



You can view this report online at : <https://www.hackerrank.com/x/tests/844637/candidates/28014267/report>

<b>Full Name:</b>	Adam	<b>100%</b> <b>75/75</b>	scored in <b>6nomads Java Developer Hiring Test</b> in 57 min 42 sec on 27 Jul 2021 00:21:29 +03
<b>Email:</b>	adam.intae.gerard@protonmail.com		
<b>Test Name:</b>	<b>6nomads Java Developer Hiring Test</b>		
<b>Taken On:</b>	27 Jul 2021 00:21:29 +03		
<b>Time Taken:</b>	57 min 42 sec/ 95 min		
<b>Work Experience:</b>	> 5 years		
<b>Contact Number:</b>	+12066125571		
<b>Resume:</b>	<a href="https://hackerrank-resumes.s3.amazonaws.com/1267170/FOg0v8yeq4WNTvge2pqsSwFB9qM9TBCo2r9A02hmO0dwMNERPsSZwkSZWEY89u2ebw/Adam_In_Tae_Gerard.pdf">https://hackerrank-resumes.s3.amazonaws.com/1267170/FOg0v8yeq4WNTvge2pqsSwFB9qM9TBCo2r9A02hmO0dwMNERPsSZwkSZWEY89u2ebw/Adam_In_Tae_Gerard.pdf</a>		
<b>Linkedin:</b>	<a href="http://www.linkedin.com/in/adamintaegerard">http://www.linkedin.com/in/adamintaegerard</a>		
<b>Invited by:</b>	Denis		
<b>Invited on:</b>	25 Jul 2021 03:57:56 +03		
<b>Skills Score:</b>	REST API (Intermediate) 75/75		
<b>Tags Score:</b>	Back-End Development 75/75		
	JSON 75/75		
	Medium 75/75		
	REST API 75/75		

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	<b>Most Active Authors</b> > Coding	50 min 8 sec	75/ 75	

**QUESTION 1**

Correct Answer

---

Score 75

**Most Active Authors** > Coding REST API Back-End Development Medium JSON

QUESTION DESCRIPTION

In this challenge, the REST API contains information about a collection of users and articles they created. Given the threshold value, the goal is to use the API to get the list of most active authors. Specifically, the list of usernames of users with submission count strictly greater than the given threshold. The list of usernames must be returned in the order the users appear in the results.

To access the collection of users perform HTTP GET request to:  
[https://jsonmock.hackerrank.com/api/article\\_users?page=<pageNumber>](https://jsonmock.hackerrank.com/api/article_users?page=<pageNumber>)

where `<pageNumber>` is an integer denoting the page of the results to return.

For example, GET request to:

`https://jsonmock.hackerrank.com/api/article_users/search?page=2`

will return the second page of the collection of users. Pages are numbered from 1, so in order to access the first page, you need to ask for page number 1.

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

- `page`: The current page of the results
- `per_page`: The maximum number of users returned per page.
- `total`: The total number of users on all pages of the result.
- `total_pages`: The total number of pages with results.
- `data`: An array of objects containing users returned on the requested page

Each user record has the following schema:

- `id`: unique ID of the user
- `username`: the username of the user
- `about`: the about information of the user
- `submitted`: total number of articles submitted by the user
- `updated_at`: the date and time of the last update to this record
- `submission_count`: the number of submitted articles that are approved
- `comment_count`: the total number of comments the user made
- `created_at`: the date and time when the record was created

### Function Description

Complete the function `getUsernames` in the editor below.

`getUsernames` has the following parameter(s):

*threshold*: integer denoting the threshold value for the number of submission count

The function must return an array of strings denoting the usernames of the users whose submission count is strictly greater than the given threshold. The usernames in the array must be ordered in the order they appear in the API response.

#### ▼ Input Format For Custom Testing

In the first line, there is an integer *threshold*.

#### ▼ Sample Case 0

##### Sample Input For Custom Testing

```
10
```

##### Sample Output

```
epaga
panny
olalonde
WisNorCan
dmmalam
replicatorblog
vladikoff
mpweiher
coloneltcb
guelo
```

##### Explanation

The threshold value is 10, so the result must contain usernames of users with the submission count

The threshold value is 10, so the result must contain usernames of users with the submission count value greater than 10. There are 10 such users and their usernames in the order they first appear in the API response are as listed in Sample Output.

