



\

## Fresh Prep Skills Test- Full Stack Developer

This is an exercise to help us get a sense of how you think. Don't worry about knowing the latest trendy tools, but don't be shy to get creative.

Your submission will be reviewed by a member of our engineering team, who will check the implementation for readability, simplicity, durability, and functional completeness.

There are 3 levels to this challenge. We know how busy life can get these days, so we hope you at least finish level 1.

Finishing level 2 will tell us that you can handle the challenge with ease and have extra time to spare. If you also finish level 3, that'll probably knock our socks off.

Lastly, a bonus point if you use standard ES6 syntax where appropriate. E.g. classes, destructuring assignment, and promises (or async-await).

### Level 1: Create a user REST API server (Required)

- Create an Ruby on Rails OR (**any framework of your choice**) REST API with this route table to act as a standalone server with own port
  - **id** (GET) Returns universally unique identifier without delay (expected < 50ms)
  - **user** (GET) Returns user details (expected response time ~300 ms)
  - **user** (POST) Returns success. Hypothetically, saving to a database (expected to fail 50%)
- Write unit tests for all endpoints

## Level 2: Create a client to consume the user endpoints (Optional)

- Create a second vue.js web app to consume the REST API above
  - Home web page should
    - Be a view template
    - Display user details including the identity
    - Content source is the user API
    - HTML and content is always displayed regardless of API failures
  - Every page load should create a new user and save it to the database. If the database saves then retry.

## Level 3: Create a REST API server using your framework (Optional)

- Use your favorite REST API framework with this route table to act as a standalone server with own port
  - **tone** (GET) This endpoint has a goal to detect the tone of the user biography (randomly return one of these three values: humorous, ironic, or cynical)
- Update the Node.js route **user** (GET) to enhance the user details with the tone property
- Use Docker so Node.js and dev environment is enclosed in the container and does not require tooling to be installed.