

COMS W4111-002/V002 (Spring 2022)

Introduction to Databases

Homework 3c: Programming

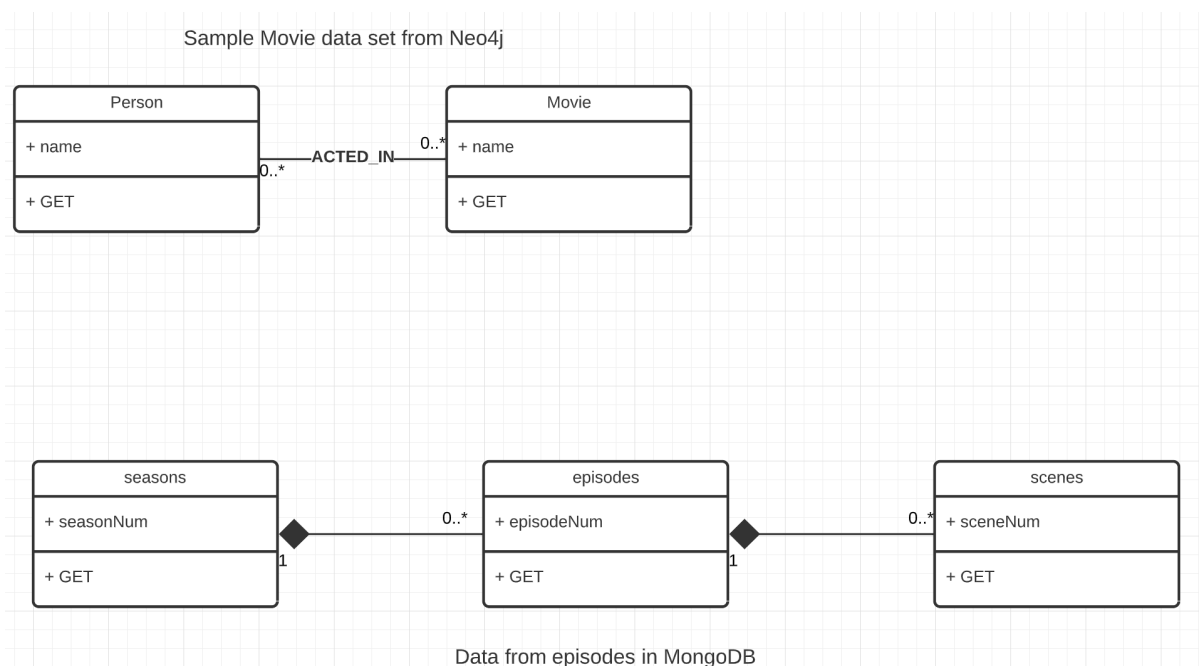
Overview

- There are three parts to HW 3:
 - 3a : Written questions
 - 3b: A set of common practical tasks.
 - 3c: Programming and Non-Programming track specific tasks.
- This notebook define part 3c Programming.

Programming Task

Resource Model/Paths

- You will extend the REST endpoint that you developed for HW2.
- You will implement the following resource model.



- This means you REST application should implement the following paths:
 - `/seasons`

- /seasons/{seasonNum}
 - /seasons/{seasonNum}/episodes
 - /seasons/seasonNum/episodes/{episodeNum}
 - /seasons/seasonNum/episodes/{episodeNum}/scenes
 - /seasons/seasonNum/episodes/{episodeNum}/scenes/sceneNum
 - /person
 - /movie
 - /person/{name}/acted_in
- Given the fact that there is "no data" in the MongoDB/JSON data for seasons, the application should simply return a list containing the episode titles and air dates.
 - You need to implement query parameter processing only for:
 - /movies
 - /episodes
 - You do not have to implement any form of project.

Suggestion

The suggested implement structure is the following:

- application.py has functions for each of the paths and calls a "resource" implementation.
- You implement a xxx_resource.py for each of the "resources." Specifically,
 - seasons_resource.py
 - episodes_resource.py
 - scenes_resource.py
 - person_resource.py
 - movie_resource.py
- Implement two data services the the above services use to interact with the backend databases, specifically:
 - mongodb_data_service.py
 - neo4j_data_service.py

Tests

- You will use requests to test your paths below.

```
In [ ]: # Tests
        # Have a cell for each test below.
        import requests
```