#### INTERVIEWER GUIDELINES

#### Python 3 Solution

```
import requests
import json

def getUsernames(threshold):
    # Write your code here
    url = 'https://jsonmock.hackerrank.com/api/article_users/search?page={number}'
    page = 1
    pages = 2
    users = []

    while page <= pages:
        req = requests.get(url.replace('{number}', str(page)))
        json_data = json.loads(req.content.decode('utf-8'))
        if page == 1:
            pages = json_data['total_pages']
            for user in json_data['data']:
                if user['submission_count'] > threshold and user['username']
not in users:
                    users.append(user['username'])
            page += 1

    return users
```

#### Python 2 Solution

```
import json
from urllib2 import urlopen

def getUsernames(threshold):
    def get_url(page=1):
        url = 'https://jsonmock.hackerrank.com/api/article_users'
        return url + '?page={}'.format(page)

    def good_entry(entry):
        return entry['submission_count'] > threshold

    mem = set()
    users = []
    page = 1
    while True:
        url = get_url(page)
        r = json.loads(urlopen(url).read())

        for user in filter(good_entry, r['data']):
            username = user['username']
            if username not in mem:
                mem.add(username)
                users.append(username)
        page += 1
        if page > r['total_pages']:
            break
    return users
```

Language used: JavaScript (Node.js)

```
1  /*
2  * Complete the 'getUsernames' function below.
3  *
4  * The function is expected to return a STRING_ARRAY.
5  * The function accepts INTEGER threshold as parameter.
6  *
7  * URL for cut and paste
8  * https://jsonmock.hackerrank.com/api/article_users?page=<pageNumber>
9  */
10
11 // Used API documentation for v16.5.0:
12 https://nodejs.org/api/https.html#https_https_request_options_callback
13 const H = require('https')
14
15 const getUsernames = threshold => new Promise((resolve, reject) => {
16     let M = {}, page = 1
17
18     // Initial query starting at page 1
19     rqst(makeUrl(page)).then(success => {
20         const J = JSON.parse(success), total_pages = J.total_pages, data =
21 J.data
22         //console.log(total_pages)
23         //console.log(data)
24
25         for (let i = 0; i < data.length; i++) {
26             // Add to hashmap
27             const D = data[i]
28             if (M[D.username]) M[D.username] += D.submission_count
29             else M[D.username] = D.submission_count
30         }
31
32         // Follow up queries for all remaining pages
33         while (page < total_pages) {
34             page++
35
36             rqst(makeUrl(page)).then(innerSuccess => {
37                 const INNER_J = JSON.parse(innerSuccess), data = INNER_J.data
38
39                 for (let i = 0; i < data.length; i++) {
40                     // Add to hashmap
41                     const D = data[i]
42                     if (M[D.username]) M[D.username] += D.submission_count
43                     else M[D.username] = D.submission_count
44                 }
45
46                 if (page === total_pages) {
47                     const K = Object.keys(M)
48                     let result = []
49
50                     //console.log(M)
51
52                     // Filter aggregated results - keys are in order
53                     for (let i = 0; i < K.length; i++) {
54                         const D = M[K[i]]
55                         if (D > threshold) result.push(K[i])
56                     }
57
58                     return resolve(result)
59                 }
60             })
61         }
62     })
63 }
```

```

61     }
62   })
63 })
64
65 const rqst = url => new Promise((resolve, reject) => {
66   return H.get(url, response => {
67     response.on("data", data => resolve(data.toString()))
68   })
69 })
70
const makeUrl = pageNumber =>
`https://jsonmock.hackerrank.com/api/article_users?page=${pageNumber}`

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Sample case	✔ Success	1	0.2846 sec	35.7 KB
TestCase 1	Easy	Sample case	✔ Success	1	0.3349 sec	35.8 KB
TestCase 2	Easy	Hidden case	✔ Success	3	0.2896 sec	35.7 KB
TestCase 3	Easy	Hidden case	✔ Success	3	0.2953 sec	36 KB
TestCase 4	Easy	Hidden case	✔ Success	3	0.2963 sec	35.9 KB
TestCase 5	Easy	Hidden case	✔ Success	4	0.2959 sec	35.6 KB
TestCase 6	Easy	Hidden case	✔ Success	10	0.2847 sec	35.8 KB
TestCase 7	Easy	Hidden case	✔ Success	10	0.3088 sec	36 KB
TestCase 8	Easy	Hidden case	✔ Success	10	0.2892 sec	36.1 KB
TestCase 9	Easy	Hidden case	✔ Success	10	0.3064 sec	35.9 KB
TestCase 10	Easy	Hidden case	✔ Success	10	0.2812 sec	35.9 KB
TestCase 11	Easy	Hidden case	✔ Success	10	0.2823 sec	35.9 KB

No Comments