Technical Test

This technical test is made to assess your skills in various domains we believe are mandatory for working efficiently with us. These are basic knowledge which are used to either prepare, explore or process data.

The objectives here are mainly to understand how you think, thus performances and portability won't be (in a moderate[1] manner) accounted.

If the subject is not specific enough, you are free to either contact me (thomas.jiang@dathena.io) or assume a hypothesis. In the latter case, this should be mentioned in the README file.

Assignment submission format

Your identifier will be the first letter of your first name followed by your last name. Thus, mine is **tjiang**.

Your results must be submitted in an archive (zip, tar, rar,...). Your archive should uncompress into a directory being your identifier. Please pay attention to the file architecture.

Both parts of this test should be read before starting, as they complete each other.

REMINDER

Your README file contains text and must not be neglected. It is your only way to defend your results; there won't be an interview for reviewing your results nor explaining your code.

It should contain all instructions required for me to evaluate your code, i.e. how to compile if required, run, and options usage if any. All explanations are welcome, as long as they are structured and well presented. A messy README won't be read.

Test Description

In Dathena, many of our products require informative dashboards packed with interactive visualizations. At the core, React is used to power our UI – but will not be strictly required for the sake of this test.

You are provided with 3 pairs of JSON files with each containing a label file and a data file. For example, for languages there are *language_labels.json* and *language_data.json* files.

The labels represent the list of instances of a particular type of data – in the case of languages, you would have English, French etc. On the other hand, the data files represent the number of documents found to have a particular language. The same logic applies to the other types of data too.

Requirements

- Create a SPA with only one screen
- You may use any JS framework or, if you prefer, vanilla JS
- If, at any point, you are unable to proceed, you may leave comments and remarks in a README file discussing the roadblocks and potential resolutions

Your Task

Build a simple dashboard to represent the data provided visually. The data should be retrieved from the JSON files directly but you may wish to write your code as though you were retrieving it from an API endpoint. The visualizations should be simple and intuitive.

They should also exist as widgets that can be moved around or hidden from the dashboard.

You may use any 3rd party libraries wherever you deem fit, just remember that they should be saved as dependencies in your *package.json* file.

It would be a bonus if you could include basic tests.

Gentle Tip

Remember to keep things simple as much as possible and that this isn't a test of your design skills ☺