Created by Steven Braun, 6/8/2015
Last updated 7/14/2015

Using the Scopus APIs and Experts XML

Documentation

This documentation provides an overview of how to work with the Scopus APIs and Experts bulk export XML, including a selection of sample PHP scripts for doing so.

For full repository contents, see http://www.github.com/braunsg/scopus-api-scripts

CONTENTS

Overview
Scopus API documentation
Getting an API key
Scopus licensing
API types
Scripts
Using the scripts
Using the data
General workflow
Experts bulk export XML

Created by Steven Braun, 6/8/2015
Last updated 7/14/2015

OVERVIEW

This package includes a set of scripts that can be used to query the Scopus API (*application program interface*). Although these particular scripts are written in PHP (http://www.php.net/) with API calls executed via cURL, any other scripting language that enables interaction with a RESTful API would produce similar results. The following sections discuss the scripts themselves as well as the skills and materials that are assumed by the user before they can be utilized.

SCOPUS API DOCUMENTATION

Scopus provides comprehensive documentation for all of their available content APIs, including information about the types of data that can be retrieved and how to build query strings. This documentation is available at http://api.elsevier.com/documentation/apis.html.

GETTING AN API KEY

In order to use any of the Scopus APIs, you must first register for an API key. This can be done through the Elsevier Developers portal at http://dev.elsevier.com/.

Once you have registered for an API key, this key should be referenced in any script used to query the APIs; those locations are documented within the scripts themselves.

SCOPUS LICENSING

The use of Scopus, including the APIs, is dependent upon the terms and conditions of institutional licensing. Refer to your institutional contact to determine whether your use of the APIs falls within your granted credentials.

API TYPES

The following APIs are utilized in the scripts included in this package:

API name	**Script description and documentation**
content/search/scopus	Performs a query against Scopus publication content and retrieves data on matching publications *Documentation* http://api.elsevier.com/documentation/SCOPUSSearchAPI.wadl
content/search/author	Performs a query against authors profiled in Scopus and retrieves data on matching author records *Documentation* http://api.elsevier.com/documentation/AUTHORSearchAPI.wadl

SCOPUS VARIABLES: DEFINITIONS

The Scopus APIs link data between them using a set of variables with specific names, described below:

Variable name	**Variable description**
Scopus author ID	A unique, proprietary identifier that is assigned to all disambiguated authors profiled in Scopus; each ID points to a unique Scopus author profile at http://www.scopus.com/authid/detail.url?authorId=[Scopus author ID] **Note:** When distinct Scopus profiles are merged into one (such as when it is found that two profiles refer to the same individual), typically the associated Scopus author IDs referring to the previously unmerged profiles become aliased and automatically forward API calls to the Scopus author ID of the merged profile.
Scopus publication ID	A unique, proprietary identifier assigned to all unique publication records in Scopus, typically indicated in API returns by **SCOPUS_ID.** **Note:** These IDs can change or disappear over time.
Scopus publication eID	A unique *electronic* identifier that provides stable and persistent reference to a unique publication record in Scopus, typically derived from the Scopus publication ID

SCRIPTS

The following scripts are included in this package:

Script name	**Base API**	**Script description**
get-author-publications.php	content/search/scopus	Retrieves data about publications authored by a specified list of people, based on known Scopus author IDs
get-author-publications_mysql.php	content/search/scopus	Retrieves data about publications authored by a specified list of people, based on known Scopus author IDs, AND injects data into a MySQL database
get-publication-data.php	content/search/scopus	Retrieves data about a specified list of publications, based on known Scopus publication eIDs (electronic identifiers)
get-publication-data_mysql.php	content/search/scopus	Retrieves data about a specified list of publications, based on known Scopus publication eIDs (electronic identifiers), AND injects data into a MySQL database

search-for-author.php	content/search/author	Searches for and retrieves data about authors profiled in Scopus, based on name, affiliation, and/or other parameters
expand-experts-data.php	N/A	Expands data in the XML dump files provided by Elsevier's Experts platform and extracts Scopus author IDs for faculty of interest

In addition to these scripts, MySQL queries that generate the generic structure of tables to ingest data extracted via the API are included. These are located in the *table-definitions/* subdirectory and are described below.

Script name	**Script description**
publication_data.mssql	Creates a table structured to hold bibliographic data about publications, such as title, publication date, and citation count
faculty_publications.mssql	Creates a table structured to hold data about faculty/researcher authorships (*i.e.,* publications)
faculty_identifiers.mssql	Creates a table structured to hold data about researcher identifiers, *e.g.,* Scopus author IDs or ORCIDs

