Automation Engineer Take Home Project

The goal of this project is to test a simple API endpoint that takes a list of users and returns some statistics about them.

What To Do

Your task is to thoroughly test the Census Toy API service using automated tests. It is not enough to simply confirm that a successful request produces a success response code or that the service returns valid JSON. **You must include automated tests that evaluate the functionality of the service.**

Your submission must have descriptions of what you tested and how. You must tell us the program, framework, or language you used and why. **We have to be able to run your code**, and to do so, we need to understand the stack you used and how your program is run.

In your submission include any test setups, scripts, and source code you created either as a link to a **Git repository or as a ZIP file**.

In your description of what you tested and how, **include descriptions of what worked with the Census Toy API service and what did not**. Format descriptions of any bugs as if you were reporting a bug and opening a new ticket for it. These descriptions should be included in your project README, a separate text file attached to your email, or a text file located in the same folder as your code.

Thank you!

The Service

The Census Toy API service you must test is available at: https://census-toy.nceng.net/prod/toy-census

There is no limit on the number of requests you may make, and **no authentication is required**.

Input Format

General

The input to the Census Toy API is a single HTTP POST request.

Each POST must have a single JSON object as its body.

Fields

The JSON body of each POST must have two fields:

- `actionType` which should be one of three values: `CountByGender`, `CountByCountry`, or `CountPasswordComplexity`
- `users` which must be an array of one or more users; each user must be formatted per the `results` list provided by https://randomuser.me/api

There is also an optional parameter:

- `top` which when non-null and greater than zero is the maximum number of results returned

The action types work like so:

- `CountByGender` will return the count of users by whatever gender strings are in the data.
- `CountByCountry` will return the count of users by whatever country strings are in the data.
- `CountPasswordComplexity` will return the passwords sorted by complexity. Complexity will be considered the number of non-alphanumeric characters in the password.

Example

```
{
  "actionType":"CountByCountry",
  "top":0,
  "users":[
    {
      "gender": "female",
      "name": {
        "title": "miss",
        "first": "lorraine",
        "last": "bryant"
      },
      "location": {
        "street": "4422 harrison ct",
        "city": "gladstone",
        "state": "tasmania",
        "postcode": 3294
      },
      "email": "lorraine.bryant@example.com",
      "login": {
        "username": "smallduck444",
        "password": "aaron",
        "salt": "fYBp4g4a",
        "md5": "5d0785427febd6d262f00929c10247e7",
        "sha1": "a12a6925740eefebebcbc79add949fa108a78bf0",
        "sha256": "61a79aebd11cd5910dd9e77dd49b3dd11df2b4b397e328697f33b86e5b082b84"
      },
      "dob": "1956-09-17 02:13:36",
      "registered": "2009-05-03 14:40:51",
      "phone": "00-3540-6154",
      "cell": "0498-678-691",
      "id": {
        "name": "TFN",
        "value": "377488473"
      },
      "picture": {
        "large": "https://randomuser.me/api/portraits/women/38.jpg",
        "medium": "https://randomuser.me/api/portraits/med/women/38.jpg",
        "thumbnail": "https://randomuser.me/api/portraits/thumb/women/38.jpg"
      },
      "nat": "AU"
    }
  ]
}
```

Output

General

The output is an ordered list of key/value pairs of names and counts formatted as JSON.

The order will be descending by the `value` property

Fields

The output will be a JSON array of objects, each of which has two properties:

- `name`: the value of the property that the `actionType` operated on; for example, for `CountByGender` this would be the potential values for `gende`
- `value`: the count generated by the action

Example

```
[
  {
    "name": "AU",
    "value": 1
  }
]
```