



Terms List API

Brought to you by IData, Inc

Table of contents

1	Introduction	3
2	Request Specification.....	3
2.1	URL	3
2.2	Parameters.....	3
2.3	Format	4
2.3.1	HTTP Query String Request Format.....	4
2.3.2	XML Request Format	6
3	Response Specification.....	7
3.1	RSS Response Format.....	7
3.2	HTML Response Format.....	9
3.3	XML Response Format.....	10
3.4	JSON Response Format.....	11

1 Introduction

The following API specification is a stable draft, but is subject to change as a result of ongoing extensions and enhancements. The terms list service returns all terms within an organization. It is the API equivalent of the 'Browse Terms' page in the Cookbook, and supports all the same options except for sorting. Without any filtering parameters, the service will return the latest approved versions of terms, or the latest version if a particular term has no approved version. An empty result set can occur in cases where the combination of filters does not produce any relevant terms.

2 Request Specification

The URL to which you submit a request tells us the institution whose terms you want to query. All Requests should be submitted over the Internet via the HTTPS protocol. Please be aware that certain versions of IE enforce a same scheme AJAX request, meaning that the page your Javascript resides in must also be in HTTPS. See this document for more details: <http://blogs.msdn.com/b/ieinternals/archive/2010/05/13/xdomainrequest-restrictions-limitations-and-workarounds.aspx>

URL

This service is provided at:

`https://<your_subdomain>.datacookbook.com/institution/terms/list`

Parameters

Authentication

- **pw** – password of the user who will be authenticating the request.
- **un** – username of user who will be authenticating the request.
- **OR** – include authentication token(s) retrieved using the service_login service as cookies in the request.

Required

- **requestType** – type of request. For term search, the request type is "list".
- **outputFormat** – Format you want the results to be returned as. Supported formats: "rss", "html" (basic, with simple classes on each element), "xml", or "json". Default is "xml".
- **jsonFunction** – Only with AJAX and JSONP. Name of function you want JSON to be passed to on return.

Optional Filters

If a supplied filter value does not exist within your organization, the service will return an empty result set. It will NOT respond with an error message such as “functional area not found”. Additionally, only one value can be provided for each filter.

- **functionalArea** – Name of the functional area to filter by.
- **status** – Equivalent to the ‘Versions’ filter in the terms browse page, this matches the state of the term version. An empty value will return the default set of latest approved term version, or latest version if none are approved. Possible values are:
 - <no value> - If you do not supply a value, the Cookbook will default to the latest approved term version, or latest version if none are approved for that term.
 - all – All terms all version.
 - latest – Latest term version regardless of state.
 - pending – Term versions in ‘pending’ state
 - in_draft – term version in ‘draft’ state
 - latest_approved – The latest approved version for terms with approved versions.
 - public_approved – Same as latest_approved, but also flagged as ‘public’
- **dataSystem** – Data System name to filter by.
- **tag** - tag name to filter by.
- **termSource** – Term Source name to filter by

Format

You can choose from one of two request formats: form input parameters or request XML.

2.3.1 HTTP Query String Request Format

If you choose to implement requests using form input parameters, a list request would be either an HTTPS GET request that has the parameters above in its query string, or an HTTPS POST request that has the parameters above stored as form inputs in the body of the request. You should only place each parameter in the request once. If you accidentally place a parameter there twice and the two instances have different values, we can't guarantee which will be used in processing your request.

NOTE: If typing a GET request directly into the browser's address bar, you should use a '+' to connect words with spaces. For example functionalArea=human+resources

Sample HTTP GET query string request:

https://idata.datacookbook.com/institution/terms/list?un=jonathanmorgan&pw=nopeeking!&requestType=list&outputFormat=json&functionalArea=human+resources&status=latest_approved

2.3.2 XML Request Format

The XML request contains the same information, but it is more structured, and is in an XML transaction dialect that is used by all of our XML-based APIs. The XML request in our dialect contains a list of request parameters and <Parameter> elements stored in a <ParameterList>. The required and optional parameters differ for each request type. Parameters are stored in a ParameterList element, one to a Parameter element. Each parameter contains a <Name> and <Value> element.

An XML request is sent as the body of an HTTPS request to the service you want to invoke.