USING THE SCRIPTS

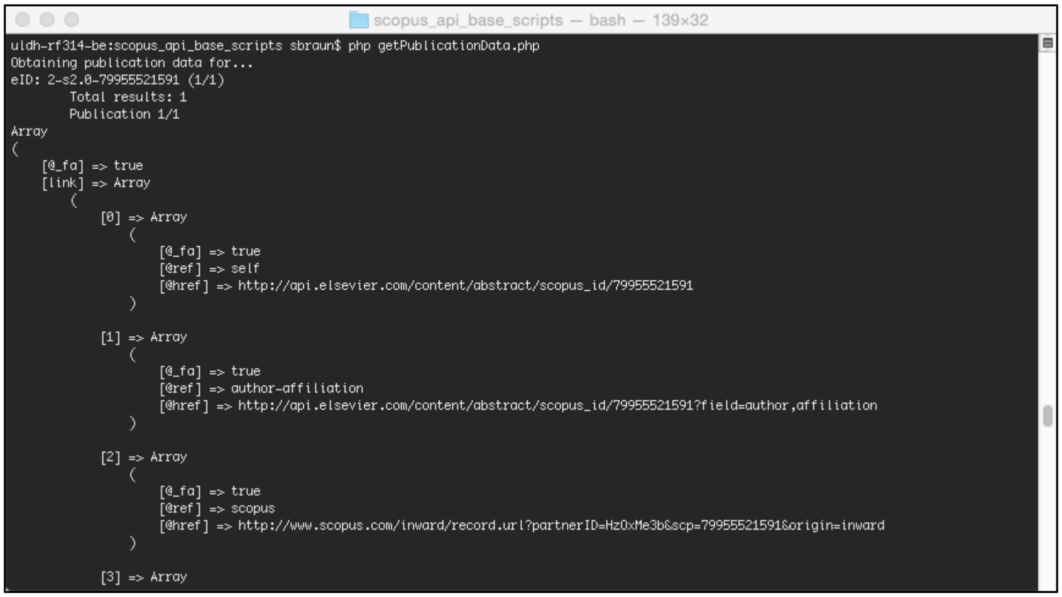
In the examples given, API queries are carried out through calls made via php cURL. Since these scripts are written in php, they may be run via any standard bash terminal.

For example, to run the script **get-publication-data.php** in terminal, navigate to the directory in which the file is located and execute the command

```
> php getPublicationData.php
```

USING THE DATA

The Scopus APIs can return data in different formats, including JSON or XML, depending on user specifications. Your choice of format will depend upon how you intend to use the data after they are retrieved. In the scripts included here, JSON is specified as the format of choice. The examples print out the retrieved API query results to the terminal screen; an example of what this looks like is shown below, using **get-publication-data.php** to retrieve data for the publication with known eID **2-s2.0-79955521591**.



```
uldh-rf314-be:scopus_api_base_scripts sbraun$ php getPublicationData.php
Obtaining publication data for...
eID: 2-s2.0-79955521591 (1/1)
        Total results: 1
        Publication 1/1
Array
(
    [@_fa] => true
    [link] => Array
        (
            [0] => Array
                (
                    [@_fa] => true
                    [@ref] => self
                    [@href] => http://api.elsevier.com/content/abstract/scopus_id/79955521591
                )

            [1] => Array
                (
                    [@_fa] => true
                    [@ref] => author-affiliation
                    [@href] => http://api.elsevier.com/content/abstract/scopus_id/79955521591?field=author,affiliation
                )

            [2] => Array
                (
                    [@_fa] => true
                    [@ref] => scopus
                    [@href] => http://www.scopus.com/inward/record.url?partnerID=HzOxMe3b&scp=79955521591&origin=inward
                )

            [3] => Array
```

One option for working with the data is to inject data into a database for storage and analysis. A popular choice for this is MySQL (https://www.mysql.com/), for which php has a reliable and powerful API (http://php.net/manual/en/book.mysqli.php) built in. For examples of this, please consult **get-publication-data_mysql.php** and **get-author-publications_mysql.php**.

GENERAL WORKFLOW

Collectively, the scripts in this package should allow the user to query the Scopus API for faculty, retrieve data about publications authored by those faculty, and inject the retrieved data into a database. Thus, a general workflow might be designed as follows:

1. **Define the faculty population.** This could be done by specifying faculty internet IDs in a CSV file, or they could be routinely extracted from an Experts XML dump. General biographic data about faculty may be stored in the table *faculty_data*.

2. **Determine Scopus author IDs for the faculty population.** These IDs may be stored in the table *faculty_identifiers* and may be routinely extracted from an Experts XML dump via the **expand-experts-data.php** script or specified in an external file.

3. **Query the Scopus API.** To pull all publications authored by faculty in the defined faculty publication and indexed in Scopus, query the API using Scopus author IDs (derived from *faculty_identifiers*) via the script **get-author-publications.php** or **get-author-publications_mysql.php**. In the case of the latter, publication data are injected into two database tables, one being *publication_data* and the other being *faculty_publications*. To retrieve data on individual publication records for which Scopus IDs are already known, use **get-publication-data.php** or **get-publication-data_mysql.php**.

The table definitions included in this package should provide a useful guide in structuring the data that are returned from the Scopus API.

Created by Steven Braun, 6/8/2015
Last updated 7/14/2015

EXPERTS BULK EXPORT XML

Elsevier's Experts platform (transitioning to Pure), which creates researcher profiles for faculty at subscribing academic institutions, generates a bulk export dump that structures all Experts data, including publications, in a single (large) XML file. Experts publication data comes directly from Scopus and has the added bonus of disambiguated authorship, meaning that for faculty profiled in Experts, their publication data is relatively clean. The Experts XML also provides a source for extracting Scopus author IDs for faculty.

The general structure of the XML is shown below. Refer to the official Experts documentation for more details about XML definitions.

```
<Experts>
        <Expert>
                <Authorships>
                        <Authorship />
                </Authorships>
                <Affiliations>
                        <Affiliation />
                </Affiliations>
        </Expert>
</Experts>
<Publications>
        <ScopusPublications>
                <Publication></Publication>
        </ScopusPublications>
</Publications>
<OrgaEntities>
        <OrgaEntity Name= “Organizations”>
                <OrgaUnits>
                        <OrgaUnit />
                </OrgaUnits>
        </OrgaEntity>
</OrgaEntities>
```

A sample script (**expand-experts-data.php**) is included that provides an example of one way to parse the XML file using PHP. The script walks through the XML file, pulls Scopus author IDs for relevant faculty, and then injects them into a MySQL database.