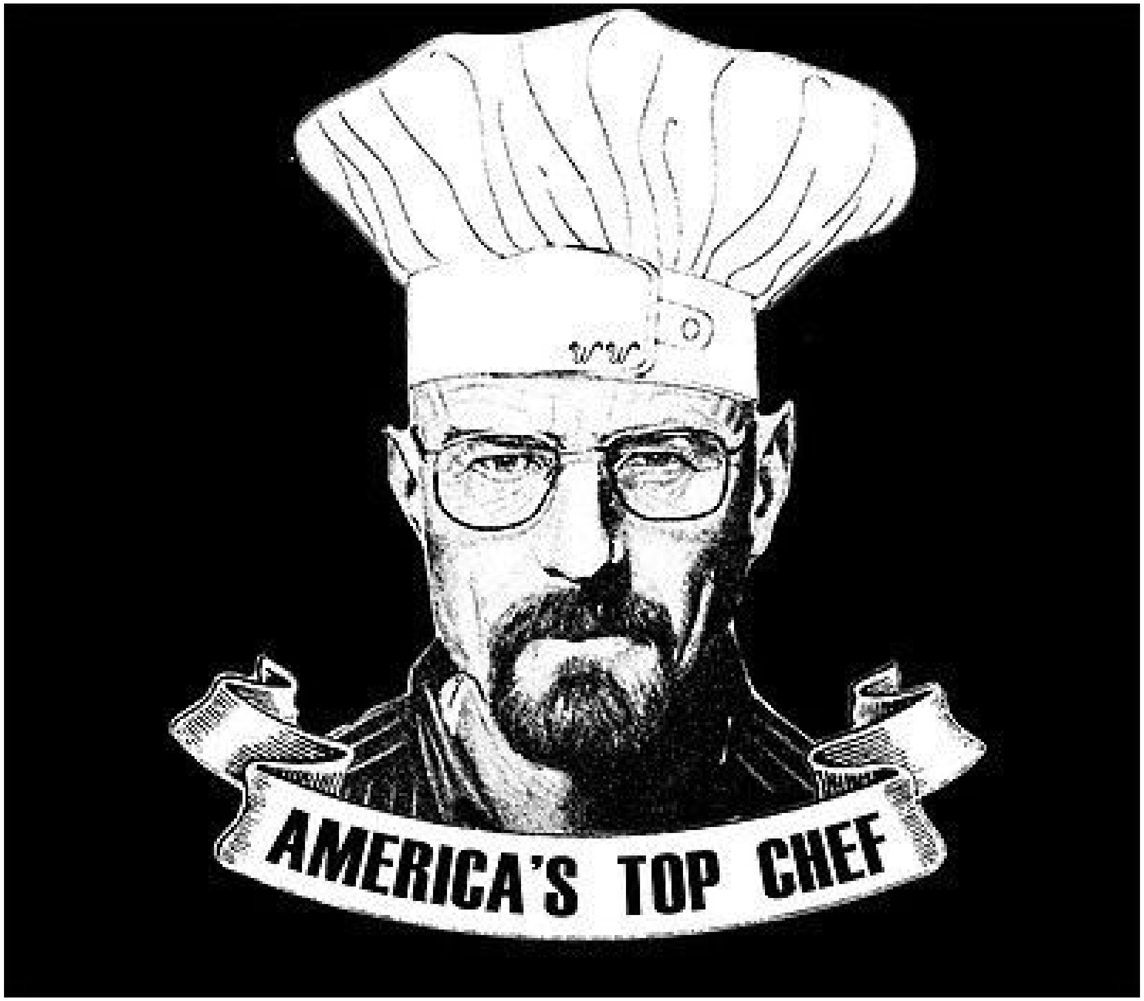
