Find out their name

Code challenge for the Junior NodeJS Engineer job opportunity

by Leonardo Rodrigues

Overview

This document has the purpose of explaining the code behind my resolution for the challenge. The main code that actually deals with the names has two parts:

1 - Generating names: where it can be only one or an array of it;

2 - Validating names: where you can insert a name and validate if it fits the challenge.

I will only explain the generating name functions (called by "generateGroupNames(generateNamesQty, nameLength)") because it is what the challenge is asking for. The function "validateNames(names)" is an extra to check manually inserted names and it can be checked on the script.

This doc will be divided into 4 steps, each one responsible for this sequence:

- 1st Step: Making a name that starts with "ma";
- 2nd Step: Inserting "di" somewhere randomly into the string;
- 3rd Step: Inserting two vowels into the string;
- 4th Step: Inserting the consonants in the rest of the string;
- 5th Step: Manipulating and returning the string as a name;
- 6th Step: Making a bigger loop that will create an array with a lot of generated names and use an API to count the syllables and check if it is ok.
- 7th Step: Download the database from the government and check if the name exists on the population or not.
- 8th Step: Making the frontend using EJS on localhost, styled with CSS using Bootstrap.

To use the main project with interface you should select the folder “Main Project” with your terminal and run these commands:

npm i

tscp

tsc

cd dist

node server.js

You should be able to check the interface pasting “localhost:3000” on your favorite browser.

If you want to check it with the console, just use the script called “simpleResolution.js”, type “npm i” on the console and “node simpleResolution.js”. Go to the last lines of code to change the parameters like quantity of names and name length, the results should be printed out on the console.

Detailed Description

1st Step:

Initially, I noticed that we can't do anything if we don't know the length of what we are trying to generate. And we know that the name needs to be at least bigger enough to match all the requirements. Considering that we have 2 letters from “ma”, 2 from “di” and 2 from vowels, we will never have less than 6 characters so this will be the lower limit, while for the upper limit, it will probably not be bigger than 11 characters.

The main function for this challenge is named “generateNames(nameLength) and the first step (called by the variable “createName1stStep”) is responsible for creating the initial array that will start with “ma” and define the rest as a variable “x”.

2nd Step:

Let's assume we want a name with 8 letters. We do know that it needs to contain "di", but not where it should be placed, so we can insert this on 5 possibilities:

Ma | - - - -

Ma - | - - -

Ma - - | - -

Ma - - - | -

Ma - - - - |

This is made with a loop that inserts this string randomly in those positions and the result will be the variable "createName2ndStep". Another important variable is "positionsLeft" that is responsible to give us the exact positions that we need to replace in the future, this is easily made by finding the "x" chars that we let there before.

3rd Step:

We take random vowels and insert them in the positions that are left. Like:

M a d i - e - e

The variable "createName3rdStep" is responsible for doing this, some other variables in between like "updatePosition" have the purpose of updating the right position in the string so we don't replace something that is already there on purpose.

4th Step:

This is the most simple step, it just add random consonants to the string on the left positions. Like:

5th Step:

This divides the returned string into an array with the positions each “nameLength”.

6th Step:

This is where the real challenge started for me, because making a syllable counter in portuguese is not that easy as english or spanish for example, because portuguese has a lot of phonetic rules.

I did some research for trying to make the counter myself, and found this document:

https://aclanthology.org/W13-4812.pdf

At first view it looks like just a couple of “if’s”, but this doc is not complete e it is referring to another paper that has the original algorithm:

https://biblioteca.sbrt.org.br/articles/2721

You can see that we need at least 20 conditionals to make this work properly, and then update with the latest portuguese rules because this is really old. As it would be a lot of work I started to looking for something already made by someone else, after a couple more hours I found this repository on github:

https://github.com/andrefs/node-stress-pt

This will allow us to count the syllables and generate at least names with the right amount of syllables (4), everytime it does not match this quantity we start over and generate another name. Of course, I also added an upper limit inside the while loop to stop if it reaches 1000 times trying generating names with no success.

7th Step:

Obviously the chances to generate a name that looks normal to us is pretty random, and not happy with my results, I started looking for something to use as a database so we could compare and validate our generated names. So I found this database from IBGE (Brazilian Institute of Geography and Statistics) from 2010 that is opened to public, you can access the SQL database here:

https://basedosdados.org/dataset/br-ibge-nomes-brasil?external_link=Visualizar

You just need to create a new project on Google Cloud and add this database.