Sample XML service transaction request:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceTransaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <ServiceRequest serviceName="list">
    <ParameterList>
      <Parameter>
        <Name>un</Name>
        <Value>jonathanmorgan</Value>
      </Parameter>
      <Parameter>
        <Name>pw</Name>
        <Value>nopeeking!</Value>
      </Parameter>
      <Parameter>
        <Name>requestType</Name>
        <Value>list</Value>
      </Parameter>
      <Parameter>
        <Name>outputFormat</Name>
        <Value>xml</Value>
      </Parameter>
      <Parameter>
        <Name>functionalArea</Name>
        <Value>Human Resources</Value>
      </Parameter>
      <Parameter>
        <Name>status</Name>
        <Value>latest_approved</Value>
      </Parameter>
    </ParameterList>
  </ServiceRequest>
</ServiceTransaction>
```

3 Response Specification

The terms list will be the body of the HTTPS response to the HTTPS request for term list. The response can be returned in a number of formats: RSS, HTML, XML, and JSON.

If you plan on accessing services using AJAX, then JSON or HTML will make more sense, though either RSS or XML would be implementable, as well.

Fields presented in all response formats

- **term name** – Name of the term.
- **id** – The version ID number.
- **term functional definition** – text explanation of the term. In the system, there are two definitions possible for each term, a functional and a technical definition. The functional definition is the text explanation of the term, in as close to plain English as possible. The technical definition is intended to contain specifics on retrieving a given term out of a database, or technical directions about interpreting it or displaying it. We don't return this information for most API requests.
- **term perma link URL** - This will be the URL of the Data Cookbook page where a user can go to see more data on the term (like the technical definitions, associated terms, tags applied to the term, where it fits in the hierarchy of terms, and the functional areas to which it belongs).

RSS Response Format

In an RSS response, an RSS document is the body of the HTTPS response, with each item containing a term that matches the submitted criteria. Our API implements RSS 2.0. For each item returned, the term name is the <title>, the term's functional description is the <description>, the last time that term was updated is stored in the <pubDate> element, and the URL to the data cookbook page for the term is in the <link> and <guid> elements.

Sample RSS service transaction response:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0">
<channel>
  <title>
    IData Data Cookbook Service Response: request type list
  </title>
  <description>
    IData Data Cookbook Service Response: request type list.
    Status Code: 0; Status Message: Success!
  </description>
  <link>
    http://www.idatainc.com/DataCookbook/RequestTypeList.aspx
  </link>
  <guid>
    http://www.idatainc.com/DataCookbook/RequestTypeList.aspx
  </guid>
</channel>
</rss>
```

```

</description>
<link>https://idata.datacookbook.com/institution/terms/list
</link>
<langage>en-us</langage>
<copyright>Copyright 2013, IData, Inc.</copyright>
<generator>IData Data Cookbook</generator>
<managingEditor>
bparish@idatainc.com (Brian Parish, President, IData, Inc.)
</managingEditor>
<webMaster>
kdezio@idatainc.com (Ken Dezio, CTO, IData Inc.)
</webMaster>
<docs>http://blogs.law.harvard.edu/tech/rss</docs>
<pubDate>
Thu, 10 Jan 2013 14:46:15 Eastern Standard Time
</pubDate>
<lastBuildDate>
Thu, 10 Jan 2013 14:46:15 Eastern Standard Time
</lastBuildDate>
<ttl>30</ttl>
<item>
  <title>Academic Program</title>
  <link>
https://idata.datacookbook.com/institution/terms/1098
</link>
  <description>
<![CDATA[
      A program of study that students seek to
      complete. The end result of completing an Academic
      Program may range from certificates to degrees. The
      programs are defined by the institution and described
      in the annually published catalog.
    ]]>
  </description>
  <pubDate>Wed, 28 Mar 2012 17:47:30 -0400</pubDate>
  <guid isPermaLink="true">
https://idata.datacookbook.com/institution/terms/1098
</guid>
</item>
<item>
  <title>Current Name</title>
  <link>
https://idata.datacookbook.com/institution/terms/214
</link>
  <description>
<![CDATA[

```



```

        A person's current full legal name
    ]]>
</description>
<pubDate>Tue, 09 Feb 2010 16:04:12 -0500</pubDate>
<guid isPermaLink="true">
https://idata.datacookbook.com/institution/terms/214
</guid>
</item>
</channel>
</rss>

```

HTML Response Format

In an HTML response, the list of matching terms is returned in a simple <div> structure in HTML, with classes and names assigned so you could target CSS to fit their appearance to your application.

