Technical Challenge: Workforce API

Description

Gold Media Tech has a developers marketplace that sends a curated list of developers to the employers in the site based on their search criteria.

Technical Challenge

Create an application (API) that takes the clients (employers) search preferences and finds the best matches for the work opening they have. The application needs to be a REST API using Node.js and the framework of your choice (no framework? That is even better), we use MySQL as out database and we prefer to see you using a relational database for this challenge. The solution should be submitted in a Github / Gitlab account, with full source code and as a bonus we expect to see a configuration file to build and run it in a Docker container.

Challenge should not take more than 3 hours to complete, and we expect you to return it in the next following days, if you see you are exceeding the 3 hour limit, you should submit your solution as it is with a short explanation of what is left and how would you complete the challenge.

Tasks

1 - Create an endpoint for employee subscription, this should receive an email address and a list of the skills the employer is looking for.

2 - Create an endpoint to fetch a list of developers for a given user account ranked by most relevant depending on the matching criteria against the employer account.

3 - Create an endpoint for developers signup, this one will receive an email address, password and a skills list / keywords cloud to build up their profile, send a confirmation email to the developer’s email address on account creation.

4 - Configure a cron job to execute every hour and send notifications to the employers’ email address if a match is found in the developers cloud.

5 - Write unit and functional testing for the different modules you will write, create functional tests for the employer endpoints, we expect to see you spinning and seeding a new database for the tests suite.

We expect the endpoints to be secure, error friendly and to return meaningful responses with their corresponding HTTP response codes, we also expect to see some usage of HTTP request headers to communicate with the different endpoints (not with all, just where it makes sense).

It is expected for the application to run with the following commands:

1 - *npm start* to launch the API
2 - *npm run test* to run functional and unit testing
3 - *npm run test:integration* to run integration testing

You should provide a list of endpoints and the methods they accept, with the method you should provide a sample payload and the response you will get with the different payloads.

Be simple, this is just a test to measure your knowledge, not everything must be perfect but it should be in working condition.

Remember, we can help you with any questions related to the challenge.

Our emails

emi@goldmediatech.com