Collection Space Stuff

REST API

intro

How CSpace works,

Use HTTP requests to connect with the server and PUT, DELETE, GET, etc those pages

URLs should follow this format: **https://<your-cspace-domain>/cspace-services/<authorityType>/<authorityRecordCSID>/items?pgSz=<pageSize>**

Base URL/CSpace Domain

The base URL connects to an individual CSpace instance. Our base URL is,

https://ucbg.collectionspace.org/cspace − services

Authority/Object Types

Then, in addition to the base URL, make a request from a specific source e.g.

orgauthorities

The following is a complete(?) list of authority types (source):

citationauthorities, conceptauthorities, locationauthorities, orgauthorities, personauthorities, placeauthorities, taxonomyauthority, workauthorities

Append the authority type to the end of the base URL e.g.

*https://ucbg.collectionspace.org/cspace − services/**orgauthorities/***

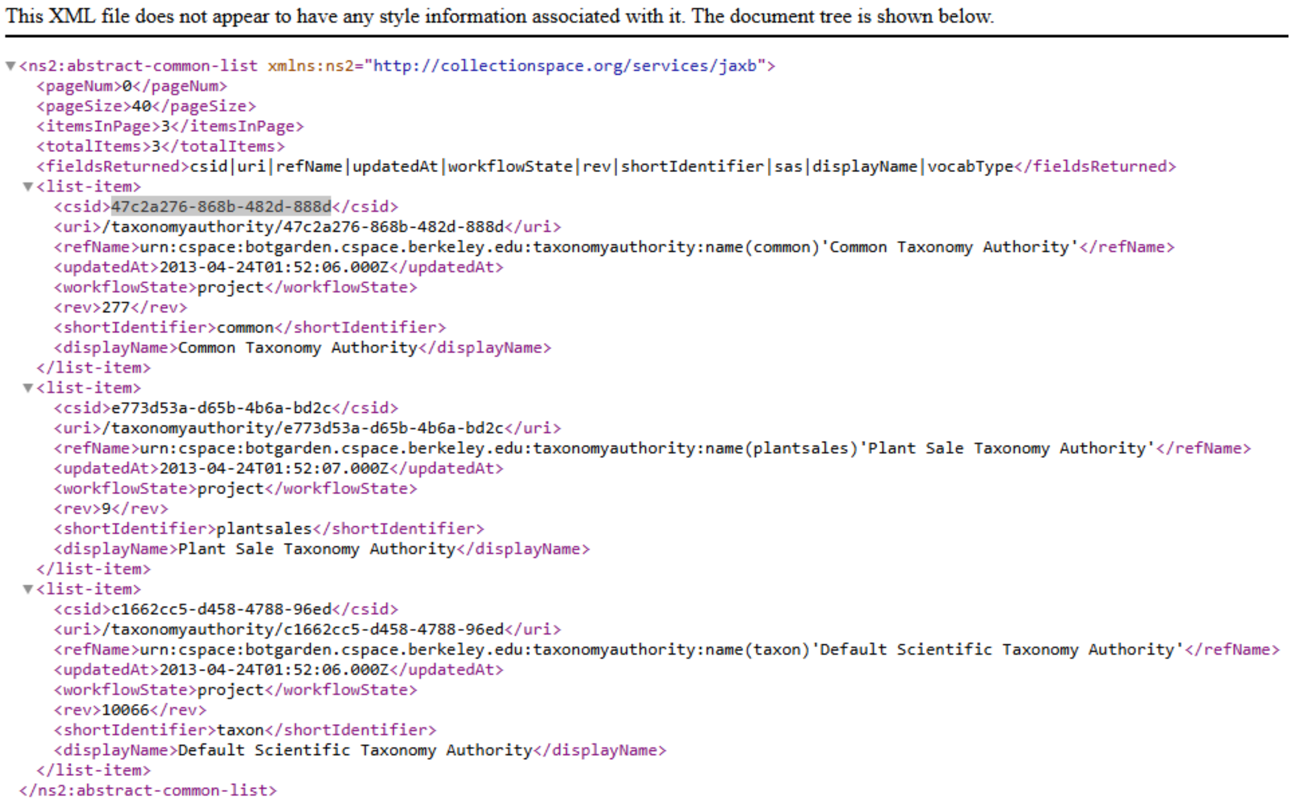
***In addition to accessing a list of objects through Authority Types, you can also access individual access types by finding the individual object CSID in the authority list. Finding the CSID of an authority type is outlined below

This is the url template you should follow to access individual objects

https://ucbg.collectionspace.org/cspace − services/collectionobjects/< *CSID*ofTheObject >

CSID/UUID

Then, add the CSID, a unique identifier assigned to every assigned to every authority record in CSpace. If you type the above URL into your browser, you will be greeted with this page. The CSID can be found between the <csid> markers. In the screenshot below, I've highlighted where the CSID can be found.



