

Publishing To ArcGIS Online via Notebooks

Written By: [MATINGO KUMBIRAI N \(https://www.linkedin.com/in/kumbirai-matingo-6910a617a/\)](https://www.linkedin.com/in/kumbirai-matingo-6910a617a/)

In this step by step *ArcGIS API for Python* tutorial we are going to publish an item from a CSV file within our machine to our ArcGIS Online Portal.

Here is a list of steps we are going to take to achieve great results in their order:

1. Login to our account [ArcGIS Online Account \(https://arcgis.com\)](https://arcgis.com)
2. Use **Pandas** to read & visualize our CSV data
3. Upload our CSV file
4. Turn CSV file into a **Feature Layer**
5. Finally, Add our new Feature Layer to a map

All these steps I have outline are going to be carried out within this same notebook. Make sure to understand the *syntax* that is required for every operation that we are going to be performing within this tutorial.

Note: You can also use the ArcGIS Pro embedded Notebooks, the API is already pre-installed on installation of the software which automatically makes it easy to perform Notebook operations for Data Science

In case you weren't here during the first tutorial, [here \(https://africansurveyors.net/introduction-to-arcgis-api-for-python/\)](https://africansurveyors.net/introduction-to-arcgis-api-for-python/) is a resource which you can follow for installation of the ArcGIS API for Python within your local machine.

You can also watch it on *YouTube*



(<https://www.youtube.com/watch?v=572QvhSUhmM>)

LET'S GET STRAIGHT TO IT

IMPORTING LIBRARIES

Just like what we mentioned in the Introductory guide, every operation within Jupyter Notebooks or that used Python Language as the basis for the operations requires some additional libraries to quicken the processes and automate these workflows.

Since we already downloaded the *Python API*, all we need to do is just call that into the Notebook and keep it ready for use within our space later down the road.

So right below, we are going to *import* the following Python libraries

- GIS
- Pandas and
- getpass

I already mentioned what the **GIS** library and the **Pandas** library does for us so I am not going to repeat that. If you have forgotten kindly refer back to the previous tutorials that were published.

getpass - The `getpass()` function prints a prompt then reads input from the user until they press return. The input is passed back as a string to the caller. [Source \(https://pymotw.com/2/getpass/\)](https://pymotw.com/2/getpass/).

So basically I need the **getpass()** function to be able to store my password for my ArcGIS Online account and send that as an authentication request. The reason why I called this function I don't want anyone reading this to be able to see my password while I record the video tutorial or while I print this document. Its also a great feature for security purposes which keeps all sensitive information secure.

So let's import these libraries into our Notebook.

In [1]:

```
from arcgis.gis import GIS
from getpass import getpass
import pandas as pd # Remember you can always shorten the library name for referal with
in the notebook.
```

Hint: Always remember to call all the libraries that you are going to be using right at the top. This should be the first thing you need to do before anything else

Authentication (Login In) to the ArcGIS Online Account

For us to get permission (be allowed) to interact with the platform, we need to be authenticated to perform the operations hence that requires that we are logged in.

So, I've created a `password` variable which is going to store the input from the `getpass()` function.

The variable `gis` is the one which is going to be used to authenticate the user based on the following parameters:

- the site URL in this case which is the URL for the ArcGIS Online Platform
- the `username` of the person and
- the `password` of the user

All this is done via the `GIS()` method and the parameters to be passed are the one's I have just mentioned above.

When you run the Cell, a small input box will appear below asking for your password

In [2]:

```
password = getpass()
gis = GIS("http://www.arcgis.com", "africansurveyors", password)
```

.....

Now that we have been authenticated, we need to move on to the next step.

Loading our Data (csv)

It's time for us to use the **Pandas** library to view, read and import our data.

- declared a variable in which you are going to store the path to the CSV file which you want to upload.
`csv`
- call the **pandas** `read_csv` function to read our data and
- the `head()` function to print out the first rows within the data that we have specified as a parameter within the function. *Remember: If the function is blank, `head()` function will print the first 6 rows as the default*

Our data from the CSV file is then printed as output within our Notebooks.

In [3]:

```
csv = r"C:\Users\Surveyor Jr\Desktop\African Surveyors Academy\ArcGIS API for Python\data\zwe_districts.csv"
districts = pd.read_csv(csv)
districts.head(20)
```

Out[3]:

