

3. JavaScript: Movie Titles List

In this challenge, the given REST API contains information about movies.

Given a year in the format YYYY, make a GET call to the API to get all the movies for this year.

To get the data, make an API GET call to the URL

'<https://jsonmock.hackerrank.com/api/movies?Year={year}>'. For example, for the year 2015, the API hit has to be <https://jsonmock.hackerrank.com/api/movies?Year=2015>.

The response to such a request is a JSON record with the following 5 fields:

- *page*: the current page of the results
- *per_page*: the maximum number of results returned per page
- *total*: the total number of results
- *total_pages*: the total number of pages with results
- *data*: an array of objects containing the desired data. Each object is a movie record having the below schema:
 - *Title* - Movie title

- *Year* - Movie year (same as the year that was queried)
- *imdbID* - Movie ID

Here is an example of a movie record:

```
{  
  "Title": "Harry & Snowman",  
  "Year": 2015,  
  "imdbID": "tt2898306"  
}
```

For each movie record, you only need the *'Title'* value. Return the data as an array of strings, where each string is a movie title. Keep the order of movies the same as returned by the API.

Please note that you only need the data from page 1. Page 1 is the default page returned on an API hit. No further page hits are required.

Function Description

Complete the function *getMovieList* in the editor below.

getMovieList has the following parameter:

int *year*: the year number you want to query

Returns:

a Promise that resolves with an array of movie titles, in the same order as returned by the API; in the case of empty data array result, the Promise resolves with an empty array

Your implementation of the function will be tested by a stubbed code on several input files. Each input file contains a year parameter for the functions call. The function *getMovieList* will be called with this parameter, and the result of their executions will be printed to the standard output by the provided code. The stubbed code prints the movie titles strings as returned by API. In case the function resolves the promise with an empty array, the stubbed code prints "No Results Found".

▼ Input Format For Custom Testing

In the first and only line, there is a year number.

▼ Sample Case 0

Sample Input For Custom Testing

```
2012
```

Sample Output

```
The Amazing Spiderman T4 Premiere Special  
Amazing Spiderman Syndrome  
The Road to Ironman  
Ironman Triathlon Special  
Boundless Maze
```

```
Maze
Harry Dean Stanton: Partly Fiction
Harry Potter and the Ten Years Later
Ain't It Cool with Harry Knowles
Creating the World of Harry Potter, Part 8: Growing Up
```

Explanation

The list of movie titles for the year 2012 is printed as returned by the API.

▼ Sample Case 1

Sample Input For Custom Testing

```
2010
```

Sample Output

```
Spiderman
The Maze
Navigating the Maze
A Place in the Maze
The Black and White Maze of the Painted Zebra
Harry Potter and the Deathly Hallows: Part 1
Harry Potter and the Forbidden Journey
Who Is Harry Nilsson (And Why Is Everybody Talkin' About Him?)
Lego Harry Potter: Years 1-4
I Am Harry Potter
```

Explanation

The list of movie titles for the year 2010 is printed as returned by the API.