This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<ns2:abstract-common-list xmlns:ns2="http://collectionspace.org/services/jaxb">
   <pageNum>0</pageNum>
   <pageSize>40</pageSize>
   <itemsInPage>3</itemsInPage>
   <totalItems>3</totalItems>
   <fieldsReturned>csid|uri|refName|updatedAt|workflowState|rev|shortIdentifier|sas|displayName|vocabType</fieldsReturned>
 ▼<list-item>
     <csid>47c2a276-868b-482d-888d</csid>
     <uri>/taxonomyauthority/47c2a276-868b-482d-888d</uri>
     <refName>urn:cspace:botgarden.cspace.berkeley.edu:taxonomyauthority:name(common)'Common Taxonomy Authority'</refName>
     <updatedAt>2013-04-24T01:52:06.000Z</updatedAt>
     <workflowState>project</workflowState>
     <rev>277</rev>
     <shortIdentifier>common</shortIdentifier>
     <displayName>Common Taxonomy Authority</displayName>
   </list-item>
 ▼<list-item>
     <csid>e773d53a-d65b-4b6a-bd2c</csid>
     <uri>/taxonomyauthority/e773d53a-d65b-4b6a-bd2c</uri>
     <refName>urn:cspace:botgarden.cspace.berkeley.edu:taxonomyauthority:name(plantsales)'Plant Sale Taxonomy Authority'</refName>
     <updatedAt>2013-04-24T01:52:07.000Z</updatedAt>
     <workflowState>project</workflowState>
     <rev>9</rev>
     <shortIdentifier>plantsales</shortIdentifier>
     <displayName>Plant Sale Taxonomy Authority</displayName>
   </list-item>
 ▼<list-item>
     <csid>c1662cc5-d458-4788-96ed</csid>
     <uri>/taxonomyauthority/c1662cc5-d458-4788-96ed</uri>
     <refName>urn:cspace:botgarden.cspace.berkeley.edu:taxonomyauthority:name(taxon)'Default Scientific Taxonomy Authority'</refName>
     <updatedAt>2013-04-24T01:52:06.000Z</updatedAt>
     <workflowState>project</workflowState>
     <rev>10066</rev>
     <shortIdentifier>taxon</shortIdentifier>
     <displayName>Default Scientific Taxonomy Authority</displayName>
   </list-item>
 </ns2:abstract-common-list>
```

If you have access to a python interpreter, you can use this code to find the CSID. Simply replace the url with the url of interest and fill in your username and password.

import requests

from requests.auth import HTTPBasicAuth

import re

url = "https://ucbg.collectionspace.org/cspace-services/taxonomyauthority"

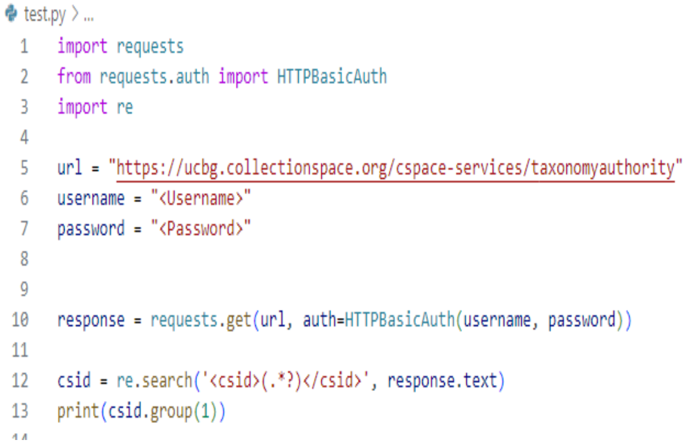
username = "<Your prod username>"

password = "<Your prod password>"

response = requests.get(url, auth=HTTPBasicAuth(username, password))

csid = re.search('<csid>(.*?)</csid>', response.text)

print(csid.group(1))



```python
import requests
from requests.auth import HTTPBasicAuth
import re

url = "https://ucbg.collectionspace.org/cspace-services/taxonomyauthority"
username = "<Username>"
password = "<Password>"


response = requests.get(url, auth=HTTPBasicAuth(username, password))