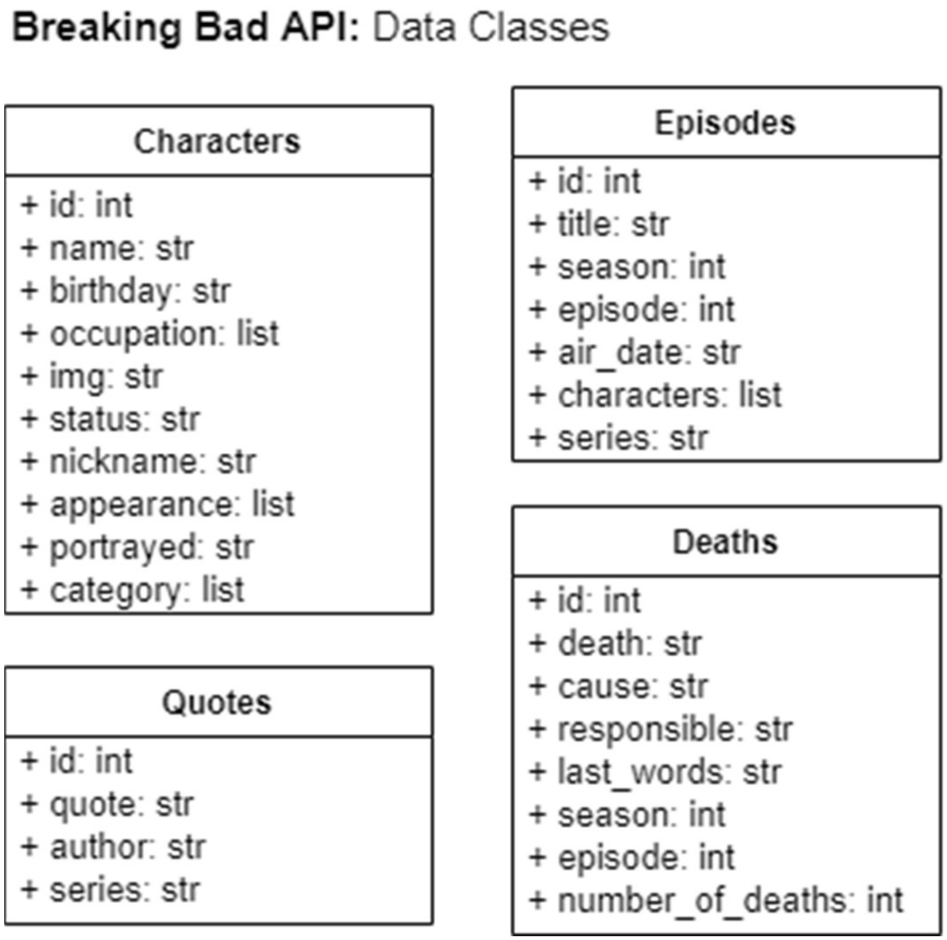
Breaking Bad API

