

[learn-co-curriculum](#) / [dsc-working-with-known-json-schemas-lab](#) Public[View license](#)

☆ 0 stars 🔗 188 forks

☆ Star

👁 Watch ▾

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

🔗 solution ▾

...

This branch is [17 commits ahead](#), [12 commits behind](#) master.[Contribute](#) ▾

hoffm386 fix objectives spacing ...

on Aug 1 ⌚ 19

[View code](#)

Working with Known JSON Schemas - Lab

Introduction

In this lab, you'll practice working with JSON files whose schema you know beforehand.

Objectives

You will be able to:

- Use the `json` module to load and parse JSON documents
- Extract data using predefined JSON schemas
- Convert JSON to a pandas dataframe

Reading a JSON Schema

Here's the JSON schema provided for a section of the NY Times API:

```
▼ {  
  status:      string  
  copyright:   string  
  num_results: integer  
  results:     ▼ [  
    ▼ {  
      display_title:  string  
      mpaa_rating:    string  
      critics_pick:   integer  
      byline:         string  
      headline:       string  
      summary_short:  string  
      publication_date: string  
      opening_date:   string  
      date_updated:   string  
      link:           ► {}  
      multimedia:     ► {}  
    }  
  ]  
}
```

or a fully expanded view:

```
▼ {
  status:      string
  copyright:   string
  num_results: integer
  results:     ▼ [
    ▼ {
      display_title: string
      mpaa_rating:   string
      critics_pick:  integer
      byline:        string
      headline:       string
      summary_short: string
      publication_date: string
      opening_date:   string
      date_updated:   string
      link:           ▼ {
        type:      string
        url:        string
        suggested_link_text: string
      }
      multimedia: ▼ {
        resource: ▼ {
          type: string
          src:  string
          height: integer
          width: integer
        }
      }
    }
  ]
}
```

You can more about the documentation [here](#).

Note that **this is a different schema than the schema used in the previous lesson**, although both come from the New York Times.

Loading the JSON Data

Open the JSON file located at `ny_times_movies.json` , and use the `json` module to load the data into a variable called `data` .

```
import json
with open('ny_times_movies.json', 'r') as f:
    data = json.load(f)
```

Run the code below to investigate its contents:

```
print("`data` has type", type(data))
print("The keys are", list(data.keys()))
```

```
`data` has type <class 'dict'>
The keys are ['status', 'copyright', 'has_more', 'num_results', 'results']
```

Loading Results

Create a variable `results` that contains the value associated with the `'results'` key.

```
results = data['results']
```

Below we display this variable as a table using pandas:

```
import pandas as pd
df = pd.DataFrame(results)
df
```

```
<style scoped> .dataframe tbody tr th:only-of-type { vertical-align: middle; }
```

```
.dataframe tbody tr th {
    vertical-align: top;
}
```

```
.dataframe thead th {
    text-align: right;
}
```

```
</style>
```

	display_title	mpaa_rating	critics_pick	byline	headline
0	Can You Ever Forgive Me	R	1	A.O. SCOTT	Review: Melissa McCarthy Is Criminally Good in...

	display_title	mpaa_rating	critics_pick	byline	headline
1	Charm City		1	BEN KENIGSBERG	Review: 'Charm City' Vividly Captures the Stre...
2	Horn from the Heart: The Paul Butterfield Story		1	GLENN KENNY	Review: Paul Butterfield's Story Is Told in 'H...
3	The Price of Everything		0	A.O. SCOTT	Review: 'The Price of Everything' Asks \$56 Bil...
4	Impulso		0	BEN KENIGSBERG	Review: 'Impulso' Goes Backstage With a Flamen...
5	Watergate		1	A.O. SCOTT	Review: 'Watergate' Shocks Anew With Its True ...
6	Barbara		1	GLENN KENNY	Review: In 'Barbara,' a Fictional Biopic of a ...
7	Over the Limit		1	JEANNETTE CATSOULIS	Review: A Russian Gymnast Goes 'Over the Limit'