Path: /seasons

```
In [4]: response = requests.get('http://192.168.0.19:5003/seasons').json()
        print(response)
```

```
[{'episodeTitle': 'Winter Is Coming', 'episodeAirDate': '2011-04-17'}, {'episodeTitle': 'The Kingsroad', 'episodeAirDate': '2011-04-24'}, {'episodeTitle': 'Lord Snow', 'episodeAirDate': '2011-05-01'}, {'episodeTitle': 'Cripples, Bastards, and Broken Things', 'episodeAirDate': '2011-05-08'}, {'episodeTitle': 'The Wolf and the Lion', 'episodeAirDate': '2011-05-15'}, {'episodeTitle': 'A Golden Crown', 'episodeAirDate': '2011-05-22'}, {'episodeTitle': 'You Win or You Die', 'episodeAirDate': '2011-05-29'}, {'episodeTitle': 'The Pointy End', 'episodeAirDate': '2011-06-05'}, {'episodeTitle': 'Baelor', 'episodeAirDate': '2011-06-12'}, {'episodeTitle': 'Fire and Blood', 'episodeAirDate': '2011-06-19'}, {'episodeTitle': 'The North Remembers', 'episodeAirDate': '2012-04-01'}, {'episodeTitle': 'The Night Lands', 'episodeAirDate': '2012-04-08'}, {'episodeTitle': 'What Is Dead May Never Die', 'episodeAirDate': '2012-04-15'}, {'episodeTitle': 'Garden of Bones', 'episodeAirDate': '2012-04-22'}, {'episodeTitle': 'The Ghost of Harrenhal', 'episodeAirDate': '2012-04-29'}, {'episodeTitle': 'The Old Gods and the New', 'episodeAirDate': '2012-05-06'}, {'episodeTitle': 'A Man Without Honor', 'episodeAirDate': '2012-05-13'}, {'episodeTitle': 'The Prince of Winterfell', 'episodeAirDate': '2012-05-20'}, {'episodeTitle': 'Blackwater', 'episodeAirDate': '2012-05-27'}, {'episodeTitle': 'Valar Morghulis', 'episodeAirDate': '2012-06-03'}, {'episodeTitle': 'Valar Dohaeris', 'episodeAirDate': '2013-03-31'}, {'episodeTitle': 'Dark Wings, Dark Words', 'episodeAirDate': '2013-04-07'}, {'episodeTitle': 'Walk of Punishment', 'episodeAirDate': '2013-04-14'}, {'episodeTitle': 'And Now His Watch Is Ended', 'episodeAirDate': '2013-04-21'}, {'episodeTitle': 'Kissed by Fire', 'episodeAirDate': '2013-04-28'}, {'episodeTitle': 'The Climb', 'episodeAirDate': '2013-05-05'}, {'episodeTitle': 'The Bear and the Maiden Fair', 'episodeAirDate': '2013-05-12'}, {'episodeTitle': 'Second Sons', 'episodeAirDate': '2013-05-19'}, {'episodeTitle': 'The Rains of Castamere', 'episodeAirDate': '2013-06-02'}, {'episodeTitle': 'Mhysa', 'episodeAirDate': '2013-06-09'}, {'episodeTitle': 'Two Swords', 'episodeAirDate': '2014-04-06'}, {'episodeTitle': 'The Lion and the Rose', 'episodeAirDate': '2014-04-13'}, {'episodeTitle': 'Breaker of Chains', 'episodeAirDate': '2014-04-20'}, {'episodeTitle': 'Oathkeeper', 'episodeAirDate': '2014-04-27'}, {'episodeTitle': 'First of His Name', 'episodeAirDate': '2014-05-04'}, {'episodeTitle': 'The Laws of Gods and Men', 'episodeAirDate': '2014-05-11'}, {'episodeTitle': 'Mockingbird', 'episodeAirDate': '2014-05-18'}, {'episodeTitle': 'The Mountain and the Viper', 'episodeAirDate': '2014-06-01'}, {'episodeTitle': 'The Watchers on the Wall', 'episodeAirDate': '2014-06-08'}, {'episodeTitle': 'The Children', 'episodeAirDate': '2014-06-15'}, {'episodeTitle': 'The Wars to Come', 'episodeAirDate': '2015-03-29'}, {'episodeTitle': 'The House of Black and White', 'episodeAirDate': '2015-04-19'}, {'episodeTitle': 'High Sparrow', 'episodeAirDate': '2015-04-26'}, {'episodeTitle': 'Sons of the Harpy', 'episodeAirDate': '2015-05-03'}, {'episodeTitle': 'Kill the Boy', 'episodeAirDate': '2015-05-10'}, {'episodeTitle': 'Unbowed, Unbent, Unbroken', 'episodeAirDate': '2015-05-17'}, {'episodeTitle': 'The Gift', 'episodeAirDate': '2015-05-24'}, {'episodeTitle': 'Hardhome', 'episodeAirDate': '2015-05-31'}, {'episodeTitle': 'The Dance of Dragons', 'episodeAirDate': '2015-06-07'}, {'episodeTitle': 'Mother's Mercy', 'episodeAirDate': '2015-06-14'}, {'episodeTitle': 'The Red Woman', 'episodeAirDate': '2016-04-24'}, {'episodeTitle': 'Home', 'episodeAirDate': '2016-05-01'}, {'episodeTitle': 'Oathbreaker', 'episodeAirDate': '2016-05-08'}, {'episodeTitle': 'Book of the Stranger', 'episodeAirDate': '2016-05-15'}, {'episodeTitle': 'The Door', 'episodeAirDate': '2016-05-22'}, {'episodeTitle': 'Blood of My Blood', 'episodeAirDate': '2016-05-29'}, {'episodeTitle': 'The Broken Man', 'episodeAirDate': '2016-06-05'}, {'episodeTitle': 'No One', 'episodeAirDate': '2016-06-12'}, {'episodeTitle': 'Battle of the Bastards', 'episodeAirDate': '2016-06-19'}, {'episodeTitle': 'The Winds of Winter', 'episodeAirDate': '2016-06-26'}, {'episodeTitle': 'Dragonstone', 'episodeAirDate': '2017-07-16'}, {'episodeTitle': 'Stormborn', 'episodeAirDate': '2017-07-23'}, {'episodeTitle': 'The Queen's Justice', 'episodeAirDate': '2017-07-30'}, {'episodeTitle': 'The Spoils of War', 'episodeAirDate': '2017-08-06'}, {'episodeTitle': 'Eastwatch', 'episodeAirDate': '2017-08-13'}, {'episodeTitle': 'Beyond the Wall', 'episodeAirDate': '2017-08-20'}, {'episodeTitle': 'The Dragon and the Wolf', 'episodeAirDate': '2017-08-27'}, {'episodeTitle': 'Winterfell', 'episodeAirDate': '2019-04-14'}, {'episodeTitle': 'A Knight of the Seven Kingdoms', 'episodeAirDate': '2019-04-21'}, {'episodeTitle': 'The Long Night', 'episodeAirDate': '2019-04-28'}, {'episodeTitle': 'The Last of the Starks', 'episodeAirDate': '2019-05-05'}, {'episodeTitle': 'The Bells', 'episodeAirDate': '2019-05-12'}, {'episodeTitle': 'The Iron Throne', 'episodeAirDate': '2019-05-19'}]
```