The Chefs

Abbey Guilliat, Sargis Abrahamyan, Daniel Brickman



Data Classes



Breaking Bad API: Data Classes

Characters

+ id: int
+ name: str
+ birthday: str
+ occupation: list
+ img: str
+ status: str
+ nickname: str
+ appearance: list
+ portrayed: str
+ category: list

Quotes

+ id: int
+ quote: str
+ author: str
+ series: str

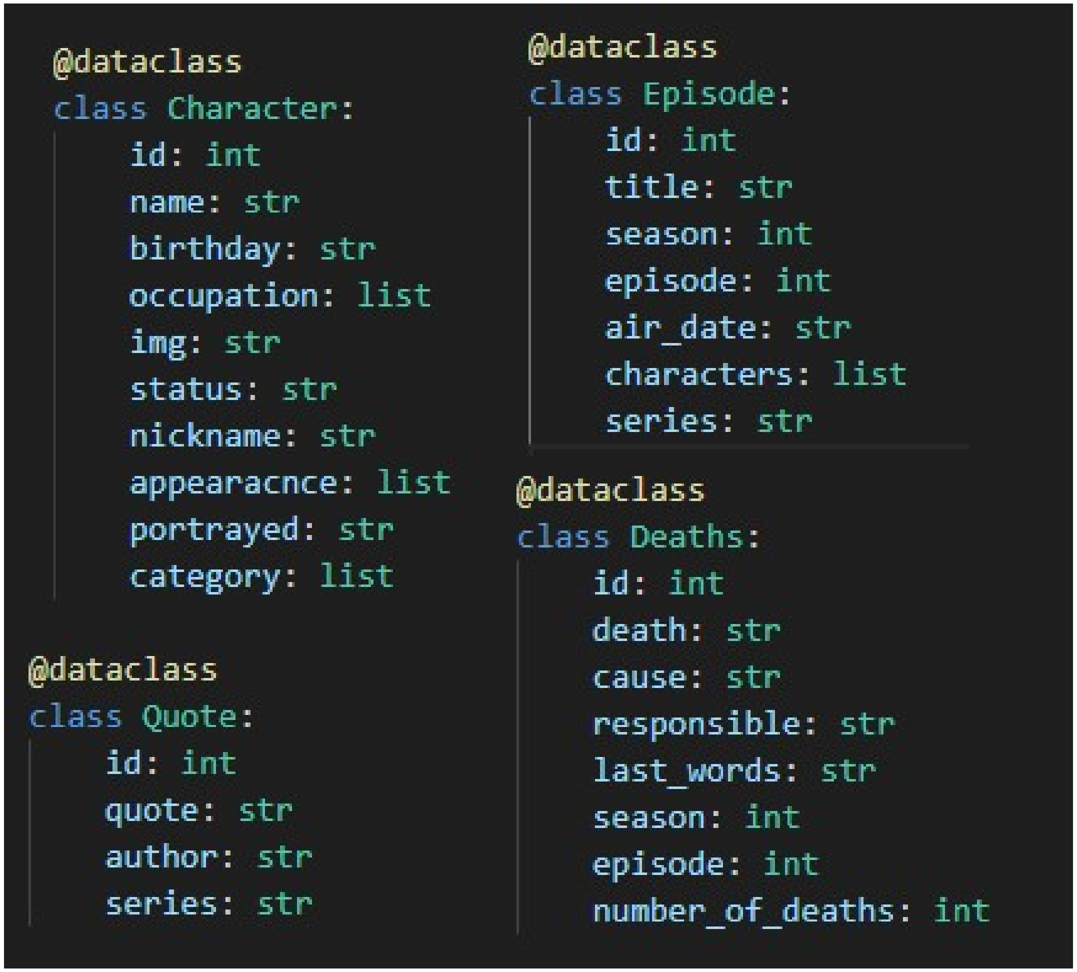
Episodes

+ id: int
+ title: str
+ season: int
+ episode: int
+ air_date: str
+ characters: list
+ series: str

Deaths

+ id: int
+ death: str
+ cause: str
+ responsible: str
+ last_words: str
+ season: int
+ episode: int
+ number_of_deaths: int

Data Classes Code



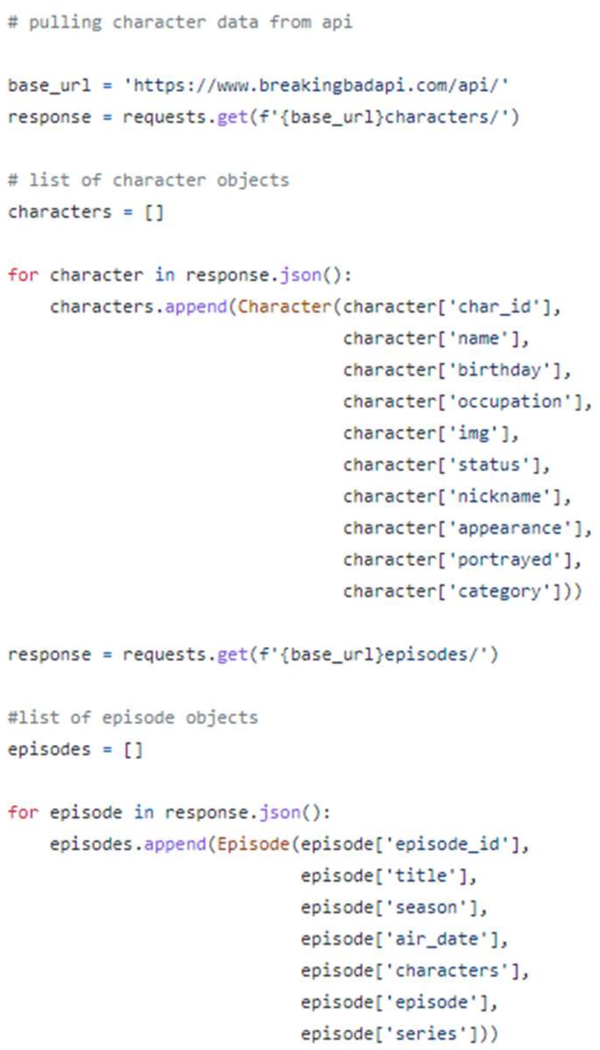
```python
@dataclass
class Character:
    id: int
    name: str
    birthday: str
    occupation: list
    img: str
    status: str
    nickname: str
    appearacnce: list
    portrayed: str
    category: list
```

```python
@dataclass
class Quote:
    id: int
    quote: str
    author: str
    series: str
```

