

## Full Stack Developer - technical task

### Scenario:

We have a prototype to deliver. This prototype will retrieve and display stored JobAd objects.

Each JobAd object has:

- its own contents
- the nr. of job applicants who have applied
- a collection of biases
  - This is a collection of key/value pairs which consist of a bias name as the key, and a bias score as the value (see JSON representations below).

### JSON representations

JSON for job ad 14:

```
{ "ad_number": 14,  
  "contents": "Looking for a qualified applicant.",  
  "applicants": 12,  
  "biases": { "gender": 2, "ethnicity": 5, "age": 2 }  
}
```

JSON for job ad 15:

```
{ "ad_number": 15,  
  "contents": "Looking for a high-pressure environment.",  
  "applicants": 12,  
  "biases": { "gender": 20, "ethnicity": 50, "age": 25 }  
}
```

JSON for job ad 16:

```
{ "ad_number": 16,  
  "contents": "Looking for a helpful individual.",  
  "applicants": 12,  
  "biases": { "gender": 50, "ethnicity": 80, "age": 1 }  
}
```

## Task

You will need to create:

1. A Python-based backend to serve the data
2. A Javascript-based frontend to retrieve the data

### Backend specifications

The backend must:

- Be written in Python
- Respond to a GET request of the following format: `/jobs/{number}` where `{number}` is the job number
- Return the appropriate JSON for the jobs listed above
- Either store the data for JSON files in a database or as files in the file system
- Respond with an appropriate HTTP error code for jobs that aren't in the above range (you'll be asked to justify which error code or codes you respond with)

The backend does *NOT* have to use a particular framework, or any framework at all. It's fine if it does (eg flask, django) but it's not a requirement.

### Frontend specifications

The frontend must:

- Be written in JavaScript or something that compiles into JavaScript (TypeScript/Coffeescript/ECMAScript etc.)
- Send an asynchronous request to the backend (GET `/jobs/{number}`) to retrieve a job listing
- Parse the output without page refresh
- Minimally output the required information
  - This job is around coding and architecture, so the display/design isn't important. The following display page will be sufficient (though you're free to embellish if you want/have time):

# Job Ad *ad\_number*

contents: *contents*

applicants: *num\_applicants*

biases:

*bias 1 name : bias 1 score*

*bias 2 name: bias 2 score*

*bias 3 name: bias 3 scoore*

...

Find job ad:

Find!

The frontend does NOT have to be in a particular framework (eg Angular, React, Vue etc.) but you may use one if you wish.

## Behavior

Inputting a job ad value into the input and pressing 'find' should send a request to the backend which will be interpreted and displayed in the frontend.

## Deliverables:

- The final code
- Instructions on how to build and run the code
- Any supplemental files

## Evaluation:

You will be evaluated on:

- Functionality of your code
- Your software decision-making process
  - Please comment your code enough to make your decision process clear

If you have any questions, please send an email back with them

## Time

This whole task should take no more than 3 hours. If you think it'll take longer, please reach out for clarification

## Additional

If you have any additional publicly available code that you'd like to submit (eg github or other public projects) and you haven't done so, please submit the relevant links in a text file with your task