Path: /seasons/{seasonNum}

```
In [7]: response = requests.get('http://192.168.0.19:5003/seasons/1').json()
        print(response)
```

```
[{'seasonNum': 1, 'episodeTitle': 'Winter Is Coming', 'episodeAirDate': '2011-04-17'},
 {'seasonNum': 1, 'episodeTitle': 'The Kingsroad', 'episodeAirDate': '2011-04-24'}, {'s
easonNum': 1, 'episodeTitle': 'Lord Snow', 'episodeAirDate': '2011-05-01'}, {'seasonNu
m': 1, 'episodeTitle': 'Cripples, Bastards, and Broken Things', 'episodeAirDate': '201
1-05-08'}, {'seasonNum': 1, 'episodeTitle': 'The Wolf and the Lion', 'episodeAirDate':
'2011-05-15'}, {'seasonNum': 1, 'episodeTitle': 'A Golden Crown', 'episodeAirDate': '2
011-05-22'}, {'seasonNum': 1, 'episodeTitle': 'You Win or You Die', 'episodeAirDate':
'2011-05-29'}, {'seasonNum': 1, 'episodeTitle': 'The Pointy End', 'episodeAirDate': '2
011-06-05'}, {'seasonNum': 1, 'episodeTitle': 'Baelor', 'episodeAirDate': '2011-06-1
2'}, {'seasonNum': 1, 'episodeTitle': 'Fire and Blood', 'episodeAirDate': '2011-06-1
9'}]
```

Path: /seasons/{seasonNum}/episodes

```
In [8]: response = requests.get('http://192.168.0.19:5003/seasons/1/episodes').json()
        print(response)
```

```
[{'episodeNum': 1, 'episodeTitle': 'Winter Is Coming', 'episodeAirDate': '2011-04-1
7'}, {'episodeNum': 2, 'episodeTitle': 'The Kingsroad', 'episodeAirDate': '2011-04-2
4'}, {'episodeNum': 3, 'episodeTitle': 'Lord Snow', 'episodeAirDate': '2011-05-01'},
 {'episodeNum': 4, 'episodeTitle': 'Cripples, Bastards, and Broken Things', 'episodeAir
Date': '2011-05-08'}, {'episodeNum': 5, 'episodeTitle': 'The Wolf and the Lion', 'episo
deAirDate': '2011-05-15'}, {'episodeNum': 6, 'episodeTitle': 'A Golden Crown', 'episo
deAirDate': '2011-05-22'}, {'episodeNum': 7, 'episodeTitle': 'You Win or You Die', 'epi
sodeAirDate': '2011-05-29'}, {'episodeNum': 8, 'episodeTitle': 'The Pointy End', 'epi
sodeAirDate': '2011-06-05'}, {'episodeNum': 9, 'episodeTitle': 'Baelor', 'episodeAirDa
te': '2011-06-12'}, {'episodeNum': 10, 'episodeTitle': 'Fire and Blood', 'episodeAirDa
te': '2011-06-19'}]
```

Path: /seasons/{seasonNum}/episodes/{episodeNum}

```
In [10]: response = requests.get('http://192.168.0.19:5003/seasons/1/episodes/2').json()
         print(response)
```

```
[{'episodeNum': 2, 'episodeTitle': 'The Kingsroad', 'episodeAirDate': '2011-04-24', 'e
pisodeDescription': "While Bran recovers from his fall, Ned takes only his daughters t
o King's Landing. Jon Snow goes with his uncle Benjen to The Wall. Tyrion joins the
m."}]
```

