geektrust.in

Job Adventures. For Geeks.

My code is my resume

"Geektrust has tie ups with some of the best startups. And all one got to do is write code, the rest is taken care by the geektrust team."

- Athira, now works at Sahaj Soft

Athira, Souranil and many more developers have solved Geektrust coding challenges to find great jobs.

- ✴ Want feedback on your coding skills and understand how valued your skills will be in the job market?
- ✴ Want to be treated as a premium candidate and directly connect with decision makers at kick-ass companies?
- ✴ Want to know which companies would be interested in you without even creating a resume?

Solve a Geektrust coding challenge and we'll give you handwritten, personal feedback on your skills. Even if you're not looking for a job.

What we look for in your code - It's not just about getting output, but how you get it. We care about how well modelled your code is, how readable, extensible, well tested it is. Have questions on the challenges or our evaluation? Ping us on the Geektrust Slack channel.

PROBLEM 1: WHATFLIX

King Shan is very fond of movies and wants to take his family for a movie. However, they can’t agree on which movie to watch. So King Shan declares a movie watching spree and takes each person to watch the movie they want. Your coding challenge is to:

1. Build a Restful web service API which will accept a search string and userID and return unique movies in the order of preference for that user.

2. Build a Restful web service API which will list out all the users and the top 3 recommended movies for each user based on their preferences

Develop the application in a language of your choice and host the service in Cloud infrastructure like AWS/Heroku and give us the url endpoints or a Postman project for the same.

API 1: INPUT

Request Type : GET

URL : http://<url>/movies/user/$userId/search?text=<text>

Where the $userId is the id of the user from the user preferences JSON file (see file) and <text> is the search text. The search text could be multiple words separated by comma, in which case it will search for all of those. Or it could be a single urlencoded entry. The search text is matched against actor, director and title fields (director_name, actor_1_name, actor_2_name, actor_3_name and movie_title in the movies data) of the movies. All matches are included in the results.

<url> is the domain name in the URL where you are hosting the solution.

For example - http://<url>/movies/user/$userId/search?text=Tom%20Hanks -> This will return all the movies matching “Tom Hanks” considering the preferences of the user

API 1: OUTPUT

An array of movies name, found based on the preferences of users, sorted in this order:

1. First show the movies matching the user’s preferences and search term. This should be further sorted on the alphabetic order of the titles. There is a chance this set of movies could be empty if there is no search result matching the user’s preferences.

2. Next show the movies matching the search term in the alphabetic order of titles (even if it does not match the user’s preferences). If #1 is empty then only these set of movies will be there.

[“Movie 1”,”Movie 2”,”Movie 3”]

