

# Pricehubble Full Stack Developer practical Test

You should not spend more than 60 minutes on this task unless you really want to. Requisit: a modern browser, a text editor and Python3 installed.

- You should use all the modern feature of JavaScript available in a modern browser: async/await, fetch, fat arrow functions, template strings, etc.
- Using libraries as jQuery or lodash is fine.
- Using heavy library like Angular or React is discouraged unless you feel this will not slow you down.
- You don't need to setup a build system for this test.
- Focus on working and clear code rather than finishing all the steps.
- Once you are finished, zip all the files together (6 minimum) and send them back to us.

## Step 1

Create a basic HTML file called `step1.html` with "Github user explorer" as a title. Then add a JavaScript tag in there that satisfy the following

1. Use the github API to fetch data on Linus Torvalds repository (by doing an Ajax call). Here is his repositories: <https://api.github.com/users/torvalds/repos>.
2. Use this data to display a list of links to all his repositories using the description and the link to the github page that can be found in the API response.
3. Do this request at page load time
4. Use semantic HTML where it make sense.

## Step 2

Copy `step1.html` to `step2.html`

Add an input field in your page that allows you to enter any username. The list of repositories should automatically update 1 second after the user has finished typing within the input. Display **an error message** if the username doesn't exist (404).

## Step 3

Copy `step2.html` to `step3.html`

Without being authenticated you will quickly reach the Github API rate limitation, and your application will stop working. To remediate this problem design a mock API **using Python** that mimicks the Github API user endpoint: `/users/[username]/repos`.

Each API response should have between 3 to 12 repositories. Values in the necessary fields should be filled with random values from a library such as [the Python lorem package](#). You only need to mock the fields that you use not the whole API.

Adapt your JavaScript so it is pointing to this new endpoint and test things are working properly.

## Step 4 (optional)

Copy `step3.html` to `step4.html`

- Improve the look of the page using your own CSS, or any CSS framework of your liking.
- Add small README file explaining how to run the API mock.
- If you feel it is necessary, add some tests.

Thank you and good luck.