Path: /seasons/{seasonNum}/episodes/{episodeNum}/scenes

```
In [11]: response = requests.get('http://192.168.0.19:5003/seasons/1/episodes/2/scenes').json()
         print(response)
```

```
[{'episodeTitle': 'The Kingsroad', 'episodeAirDate': '2011-04-24', 'scenes': [{'sceneS
tart': '0:02:22', 'sceneEnd': '0:05:28', 'location': 'The Dothraki Sea', 'characters':
[{'name': 'Jorah Mormont'}, {'name': 'Daenerys Targaryen', 'title': 'Khaleesi'}, {'nam
e': 'Khal Drogo', 'title': 'Khal'}, {'name': 'Viserys Targaryen'}, {'name': 'Doreah'},
{'name': 'Irri'}, {'name': 'Jhiqui'}]}, {'sceneStart': '0:05:28', 'sceneEnd': '0:07:0
5', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Tyr
ion Lannister'}, {'name': 'Joffrey Baratheon', 'weapon': [{'action': 'has', 'name': "L
ion's Tooth"}]}, {'name': 'Sandor Clegane'}]}, {'sceneStart': '0:07:05', 'sceneEnd':
'0:09:18', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'nam
e': 'Tyrion Lannister'}, {'name': 'Jaime Lannister'}, {'name': 'Cersei Lannister'},
{'name': 'Tommen Baratheon'}, {'name': 'Myrcella Baratheon'}]}, {'sceneStart': '0:09:1
8', 'sceneEnd': '0:11:44', 'location': 'The North', 'subLocation': 'Winterfell', 'char
acters': [{'name': 'Catelyn Stark'}, {'name': 'Bran Stark'}, {'name': 'Cersei Lanniste
```

