:kylw0221",
      "wt":"json"}},
  "response":{"numFound":1,"start":0,"docs":[
      {
        "id":"kylw0221",
        "tid":"ctg37j00",
        "collection":["Depositions and Trial Testimony (DATTA)"],
        "availability":["no restrictions",
          "public"],
        "case":["Engle Progeny; Andy R. Allen, Sr. and Patricia L.
Allen, Case No. 16-2007-CA-008311-BXXX-MA, Case No. 2008-CA-15000"],
        "titie":"In Re: Engle Progeny Cases Tobacco Litigation.
Pertains to Andy R. Allen, Sr., as Personal Representative for the
Estate of Patricia L. Allen. Jury Trial",
        "documentdate":"2014 November 25",
        "type":["trial transcript"],
        "pages":155,
        "bates":"figlarj20141125",
        "witness":"Figlar, James, Ph.D.",
        "dateaddeducsf":"2015 March 05"}]
  }}
```

Searches

Users can also run queries against the Solr server and extract results in the desired format. In order to prevent user issued queries from overloading the Solr sever, we allow the retrieval of 100 records at a time, users can page through the results by appending ***&start=[number]*** to the request. Deep paging using start is expensive for a Solr server, so allow maximum of 10,000 pages using start. If you wish to access further, you will need to use cursorMark.

The parameters of interest are:

Parameter	Description	Comment
Q	query	The default is to return all records. If no q is passed in or q=* or q=*:*, all records are returned. Please see the "Query Notes" section below for more details on how to query the Solr server. Use Boolean operators to refine your query: AND, OR, NOT.
Wt	writer type	xml (default) json python ruby php csv
Start	start	For paging through results. 100 records are returned at a time. **NOTE**: Using start is costly for solr server when paging deeply. We have a restriction on start value that cannot be bigger than 10,000. If you

		wish to page beyond 10,000 pages, you have to use cursorMark. See solr documentation: https://solr.apache.org/guide/8_5/pagination-of-results.html

For example, to query for all documents with author Glantz and extract the data in xml format:

https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=author:glantz

The response looks like:

```
<response>
<lst name="responseHeader">
<int name="status">0</int>
<int name="QTime">2</int>
<lst name="params">
<str name="q">author:glantz</str>
</lst>
</lst>
<result name="response" numFound="1189" start="0">
<doc>
<str name="id">qpnl0052</str>
<str name="tid">dpl03c00</str>
<arr name="collection">
<str>RJ Reynolds</str>
</arr>
<str name="area">
OPERATIONS; ENGINEERING; BOHANON HR; SR PRINCIPAL ENGINEER
</str>
<arr name="availability">
<str>public</str>
<str>no restrictions</str>
</arr>
<str name="topic">CTR/TIRC/TI; TOBACCO INSTITUTE</str>
<str name="box">NA; RJR9941</str>
<arr name="case">
<str>US RESEARCH AND MANUFACTURING DOCUMENT PRODUCTION</str>
</arr>
<str name="title">
ECNOMIC IMPACT OF GOVERNMENT MANDATED SMOKING RESTRICTIONS ON THE
RESTAURANT INDUSTRY.
</str>
<arr name="author">
<str>RJR</str>
<str>BOHANON H</str>
<str>GLANTZ S</str>
<str>TI</str>
<str>GLANTZ & SMITH</str>
<str>OSHA</str>
<str>WAXMAN H</str>
<str>SENATE</str>
<str>EPA</str>
<str>CRAIG</str>
```

