

# Cloning Apps and Items with ArcGIS API for Python

John Yaist

Raul Jimenez Ortega



# Topics

- **Cloning Content Scenario**
- **Demo – Cloning Hosted Feature Layers and Webmaps**
- **Discussion – clone\_items() function and key parameters**
- **Demo – Cloning Dashboards and Web Mapping Applications**



# Staging



# Production



# Demo:

Cloning Feature Layers and Webmaps





# Accessing the function

Content Manager



# ContentManager

- Accessed from the *content* property of a *GIS* object

```
gis = GIS(profile="your_ent_admin_profile")  
content_manager = gis.content
```

```
type(content_manager)
```

```
arcgis.gis.ContentManager
```

- *clone\_items()* is a method on the ContentManager

```
clone_result = gis.content.clone_items(items=[item_list]  
                                       search_existing_items=True)
```

[https://developers.arcgis.com/python/api-reference/arcgis.gis.toc.html#arcgis.gis.ContentManager.clone\\_items](https://developers.arcgis.com/python/api-reference/arcgis.gis.toc.html#arcgis.gis.ContentManager.clone_items)

# clone\_items()

```
clone_items(items, folder=None, item_extent=None, use_org_basemap=False, copy_data=True,  
search_existing_items=True, item_mapping=None, group_mapping=None, owner=None)
```

Clone content to the GIS by creating new items.

Cloning an item will create a copy of the item and for certain item types a copy of the item dependencies in the GIS.

For example a web application created using Web AppBuilder or a Configurable App Template which is built from a web map that references one or more hosted feature layers. This function will clone all of these items to the GIS and swizzle the paths in the web map and web application to point to the new layers.

This creates an exact copy of the application, map, and layers in the GIS.

## Key Parameters

- *search\_existing\_items*  
default value = True
- *item\_mapping*  
optional dictionary





## *search\_existing\_items*

```
clone_result = gis.content.clone_items(items=[item_list]  
                                       search_existing_items=True)|
```

- uses the item *typeKeyword* property to label result
  - cloned items have a typeKeyword formatted like below added:  
*source-<itemId>*
  - example: *Feature Layer* item id e9541fd18bdc439dbf84ef7e929475ac.  
*cloned item typeKeyword: source-e9541fd18bdc439dbf84ef7e929475ac*

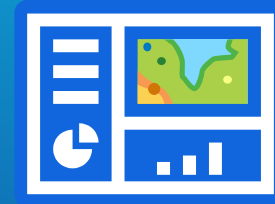
## *item\_mapping*

```
clone_result = gis.content.clone_items(items=[item_list]  
                                       item_mapping={source.id:target.id})
```

- uses an existing item in the *target* GIS
  - useful for items comprised of items:
    - Web Mapping Applications
    - ArcGIS Dashboards
- *clone\_items()*
  - does not automatically clone all items comprising more complex item types

# Demo:

Cloning ArcGIS Dashboards and Web Mapping Applications





# Conclusion

- *search\_existing\_items=True* – prevents duplication of previously cloned items
- Webmaps – Feature Layers are cloned by default
- Feature Layers – data is cloned by default
- cloning Web Mapping Applications and Dashboards
  - use *item\_mapping* to control which items serve particular roles



# ArcGIS API for Python Cloning Content Guide

- <https://developers.arcgis.com/python/guide/cloning-content/>





**Please share your feedback via the survey  
available at the link below the presentation.**



# Group Export and Import APIs

Available on **ArcGIS Enterprise >= 10.8.1** (July 2020)

**ArcGIS REST API**

Documentation / REST API / Users, groups, and items

## Export Group Content

URL: `https://<group-url>/export(POST only)`  
Required Capability: Administrator | Group Owner  
Version Introduced: 10.8.1

**Description**

Note: This operation is only supported in ArcGIS Enterprise.

This operation allows you to extract all supported items, or set of supported items, from a

**In this topic**

- Description
- Supported items
- Additional considerations
- Request parameters
- Example usage
- Accessing results
- JSON Response example

We'd love to hear your feedback

Was this page helpful? ☐ Yes ☐ No

URL: `https://<group-url>/export(POST only)`  
Required Capability: Administrator | Group Owner  
Version Introduced: 10.8.1

**ArcGIS REST API**

Documentation / REST API / Users, groups, and items

## Import Group Content

URL: `https://<group-url>/import(POST only)`  
Required Capability: Administrator | Group Owner  
Version Introduced: 10.8.1

**Description**

Note: This operation is only supported in ArcGIS Enterprise.

