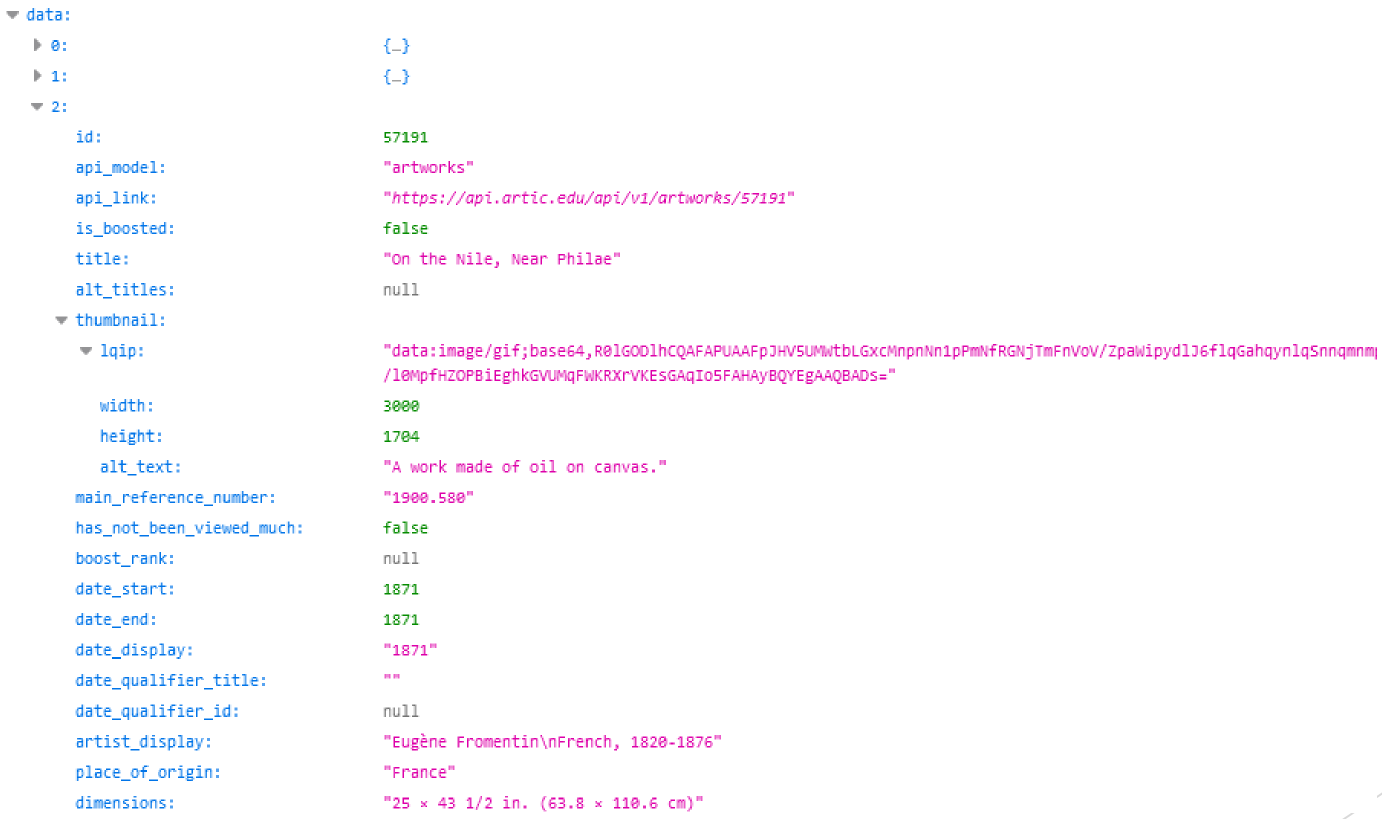
Data Classes and APIs

Annie, Ben H., Lindsey

Art Institute of Chicago API

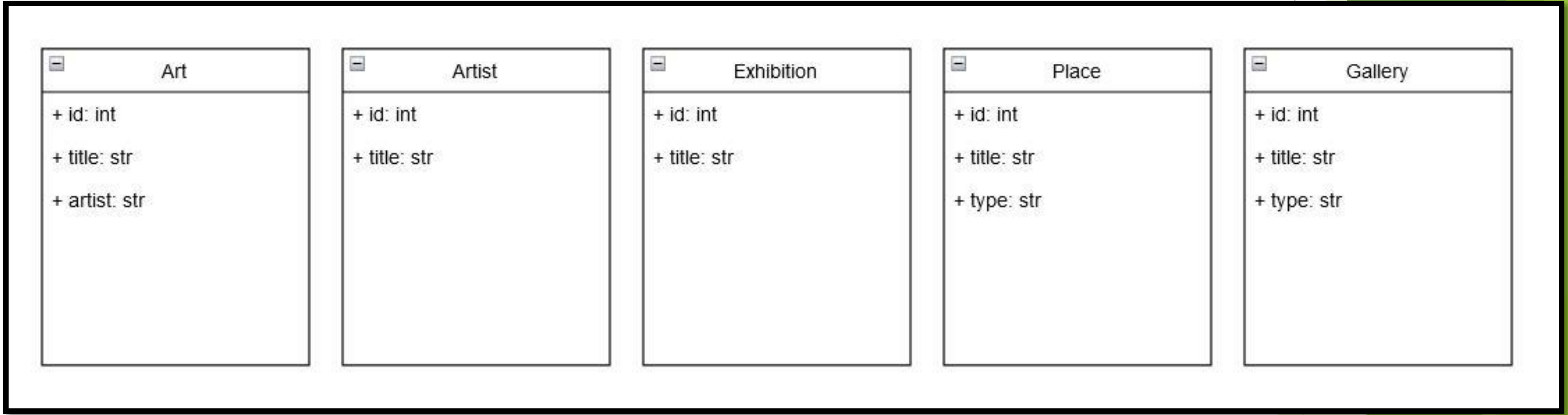
https://api.artic.edu/api

https://api.artic.edu/api/v1/artworks



```
▾ data:
  ▸ 0:                                  {…}
  ▸ 1:                                  {…}
  ▾ 2:
      id:                               57191
      api_model:                        "artworks"
      api_link:                         "https://api.artic.edu/api/v1/artworks/57191"
      is_boosted:                       false
      title:                            "On the Nile, Near Philae"
      alt_titles:                       null
    ▾ thumbnail:
      ▾ lqip:                           "data:image/gif;base64,R0lGODlhCQAFAPUAAFpJHV5UMWtbLGxcMnpnNn1pPmNfRGNjTmFnVoV/ZpaWipydlJ6flqGahqynlqSnnqmnm
                                        /l0MpfHZOPBiEghkGVUMqFWKRXrVKEsGAqIo5FAHAyBQYEgAAQBADs="
        width:                          3000
        height:                         1704
        alt_text:                       "A work made of oil on canvas."
      main_reference_number:            "1900.580"
      has_not_been_viewed_much:         false
      boost_rank:                       null
      date_start:                       1871
      date_end:                         1871
      date_display:                     "1871"
      date_qualifier_title:             ""
      date_qualifier_id:                null
      artist_display:                   "Eugène Fromentin\nFrench, 1820-1876"
      place_of_origin:                  "France"
      dimensions:                       "25 × 43 1/2 in. (63.8 × 110.6 cm)"
```