Then, aiming to select only the names that starts with “ma” and contains “di” you add the following command to the console:

```
SELECT nome FROM `basedosdados.br_ibge_nomes_brasil.quantidade_municipio_nome_2010`
WHERE nome like "Ma%di%"
```

You will get a list with 76 names, from a total of 1959116 names. The problem is: they are repeating themselves because it counts all the people that have those names, even repeating. As it is not a lot of data, you can just download it and make a simple script to delete the duplicated names with:

let uniqueChars = [...new Set(rawNames)];

It will be 19 names left. Now we just need to apply our syllable counter and see how much of them has 4 syllables and 4 vowels. This will leave only 5 options:

namesVerifiedIBGE = [‘Mamedio’, ‘Madiana’, ‘Madilene’, ‘Marilandi’, ‘Marindia’]

Well this is very disappointing, because the chances for our generator to create something like this is very remote… As there is 2 names with 7 letters, I decided to create a while loop that will

keep generating names until we find something like ‘Mamedio’ or ‘Madiana’. I runned 6 times and that is how much interactions we needed:

1st try:

7619 to find Mamedio

2nd try:

526 to find Madiana

3rd try:

1185 to find Madiana

4th try:

372 to find Madiana

5th try:

547 to find Madiana

6th try:

3174 to find Mamedio

We can see a lot of names with weird combinations like repeating letters, “q-(anything that is not u)”, “x - (anything that is not a vowel)”, and also a lot of repeating consonants and vowels (‘aa’,bb’,cc’,...) that don’t make sense. Most consonants don’t fit well with another consonant, but we can’t just make a general loop with this condition because there are only specific cases where this happens. So I added a function called “fixingWeirdCombinations()” that takes care of this a bit.

8th Step:

Now that the algorithm is finished we could stop the development. But why not make an interface that makes the use of the algorithm way more friendly-use? Since the job opportunity asks for NodeJS I decided to use it to make the backend, including Express to take care of the routes, the standard library HTTP that will make the request from the government API (yes I could have used that array I made with the direct access to SQL database but after a while I realized that the API has more names than the that database, sadly we can't access to all the names in the API though). The HTML will be printed out with EJS and the CSS is styled with the help of bootstrap.

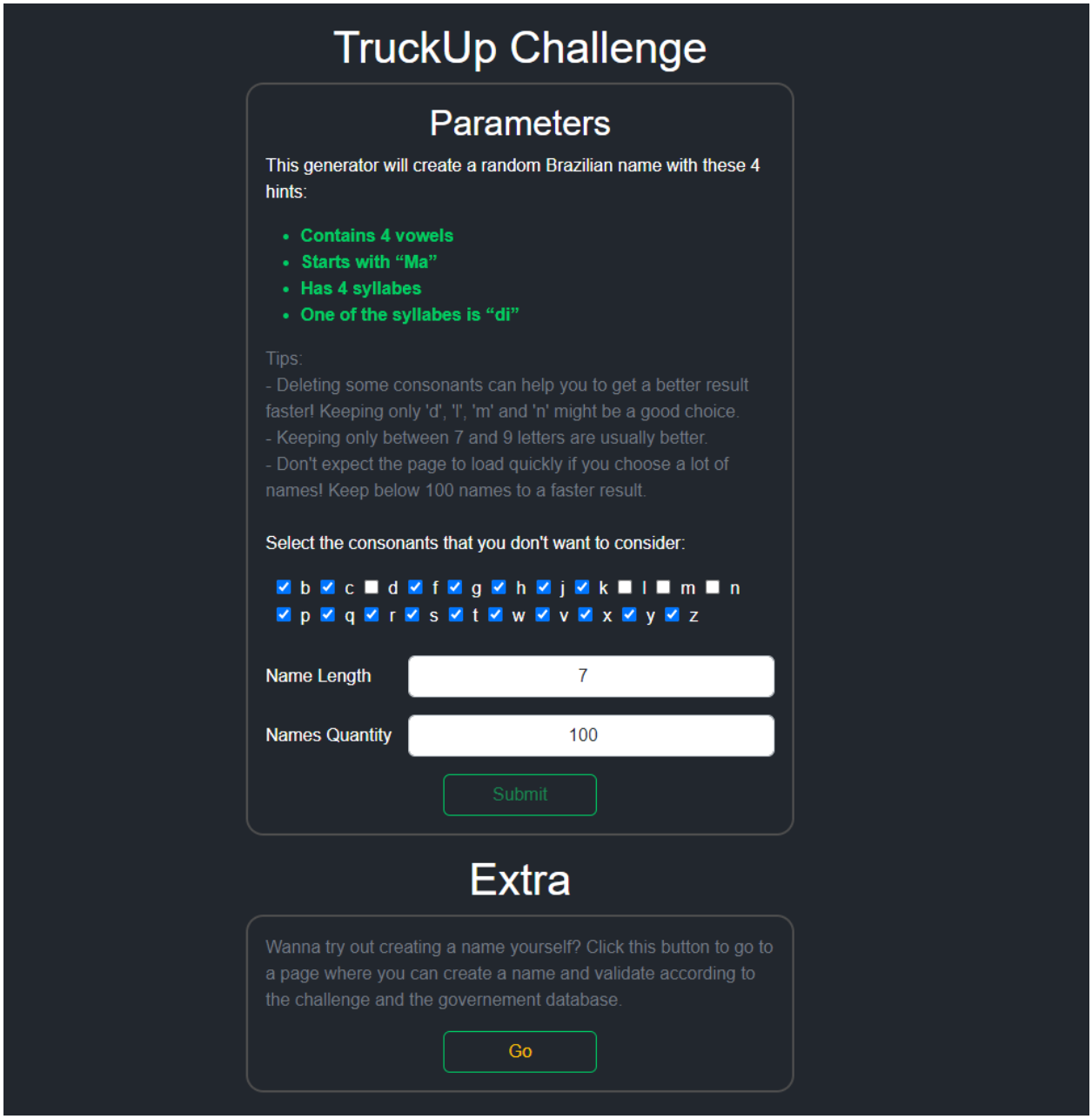
You can check IBGE's API here:

https://servicodados.ibge.gov.br/api/docs/nomes?versao=2

https://www.ibge.gov.br/censo2010/apps/nomes/#/search

Basically it has 4 pages divided in 2 main parts: home / results (take care of the challenge itself) and the extra / extra-results that verify an inserted name manually with the IBGE API.

They look like this:


TruckUp Challenge
Parameters
This generator will create a random Brazilian name with these 4 hints:
Contains 4 vowels
Starts with “Ma”
Has 4 syllabes
One of the syllabes is “di”
Tips:
- Deleting some consonants can help you to get a better result faster! Keeping only 'd', 'l', 'm' and 'n' might be a good choice.
- Keeping only between 7 and 9 letters are usually better.
- Don't expect the page to load quickly if you choose a lot of names! Keep below 100 names to a faster result.
Select the consonants that you don't want to consider:
b c d f g h j k l m n
p q r s t w v x y z
Name Length
7
Names Quantity
100
Submit
Extra
Wanna try out creating a name yourself? Click this button to go to a page where you can create a name and validate according to the challenge and the governement database.
Go

TruckUp Challenge

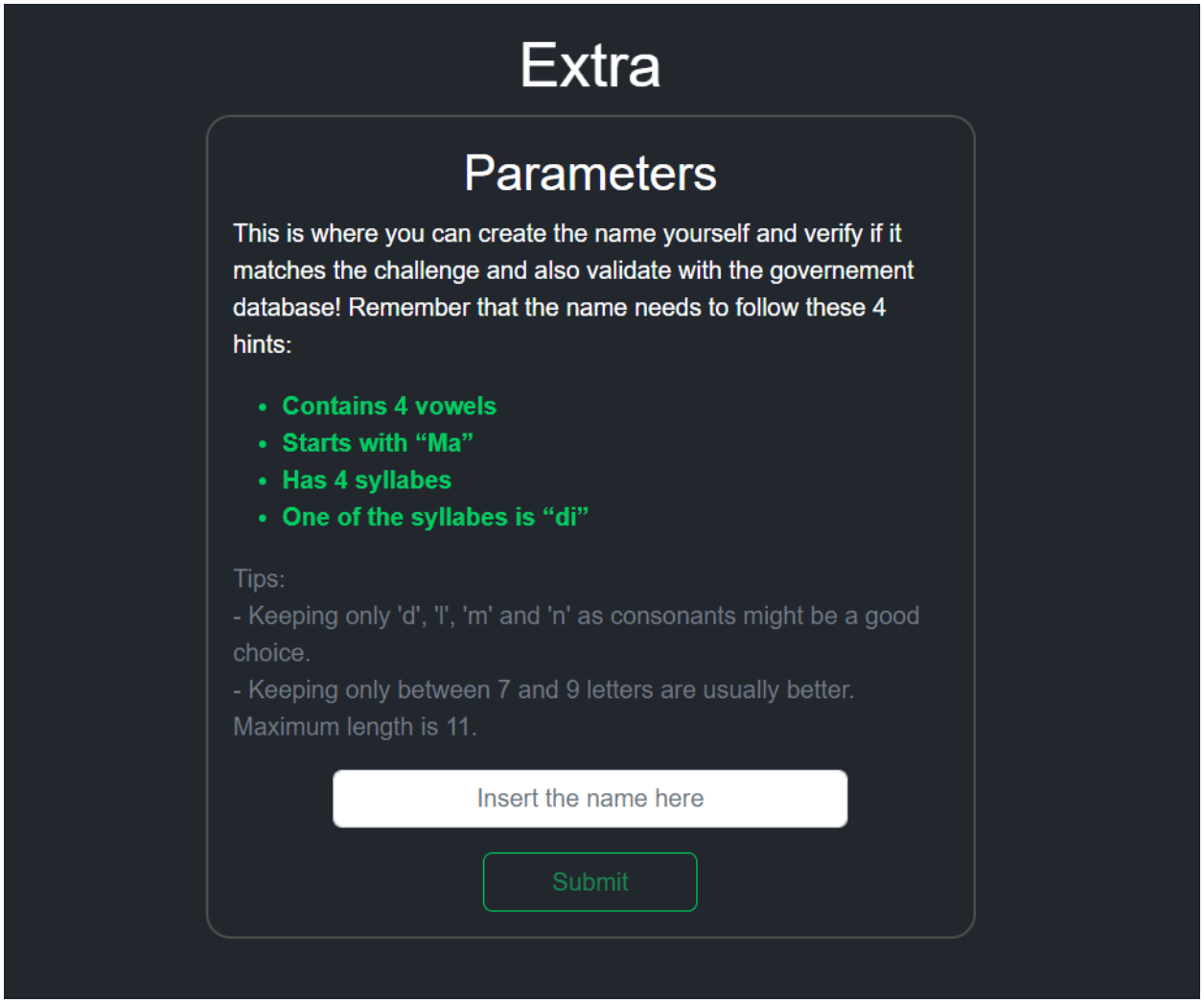
Results

Generated names:

Maoadim,Madieen,Maeldie,Maenadi,Manodio,
,Madioma,Maedili,Malaedi,Maodiel,Maodila,
,Madioon,Madiane,Maledio,Maoamdi,Madianu,
,Mamdioe,Maoedin,Manodia,Manadio,Madiele,
,Maeadim,Maedimo,Maloedi,Manidia,Mamodie,