This operation imports supported content items from an export package (generated using the

**In this topic**

- Description
- Request parameters
- Upload .epk item to receiving environment
- Example usage
- Accessing results
- JSON Response examples

We'd love to hear your feedback

Was this page helpful? ☐ Yes ☐ No

URL: `https://<group-url>/import(POST only)`  
Required Capability: Administrator | Group Owner  
Version Introduced: 10.8.1

*Simplify the flow of content migration **between groups.***

# Group Export and Import

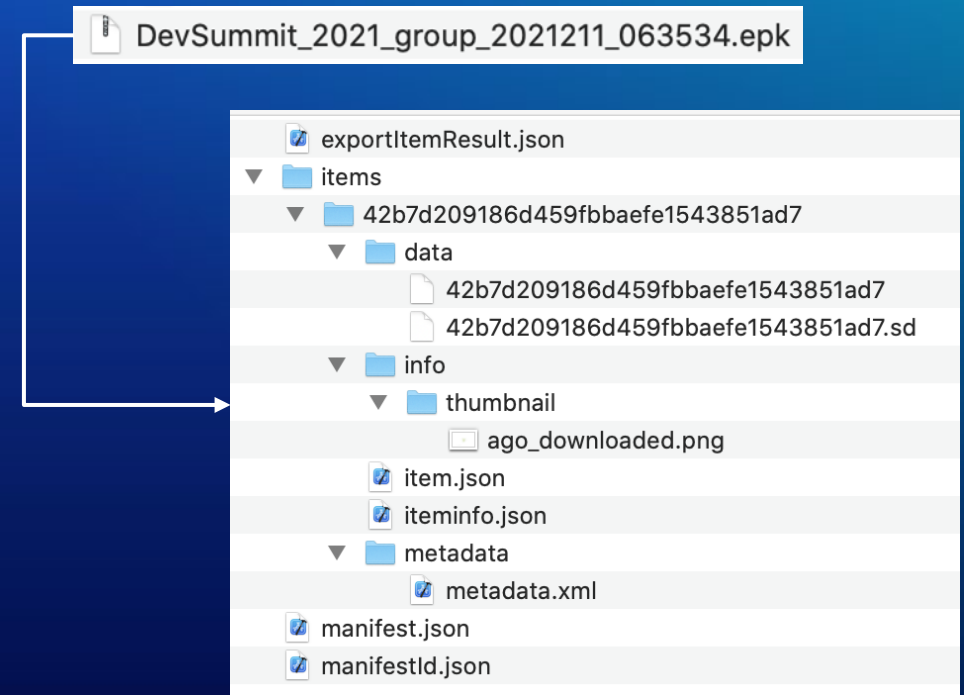
How the operations work

## Export:

- **Generate an export package (.epk)** containing all supported items<sup>1</sup> that have been shared **with the group**.
- That epk can be uploaded to a **separate** ArcGIS Enterprise **organization**.

## Import:

- Support **content items from an export** into the **selected group**.
- When the items are imported, their **item IDs are maintained**, and URLs are updated to reflect the new environment
- Before running the operation, the .epk must be added as an item. And after the operations complete, the package will be deleted.



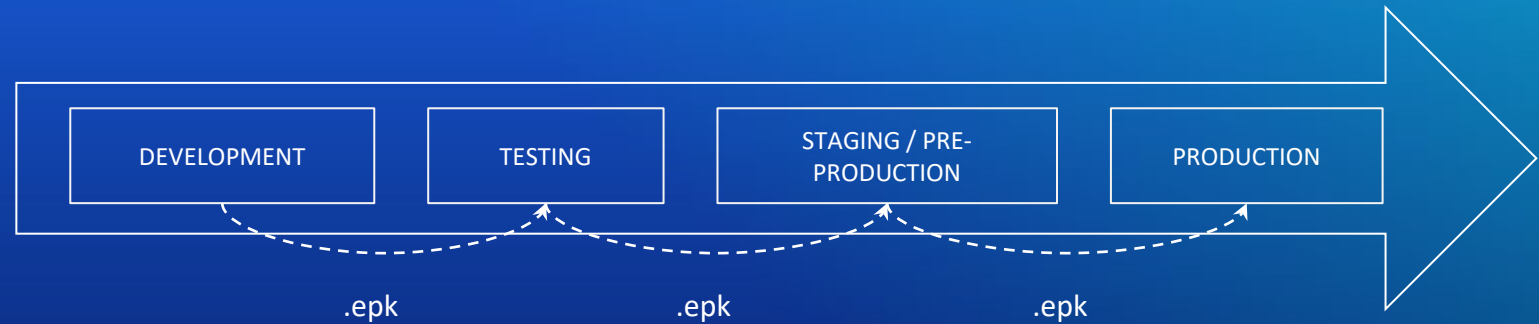
<sup>1</sup> Supported items types are listed in the docs (+30)

# Group Export and Import

When would you use them? Example scenarios

## Scenario 1)

Move content between  
**deployment environments**



## Scenario 2)

Moving content between  
**disconnected environments**



## Scenario 3)

Select content that needs to be  
backed up and restored or  
archive (**partial backups**).



