Data API Hands On Instructions for Data Search and Fetch

This tutorial helps you get started using HTRC **Data API** and HTRC **Solr Proxy** with our demo applications. The demo application sends a Solr query string to HTRC Solr Proxy to get a list of volume id, obtains an Oauth2 token from Oauth2 authentication server, and sends the list of volume id with the Oauth2 token to Data API to retrieve data. There are three demo applications which make three different RESTful calls to Data API, i.e., volume retrieve, page retrieve, and token count. We have a Python version and a Java version for the demo. ***For simplicity, we focus on the Python version.*** We also provide documents for you to explore the Java version if you are interested in.

1. *Download demo code and prepare*

1) Download the demo archive from http://bit.ly/15JIVcW (http://wiki.htrc.illinois.edu/pages/viewpageattachments.action?pageId=1769480). Depending on your programming language preference, choose either the Python or the Java version.

The Python demo runs in **Python 2.7.5**.

2) Unzip the archive.

2. *For Python version of the demo*

**Text in italic is the file name. Text in italic and bold is the variable name defined in the source code file.

The 3 Python scripts in the folder, *UnCampVolumeDemo.py, UnCampPageDemo.py* and *UnCampTokenCountDemo.py,* are applications interacting with Data API and Solr Proxy.

Script	What does it do?
UnCampVolumeDemo.py	Sends a search query to Solr proxy to obtain a list of volume ids, and then fetches volume data from Data API based on the volume ids.
UnCampPageDemo.py	Similar to *UnCampVolumeDemo.py* but fetches data on page level: it sends a search query to Solr proxy to obtain a list of volume ids, and for each volume it fetches page-level data for user designated pages (pages 1, 5, 10 in this example) using Data API.
UnCampTokenCountDemo.py	First sends a search query to Solr proxy to obtain volume ids, and then gets word frequency count for each volume

The uncamputils.py script is *not* a demo script. It includes service end point constants and helper methods which you do not need to change.

1) **Run** *UnCampVolumeDemo.py*

The command is:

```
> python UnCampVolumeDemo.py volume.zip
```

Parameters

Change these parameter values to fit your need.

- Data API parameter.
  In line 23: VOLUME_PARAMETERS = {*'mets'* : *'true'*}
  This allows Data API to return mets record along with the volume content.
- Data API parameter.
  In line 24: VOLUME_PARAMETERS = {*'mets'* : *'true'*, *'concat'* : *'true'*}
  This allows Data API to return mets record along with the volume content and concatenate all the pages into one single text file per volume.
- Solr proxy parameter.
  In line 27: SOLR_METADATA_REQUEST = {*'q'* : *'title:peace AND author:Bill'*}
  This changes the query string sent to Solr Proxy. You will get different volume id for this query string.
- Solr proxy parameter
  In line 28: SOLR_METADATA_REQUEST = {*'q'* : *'publishDate:1884 AND author:Dickens', 'start'* : *'0'*, *'rows'* : *'1'*}
  This shows you how to control the number of results returned from Solr Proxy. It means only returning 1 result for this query.

Output

The script generates a .zip file named “volume.zip”. The zip file has multiple folders. Each folder uses a volume id as its name and contains all the page contents as separated files. Below is an output example with default parameters:

```
loc.ark+=13960=t2s47863z                <= volume id
        | 00000001.txt                  <= page
        | 00000002.txt
        |…
uc2.ark+=13960=t9w094g4m
        | 00000001.txt
        | 00000002.txt
        |…
```

2) **Run** *UnCampPageDemo.py*

The command is:

```
> python UnCampPageDemo.py page.zip
```

Parameters

- Data API parameter. It shares the same setting with as VOLUME_PARAMETERS in the above *UnCampVolumeDemo.py* script.
  In line 23: PAGE_PARAMETERS = {*'mets'* : *'true'*}
  This allows Data API to return mets record along with the volume content.
- Data API parameter. It shares the same setting with as VOLUME_PARAMETERS in the above *UnCampVolumeDemo.py* script. Note that *'mets'* and *'concat'* cannot be both set to true.
  In line 24: PAGE_PARAMETERS = {*'mets'* : *false*, *'concat'* : *'true'*}
  This allows Data API to return mets record along with the volume content and concatenate all the pages into one single text file per volume.
- Solr Proxy parameter.
  In lines 28, 29. It shares the same setting as the above *UnCampVolumeDemo.py* script.
- Page id. In line 129 (hardcoded). Input a list of page ids of which the content you want to return.

Output

The output is a .zip file named "page.zip", with a similar folder structure as the above *UnCampVolumeDemo.py* script.

3) **Run** *UnCampTokenCountDemo.py*

The command is:

```
> python UnCampTokenCountDemo.py tokencount.zip
```

Parameters

- Solr proxy parameter. It uses the same setting as the previous 2 scripts. In lines 27, 28.
- Token count parameter.
  In lines 22, 23.
  TOKENCOUNT_PARAMETERS = {*'level'* : *'volume'*, *'sortBy'* : *'count'*, *'sortOrder'* : *'asc'*}
  This allows Data API to return token count in volume level and sort the result by count in ascending order.

Output

The script generates a .zip file called "tokencount.zip", each text file inside which contains word frequency counts for a volume. The text files are named by volume ids.

3. *For Java version of the demo*

1) The zip file includes compiled jar file. There is a README.txt file under the unzipped folder show you how to run different demos through scripts provided.
2) The Solr query string can be modified in the scripts. Any other parameters changes of Data API or Solr Proxy require modifications of the code.
3) The zip file also includes a build.xml file which you can use Apache Ant to recompile the source code by typing "ant jar".
4) Under the src/htrc/uncamp folder, there are UnCampVolumeDemo.java, UnCampPageDemo.java, and UnCampTokenCountDemo.java files which you should make your changes to. HTRCUtils.java provides helper methods which you don't need to change. Look for the methods ***getVolumeIdFromSolrProxy()*** and ***getDataFromDataAPI()*** which interacts with Solr Proxy and Data API respectively.

For detailed information about parameters usage in Data API, please refer to HTRC Data API User Guide
http://wiki.htrc.illinois.edu/display/COM/HTRC+Data+API+Users+Guide

For detailed information about parameters usage in Solr Proxy, please refer to
http://wiki.apache.org/solr/CommonQueryParameters
http://lucene.apache.org/core/3_6_0/queryparsersyntax.html