

API homework assignment.

Using an automation framework of your choice.

1. Share a repository with your coding challenge (You should be able to set up a free account on github, bitbucket, sourceforge, etc. to share your coding challenge.)
2. Name it with your "firstname_lastname_teladoc_challenge"
3. Update your README to show how to run your scripts

This assignment will contain two problems:

1. Counting Movies - a basic json parsing exercise
2. Movie Titles - a more advanced json parsing exercise with paging

The details for each homework assignment are as follows:

Counting Movies

In this challenge, you will write an *HTTP GET* method to retrieve information from a movie database. You will be given a search term, and your function must return the value of the *total* field in the returned JSON object.

Function Description

Complete the function *getNumberOfMovies*. The function must return the value of the *total* field in the returned JSON object.

getNumberOfMovies has the following parameter(s):

substr: the string to search for in the movie database

It must query *https://jsonmock.hackerrank.com/api/movies/search/?Title=[substr]* and return the total number of movie titles having the substring *substr* in their title. The query response is a JSON object with the following five fields:

- *page*: The current page.
- *per_page*: The maximum number of results per page.

- *total*: The total number of movies having the substring *substr* in their title.
- *total_pages*: The total number of pages which must be queried to get all the results.
- *data*: An array of JSON objects containing movie information where the *Title* field denotes the title of the movie.

Constraints

- $0 < |substr| < 20$

Sample Input

```
maze
```

Sample Output

```
97
```

Explanation

The value of *substr* is *maze*, so our query is

<https://jsonmock.hackerrank.com/api/movies/search/?Title=maze> and the response is:

```
{
  page:1,
  per_page:10,
  total:97,
  total_pages:10,
  data:[
    {
      Poster:"https://images-na.ssl-images-
amazon.com/images/M/MV5BMjUyNTA3MTAyM15BMl5BanBnXkFtZTgwOTEyMTkyMjE@._V1_S
X300.jpg",
      Title:"The Maze Runner",
      Type:"movie",
      Year:2014,
      imdbID:"tt1790864"
    },
    {
```

```
      Poster:"https://images-na.ssl-images-
amazon.com/images/M/MV5BMjE3MDU2NzQyMl5BMl5BanBnXkFtZTgwMzQxMDQ3NTE@._V1_S
X300.jpg",
      Title:"Maze Runner: The Scorch Trials",
      Type:"movie",
      Year:2015,
      imdbID:"tt4046784"
    },
    {
      Poster:"https://images-na.ssl-images-
amazon.com/images/M/MV5BMjExOTkxMTIzNl5BMl5BanBnXkFtZTgwNjcxNzY2NTE@._V1_S
X300.jpg",
      Title:"Into the Grizzly Maze",
      Type:"movie",
      Year:2015,
      imdbID:"tt1694021"
    },
    {
      Poster:"https://images-na.ssl-images-
amazon.com/images/M/MV5BMTIwNDg4MjIyMV5BMl5BanBnXkFtZTcwNDEwMzkxMQ@@._V1_S
X300.jpg",
      Title:"Hercules in the Maze of the Minotaur",
      Type:"movie",
      Year:1994,
      imdbID:"tt0110018"
    },
    {
      Poster:"http://ia.media-
imdb.com/images/M/MV5BNTRkNzI4YTktNTBmMi00OWNlLWExODctMTA0ODdiNDNjZTkzXkEy
XkFqcGdeQXVyMTk3MTY0NDc@._V1_SX300.jpg",
      Title:"The Crystal Maze",
      Type:"series",
      Year:1990,
      imdbID:"tt0098774"
    },
    {
      Poster:"https://images-na.ssl-images-
amazon.com/images/M/MV5BMTk0MTAxODI3Nl5BMl5BanBnXkFtZTcwMjI1NTk5Mw@@._V1_S
X300.jpg",
      Title:"The Maze",
```

```

        Type:"movie",
        Year:2010,
        imdbID:"tt1675758"
    },
    {
        Poster:"https://images-na.ssl-images-
amazon.com/images/M/MV5BMTkwNzE5MTA1NF5BMl5BanBnXkFtZTcwODQ3NzkxMQ@@._V1_S
X300.jpg",
        Title:"Maze",
        Type:"movie",
        Year:2000,
        imdbID:"tt0246072"
    },
    {
        Poster:"https://images-na.ssl-images-
amazon.com/images/M/MV5BNDY4NDAXNjgzOF5BMl5BanBnXkFtZTcwNDEyMjAzMQ@@._V1_S
X300.jpg",
        Title:"Iron Maze",
        Type:"movie",
        Year:1991,
        imdbID:"tt0102128"
    },
    {
        Poster:"https://images-na.ssl-images-
amazon.com/images/M/MV5BMTg2Mjk1MDE3OF5BMl5BanBnXkFtZTcwOTk2OTYxMg@@._V1_S
X300.jpg",
        Title:"The Maze",
        Type:"movie",
        Year:1953,
        imdbID:"tt0046057"
    },
    {
        Poster:"http://ia.media-
imdb.com/images/M/MV5BNmI4YWUyNzctYjcyYy00OTdkLTljOTgtOGQxNDJhNjNhODZkXkEy
XkFqcGdeQXVyNTUzNzE0ODE@._V1_SX300.jpg",
        Title:"Maze Runner: The Burn Trials",
        Type:"movie",
        Year:2015,
        imdbID:"tt4844320"
    }
}

```

```
]
}
```

We then return the value of the *total* field as our answer, 97.

Movie Titles

To solve this challenge, you are required to write an *HTTP GET* method to retrieve information from a movie database.

Function Description

Given a string *substr*, *getMovieTitles* must perform the following tasks:

1. Query
`https://jsonmock.hackerrank.com/api/movies/search/?Title=substr`(replace *substr*).
2. Initialize the *titles* array to store `total` string elements. Store the `Title` of each movie meeting the search criteria in the *titles* array.
3. Sort *titles* in ascending order and return it as your answer.