csid = re.search('<csid>(.*?)</csid>', response.text)
print(csid.group(1))
```

Items/Page Size/Page Number

Finally, append the items and page size variable to your url

$$items?\,pgSz =< number\ of\ entries\ to\ display > \&\text{pgNum} =< \text{page Number} >$$

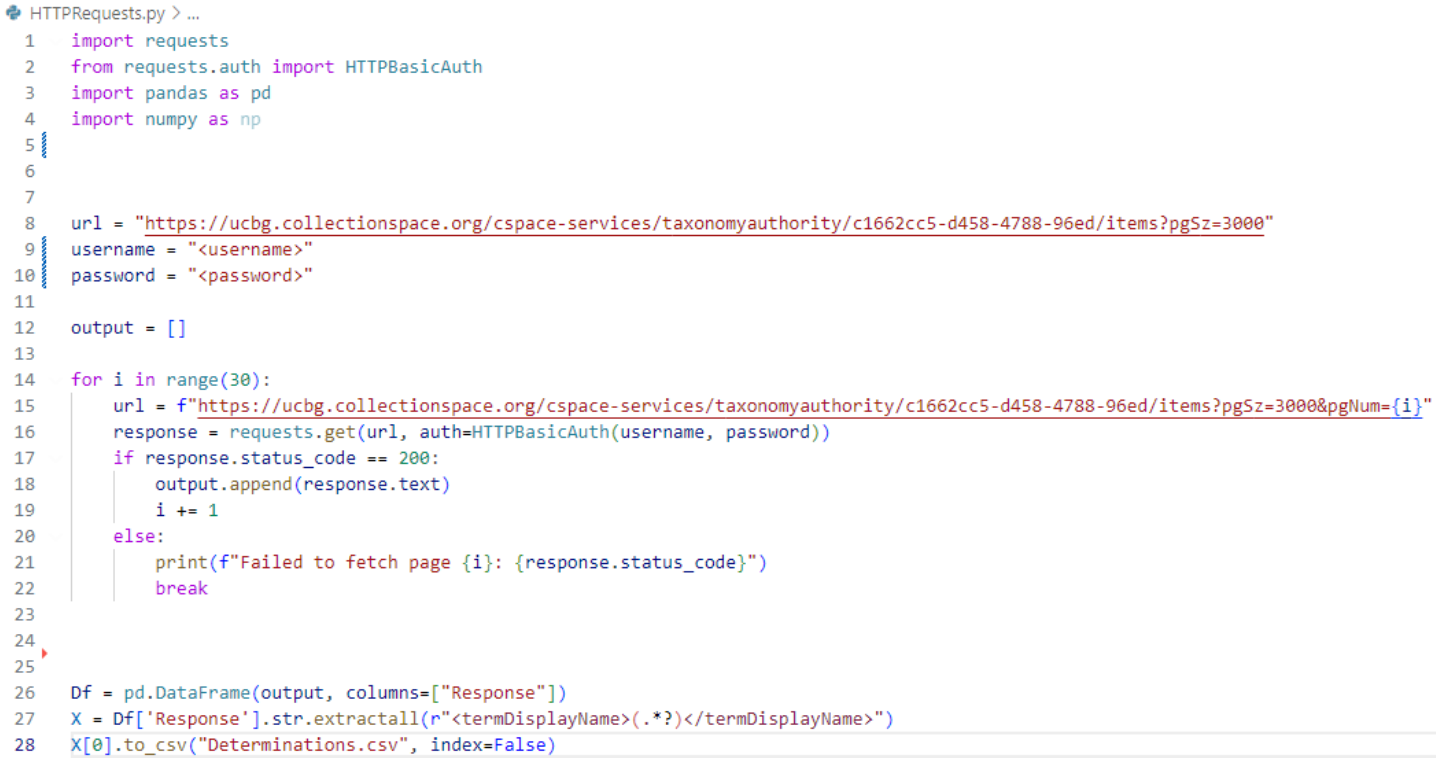
Your final URL should look like this

$$https://ucbg.collectionspace.org/cspace - services/taxonomyauthority/47c2a276 - 868b - 482d - 888d/items?\,pgSz = 100\&\text{pgNum} = 0$$

Doing Stuff

Example

The following is an example of a python program which utilizes URL construction to return all the determinations in the database



```python
import requests
from requests.auth import HTTPBasicAuth
import pandas as pd
import numpy as np



url = "https://ucbg.collectionspace.org/cspace-services/taxonomyauthority/c1662cc5-d458-4788-96ed/items?pgSz=3000"
username = "<username>"
password = "<password>"

output = []

for i in range(30):
    url = f"https://ucbg.collectionspace.org/cspace-services/taxonomyauthority/c1662cc5-d458-4788-96ed/items?pgSz=3000&pgNum={i}"
    response = requests.get(url, auth=HTTPBasicAuth(username, password))
    if response.status_code == 200:
        output.append(response.text)
        i += 1
    else:
        print(f"Failed to fetch page {i}: {response.status_code}")
        break



Df = pd.DataFrame(output, columns=["Response"])
X = Df['Response'].str.extractall(r"<termDisplayName>(.*?)</termDisplayName>")
X[0].to_csv("Determinations.csv", index=False)
```