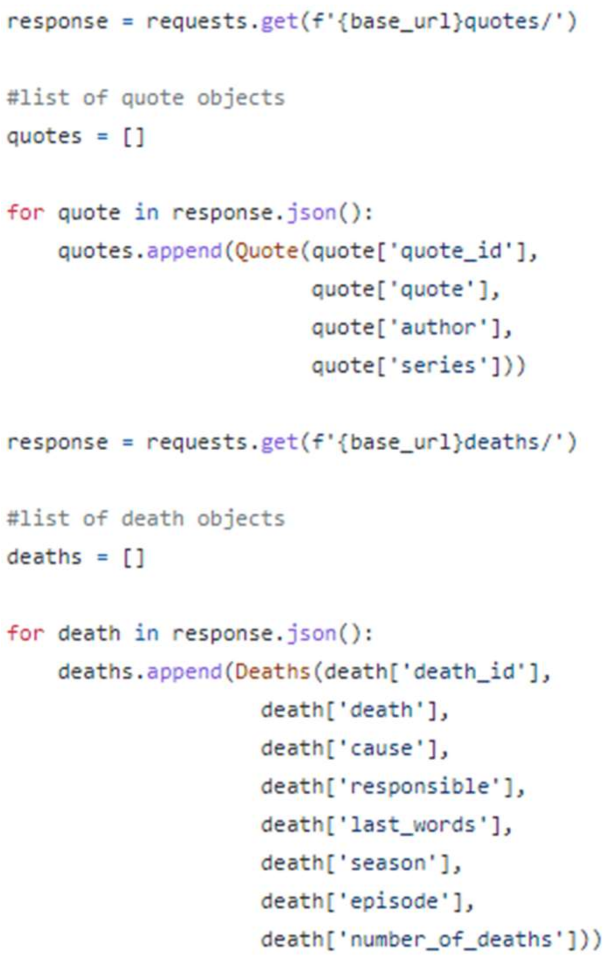
```python
@dataclass
class Episode:
    id: int
    title: str
    season: int
    episode: int
    air_date: str
    characters: list
    series: str
```

```python
@dataclass
class Deaths:
    id: int
    death: str
    cause: str
    responsible: str
    last_words: str
    season: int
    episode: int
    number_of_deaths: int
```

Loading Data Code

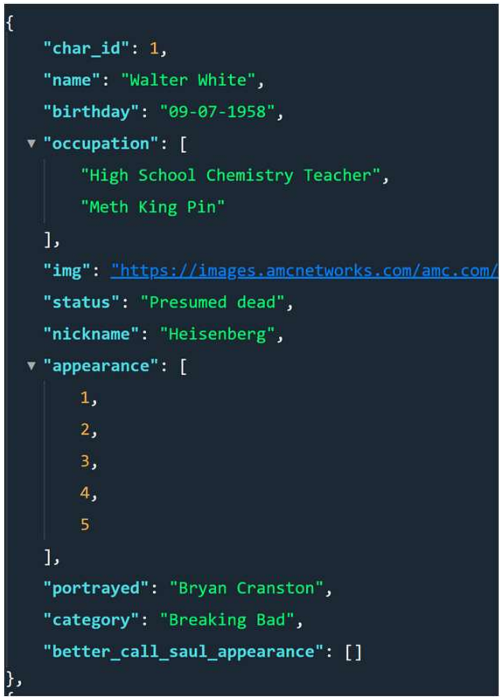


```
# pulling character data from api

base_url = 'https://www.breakingbadapi.com/api/'
response = requests.get(f'{base_url}characters/')

# list of character objects
characters = []

for character in response.json():
    characters.append(Character(character['char_id'],
                                character['name'],
                                character['birthday'],
                                character['occupation'],
                                character['img'],
                                character['status'],
                                character['nickname'],
                                character['appearance'],
                                character['portrayed'],
                                character['category']))

response = requests.get(f'{base_url}episodes/')

#list of episode objects
episodes = []

for episode in response.json():
    episodes.append(Episode(episode['episode_id'],
                            episode['title'],
                            episode['season'],
                            episode['air_date'],
                            episode['characters'],
                            episode['episode'],
                            episode['series']))
```