	ID_0	ISO	NAME_0	ID_1	NAME_1	ID_2	NAME_2	TYPE_2	ENGTYPE_2	NL_
0	256	ZWE	Zimbabwe	1	Bulawayo	1	Bulawayo	District	District	
1	256	ZWE	Zimbabwe	2	Harare	2	Harare	District	District	
2	256	ZWE	Zimbabwe	3	Manicaland	3	Buhera	District	District	
3	256	ZWE	Zimbabwe	3	Manicaland	4	Chimanimani	District	District	
4	256	ZWE	Zimbabwe	3	Manicaland	5	Chipinge	District	District	
5	256	ZWE	Zimbabwe	3	Manicaland	6	Makoni	District	District	
6	256	ZWE	Zimbabwe	3	Manicaland	7	Mutare	District	District	
7	256	ZWE	Zimbabwe	3	Manicaland	8	Mutasa	District	District	
8	256	ZWE	Zimbabwe	3	Manicaland	9	Nyanga	District	District	
9	256	ZWE	Zimbabwe	4	Mashonaland Central	10	Bindura	District	District	
10	256	ZWE	Zimbabwe	4	Mashonaland Central	11	Centenary	District	District	
11	256	ZWE	Zimbabwe	4	Mashonaland Central	12	Guruve	District	District	
12	256	ZWE	Zimbabwe	4	Mashonaland Central	13	Mazowe	District	District	
13	256	ZWE	Zimbabwe	4	Mashonaland Central	14	Mount Darwin	District	District	
14	256	ZWE	Zimbabwe	4	Mashonaland Central	15	Rushinga	District	District	
15	256	ZWE	Zimbabwe	4	Mashonaland Central	16	Shamva	District	District	
16	256	ZWE	Zimbabwe	5	Mashonaland East	17	Chikomba	District	District	
17	256	ZWE	Zimbabwe	5	Mashonaland East	18	Goromonzi	District	District	
18	256	ZWE	Zimbabwe	5	Mashonaland East	19	Marondera	District	District	
19	256	ZWE	Zimbabwe	5	Mashonaland East	20	Mudzi	District	District	

Columns with no value in them will be displayed as **NaN**

We just want to see how many rows and how many columns are contained within our dataset. To do that we just call in `shape` after our variable and our information is displayed below in the format: (rows, columns)

In [4]:

```
districts.shape
```

Out[4]:

```
(60, 13)
```

Publishing An Item

We have seen what our data looks like and we are sure its the right dataset that we want to use. For some, if you would like to make changes to the data before publishing you could also do that again. (We will talk about how you can edit data using Pandas in another tutorial)

Now we need to publish an item to the ArcGIS Online platform based on our CSV file which is still within our local machine.

To do that, we need to declare a variable here which we just named `item_csv` which is going to store some property information for us. We are just going to put *title* as the only property for now.

We then have to call again another variable and this time we will name it `csv_item` in which we will call the **gis** library and give it a command that we need to **add** some **content** with these properties (`item_csv`) using this data (`csv`).

Remember that we already declared variables which contain the paths and data that we need to upload so we just call the variables instead of typing a huge piece of code which is just executed once

Lets call that variable `csv_item` to show us the output of our newly uploaded item within our ArcGIS Online platform.

In [5]:

```
item_csv = {'title': "ZWE Districts"}  
csv_item = gis.content.add(item_properties=item_csv, data=csv)  
csv_item
```

Out[5]:



ZWE Districts

(<http://africansurveyors.maps.arcgis.com/home/item.html?id=14a53c6f5ec845c7872327df2b4afb15>)



CSV by africansurveyors
Last Modified: March 04, 2021
0 comments, 0 views

(<http://africansurveyors.maps.arcgis.com/home/item.html?id=14a53c6f5ec845c7872327df2b4afb15>)

Create a Feature Layer

Since we have already uploaded our data as an item, it only takes two lines of code to create a **Feature Layer** from that item.

- we create a variable which contains the name of our item and the **publish** function which has NO parameters at the moment
- we just call our variable for us to verify that the item is now a **Feature Layer**

WE HAVE A FEATURE LAYER. GREAT!

*Notice the Difference in MetaData between the **Out[5]** and **Out[6]** cell blocks*

In [6]:

```
districts_item = csv_item.publish()  
districts_item
```

Out[6]:



ZWE Districts

(<http://africansurveyors.maps.arcgis.com/home/item.html?id=3b35d36b622f4a82936af18d36de1fcd>).



Feature Layer Collection by africansurveyors

Last Modified: March 04, 2021

0 comments, 0 views

(<http://africansurveyors.maps.arcgis.com/home/item.html?id=3b35d36b622f4a82936af18d36de1fcd>).

In this step, we just want to verify that we really uploaded this feature layer to the ArcGIS Online platform. Let's just run a basic search query.

In case you are lost - refer to the previous tutorial in the video at the beginning to understand more about searching with the Python API

OK. So our Feature Layer has been published we just confirmed it. If yours didn't publish, try checking that all the cells above have been ran without any errors and check especially for any typos that you might have made.