r']]], {'sceneStart': '0:11:44', 'sceneEnd': '0:13:26', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Jon Snow'}, {'name': 'Jaime Lannister'}]}], {'sceneStart': '0:13:26', 'sceneEnd': '0:15:43', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Arya Stark', 'weapon': [{'action': 'receives', 'name': 'Needle'}]}], {'name': 'Nymeria'}, {'name': 'Jon Snow', 'weapon': [{'action': 'gives', 'name': 'Needle'}]}]}], {'sceneStart': '0:15:43', 'sceneEnd': '0:19:22', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Catelyn Stark'}, {'name': 'Jon Snow'}, {'name': 'Bran Stark'}, {'name': 'Eddard Stark'}]}], {'sceneStart': '0:19:22', 'sceneEnd': '0:20:07', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Jon Snow'}, {'name': 'Robb Stark'}]}], {'sceneStart': '0:20:07', 'sceneEnd': '0:21:51', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Tyrion Lannister'}, {'name': 'Benjen Stark'}, {'name': 'Eddard Stark'}, {'name': 'Jon Snow'}]}], {'sceneStart': '0:21:51', 'sceneEnd': '0:25:10', 'location': 'The North', 'subLocation': 'The Kingsroad South to King's Landing', 'characters': [{'name': 'Robert Baratheon', 'title': 'King'}, {'name': 'Eddard Stark'}]}], {'sceneStart': '0:25:10', 'sceneEnd': '0:25:56', 'location': 'The Dothraki Sea', 'subLocation': 'Dothraki Camp', 'characters': [{'name': 'Khal Drogo', 'title': 'Khal', 'sex': {'with': 'Daenerys Targaryen'}, 'when': 'present', 'type': 'rape'}], {'name': 'Daenerys Targaryen', 'title': 'Khaleesi', 'sex': {'with': 'Khal Drogo'}, 'when': 'present', 'type': 'rape'}], {'name': 'Drogon', 'born': False}, {'name': 'Rhaegal', 'born': False}, {'name': 'Viserion', 'born': False}]}, {'sceneStart': '0:25:56', 'sceneEnd': '0:29:16', 'location': 'The North', 'subLocation': 'North to the Wall', 'characters': [{'name': 'Jon Snow'}, {'name': 'Ghost'}, {'name': 'Benjen Stark'}, {'name': 'Tyrion Lannister'}, {'name': 'Rast'}]}], {'sceneStart': '0:29:16', 'sceneEnd': '0:32:28', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Catelyn Stark'}, {'name': 'Bran Stark'}, {'name': 'Maester Luwin'}, {'name': 'Robb Stark'}, {'name': 'Catspaw Assassin', 'alive': False, 'mannerOfDeath': 'Mauling', 'killedBy': 'Summer', 'weapon': [{'action': 'has', 'name': 'Valyrian Steel Dagger'}]}], {'name': 'Summer'}]}], {'sceneStart': '0:32:28', 'sceneEnd': '0:34:21', 'location': 'The Dothraki Sea', 'subLocation': 'Dothraki Camp', 'characters': [{'name': 'Daenerys Targaryen', 'title': 'Khaleesi'}, {'name': 'Doreah'}, {'name': 'Irri'}, {'name': 'Jhiqui'}, {'name': 'Drogon', 'born': False}, {'name': 'Rhaegal', 'born': False}, {'name': 'Viserion', 'born': False}]}, {'sceneStart': '0:34:21', 'sceneEnd': '0:34:53', 'location': 'The Wall', 'subLocation': 'The Gift', 'characters': [{'name': 'Benjen Stark'}, {'name': 'Jon Snow'}, {'name': 'Tyrion Lannister'}]}], {'sceneStart': '0:34:53', 'sceneEnd': '0:36:12', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Catelyn Stark'}]}], {'sceneStart': '0:36:12', 'sceneEnd': '0:38:01', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Catelyn Stark'}, {'name': 'Robb Stark'}, {'name': 'Maester Luwin'}, {'name': 'Rodrik Cassel', 'weapon': [{'action': 'has', 'name': 'Valyrian Steel Dagger'}]}], {'name': 'Theon Greyjoy'}]}], {'sceneStart': '0:38:01', 'sceneEnd': '0:38:54', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Catelyn Stark'}, {'name': 'Bran Stark'}]}], {'sceneStart': '0:38:54', 'sceneEnd': '0:40:37', 'location': 'The Dothraki Sea', 'subLocation': 'Dothraki Camp', 'characters': [{'name': 'Daenerys Targaryen', 'title': 'Khaleesi'}, {'name': 'Doreah'}]}], {'sceneStart': '0:40:37', 'sceneEnd': '0:42:47', 'location': 'The Dothraki Sea', 'subLocation': 'Dothraki Camp', 'characters': [{'name': 'Drogon', 'born': False}, {'name': 'Rhaegal', 'born': False}, {'name': 'Viserion', 'born': False}, {'name': 'Khal Drogo', 'title': 'Khal', 'sex': {'with': 'Daenerys Targaryen'}, 'when': 'present', 'type': 'love'}], {'name': 'Daenerys Targaryen', 'title': 'Khaleesi', 'sex': {'with': 'Khal Drogo'}, 'when': 'present', 'type': 'love'}]}], {'sceneStart': '0:42:47', 'sceneEnd': '0:44:38', 'location': 'The Riverlands', 'subLocation': 'Crossroads Inn', 'characters': [{'name': 'Sansa Stark'}, {'name': 'Lady'}, {'name': 'Joffrey Baratheon', 'weapon': [{'action': 'has', 'name': "Lion's Tooth"}]}, {'name': 'Sandor Clegane'}, {'name': 'Ilyn Payne'}]}], {'sceneStart': '0:44:38', 'sceneEnd': '0:47:21', 'location': 'The Riverlands', 'subLocation': 'Crossroads Inn', 'characters': [{'name': 'Arya Stark'}, {'name': 'Mycah'}, {'name': 'Joffrey Baratheon', 'weapon': [{'action': 'loses', 'name': "Lion's Tooth"}]}, {'name': 'Sansa Stark'}, {'name': 'Nymeria'}]}], {'sceneStart': '0:47:21', 'sceneEnd': '0:48:12', 'location': 'The Riverlands', 'subLocation': 'Crossroads Inn', 'characters': [{'name': 'Arya Stark'}, {'name': 'Nymeria'}]}], {'sceneStart': '0:48:12', 'sceneEnd': '0:48:50', 'location': 'The Riverlands', 'subLocation': 'Crossroads Inn', 'characters': [{'name': 'Eddard Stark'}, {'name': 'Jory Cassel'}]}], {'sceneStart': '0:48:50', 'sceneEnd': '0:52:59', 'location': 'The Riverlands', 'subLocation': 'Crossroads Inn', 'characters': [{'name': 'Eddard Stark'}, {'name': 'Arya Stark'}, {'name': 'Joffrey Baratheon'}, {'name': 'Cersei Lanni

```
ster'}, {'name': 'Robert Baratheon', 'title': 'King'}, {'name': 'Ilyn Payne'}, {'name': 'Sansa Stark'}, {'name': 'Jory Cassel'}]], {'sceneStart': '0:52:59', 'sceneEnd': '0:53:30', 'location': 'The Riverlands', 'subLocation': 'Crossroads Inn', 'characters': [{'name': 'Sandor Clegane'}, {'name': 'Mycah', 'alive': False, 'mannerOfDeath': 'Horse', 'killedBy': ['Sandor Clegane']}], {'name': 'Eddard Stark'}]], {'sceneStart': '0:53:30', 'sceneEnd': '0:53:41', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Bran Stark'}, {'name': 'Summer'}]], {'sceneStart': '0:53:41', 'sceneEnd': '0:54:09', 'location': 'The Riverlands', 'subLocation': 'Crossroads Inn', 'characters': [{'name': 'Lady'}, {'name': 'Eddard Stark'}]], {'sceneStart': '0:54:09', 'sceneEnd': '0:54:25', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Bran Stark'}, {'name': 'Summer'}]], {'sceneStart': '0:54:25', 'sceneEnd': '0:54:35', 'location': 'The Riverlands', 'subLocation': 'Crossroads Inn', 'characters': [{'name': 'Eddard Stark'}, {'name': 'Lady', 'alive': False, 'mannerOfDeath': 'Stab', 'killedBy': ['Eddard Stark']}]], {'sceneStart': '0:54:35', 'sceneEnd': '0:54:44', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Bran Stark'}, {'name': 'Summer'}]]]]]]
```

Path:

/seasons/{seasonNum}/episodes/{episodeNum}/scenes/{sceneNum}



```
In [12]: response = requests.get('http://192.168.0.19:5003/seasons/1/episodes/2/scenes/3').json()
print(response)
```

```
{'sceneStart': '0:09:18', 'sceneEnd': '0:11:44', 'location': 'The North', 'subLocation': 'Winterfell', 'characters': [{'name': 'Catelyn Stark'}, {'name': 'Bran Stark'}, {'name': 'Cersei Lannister'}]}
```

Path: /person

```
In [13]: response = requests.get('http://192.168.0.19:5003/person').json()
print(response)
```

```
[{'born': 1964, 'name': 'Keanu Reeves'}, {'born': 1967, 'name': 'Carrie-Anne Moss'}, {'born': 1961, 'name': 'Laurence Fishburne'}, {'born': 1960, 'name': 'Hugo Weaving'}, {'born': 1967, 'name': 'Lilly Wachowski'}, {'born': 1965, 'name': 'Lana Wachowski'}, {'born': 1952, 'name': 'Joel Silver'}, {'born': 1978, 'name': 'Emil Eifrem'}, {'born': 1975, 'name': 'Charlize Theron'}, {'born': 1940, 'name': 'Al Pacino'}, {'born': 1944, 'name': 'Taylor Hackford'}, {'born': 1962, 'name': 'Tom Cruise'}, {'born': 1937, 'name': 'Jack Nicholson'}, {'born': 1962, 'name': 'Demi Moore'}, {'born': 1958, 'name': 'Kevin Bacon'}, {'born': 1966, 'name': 'Kiefer Sutherland'}, {'born': 1971, 'name': 'Noah Wyle'}, {'born': 1968, 'name': 'Cuba Gooding Jr.'}, {'born': 1957, 'name': 'Kevin Pollak'}, {'born': 1943, 'name': 'J.T. Walsh'}, {'born': 1967, 'name': 'James Marshall'}, {'born': 1948, 'name': 'Christopher Guest'}, {'born': 1947, 'name': 'Rob Reiner'}, {'born': 1961, 'name': 'Aaron Sorkin'}, {'born': 1957, 'name': 'Kelly McGillis'}, {'born': 1959, 'name': 'Val Kilmer'}, {'born': 1962, 'name': 'Anthony Edwards'}, {'born': 1933, 'name': 'Tom Skerritt'}, {'born': 1961, 'name': 'Meg Ryan'}, {'born': 1944, 'name': 'Tony Scott'}, {'born': 1941, 'name': 'Jim Cash'}, {'born': 1969, 'name': 'Renée Zellweger'}, {'born': 1962, 'name': 'Kelly Preston'}, {'born': 1974, 'name': 'Jerry O'Connell'}, {'born': 1970, 'name': 'Jay Mohr'}, {'born': 1961, 'name': 'Bonnie Hunt'}, {'born': 1971, 'name': 'Regina King'}, {'born': 1996, 'name': 'Jonathan Lipnicki'}, {'born': 1957, 'name': 'Cameron Crowe'}, {'born': 1970, 'name': 'River Phoenix'}, {'born': 1971, 'name': 'Corey Feldman'}, {'born': 1972, 'name': 'Wil Wheaton'}, {'born': 1966, 'name': 'John Cusack'}, {'born': 1942, 'name': 'Marshall Bell'}, {'born': 1963, 'name': 'Helen Hunt'}, {'born': 1963, 'name': 'Greg Kinnear'}, {'born': 1940, 'name': 'James L. Brooks'}, {'born': 1960, 'name': 'Annabella Sciorra'}, {'born': 1929, 'name': 'Max von Sydow'}, {'born': 1942, 'name': 'Werner Herzog'}, {'born': 1951, 'name': 'Robin Williams'}, {'born': 1956, 'name': 'Vincent Ward'}, {'born': 1970, 'name': 'Ethan Hawke'}, {'born': 1971, 'name': 'Rick Yune'}, {'born': 1940, 'name': 'James Cromwell'}, {'born': 1953, 'name': 'Scott Hicks'}, {'born': 1968, 'name': 'Parker Posey'}, {'born': 1973, 'name': 'Dave Chappelle'}, {'born': 1967, 'name': 'Steve Zahn'},
```

```
{'born': 1956, 'name': 'Tom Hanks'}, {'born': 1941, 'name': 'Nora Ephron'}, {'born': 1956, 'name': 'Rita Wilson'}, {'born': 1953, 'name': 'Bill Pullman'}, {'born': 1949, 'name': 'Victor Garber'}, {'born': 1962, 'name': 'Rosie O'Donnell'}, {'born': 1950, 'name': 'John Patrick Stanley'}, {'born': 1956, 'name': 'Nathan Lane'}, {'born': 1948, 'name': 'Billy Crystal'}, {'born': 1956, 'name': 'Carrie Fisher'}, {'born': 1949, 'name': 'Bruno Kirby'}, {'born': 1977, 'name': 'Liv Tyler'}, {'born': 1970, 'name': 'Brooke Langton'}, {'born': 1930, 'name': 'Gene Hackman'}, {'born': 1968, 'name': 'Orlando Jones'}, {'born': 1950, 'name': 'Howard Deutch'}, {'born': 1974, 'name': 'Christian Bale'}, {'born': 1954, 'name': 'Zach Grenier'}, {'born': 1931, 'name': 'Mike Nichols'}, {'born': 1930, 'name': 'Richard Harris'}, {'born': 1930, 'name': 'Clint Eastwood'}, {'born': 1947, 'name': 'Takeshi Kitano'}, {'born': 1968, 'name': 'Dina Meyer'}, {'born': 1958, 'name': 'Ice-T'}, {'born': 1953, 'name': 'Robert Longo'}, {'born': 1966, 'name': 'Halle Berry'}, {'born': 1949, 'name': 'Jim Broadbent'}, {'born': 1965, 'name': 'Tom Tykwer'}, {'born': 1969, 'name': 'David Mitchell'}, {'born': 1961, 'name': 'Stefan Arndt'}, {'born': 1939, 'name': 'Ian McKellen'}, {'born': 1976, 'name': 'Audrey Tautou'}, {'born': 1971, 'name': 'Paul Bettany'}, {'born': 1954, 'name': 'Ron Howard'}, {'born': 1981, 'name': 'Natalie Portman'}, {'born': 1946, 'name': 'Stephen Rea'}, {'born': 1940, 'name': 'John Hurt'}, {'born': 1967, 'name': 'Ben Miles'}, {'born': 1985, 'name': 'Emile Hirsch'}, {'born': 1960, 'name': 'John Goodman'}, {'born': 1946, 'name': 'Susan Sarandon'}, {'born': 1966, 'name': 'Matthew Fox'}, {'born': 1980, 'name': 'Christina Ricci'}, {'born': 1982, 'name': 'Rain'}, {'name': 'Naomie Harris'}, {'born': 1957, 'name': 'Michael Clarke Duncan'}, {'born': 1953, 'name': 'David Morse'}, {'born': 1968, 'name': 'Sam Rockwell'}, {'born': 1955, 'name': 'Gary Sinise'}, {'born': 1959, 'name': 'Patricia Clarkson'}, {'born': 1959, 'name': 'Frank Darabont'}, {'born': 1938, 'name': 'Frank Langella'}, {'born': 1969, 'name': 'Michael Sheen'}, {'born': 1960, 'name': 'Oliver Platt'}, {'born': 1944, 'name': 'Danny DeVito'}, {'born': 1965, 'name': 'John C. Reilly'}, {'born': 1950, 'name': 'Ed Harris'}, {'born': 1955, 'name': 'Bill Paxton'}, {'born': 1967, 'name': 'Philip Seymour Hoffman'}, {'born': 1943, 'name': 'Jan de Bont'}, {'born': 1951, 'name': 'Robert Zemeckis'}, {'born': 1932, 'name': 'Milos Forman'}, {'born': 1946, 'name': 'Diane Keaton'}, {'born': 1949, 'name': 'Nancy Meyers'}, {'born': 1958, 'name': 'Chris Columbus'}, {'born': 1967, 'name': 'Julia Roberts'}, {'born': 1954, 'name': 'Madonna'}, {'born': 1956, 'name': 'Geena Davis'}, {'born': 1963, 'name': 'Lori Petty'}, {'born': 1943, 'name': 'Penny Marshall'}, {'name': 'Paul Blythe'}, {'name': 'Angela Scope'}, {'name': 'Jessica Thompson'}, {'name': 'James Thompson'}]
```

