Scrub Personal Identifying Information Problem

Application event tracking to external services often involves meta data about the user who took the action. When this event information is sent to a third party, we want to scrub all personally identifying data reported, without losing information about what fields were actually recorded.

In order to do this on different platforms, it is useful to implement a method called `scrub()` that will take a JSON object and an array of sensitive fields as an input and returns a modified JSON object that replaces alphanumeric characters with an asterisk ("*") from values corresponding to keys matching the sensitive field names.

For example, if the sensitive fields we want to "scrub" are `name`, `phone`, and `email`, a JSON input of:

```
{
  "name": "Kelly Doe",
  "email": "kdoe@example.com",
  "id": 12324,
  "phone": "5551234567"
}
```

should return:

```
{
  "name": "***** ***",
  "email": "****@*******.***",
  "id": 12324,
  "phone": "**********"
}
```

Requirements

You will need to implement a command line executable that takes two arguments: a text file with a list of sensitive fields and a JSON file of user data to scrub. Calling `./scrub sensitive_fields.txt input.json` should output a scrubbed JSON version of `input.json` with the keys in `sensitive_fields.txt` "scrubbed".

Any valid JSON input should be able to be handled. Value types for sensitive keys should be handled as follows:

- `String`: replace alphanumeric characters with "*"
- `Number`: convert to string and replace alphanumeric characters with "*"
- `Boolean`: replace entire value with "-"
- `Array`: each value of the array should be evaluated as described by other field types

- `Object`: if the key matches a sensitive field, all values of the nested object should be scrubbed as described by other field types. If the nested object does not correspond to a sensitive key, each key/value pair of the nested object should be evaluated as described by other field types
- `null`: value should be unmodified

Tests

A handful of example test "scrubs" are in the `/tests/` directory. Each subdirectory has an `input.json`, `sensitive_fields.txt`, and corresponding `output.json` that is expected.

You are not expected to have completed all tests; simply work through accomodating them incrementally in the time allotted.

Assignment and time expectations

You can find the assignment in this Google Drive folder. Please download the contents so that you can work with the code.

We recommend you spend no more than 2.5 hours on the project.

In case the above link is broken for you, here's the full URL:
https://drive.google.com/drive/folders/13UhcB1EdIvYSwICxmCu4dI0dpeQ5vbsS?usp=sharing

Submission

To submit your solution, please upload a text file or `README` to Greenhouse that includes the following:

1. A link to your code that we can access, such as a Google Drive folder or a Github repo
2. A description of how to run your code
3. Anything else we should know

If you're having any trouble at all with Greenhouse, feel free to email us the file instead.