# Group Export and Import

Some considerations to take into account

- **Considerations** about this version of the APIs (10.8.1).
  - **It is still available:**
    - Through **REST API** and **ArcGIS for Python API** (no Portal UI/UX yet)
    - In **ArcGIS Enterprise** (no in ArcGIS Online at this time)
- **Other technical considerations**
  - Only available to **Administrators** and **Group Owners**.
  - An .epk has a **5GB file limit**.
  - For items with **dependencies** (like web maps), items and dependencies **must be shared** with the group.

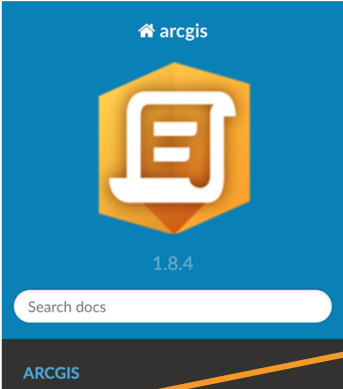


# Group Export and Import

[NEW] GroupMigrationManager class since ArcGIS API for Python 1.8.1

This new class is derived of a Group object and encapsulate the operations:

REST API (operation)	ArcGIS API for Python (method)
/export	create
/import	load



arcgis

1.8.4

Search docs

ARCGIS

- arcgis.gis
  - GIS
  - Item
  - User
  - Group
  - Datastore
  - Role
  - Layer
  - GroupApplication
  - CategorySchemaManager
  - ContentManager

## GroupMigrationManager

`class arcgis.gis.GroupMigrationManager(group)`

Bases: `object`

This manager class allows groups to export and import data to and from EPK files.

**create(items=None, future: bool = True)**

Exports a Group content to a EPK Package Item.

EPK Items are intended to migrate content from an enterprise deployment to a new enterprise. Once an EPK Item is created using this method, you can use the *load* to ingest the package's content into the target enterprise. If your package contains web maps, web-mapping applications, and/or associated web layers, during the import operation, the method will take care of unifying the service URLs and item IDs correctly.

**create(items=None, future: bool = True)**

Exports a Group content to a EPK Package Item.

EPK Items are intended to migrate content from an enterprise deployment to a new enterprise. Once an EPK Item is created using this method, you can use the *load* to ingest the

future	Optional Boolean. When True, the operation will return a Job object and return the results asynchronously.
--------	--

**Returns:** Item –or– Job when future=True

**load(epk\_item, item\_ids: list = None, overwrite: bool = True, future: bool = True)**

Imports the EPK content into the current Group.

# Group Export and Import

## DEMO TIME!

Importing a group on 6 lines of code

```
cloneMigrations - export.py
export.py
1 import os
2 from arcgis.gis import GIS
3
4 print("Connecting to source Portal...")
5 source = GIS("https://mormont.esri.es/portal", os.environ["MORMONT_USER"], os.environ["MORMONT_PWD"])
6
7 print("Getting group from source...")
8 dev_group = source.groups.search("DevSummit 2021 group")[0]
9 print(dev_group)
10 print("Building epk file...")
11
12 epk_job = dev_group.migration.create(future=True) # Async call
13 epk_file = epk_job.result()
14
15 print("Downloading export package...")
16 fp = epk_file.download()
17 print("Export operation finished:")
18 print(fp)
```

Exporting a group on 4 lines of code

```
cloneMigrations - import.py
import.py
1 import os
2 from arcgis.gis import GIS
3 from utils import clean_data
4
5 print("Connecting to target Portal...")
6 target = GIS("https://formacionutility.esri.es/portal", os.environ["UTILITY_USER"], os.environ["UTILITY_PWD"])
7
8 print("Cleaning data from previous executions")
9 clean_data()
10
11 print("Adding export package to Target Portal...")
12 pitem = target.content.add({'title': "DEVtoTEST Migration Package", "tags": ['Migration', 'TEST', 'Cloning'], 'type': 'Export_Package'}, data=fp)
13 print(pitem)
14
15 print("Creating target group...")
16 new_group = target.groups.create(title="DevSummit 2021 backup group", tags="Migration,TEST,Cloning")
17
18 print("Sharing Export Package (.epk) to the target group...")
19 pitem.share(groups=[new_group])
20
21 print("Inspecting the epk...")
22 m = new_group.migration
23 resp = m.inspect(pitem)
24 from pprint import pprint
25 pprint(resp)
26
27 print("Executing import operation...")
28 res = m.load(pitem)
29 print("Import operation finished")
30 print(res)
```

