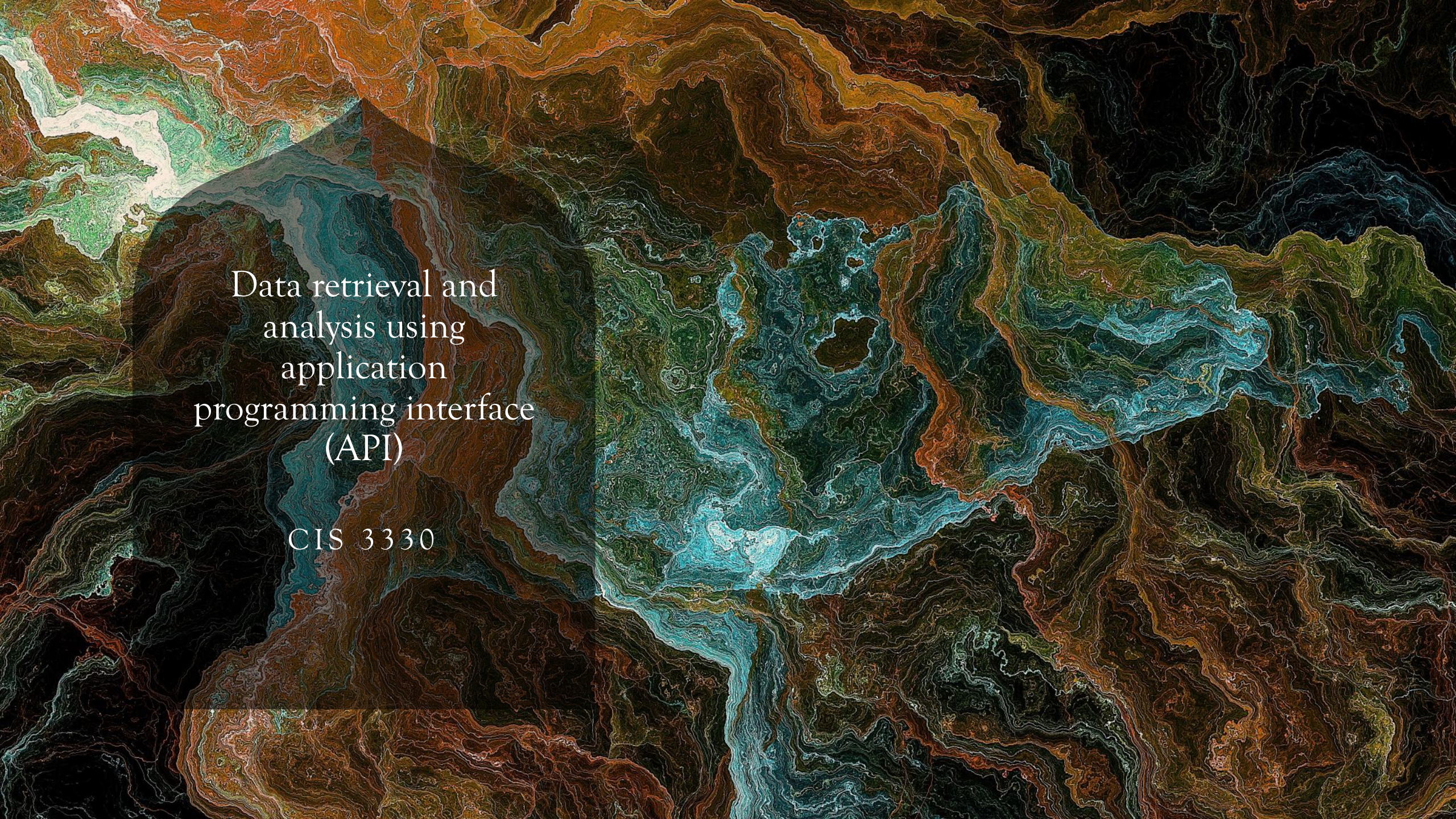

Data retrieval and analysis using application programming interface (API)
CIS 3330

Application Programming Interface (API)

An API can be defined as a technology layer that allows applications to interact with each other without giving direct access to programming code or code logic .

APIs incorporate rules and procedures that other applications must follow in order to integrate their products.

APIs for data retrieval can control the access and level of detail of outsiders to data

APIs for data analysis can limit the amount of data or requests you can send.

In data analysis and retrieval APIs often include quotas and limitations.

The pricing for APIs can be done by number requests or the complexity of requests.

OpenAI Research Product Developers Safety Company

CIS 3330 - Dr. Villacis Calderon

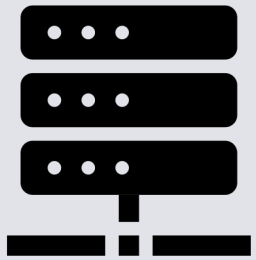
https://openai.com/pricing

Pricing

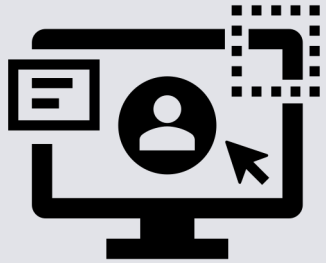
Simple and flexible. Only pay for what you use.