	display_title	mpaa_rating	critics_pick	byline	headline
8	The Kindergarten Teacher	R	1	JEANNETTE CATSOULIS	Review: The Disturbing Obsession of 'The Kinde...
9	Classical Period		1	BEN KENIGSBURG	Review: In 'Classical Period,' a Deep Dive — R...
10	Bad Times at the El Royale	R	0	MANOHLA DARGIS	Review: Hard-Boiled Play in 'Bad Times at the ...
11	Beautiful Boy	R	0	A.O. SCOTT	Review: In 'Beautiful Boy,' a Writer Confronts...
12	The Oath	R	0	GLENN KENNY	Review: In 'The Oath,' a Pledge of Allegiance ...
13	Bikini Moon		0	KEN JAWOROWSKI	Review: 'Bikini Moon' Finds a Documentary Crew...
14	Goosebumps 2: Haunted Halloween	PG	0	TEO BUGBEE	Review: 'Goosebumps 2: Haunted Halloween' Is T...

	display_title	mpaa_rating	critics_pick	byline	headline
15	The Sentence		0	KEN JAWOROWSKI	Review: In 'The Sentence,' a Woman Gets Prison...
16	All Square		0	GLENN KENNY	Review: In 'All Square,' Taking Big Bets on Yo...
17	Sadie		0	KEN JAWOROWSKI	Review: The Drama 'Sadie' Finds a Teenager in ...
18	After Everything		0	TEO BUGBEE	Review: In 'After Everything,' a Young Love Bl...
19	First Man	PG-13	0	A.O. SCOTT	Review: 'First Man' Takes a Giant Leap for Man...

Data Analysis

Now that you have a general sense of the data, answer some questions about it.

How many results are in the file?

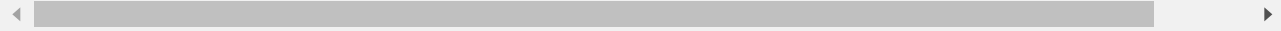
The metadata says this:

```
data['num_results']
```

20

Double-check that by looking at `results` . Does it line up?

```
print("The length of `results` is", len(results))
print("That length equals the 'num_results value?', len(results) == data['num_resul
```



```
The length of `results` is 20
That length equals the 'num_results value?' True
```

```
"""
Yes, the length of the `results` list matches the 'num_results'
reported by the metadata
"""
```

How many unique critics are there?

A critic's name can be identified using the `'byline'` key. Assign your answer to the variable `unique_critics` .

```
# Base Python solution:
unique_critics_set = set()
for result in results:
    unique_critics_set.add(result["byline"])
unique_critics = len(unique_critics_set)
```

```
# Pandas solution
unique_critics = df["byline"].nunique()
```

```
unique_critics
```