In [7]:

```
districts_lyr = gis.content.search(query="Districts*", item_type="Feature Layer")[0]
districts_lyr
```

Out[7]:

**ZWE Districts**

(<http://africansurveyors.maps.arcgis.com/home/item.html?id=3b35d36b622f4a82936af18d36de1fcd>)



Feature Layer Collection by africansurveyors

Last Modified: March 04, 2021

0 comments, 0 views

(<http://africansurveyors.maps.arcgis.com/home/item.html?id=3b35d36b622f4a82936af18d36de1fcd>)

Add Layer to Map

We're getting ready to add our newly added layer to the map. Before we do that I am just going to show you one important thing that has to do with Feature Layers.

All feature layers within the ArcGIS Platform have an *Index*. This is just the number contained at the end of the URL for the whole feature layer. This Index helps us identify the dataset which we want to use.

*Imagine if we had published a **Geodatabase File** with multiple layers all combined and attached to one URL. We would need to know which is which because we could have line data and point data within the Geodatabase file and they have the same URL. The only thing to separate them is the Index at the end of the URL after the trailing slash (/).*

These Indexes start at Zero(0). So if we have just one feature within a layer, that automatically means that its index is Zero. So I am just going to use that here and so that you know how Layer Indexes work.

In [8]:

```
districts_lyr = districts_lyr.layers[0]
```

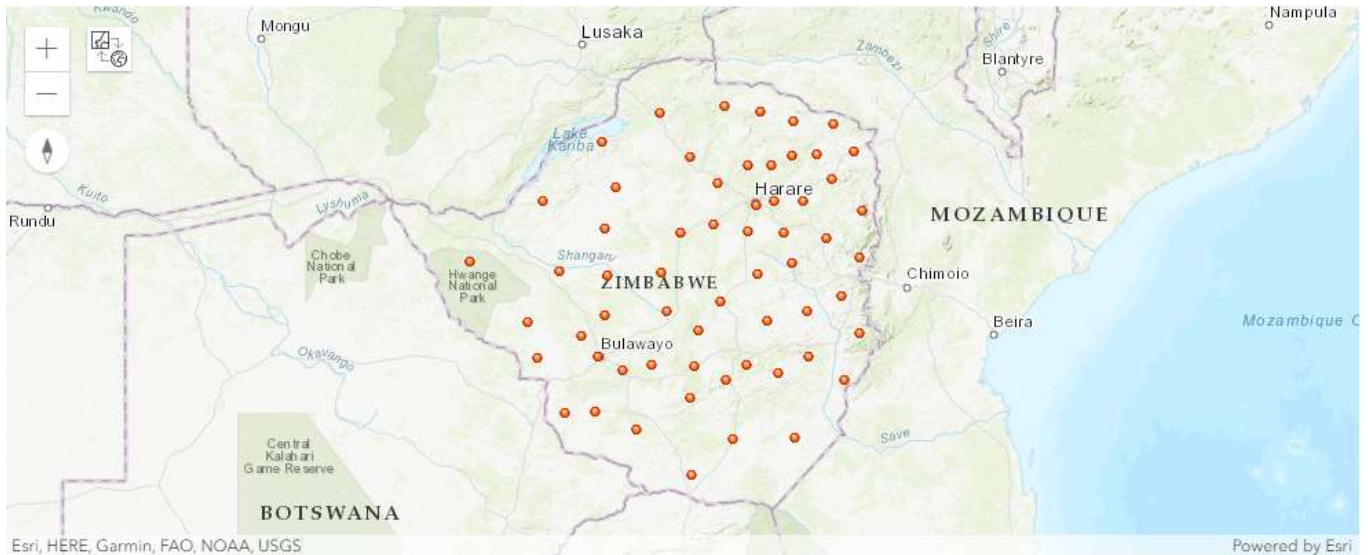
Call the Map with Location

We have called a map for use to have a basemap in order to easily visualize the data which we are about to add.

Remember from our last tutorial, we discussed on how we can add a basemap to these maps. In my case, I am just going to leave that as default.

In [9]:

```
map = gis.map("Zimbabwe", 6)
map
```

Finally, our last step is to load the layer onto the map by simply calling the `add_layer()` function like what we did previous in the last tutorial again.

The variable in which we selected an index for the layer is the one which is going to be our parameters for this function.

Wait a few seconds, depending on your network connection and your Feature Layer will be plotted onto your map.

In [10]:

```
map.add_layer(districts_lyr)
```

If you didn't face any errors along the way and managed to get the data onto the map then;

CONGRATULATIONS



You've managed to complete your first step in **#GISAutomation**

For anyone having trouble or fails to understand this tutorial, I am reachable via [LinkedIn](https://www.linkedin.com/in/kumbirai-matingo-6910a617a/) (<https://www.linkedin.com/in/kumbirai-matingo-6910a617a/>). Just send me a direct message and I will be sure to respond to any questions relating to the tutorials that you might have.

About Author



- 3rd Year BSc Hons in Surveying & Geomatics
- Interested in GIS for Health and Land Administration, Spatial Data Science and Programming
- currently working on a book titled: **GIS Step by Step: A Practical Guide to GIS**