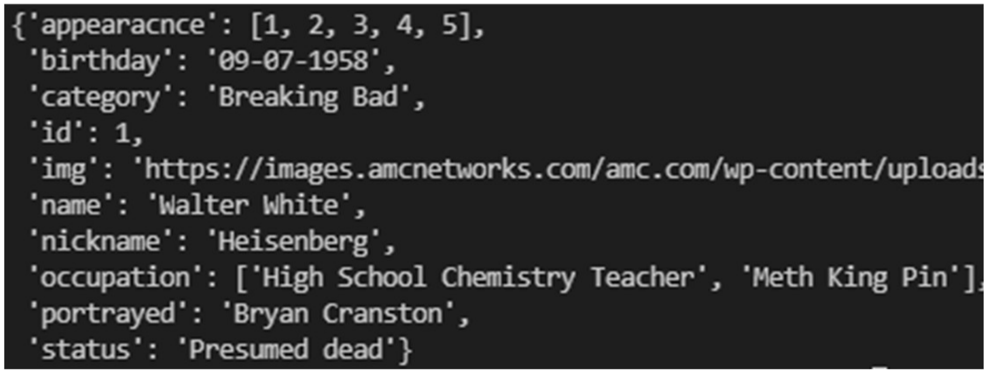


```
response = requests.get(f'{base_url}quotes/')

#list of quote objects
quotes = []

for quote in response.json():
    quotes.append(Quote(quote['quote_id'],
                        quote['quote'],
                        quote['author'],
                        quote['series']))

response = requests.get(f'{base_url}deaths/')

#list of death objects
deaths = []

for death in response.json():
    deaths.append(Deaths(death['death_id'],
                         death['death'],
                         death['cause'],
                         death['responsible'],
                         death['last_words'],
                         death['season'],
                         death['episode'],
                         death['number_of_deaths']))
```

API vs. The Chefs



```
{
    "char_id": 1,
    "name": "Walter White",
    "birthday": "09-07-1958",
  ▼ "occupation": [
        "High School Chemistry Teacher",
        "Meth King Pin"
    ],
    "img": "https://images.amcnetworks.com/amc.com/
    "status": "Presumed dead",
    "nickname": "Heisenberg",
  ▼ "appearance": [
        1,
        2,
        3,
        4,
        5
    ],
    "portrayed": "Bryan Cranston",
    "category": "Breaking Bad",
    "better_call_saul_appearance": []
},
```



```
{'appearacnce': [1, 2, 3, 4, 5],
 'birthday': '09-07-1958',
 'category': 'Breaking Bad',
 'id': 1,
 'img': 'https://images.amcnetworks.com/amc.com/wp-content/uploads
 'name': 'Walter White',
 'nickname': 'Heisenberg',
 'occupation': ['High School Chemistry Teacher', 'Meth King Pin'],
 'portrayed': 'Bryan Cranston',
 'status': 'Presumed dead'}
```

Differences
Challenges
Takeaways

- Differences
  - Search for things using the URL in the API
  - Search through a table in our code
- Challenges
  - Finding a suitable API
- Takeaways
  - How to model data classes based on API output
  - How to pull data from an API

Source

https://breakingbadapi.com/documentation