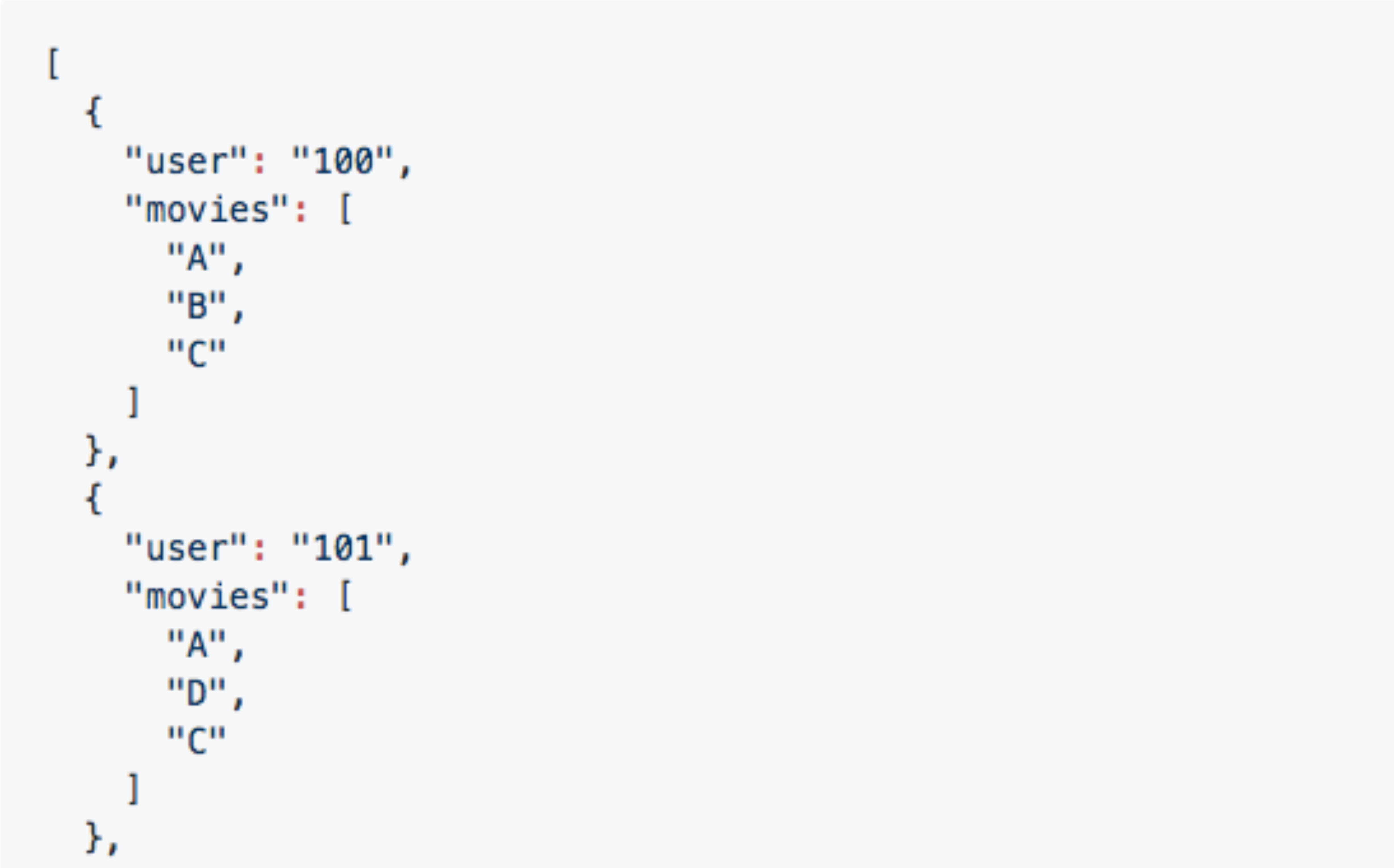
API 2: INPUT

Request Type : GET

URL : http://<url>/movies/users

API 2: OUTPUT

A json array of userids and top 3 movies recommended. The movie names are sorted in the alphabetic order of titles. If there is no recommendations then it can be an empty array.



```
[
  {
    "user": "100",
    "movies": [
      "A",
      "B",
      "C"
    ]
  },
  {
    "user": "101",
    "movies": [
      "A",
      "D",
      "C"
    ]
  },
```

And so on and so forth.

DATA SETS

We use the Kaggle TMDB data set for this problem. Complete movie data set can be downloaded from here.

The user preferences data file can also be downloaded from here.

WHAT WE LOOK FOR IN YOUR CODE

1. Overall architecture. Please submit documentation to explain your choices and the complete architecture.
2. Code scalability. Can it handle 1000s of hits a second?
3. Clean code, RESTful semantics, and object modelling
4. Bonus points for automating the deployment of your solution
5. When the instructions are not clear or there are multiple solution possibilities, make a choice that you find reasonable and practical.

SUBMITTING CODE

1. Please compress the file before upload. We accept .zip, .rar, .gz and .gzip

2. Name of the file should have the name of the problems you have solved TechArchitectSet1.zip in this case

3. Upload the file in a way that makes it easy for us to get it running. This will factor into your evaluation.

4. We advise not to put your personal details in your solution as we maintain your anonymity with a company until there is genuine interest from them.