7

This code checks your answer.

```
assert unique_critics == 7
```


Flattening Data

Create a list `review_urls` that contains the URL for each review. This can be found using the `'url'` key nested under `'link'`.

```
# First, exploring the structure a bit more to make  
# sure we understand it
```

```
results[0]['link']
```

```
{'type': 'article',  
 'url': 'http://www.nytimes.com/2018/10/16/movies/can-you-ever-forgive-me-review-melissa-mccarthy.html',  
 'suggested_link_text': 'Read the New York Times Review of Can You Ever Forgive Me'}
```

```
# In base Python, we can make the list with list comprehension  
review_urls = [result['link']['url'] for result in results]  
review_urls
```

```
['http://www.nytimes.com/2018/10/16/movies/can-you-ever-forgive-me-review-melissa-mccarthy.html',  
 'http://www.nytimes.com/2018/10/16/movies/charm-city-review-baltimore.html',  
 'http://www.nytimes.com/2018/10/16/movies/horn-from-the-heart-review-paul-butterfield.html',  
 'http://www.nytimes.com/2018/10/16/movies/the-price-of-everything-review-documentary.html',  
 'http://www.nytimes.com/2018/10/16/movies/impulso-review-documentary.html',  
 'http://www.nytimes.com/2018/10/11/movies/watergate-review-documentary.html',  
 'http://www.nytimes.com/2018/10/11/movies/barbara-review.html',  
 'http://www.nytimes.com/2018/10/11/movies/over-the-limit-review.html',  
 'http://www.nytimes.com/2018/10/11/movies/the-kindergarten-teacher-review.html',  
 'http://www.nytimes.com/2018/10/11/movies/classical-period-review.html',  
 'http://www.nytimes.com/2018/10/11/movies/bad-times-at-the-el-royale-review.html',  
 'http://www.nytimes.com/2018/10/11/movies/beautiful-boy-review-steve-carell.html',  
 'http://www.nytimes.com/2018/10/11/movies/the-oath-review-tiffany-haddish.html',  
 'http://www.nytimes.com/2018/10/11/movies/bikini-moon-review.html',  
 'http://www.nytimes.com/2018/10/11/movies/goosebumps-2-haunted-halloween-review.html',  
 'http://www.nytimes.com/2018/10/11/movies/the-sentence-review.html',  
 'http://www.nytimes.com/2018/10/11/movies/all-square-review.html',
```

```
'http://www.nytimes.com/2018/10/11/movies/sadie-review.html',  
'http://www.nytimes.com/2018/10/11/movies/after-everything-review.html',  
'http://www.nytimes.com/2018/10/10/movies/first-man-review-ryan-gosling-damien-  
chazelle.html']
```

Alternatively, we can use pandas with a lambda function

☰ README.md

```
['http://www.nytimes.com/2018/10/16/movies/can-you-ever-forgive-me-review-  
melissa-mccarthy.html',  
'http://www.nytimes.com/2018/10/16/movies/charm-city-review-baltimore.html',  
'http://www.nytimes.com/2018/10/16/movies/horn-from-the-heart-review-paul-  
butterfield.html',  
'http://www.nytimes.com/2018/10/16/movies/the-price-of-everything-review-  
documentary.html',  
'http://www.nytimes.com/2018/10/16/movies/impulso-review-documentary.html',  
'http://www.nytimes.com/2018/10/11/movies/watergate-review-documentary.html',  
'http://www.nytimes.com/2018/10/11/movies/barbara-review.html',  
'http://www.nytimes.com/2018/10/11/movies/over-the-limit-review.html',  
'http://www.nytimes.com/2018/10/11/movies/the-kindergarten-teacher-review.html',  
'http://www.nytimes.com/2018/10/11/movies/classical-period-review.html',  
'http://www.nytimes.com/2018/10/11/movies/bad-times-at-the-el-royale-  
review.html',  
'http://www.nytimes.com/2018/10/11/movies/beautiful-boy-review-steve-  
carell.html',  
'http://www.nytimes.com/2018/10/11/movies/the-oath-review-tiffany-haddish.html',  
'http://www.nytimes.com/2018/10/11/movies/bikini-moon-review.html',  
'http://www.nytimes.com/2018/10/11/movies/goosebumps-2-haunted-halloween-  
review.html',  
'http://www.nytimes.com/2018/10/11/movies/the-sentence-review.html',  
'http://www.nytimes.com/2018/10/11/movies/all-square-review.html',  
'http://www.nytimes.com/2018/10/11/movies/sadie-review.html',  
'http://www.nytimes.com/2018/10/11/movies/after-everything-review.html',  
'http://www.nytimes.com/2018/10/10/movies/first-man-review-ryan-gosling-damien-  
chazelle.html']
```

The following code will check your answer:

```
# review_urls should be a list  
assert type(review_urls) == list  
  
# The length should be 20, same as the length of reviews  
assert len(review_urls) == 20
```



```
# The data type contained should be string
assert type(review_urls[0]) == str and type(review_urls[-1]) == str

# Spot checking a specific value
assert review_urls[6] == 'http://www.nytimes.com/2018/10/11/movies/barbara-review.ht
```

Summary

Well done! In this lab you continued to practice extracting and transforming data from JSON files with known schemas.

Releases

No releases published

Packages

No packages published

Contributors 5



Languages

● Jupyter Notebook 100.0%