Contact sales Learn more ↓

How to use an API



Find a service that you want to integrate to your code



Request access to the API

Implement the API functionalities in your code or workflow

Yelp Fusion API

Yelp Fusion API

- "The Yelp Fusion API allows you to get the best local content and user reviews from millions of businesses around the world."
- In this class we will learn how to obtain a list of business and how to obtain three reviews from the businesses.
- We are going to use two features of the API

  Business search – https://docs.developer.yelp.com/reference/v3_business_search

  Reviews – https://docs.developer.yelp.com/reference/v3_business_reviews
- We will learn how to use both features by implementing code using the **yelpapi** package (simpler approach) and the **request** package (advanced approach).

Searching for businesses using the Yelp API

Business search – yelpapi

```
from yelpapi import YelpAPI

api_key ="INSERT YOUR KEY"
yelp_api = YelpAPI(api_key)
search_term = "tacos"
location_term = "El Paso, TX"
search_results = yelp_api.search_query(
    term=search_term, location=location_term,
    sort_by='rating', limit=20)
# List of results in search_results['businesses']
print(search_results)
```

Business search – yelpapi

Searching for the next 20 restaurant

```
from yelpapi import YelpAPI
api_key ="INSERT YOUR KEY"
yelp_api = YelpAPI(api_key)
search_term = "tacos"
location_term = "El Paso, TX"

search_results = yelp_api.search_query(
    term=search_term, location=location_term,
    sort_by='rating', limit=20, offset=20
    )
# List of results in search_results['businesses']
print(search_results)
```

https://docs.developer.yelp.com/reference/v3_business_search

Business search – request

```python
import requests
import urllib.parse
import json

search_term = "tacos"
search_term = urllib.parse.quote_plus(search_term)
location = "El Paso, TX"
location = urllib.parse.quote_plus(location)
sort_by = "rating" # best_match, rating, review_count, distance
limit = 20 # number
url = f"https://api.yelp.com/v3/businesses/search?location={loca
headers = {
    "accept": "application/json",
    "Authorization": "Bearer INSERT YOUR KEY"
}
response = requests.get(url, headers=headers)
response_json = json.loads(response.text)
# List of results in response_json['businesses']
print(response_json['businesses'])
```

```
url=f"https://api.yelp.com/v3/businesses/search?location={location}&term={search_te
rm}&sort_by={sort_by}&limit={limit}"
```

Using pandas to handle API responses

yelpapi approach

```
# List of results in search_results['businesses']
print(search_results)
#Converting list of json objects to Pandas Data Frame
result_df = pd.DataFrame.from_dict(search_results['businesses'])
print(result_df)
result_df.to_csv("yelpapi_businesses_results.csv")
```

request approach

```
response = requests.get(url, headers=headers)
response_json = json.loads(response.text)
# List of results in response_json['businesses']
print(response_json['businesses'])
#Converting list of json objects to Pandas Data Frame
result_df = pd.DataFrame.from_dict(response_json['businesses'])
print(result_df)
result_df.to_csv("request_businesses_results.csv")
```

Searching for businesses reviews using Yelp API

Business search – yelpapi

```
id_for_reviews = "taconeta-el-paso" #use alias from data
review_response = yelp_api.reviews_query(id=id_for_reviews)
# List of reviews in review_response['reviews']
print(review_response)
#Text is in element (review) text
for review in review_response['reviews']:
    print(review['text'])
```

Business search – request

```python
import requests
import json
import pandas as pd

id_for_reviews ="taconeta-el-paso"
limit = 20
sort_by = "newest" # newest or yelp_sort
url = f"https://api.yelp.com/v3/businesses/{id_for_reviews}/reviews?limit={limit}&sort_by={sort_by}"
headers = {
    "accept": "application/json",
    "Authorization": "Bearer INSERT YOUR KEY"
}
response = requests.get(url, headers=headers)
response_json = json.loads(response.text)
print(response_json)
```

Using pandas to handle API responses

yelpapi approach

```
id_for_reviews = "taconeta-el-paso" #use alias from data
review_response = yelp_api.reviews_query(id=id_for_reviews)
# List of reviews in review_response['reviews']
print(review_response)
#Text is in element (review) text
for review in review_response['reviews']:
    print(review['text'])
#Converting list of json objects to Pandas Data Frame
result_df = pd.DataFrame.from_dict(review_response['reviews'])
print(result_df)
result_df.to_csv(f"{id_for_reviews}_yelpapi_businesses_results.csv")
```

request approach

```
response = requests.get(url, headers=headers)
response_json = json.loads(response.text)
print(response_json)
#Converting list of json objects to Pandas Data Frame
result_df = pd.DataFrame.from_dict(response_json['reviews'])
print(result_df)
result_df.to_csv(f"{id_for_reviews}_request_reviews_results.csv")
```