Sample HTML service transaction response:

```

<div class="IDATA_services">

    <div class="IDATA_DC_TermList">

        <div class="IDATA_DC_Term" name="Academic Program"
id="5920">
        <div class="IDATA_DC_TermName">Academic Program</div>
        <div class="IDATA_DC_FunctionalDefinition">A program of
study that students seek to complete. The end result of
completing an Academic Program may range from certificates
to degrees. The programs are defined by the institution
and described in the annually published catalog.</div>
        <div class="IDATA_DC_URL"><a
href="http://idatau.datacookbook.local/institution/terms/10
98">http://idatau.datacookbook.local/institution/terms/1098
</a></div>
        </div>

        <div class="IDATA_DC_Term" name="Current Name" id="1445">
        <div class="IDATA_DC_TermName">Current Name</div>
        <div class="IDATA_DC_FunctionalDefinition">A person's
current full legal name</div>
        <div class="IDATA_DC_URL"><a
href="http://idatau.datacookbook.local/institution/terms/21
4">http://idatau.datacookbook.local/institution/terms/214</
a></div>
        </div>
    </div>

```

</div>

</div>

XML Response Format

An XML response contains the same term information as the other options, but also includes more detail than the other options, including response status and message and potentially a re-cap of the request information. If no matches are found, there will be either no <TermList> element or an empty <TermList> element in the response.

Sample XML service transaction response:

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiceTransaction xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
  <ServiceResponse serviceName="list">
    <ResponseStatus>
      <ResponseCode>0</ResponseCode>
      <ResponseMessage>Success!</ResponseMessage>
    </ResponseStatus>
    <TermList>
      <Term>
        <id type="integer">5920</id>
        <name>Academic Program</name>
        <perma-link-url>
          http://idatau.datacookbook.local/institution/terms/109
          8
        </perma-link-url>
        <term-functional-definition>
          A program of study that students seek to complete.
          The end result of completing an Academic Program may
          range from certificates to degrees. The programs are
          defined by the institution and described in the
          annually published catalog.Edits were made to funcdef
        </term-functional-definition>
      </Term>
      <Term>
        <id type="integer">1445</id>
        <name>Current Name</name>
        <perma-link-url>
          http://idatau.datacookbook.local/institution/terms/214
        </perma-link-url>
        <term-functional-definition>
          A person's current full legal name
```

```

        </term-functional-definition>
    </Term>
</TermList>
</ServiceResponse>
</ServiceTransaction>

```

JSON Response Format

JSON is generally used to support a Javascript AJAX implementations of API requests from within a browser. The cookbook supports the JSONP specification of providing a callback function in order to overcome browser same-origin restrictions. Your request parameters must include:

- **jsonFunction** – Name of function you want JSON to be passed to on return.

This callback function will receive the cookbook results formatted as JSON. The JSON response contains the same granularity and level of detail found in the XML response. The exact request implementation will vary depending on which (if any) javascript framework you are using.

Below is a sample JSONP response. The callback function the consumer would have to implement is named "processServiceResponse()".

NOTE: Certain versions of IE enforce a same scheme AJAX request, meaning that the page your Javascript resides in must also be in HTTPS. See this document for more details: <http://blogs.msdn.com/b/ieinternals/archive/2010/05/13/xdomainrequest-restrictions-limitations-and-workarounds.aspx>

Sample JSONP service transaction response:

```

processServiceResponse (
{
    "ServiceName": "list",
    "ResponseStatus": {
        "ResponseCode": 0,
        "ResponseMessage": "Success!"
    },
    "TermList": [
        {

```

```

"Term": {
    "name": "Academic Program",
    "id": 5920,
    "term_functional_definition": "A program of study
that students seek to complete. The end result
of completing an Academic Program may range from
certificates to degrees. The programs are
defined by the institution and described in the
annually published catalog.\r\n\r\nEdits were
made to funcdef",
    "perma_link_url":
"http://idatau.datacookbook.local/institution/terms/1098"
}

},
{
    "Term": {
        "name": "Address",
        "id": 16968,
        "term_functional_definition": "This represents an
address of an individual or an organization. It
is comprised of a street address, city, state or
province, zip code. It also may contain other
elements such as county or mail courier route.",
        "perma_link_url":
"http://idatau.datacookbook.local/institution/terms/7749"
    }
},
{
    "Term": {
        "name": "Current Name",
        "id": 1445,
        "term_functional_definition": "A person's current
full legal name",
        "perma_link_url":
"http://idatau.datacookbook.local/institution/terms/214"
    }
}

```

}
]
}
) ;