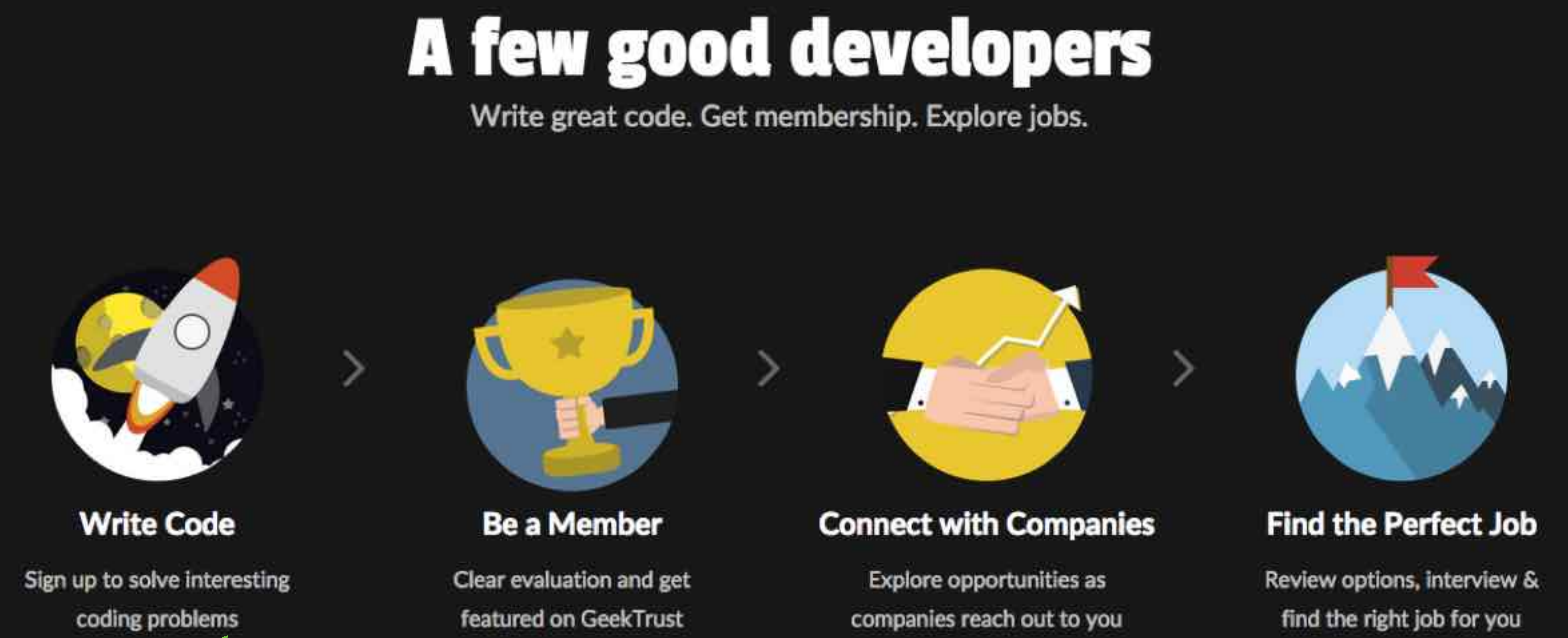
NEED HELP?

1. We know it’s possible you may have more questions on the coding challenges. So we’re available on our Slack chat channel for you. You can come in and ask questions and get all the support you need.

2. Just send a mail to devs@geektrust.in saying ‘Add to Slack’ and we’ll send you an invite to join our channel, and we will help you out.

3. *You can also submit incomplete solutions if you want feedback to make sure you’re going in the right direction. Just make sure to add INCOMPLETE as a prefix to the filename you’re uploading e.g - INCOMPLETE-problem12.zip*

WHAT NEXT?



A few good developers

Write great code. Get membership. Explore jobs.

Write Code

Sign up to solve interesting coding problems

✓

Be a Member

Clear evaluation and get featured on GeekTrust

Connect with Companies

Explore opportunities as companies reach out to you

Find the Perfect Job

Review options, interview & find the right job for you