# ArcGIS API for Python

Keep diving: [developers.arcgis.com/python](https://developers.arcgis.com/python) | [github.com/Esri/arcgis-python-api](https://github.com/Esri/arcgis-python-api) | [esri.com/arcgis-blog/?s=#&products=api-python](https://esri.com/arcgis-blog/?s=#&products=api-python)

The image shows two overlapping screenshots. The top screenshot is the ArcGIS Developers website, with the 'Guide' link in the top navigation bar highlighted by a purple box. The bottom screenshot is the GitHub repository for 'Esri/arcgis-python-api', with the 'Table of Contents' section highlighted by a purple box. The 'Table of Contents' includes links to 'Introduction', 'Import libraries', 'Connect to Web GIS Systems', 'Get Started with Cloning', 'Cloning Multiple Items Simultaneously', 'Helper Functions', 'The Cloning Process' (with sub-links for 'Feature Layers', 'Web Maps', 'ArcGIS Dashboards', and 'Story Maps'), and 'Conclusion'.

## Python for Geographers

### On Demand

#### ESRI TECHNICAL SESSION

Would you like to accent your GIS skills with Python but are not sure where to get started? Learn the basics of Python-speak in the ArcGIS notebooks environment where we'll use packages to make GIS workflows reproducible through code. We will work with spatial data and make relevant maps using open source packages as well as the ArcGIS API for Python. To conclude we'll discuss useful resources as you delve deeper into the universe of Python for Data Science and GIS. Keywords: Computational geography, Introduction, Reproducible science

Manushi Majumdar

## Deploying Apps and Services with ArcGIS API for Python

### On Demand

#### ESRI TECHNICAL SESSION

This talk aimed for GIS developers demonstrates two different capabilities of the Python API. First, we show how the apps module of the Python API can be to build elegant GIS web apps such as Dashboards and StoryMaps that can be deployed easily in your GIS. Next we demonstrate how light weight web APIs such as FAAS (AWS Lambda for instance) be built using the API. We finish by demonstrating how even complex applications such as models for deep learning inferencing can be deployed in this fashion. Keywords: App development, Cloud GIS, FaaS, Lambda, GeoLambda

Akhil Negi, Divyansh Jha, Atma Mani

The image shows a YouTube search results page for 'ArcGIS API for Python'. The search bar at the top contains the text 'ArcGIS API for Python'. Below the search bar, there is a table of results. The first result is a video titled 'Python Map Automation and Beyond' by ArcGIS, with a duration of 59:02. The second result is a video titled 'Performing Analysis with the ArcGIS API for Python' by ArcGIS, with a duration of 1:00:34. The third result is a video titled 'Python Across the ArcGIS Platform' by Esri Events, with a duration of 36:31. The fourth result is a video titled 'ArcGIS API for Python: Mapping, Visualization, and Analysis' by Esri Events, with a duration of 54:01. The fifth result is a video titled 'Using Python and Notebooks in ArcGIS Enterprise' by Esri Events, with a duration of 36:22. The sixth result is a video titled 'Processing Drone Imagery using the ArcGIS API for Python' by Esri Events, with a duration of 55:10. The seventh result is a video titled 'ArcGIS API for Python: Defining Templates' by Esri Events, with a duration of 36:22.

The image shows a screenshot of the ArcGIS Developers website, specifically the 'ArcGIS API for Python' section. It displays a list of videos related to the API. The videos are listed in a table with columns for video number, video title, duration, and presenter. The videos are: 1. 'Python Map Automation and Beyond' by ArcGIS (59:02), 2. 'Performing Analysis with the ArcGIS API for Python' by ArcGIS (1:00:34), 3. 'Python Across the ArcGIS Platform' by Esri Events (36:31), 4. 'ArcGIS API for Python: Mapping, Visualization, and Analysis' by Esri Events (54:01), 5. 'Using Python and Notebooks in ArcGIS Enterprise' by Esri Events (36:22), 6. 'Processing Drone Imagery using the ArcGIS API for Python' by Esri Events (55:10), and 7. 'ArcGIS API for Python: Defining Templates' by Esri Events (36:22).



**Please share your feedback via the survey  
available at the link below the presentation.**





esri®

THE  
SCIENCE  
OF  
WHERE®