rithmschool / yodlr-take-home Public

Notifications Fork 109 Star 3

Code Issues Pull requests 6 Actions Projects Security Insights

master 1 branch 0 tags Go to file Code

elie Update README.md d8eb44b on Apr 26, 2021 3 commits

lib	init	5 years ago
public	init	5 years ago
routes	init	5 years ago
.gitignore	init	5 years ago
README.md	Update README.md	2 years ago
index.js	add cors	5 years ago
init_data.json	init	5 years ago
license	init	5 years ago
package-lock.json	add cors	5 years ago
package.json	add cors	5 years ago

README.md

Yodlr Front End Engineer Code/Design Challenge

Hello!

We're excited that you're interested in joining the Yodlr team. In the past, we have brought potential engineering candidates into our office for a full-day technical interview. This interview would include whiteboard programming exercises, code reviews, and other thought exercises. Unfortunately, onsite technical interviews are often biased against people who are not comfortable being put on the spot. Not to mention, who *actually* codes on a whiteboard in real life? In short, we realized that we were evaluating candidates in an unusual situation.

So instead, we've come up with this relatively open-ended programming/design challenge that will allow you to demonstrate your skills from the comfort of your own workspace. In addition, we know your time is valuable, so please feel free to use your completed work as a portfolio piece.

We wish you the best of luck and can't wait to see what you create!

Thanks,
Team Yodlr (Jared, Tom, & Ross)

Overview

We have provided you with a simple NodeJS application server for user registration and administration. This app does two things:

- Hosts static content from the 'public' directory
- Serves a JSON REST API for CRUD operations on users stored in memory

We would like for you to build a user interface for user registration and one for administration of existing users. To get you started, we have created placeholder pages (signup.html and admin.html) that you can build on.

You may use any front-end technologies you would like to create these user interfaces. This nodejs application is currently geared towards client-side rendering for systems like AngularJS, BackboneJS, EmberJS, jQuery, etc. but if you are more familiar with server-side templating (Handlebars, Jade, Swig, EJS, etc) please feel free to rework the application as needed.

In terms of design & layout, we leave that entirely up to you. We are familiar with Bootstrap but have also heard great things about Foundation. We recommend using whatever you're most comfortable with.

Getting Started

To use this application, you will need to download and install NodeJS.

Once you have NodeJS installed, you have two choices for downloading this source code:

1. Download & extract a zip file of the source
2. Fork this repository and git clone your fork

Next, you need to install the package dependencies by running the following command in the top-level directory of this source tree:

```
npm install
```

Once the dependancies are installed, you can start the application server by running

```
npm start
```

Once the server is running, you can access the start page (index.html) by opening your browser to http://localhost:3000.

To stop the server, press CTRL-C.

REST API

The Users JSON REST API is exposed at http://localhost:3000/users.

On server start, user data is read into memory from init_data.json. All subsequent actions are done against this memory store. Stopping and starting the server will re-initialize data from init_data.json.

API Endpoints

- **/users**
  HTTP GET: returns array of all users
  HTTP POST: creates a new user, returns the created user data
- **/users/:id**
  HTTP GET: returns the user with given id (numeric, auto-incrementing). HTTP 404 if user not found
  HTTP PUT: updates the user with given id and returns updated record. HTTP 404 if user not fund.
  HTTP DELETE: removes the users with given id, returns nothing (HTTP 204)

Here is an example of results returned from HTTP GET on /users:

```
[{"id":1,"email":"kyle@getyodlr.com","firstName":"Kyle","lastName":"White","state":"active"},
{"id":2,"email":"jane@getyodlr.com","firstName":"Jane","lastName":"Stone","state":"active"},
{"id":3,"email":"lilly@getyodlr.com","firstName":"Lilly","lastName":"Smith","state":"pending"},
{"id":4,"email":"fred@getyodlr.com","firstName":"Fred","lastName":"Miles","state":"pending"},
{"id":5,"email":"alex@getyodlr.com","firstName":"Alexandra","lastName":"Betts","state":"pending"}
```

Requirements

We kindly ask that you spend no more than 6-8 hours on this challenge.

At a minimum, there are three things we would like to see:

- Users should be able to register
- Admin page should list all users
- Design/layout of content

We will of course examine your code for readability, architectural decisions, and modularity. If/when you meet with us, be prepared to talk about why and how you built your interfaces.

Idea inspiration

If you have additional time after completing the requirements (we *think* you should), then we'd love to see what else you can do. Here are some ideas to get you started (but please don't limit yourself to these!).

- Experiment with alternative designs (A/B Testing is important for registration!)
- Signup form validation
- Automated testing
- Dynamic data on Admin page (no need to refresh to status changes)
- Sorting/Searching of users
- Admin button to activate accounts (set user status to 'active')
- Admin creation of new accounts
- Optimize assets (minimize and/or bundle css/js)
- Authentication/Authorization

To be perfectly clear, we don't expect that anyone could complete *all* of these in 6-8 hours. This is simply a list of ideas to inspire you.

License

We have licensed this project under the MIT license so that you may use this for a portfolio piece (or anything else!).

About

No description, website, or topics provided.

Readme
MIT license
3 stars
2 watching
109 forks
Report repository

Releases

No releases published

Packages

No packages published

Contributors 2

hueter Michael Hueter
elie Elie Schoppik

Languages

JavaScript 81.1% HTML 18.9%