Path: /movie

```
In [14]: response = requests.get('http://192.168.0.19:5003/movie').json()
print(response)
```

```
{'tagline': 'Welcome to the Real World', 'title': 'The Matrix', 'released': 1999},
{'tagline': 'Free your mind', 'title': 'The Matrix Reloaded', 'released': 2003}, {'tagline': 'Everything that has a beginning has an end', 'title': 'The Matrix Revolutions', 'released': 2003}, {'tagline': 'Evil has its winning ways', 'title': 'The Devil's Advocate', 'released': 1997}, {'tagline': 'In the heart of the nation's capital, in a courthouse of the U.S. government, one man will stop at nothing to keep his honor, and one will stop at nothing to find the truth.', 'title': 'A Few Good Men', 'released': 1992}, {'tagline': 'I feel the need, the need for speed.', 'title': 'Top Gun', 'released': 1986}, {'tagline': 'The rest of his life begins now.', 'title': 'Jerry Maguire', 'released': 2000}, {'tagline': 'For some, it's the last real taste of innocence, and the first real taste of life. But for everyone, it's the time that memories are made of.', 'title': 'Stand By Me', 'released': 1986}, {'tagline': 'A comedy from the heart that goes for the throat.', 'title': 'As Good as It Gets', 'released': 1997}, {'tagline': 'After life there is more. The end is just the beginning.', 'title': 'What Dreams May Come', 'released': 1998}, {'tagline': 'First loves last. Forever.', 'title': 'Snow Falling on Cedars', 'released': 1999}, {'tagline': 'At odds in life... in love online.', 'title': 'You've Got Mail', 'released': 1998}, {'tagline': 'What if someone you never met, someone you never saw, someone you never knew was the only someone for you?', 'title': 'Sleepless in Seattle', 'released': 1993}, {'tagline': 'A story of love, lava and burning desire.', 'title': 'Joe Versus the Volcano', 'released': 1990}, {'tagline': 'Can two friends sleep together and still love each other in the morning?', 'title': 'When Harry Met Sally', 'released': 1998}, {'tagline': 'In every life there come
```

```
s a time when that thing you dream becomes that thing you do', 'title': 'That Thing Yo
u Do', 'released': 1996}, {'tagline': 'Pain heals, Chicks dig scars... Glory lasts for
ever', 'title': 'The Replacements', 'released': 2000}, {'tagline': "Based on the extra
ordinary true story of one man's fight for freedom", 'title': 'RescueDawn', 'release
d': 2006}, {'tagline': 'Come as you are', 'title': 'The Birdcage', 'released': 1996},
{'tagline': "It's a hell of a thing, killing a man", 'title': 'Unforgiven', 'release
d': 1992}, {'tagline': 'The hottest data on earth. In the coolest head in town', 'titl
e': 'Johnny Mnemonic', 'released': 1995}, {'tagline': 'Everything is connected', 'titl
e': 'Cloud Atlas', 'released': 2012}, {'tagline': 'Break The Codes', 'title': 'The Da
Vinci Code', 'released': 2006}, {'tagline': 'Freedom! Forever!', 'title': 'V for Vende
tta', 'released': 2006}, {'tagline': 'Speed has no limits', 'title': 'Speed Racer', 'r
eleased': 2008}, {'tagline': 'Prepare to enter a secret world of assassins', 'title':
'Ninja Assassin', 'released': 2009}, {'tagline': "Walk a mile you'll never forget.",
'title': 'The Green Mile', 'released': 1999}, {'tagline': '400 million people were wai
ting for the truth.', 'title': 'Frost/Nixon', 'released': 2008}, {'tagline': "He did
n't want law. He wanted justice.", 'title': 'Hoffa', 'released': 1992}, {'tagline': 'H
ouston, we have a problem.', 'title': 'Apollo 13', 'released': 1995}, {'tagline': "Do
n't Breathe. Don't Look Back.", 'title': 'Twister', 'released': 1996}, {'tagline': 'At
the edge of the world, his journey begins.', 'title': 'Cast Away', 'released': 2000},
{'tagline': "If he's crazy, what does that make you?", 'title': "One Flew Over the Cuc
koo's Nest", 'released': 1975}, {'title': "Something's Gotta Give", 'released': 2003},
{'tagline': "One robot's 200 year journey to become an ordinary man.", 'title': 'Bicen
tennial Man', 'released': 1999}, {'tagline': "A stiff drink. A little mascara. A lot o
f nerve. Who said they couldn't bring down the Soviet empire.", 'title': "Charlie Wils
on's War", 'released': 2007}, {'tagline': 'This Holiday Season... Believe', 'title': 'T
he Polar Express', 'released': 2004}, {'tagline': 'Once in a lifetime you get a chance
to do something different.', 'title': 'A League of Their Own', 'released': 1992}]
```