The query response from the website is a JSON response with the following five fields:

- `page`: The current page.
- `per_page`: The maximum number of results per page.
- `total`: The total number of movies in the search result.
- `total_pages`: The total number of pages which must be queried to get all the results.
- `data`: An array of JSON objects containing movie information where the `Title` field denotes the title of the movie.

In order to get all results, you may have to make multiple page requests. To request a page by number, your query should read

`https://jsonmock.hackerrank.com/api/movies/search/?Title=substr&page=pageNumber`, replacing *substr* and *pageNumber*.

Sample Input 0

```
spiderman
```

Sample Output

```
Amazing Spiderman Syndrome
```

Fighting, Flying and Driving: The Stunts of Spiderman 3
Hollywood's Master Storytellers: Spiderman Live
Italian Spiderman
Spiderman
Spiderman
Spiderman 5
Spiderman and Grandma
Spiderman in Cannes
Superman, Spiderman or Batman
The Amazing Spiderman T4 Premiere Special
The Death of Spiderman
They Call Me Spiderman

Explanation

For this example, we want *all* the movie titles containing the substring `spiderman`. The response for the query

`https://jsonmock.hackerrank.com/api/movies/search/?Title=spiderman&page=1` is:

```
{
  "page": "1",
  "per_page": 10,
  "total": 13,
  "total_pages": 2,
  "data": [
    {
      "Poster": "https://images-na.ssl-images-
amazon.com/images/M/MV5BYjFhN2RjZTctMzA2Ni00NzE2LWJmYjMtNDAYYTl1OTkyMmY3Xk
EyXkFqcGdeQXVyNTA0OTU0OTQ@._V1_SX300.jpg",
      "Title": "Italian Spiderman",
      "Type": "movie",
      "Year": 2007,
      "imdbID": "tt2705436"
    },
    {
```

```
    "Poster": "https://images-na.ssl-images-  
amazon.com/images/M/MV5BMjQ4MzcxNDU3N15BMl5BanBnXkFtZTgwOTE1MzMxNzE@._V1_S  
X300.jpg",  
    "Title": "Superman, Spiderman or Batman",  
    "Type": "movie",  
    "Year": 2011,  
    "imdbID": "tt2084949"  
  },  
  {  
    "Poster": "N/A",  
    "Title": "Spiderman",  
    "Type": "movie",  
    "Year": 1990,  
    "imdbID": "tt0100669"  
  },  
  {  
    "Poster": "N/A",  
    "Title": "Spiderman",  
    "Type": "movie",  
    "Year": 2010,  
    "imdbID": "tt1785572"  
  },  
  {  
    "Poster": "N/A",  
    "Title": "Fighting, Flying and Driving: The Stunts of Spiderman 3",  
    "Type": "movie",  
    "Year": 2007,  
    "imdbID": "tt1132238"  
  },  
  {  
    "Poster": "http://ia.media-  
imdb.com/images/M/MV5BMjE3Mzg0MjAxMl5BMl5BanBnXkFtZTcwNjIyODg5Mg@@._V1_SX3  
00.jpg",  
    "Title": "Spiderman and Grandma",  
    "Type": "movie",  
    "Year": 2009,
```



```
    "imdbID": "tt1433184"
  },
  {
    "Poster": "N/A",
    "Title": "The Amazing Spiderman T4 Premiere Special",
    "Type": "movie",
    "Year": 2012,
    "imdbID": "tt2233044"
  },
  {
    "Poster": "N/A",
    "Title": "Amazing Spiderman Syndrome",
    "Type": "movie",
    "Year": 2012,
    "imdbID": "tt2586634"
  },
  {
    "Poster": "N/A",
    "Title": "Hollywood's Master Storytellers: Spiderman Live",
    "Type": "movie",
    "Year": 2006,
    "imdbID": "tt2158533"
  },
  {
    "Poster": "N/A",
    "Title": "Spiderman 5",
    "Type": "movie",
    "Year": 2008,
    "imdbID": "tt3696826"
  }
]
}
```

The response for the query

<https://jsonmock.hackerrank.com/api/movies/search/?Title=spiderman&page=2> is:

```
{
  "page": "2",
  "per_page": 10,
  "total": 13,
  "total_pages": 2,
  "data": [
    {
      "Poster": "N/A",
      "Title": "They Call Me Spiderman",
      "Type": "movie",
      "Year": 2016,
      "imdbID": "tt5861236"
    },
    {
      "Poster": "N/A",
      "Title": "The Death of Spiderman",
      "Type": "movie",
      "Year": 2015,
      "imdbID": "tt5921428"
    },
    {
      "Poster": "https://images-na.ssl-images-
amazon.com/images/M/MV5BZDlmMGQwYmItNTNmOS00OTNkLTkxNTYtNDM3ZWVlMWUyZDIzXk
EyXkFqcGdeQXVyMTA5Mzk5Mw@@._V1_SX300.jpg",
      "Title": "Spiderman in Cannes",
      "Type": "movie",
      "Year": 2016,
      "imdbID": "tt5978586"
    }
  ]
}
```

The values of the Title field for each movie in each response page in the order received are:

Italian Spiderman
Superman, Spiderman or Batman
Spiderman
Spiderman
Fighting, Flying and Driving: The Stunts of Spiderman 3
Spiderman and Grandma
The Amazing Spiderman T4 Premiere Special
Amazing Spiderman Syndrome
Hollywood's Master Storytellers: Spiderman Live
Spiderman 5
They Call Me Spiderman
The Death of Spiderman
Spiderman in Cannes

We then sort the array in ascending order, and return it as our answer.