Diagram of Classes



Art
+ id: int
+ title: str
+ artist: str

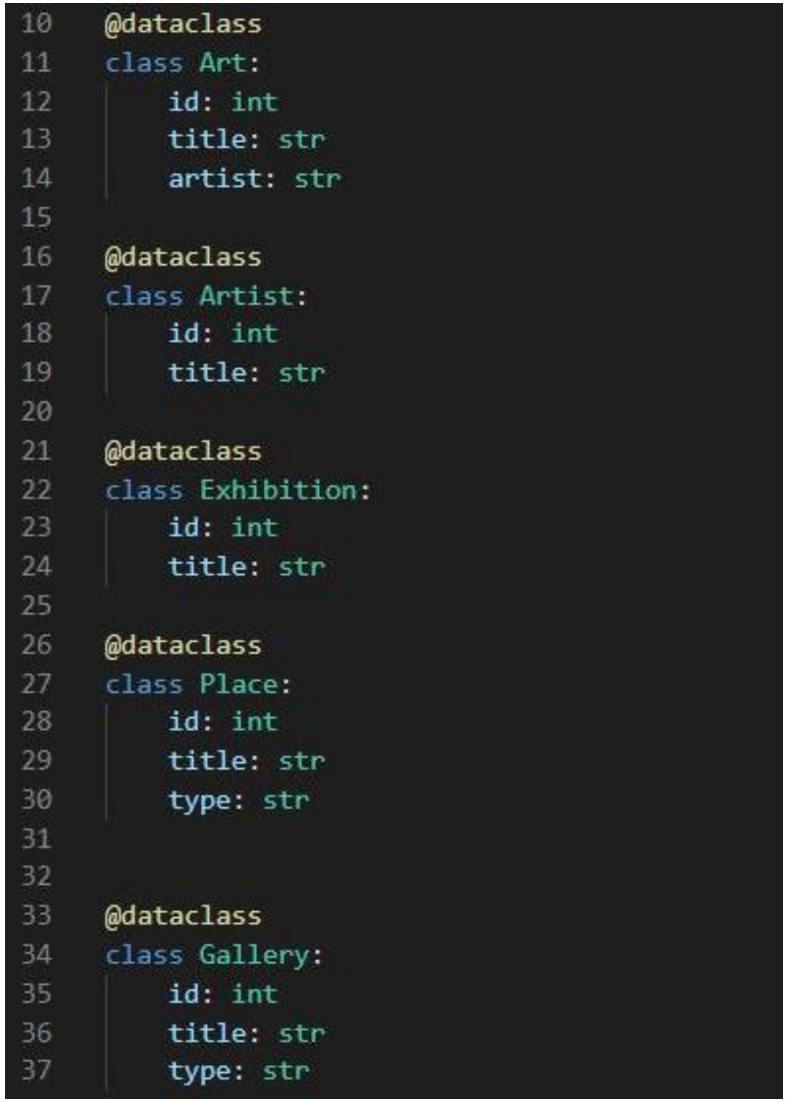
Artist
+ id: int
+ title: str

Exhibition
+ id: int
+ title: str

Place
+ id: int
+ title: str
+ type: str

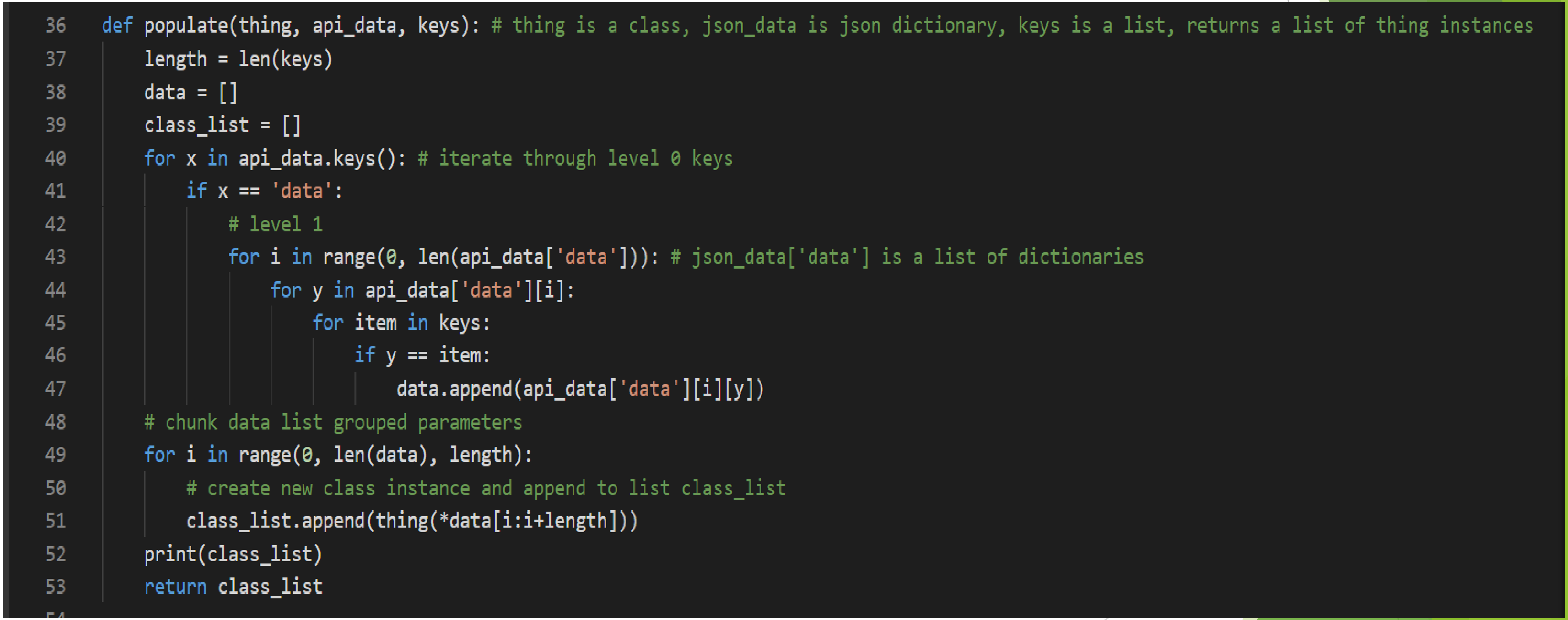
Gallery
+ id: int
+ title: str
+ type: str