Path: /person/{name}/acted_in

In [18]:

```
response = requests.get('http://192.168.0.19:5003/person/Tom Hanks/acted_in').json()
print(response)
```

```
[{'tagline': 'Houston, we have a problem.', 'title': 'Apollo 13', 'released': 1995},
{'tagline': 'At odds in life... in love on-line.', 'title': "You've Got Mail", 'releas
ed': 1998}, {'tagline': 'Once in a lifetime you get a chance to do something differen
t.', 'title': 'A League of Their Own', 'released': 1992}, {'tagline': 'A story of lov
e, lava and burning desire.', 'title': 'Joe Versus the Volcano', 'released': 1990},
{'tagline': 'In every life there comes a time when that thing you dream becomes that t
hing you do', 'title': 'That Thing You Do', 'released': 1996}, {'tagline': 'Break The
Codes', 'title': 'The Da Vinci Code', 'released': 2006}, {'tagline': 'Everything is co
nnected', 'title': 'Cloud Atlas', 'released': 2012}, {'tagline': 'At the edge of the w
orld, his journey begins.', 'title': 'Cast Away', 'released': 2000}, {'tagline': "Walk
a mile you'll never forget.", 'title': 'The Green Mile', 'released': 1999}, {'taglin
e': 'What if someone you never met, someone you never saw, someone you never knew was
the only someone for you?', 'title': 'Sleepless in Seattle', 'released': 1993}, {'tagl
ine': 'This Holiday Season... Believe', 'title': 'The Polar Express', 'released': 200
4}, {'tagline': "A stiff drink. A little mascara. A lot of nerve. Who said they could
n't bring down the Soviet empire.", 'title': "Charlie Wilson's War", 'released': 200
7}]
```

Submission Format

- The TAs will post final submission format and location.
- The submission will include:
 - A PDF of this notebook.
 - A zip file containing this notebook and all of your Python code.

- You can use the application structure from HW2 as the basis for your directory.

In []: