Data Classes and APIs

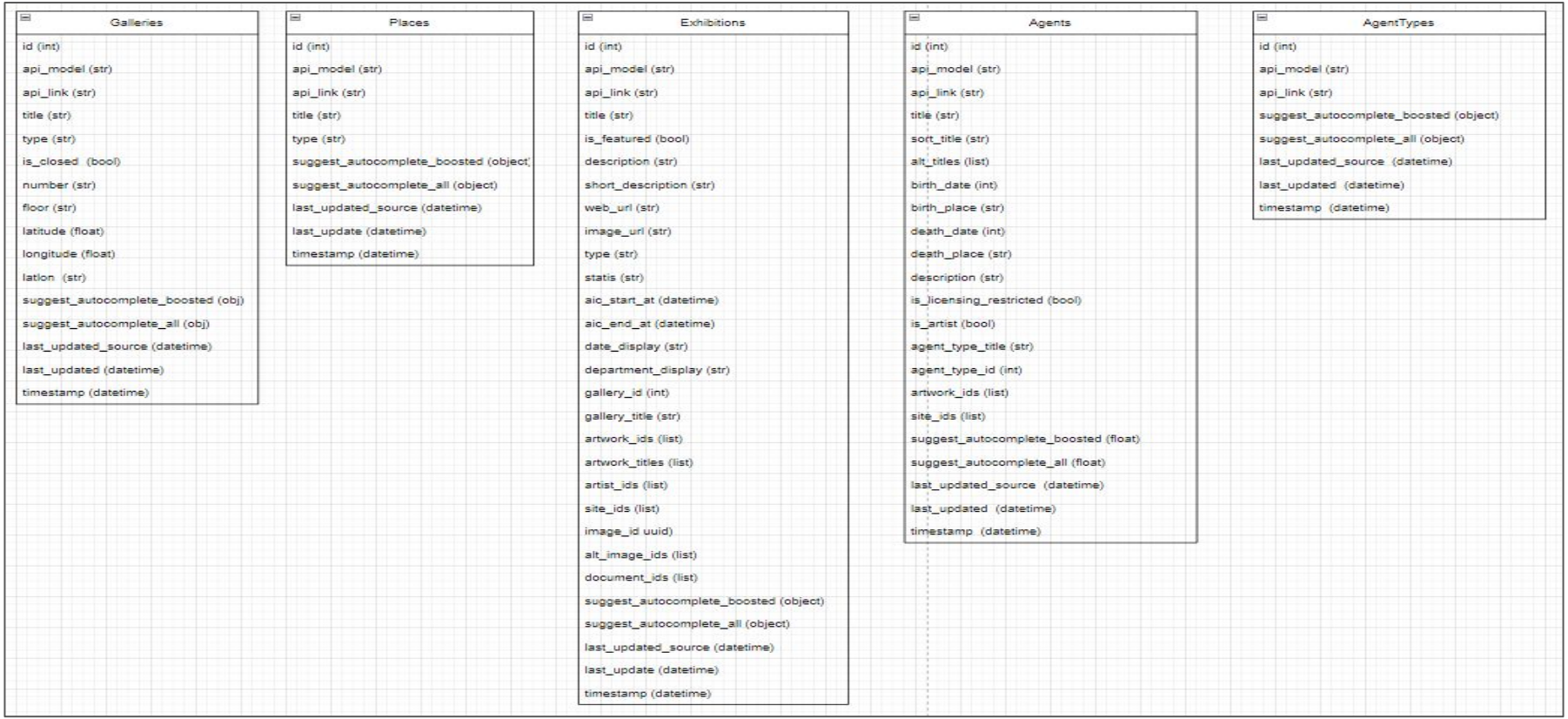
Darrell Gerber, Raymond Sepulveda, Amy Yucus

Github: https://github.com/dgitall/GroupExerciseWebAPI

Web API Selection

- Some APIs returned little data for us to use
- Some APIs had poor documentation
- Art Institute of Chicago API was selected
- Data Classes:
  - Galleries
  - Places
  - Exhibitions
  - Agents
  - Agent Types

Initial Data Model



Coding the Classes: Galleries, for example

Galleries.py - GroupExerciseWebAPI -

Galleries.py > Galleries > read

You, 44 minutes ago | 1 author (You)

```python
from dataclasses import dataclass, fields
import datetime as dt

@dataclass
class Galleries:
    """ Galleries class"""
    def __init__(self):
        id = 0
        api_model = ' '
        api_link = ' '
        title = ' '
        type = ' '
        is_closed = False
        number = ' '
        floor = ' '
        latitude = 0
        longitude = 0
        latlon = ' '
        suggest_autocomplete_boosted = ' '
        suggest_autocomplete_all = ' '
        last_updated_source = dt.datetime(1970, 1, 1)
        last_updated = dt.datetime(1970, 1, 1)
        timestamp = dt.datetime(1970, 1, 1)

    id: int
    api_model: str
    api_link: str
    title: str
    type: str
    is_closed: bool
    number: str
    floor: str
    latitude: float
    longitude: float
    latlon: str
```

Galleries.py - GroupExerciseWebAPI - Visual Studio Code

Galleries.py > Galleries > read

```python
    floor: str
    latitude: float
    longitude: float
    latlon: str
    suggest_autocomplete_boosted: str
    suggest_autocomplete_all: str
    last_updated_source: dt.datetime
    last_updated: dt.datetime
    timestamp: dt.datetime


    def read(self, jsondata):
        status = 0
        dateformat = "%Y-%m-%d%T%H:%M:%S%z"
        if(jsondata is not None):
            self.id = int(jsondata['id'])
            self.api_model = jsondata['api_model']
            self.api_link = jsondata['api_link']
            self.title = jsondata['title']
            self.type = jsondata['type']
            self.is_closed = bool(jsondata['is_closed'])
            self.number = jsondata['number']
            self.floor = jsondata['floor']
            if jsondata['latitude'] == None:
                self.latitude = -999999
            else:
                self.latitude = float(jsondata['latitude'])
            if jsondata['longitude'] == None:
                self.longitude = -999999
            else:
                self.longitude = float(jsondata['longitude'])
            self.latlon = jsondata['latlon']
            if jsondata.get('suggest_autocomplete_boosted') == None:
                self.suggest_autocomplete_boosted = ' '
            else:
                self.suggest_autocomplete_boosted = jsondata['suggest_autocomplete_boosted']
            if jsondata.get('suggest_autocomplete_all') == None:
```

Galleries.py - GroupExerciseWebAPI - Visual Studio Code

Galleries.py > Galleries > read

```python
            self.id = int(jsondata['id'])
            self.api_model = jsondata['api_model']
            self.api_link = jsondata['api_link']
            self.title = jsondata['title']
            self.type = jsondata['type']
            self.is_closed = bool(jsondata['is_closed'])
            self.number = jsondata['number']
            self.floor = jsondata['floor']
            if jsondata['latitude'] == None:
                self.latitude = -999999
            else:
                self.latitude = float(jsondata['latitude'])
            if jsondata['longitude'] == None:
                self.longitude = -999999
            else:
                self.longitude = float(jsondata['longitude'])
            self.latlon = jsondata['latlon']
            if jsondata.get('suggest_autocomplete_boosted') == None:
                self.suggest_autocomplete_boosted = ' '
            else:
                self.suggest_autocomplete_boosted = jsondata['suggest_autocomplete_boosted']
            if jsondata.get('suggest_autocomplete_all') == None:
                self.suggest_autocomplete_all = ' '
            else:
                self.suggest_autocomplete_all = jsondata['suggest_autocomplete_all']
            self.last_updated_source = dt.datetime.strptime(jsondata['last_updated_source'], dateformat)
            self.last_updated = dt.datetime.strptime(jsondata['last_updated'], dateformat)
            self.timestamp = dt.datetime.strptime(jsondata['timestamp'], dateformat)
        else:
            status = -1
            You, 3 hours ago • Gallery API data pull
        return status

    def __str__(self):
        return self.title
```

__init__: default values

Parameters: start

Parameters: continued

Read(json) function: to transfer data from the JSON block to the parameters

Read(json): continued

__str__: return title

Coding the API calls: e.g. Galleries

- Load classes from individual files
- Setup base pieces of URL
- Loop through all of the page requests
  - First time get the total pages
  - Loop through each block of gallery data
  - Call the read() function in each class
  - Append to list if successful
- Print list

MainApp.py - GroupExerciseWebAPI - Visual Studio Code

MainApp.py | Exhibitions.py | Agents.py | AgentTypes.py | Places.py | Artwork.p

MainApp.py > agents_list

You, 10 minutes ago | 2 authors (You and others)

```python
## Main application file for the AIC API group project
import requests
from pprintpp import pprint as pp

import Galleries
import Places
import Agents
import AgentTypes
import Exhibitions

baseURL = "https://api.artic.edu/api/v1/"
limitURL = "?limit=10"
pageURL = "&page="

################################
## Galleries

galleries_list = []
totalpages = 1
page = 0
while page < totalpages:
    galleriesURL = f"{baseURL}galleries{limitURL}{pageURL}{page+1}"
    response = requests.get(galleriesURL)
    galleries_json = response.json()

    if totalpages == 1:
        totalpages = int(galleries_json['pagination']['total_pages'])

    # pull out the data package from the json payload
    # it returns an iterable list of
    data = galleries_json['data']

    for galleryData in data:
        gallery = Galleries.Galleries()
        result = gallery.read(galleryData)
        if (result == 0):
            galleries_list.append(gallery)

    page += 1

pp(galleries_list)
```

TERMINAL | PROBLEMS 1 | OUTPUT | DEBUG CONSOLE

```
tetime(1976, 9, 2, 11, 20, tzinfo=datetime.timezone(datetime.timedelta(days=-1, seconds=68400))), last_updated
timedelta(days=-1, seconds=68400))), timestamp=datetime.datetime(2022, 1, 6, 14, 3, 19, tzinfo=datetime.timezo
    Galleries(id=2147477076, api_model='galleries', api_link='https://api.artic.edu/api/v1/galleries/214747707
osed=False, number=None, floor='1', latitude=41.880504, longitude=-87.623648, latlon='41.880504,-87.623648', s
urce=datetime.datetime(1976, 9, 2, 11, 20, tzinfo=datetime.timezone(datetime.timedelta(days=-1, seconds=68400)
mezone(datetime.timedelta(days=-1, seconds=68400))), timestamp=datetime.datetime(2022, 1, 6, 14, 3, 19, tzinfo
    Galleries(id=2147476004, api_model='galleries', api_link='https://api.artic.edu/api/v1/galleries/214747600
or='1', latitude=41.879239947372, longitude=-87.6222917258, latlon='41.879239947372,-87.6222917258', suggest_a
etime.datetime(1976, 9, 2, 11, 20, tzinfo=datetime.timezone(datetime.timedelta(days=-1, seconds=68400))), last
atetime.timedelta(days=-1, seconds=68400))), timestamp=datetime.datetime(2022, 1, 6, 14, 3, 19, tzinfo=datetim
    Galleries(id=2147473659, api_model='galleries', api_link='https://api.artic.edu/api/v1/galleries/214747365
e, number=None, floor='1', latitude=41.879586, longitude=-87.62398, latlon='41.879586,-87.62398', suggest_auto
me.datetime(1976, 9, 2, 11, 20, tzinfo=datetime.timezone(datetime.timedelta(days=-1, seconds=68400))), last_up
ime.timedelta(days=-1, seconds=68400))), timestamp=datetime.datetime(2022, 1, 6, 14, 3, 19, tzinfo=datetime.ti
    Galleries(id=2147473950, api_model='galleries', api_link='https://api.artic.edu/api/v1/galleries/214747395
```

Darrell | Live Share | 1.83 KB | 608ms | 4 hrs 9 mins | tabnine

What did we learn

- Check for empty entries for all classes in the beginning instead of fixing them at the end.
- Split URLs into variables
  - Especially when working with multiple pages.
- Utilize ‘Try’ and ‘Except’ blocks more
  - Primarily for GET requests, so that it can handle errors if and when they occur
- __init__ functions may be necessary to specify, even in data classes, depending on what API is being interacted with