```
<str>NCSU</str>
<str>NRA</str>
</arr>
<str name="documentdate">1995 April 19</str>
<arr name="type">
<str>report</str>
</arr>
<int name="pages">24</int>
<arr name="mentioned">
<str>LIST OF FOOTNOTES</str>
<str>HANSEN & LOTT</str>
<str>MADD</str>
</arr>
<str name="description">Marginalia; Y</str>
<str name="bates">525616666-525616689</str>
<str name="minnesotarequestnumber">US RESEARCH AND MANUFACTURING
DOCUMENT PRODUCTION</str>
<str name="dateshipped">2015 August 06</str>
<str name="dateaddeducsf">2003 January 15</str>
<str name="dateaddedindustry">2002 October 15</str>
<str name="datemodifiedindustry">2012 April 17; 2015 August 06</str>
</doc>
...
```

The attribute ***numFound="1189"*** shows that there are 1189 records that match this query. Since we only return 100 records at a time, paging is needed to access all records.

To see the next 100 records, append &start=100 to the url:
https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=author:glantz&start=100

To see the remaining 32 records, append &start=200 to the urls:
https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=author:glantz&start=200

If the user wants to find **author:glantz** only in the tobacco industry, then you should add ***AND industry:tobacco*** to the query.
https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=(author:glantz AND industry:tobacco)

If the user wants to find all records from the Brown & Williamson collection with type letter but NOT with brand Kool. Please note that you cannot use & directly in the URL query. You must substitute it with %26. (Also see the Ampersand Section under Query Notes.) This is because & is a special URL character. The query should look like this:
https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=((collection:"brown %26 williamson" AND type:letter) NOT brand:kool)

Example with cursorMark deep paging

- You must have a sort order. This example uses sort by id desc
- Initial request must pass cursorMark=*
- Response will have a nextCursorMark value, let's call it XXX
- Subsequent requests pass nextCursorMark value into &cursorMark=XXX

Initial Request:
https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=tobacco&wt=json&cursorMark=*&sort=id%20desc

At the end of the response, you will see:
```
"nextCursorMark":"AoEoenp5eDAyMTY="}
```

Next Request:
https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=tobacco&wt=json&cursorMark=AoEoenp5eDAyMTY=&sort=id%20desc

Pseudo code for looping through search results using cursorMark:

```
// NOTE: This is not real code.
//       This is pseudo code to demo the logic of how to loop through
//       the result set using cursorMark.
//       Please translate to your programming language of choice.

$params = [ q => $some_query, sort => 'id asc', cursorMark => '*' ]
$done = false
while (not $done) {
  $results = fetch_solr($params)

  // write logic here to do something with $results

  if ($params[cursorMark] == $results[nextCursorMark]) {
    $done = true
  }
  $params[cursorMark] = $results[nextCursorMark]
}
```

Query Notes

The Industry Documents Library (IDL) Solr server does not have the syntax sugar that the Industry Documents Library (IDL) website adds for the user. But most of the queries entered on the website can be directly issued to the Solr server **with a few exceptions**.

Dates:
The date fields are stored as a string in the format of "4-digit-year month 2-digit-day", for example "2001 November 09".

Searching for an exact date can be done on the date field, for example
q=documentdate:"2001 November 09".

However, range searches do not work on string fields. We also store the date fields in ISO format that is meant for date range searches. If one wants to search for documents dated between 2001 and 2011, the search must be done on the ***documentdateiso*** field ***(q=documentdateiso:[2001-01-01T00:00:00Z TO 2011-12-31T00:00:00Z]***).

Bates search:
When searching for a bates number on the Solr server, please use the ***batesexpanded*** field code (***q=batesexpanded:XXXXX***). This is because bates number could cover a range. One could find a single bates number within the bates range by querying the ***batesexpanded*** field.

cited:yes/cited:no:
The query ***cited:yes*** works on the website, but it needs to be issued as ***q=cited:**** with the Solr server directly.
Similarly ***cited:no*** needs to be issued as ***q=–cited:**** or ***q=NOT cited:**** to the Solr server.

Ampersand &
Some field values have Ampersand in them, for example Brown & Williamson. Ampersand is a special character in the URL and when searching for values with ampersand, please use %26.

For example: https://metadata.idl.ucsf.edu/solr/ltdl3/query?q=collection:"brown %26 williamson"

Operators:
Use operators AND, OR, NOT to refine your search. Always remember to use parentheses to indicate how you want your query to be interpreted.
For example:
q=author:glantz AND type:letter OR brand:kool

The above query will give you unexpected results if you don't add parenthesis to the query.

You **MUST** use parentheses to explicitly indicate how you want the query to be executed:

q=((author:glantz AND type:letter) OR brand:kool)
OR
q=(author:glantz AND (type:letter OR brand:kool))