Code (Classes)



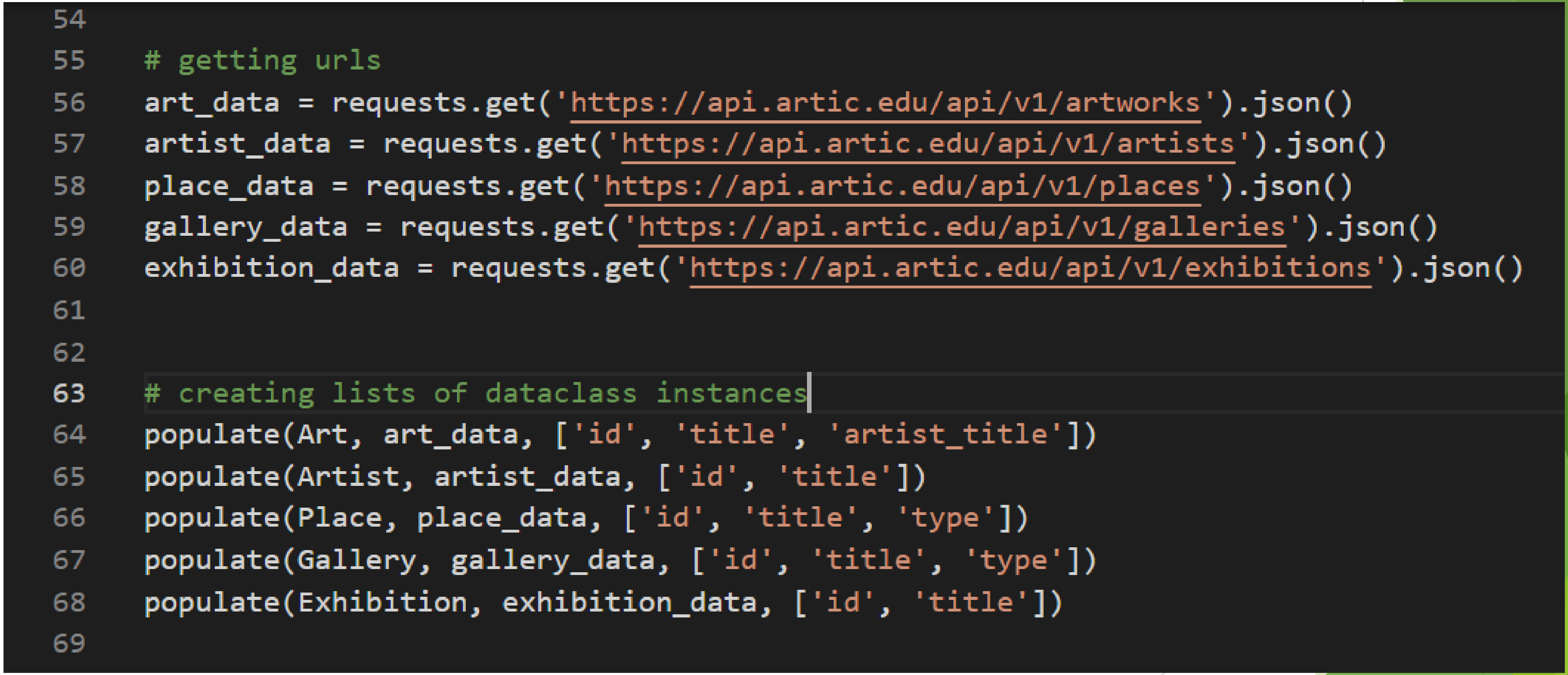
```python
@dataclass
class Art:
    id: int
    title: str
    artist: str

@dataclass
class Artist:
    id: int
    title: str

@dataclass
class Exhibition:
    id: int
    title: str

@dataclass
class Place:
    id: int
    title: str
    type: str


@dataclass
class Gallery:
    id: int
    title: str
    type: str
```

Code (Function)



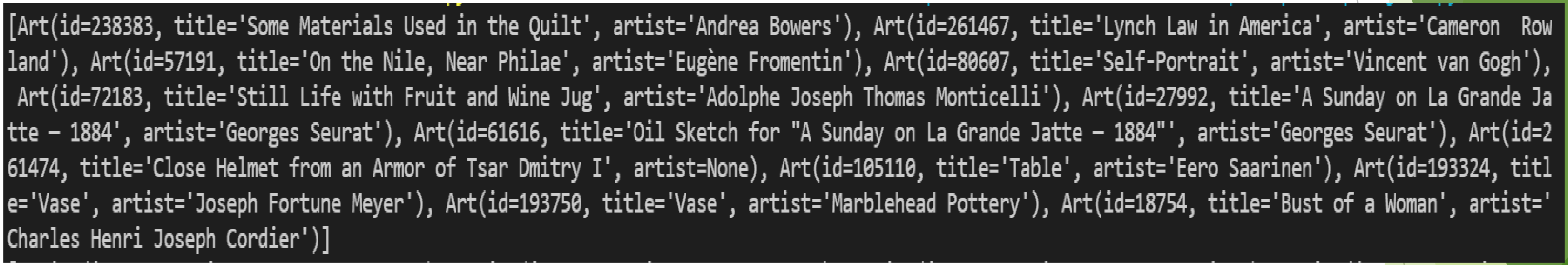
```python
def populate(thing, api_data, keys): # thing is a class, json_data is json dictionary, keys is a list, returns a list of thing instances
    length = len(keys)
    data = []
    class_list = []
    for x in api_data.keys(): # iterate through level 0 keys
        if x == 'data':
            # level 1
            for i in range(0, len(api_data['data'])): # json_data['data'] is a list of dictionaries
                for y in api_data['data'][i]:
                    for item in keys:
                        if y == item:
                            data.append(api_data['data'][i][y])
    # chunk data list grouped parameters
    for i in range(0, len(data), length):
        # create new class instance and append to list class_list
        class_list.append(thing(*data[i:i+length]))
    print(class_list)
    return class_list
```

Code (Execution)



```python

# getting urls
art_data = requests.get('https://api.artic.edu/api/v1/artworks').json()
artist_data = requests.get('https://api.artic.edu/api/v1/artists').json()
place_data = requests.get('https://api.artic.edu/api/v1/places').json()
gallery_data = requests.get('https://api.artic.edu/api/v1/galleries').json()
exhibition_data = requests.get('https://api.artic.edu/api/v1/exhibitions').json()


# creating lists of dataclass instances
populate(Art, art_data, ['id', 'title', 'artist_title'])
populate(Artist, artist_data, ['id', 'title'])
populate(Place, place_data, ['id', 'title', 'type'])
populate(Gallery, gallery_data, ['id', 'title', 'type'])
populate(Exhibition, exhibition_data, ['id', 'title'])

```

Code (Output)



```
[Art(id=238383, title='Some Materials Used in the Quilt', artist='Andrea Bowers'), Art(id=261467, title='Lynch Law in America', artist='Cameron  Row
land'), Art(id=57191, title='On the Nile, Near Philae', artist='Eugène Fromentin'), Art(id=80607, title='Self-Portrait', artist='Vincent van Gogh'),
 Art(id=72183, title='Still Life with Fruit and Wine Jug', artist='Adolphe Joseph Thomas Monticelli'), Art(id=27992, title='A Sunday on La Grande Ja
tte – 1884', artist='Georges Seurat'), Art(id=61616, title='Oil Sketch for "A Sunday on La Grande Jatte – 1884"', artist='Georges Seurat'), Art(id=2
61474, title='Close Helmet from an Armor of Tsar Dmitry I', artist=None), Art(id=105110, title='Table', artist='Eero Saarinen'), Art(id=193324, titl
e='Vase', artist='Joseph Fortune Meyer'), Art(id=193750, title='Vase', artist='Marblehead Pottery'), Art(id=18754, title='Bust of a Woman', artist='
Charles Henri Joseph Cordier')]
```

Issues We Encountered

- Understanding the data structure and accessing information in it
  - Drilling down into dictionaries
- Acquiring desired data and populating it into classes
- Iterating across the dictionaries
- Function enabled repeatability

Conclusion

- Accomplished:
  - Created a program that defines multiple dataclasses and utilized an API to instantiate them.
- Next steps:
  - Add methods to utilize data
  - Inheritance
  - Magic Method overrides
  - Expand the data held by classes