,Madioel,Madieme,Maeamdi,Maniedi,Madieni,
,Maeoldi,Madiema,Maendio,Maliadi,Maoldia,
,Madinia,Maondie,Madiane,Madiena,Malidia,
,Maeandi,Madioel,Manoadi,Maeamdi,Mamedia,
,Madiene,Madieno,Malodia,Maedila,Madiala,
,Madiena,Madiolo,Maladia,Madimia,Maoedil,
,Maodino,Maomadi,Mamadia,Madimia,Madinio,
,Maladio,Madinie,Maedile,Mamdioe,Maoamdi,
,Madienu,Maneadi,Madiaon,Manedie,Malodie,
,Madioma,Madiome,Madioan,Madiuno,Madiloa,
,Madiala,Malodie,Maomedi,Madineo,Maediam,
,Madiaom,Maoldio,Maodile,Maendio,Madioim,
,Madiamu,Madioal,Maliedi,Maemdia,Madinoe,
,Madiomi,Mamoedi,Madiema,Mamiudi,Maodien,
,Madiela,Maedine,Maodial,Maodiem,Maodine,

According to IBGE (Brazilian Institute of Geography and Statistics) these names were found on the government database (with total of people with it):

Madiane | 121
Madiane | 121
Mamedia | 157


Extra
Parameters
This is where you can create the name yourself and verify if it matches the challenge and also validate with the governement database! Remember that the name needs to follow these 4 hints:
Contains 4 vowels
Starts with “Ma”
Has 4 syllabes
One of the syllabes is “di”
Tips:
- Keeping only 'd', 'l', 'm' and 'n' as consonants might be a good choice.
- Keeping only between 7 and 9 letters are usually better. Maximum length is 11.
Insert the name here
Submit

Extra Results

Mamedio was validated. And it was found on the Brazilian government database that there was 815 people with this name in Brazil until the year 2010.