Technical Task - DataCleaner

Introduction

Many different data sources are collected by NHS Digital. Some of the collected data relates to individuals, and a service named "National opt-out" is used for people to set preferences on how their data can be used.

The National Opt-Out service needs a new API to clean data. It is to be called the "DataCleaner". It will make use of an existing service - the *PreferenceProvider* API. The *PreferenceProvider* returns information about what preferences an individual has set.

See below to see how the components of the service will fit together:

```
[Client] -> [DataCleaner API] -> [PreferenceProvider API]
```

This task is to write the *DataCleaner* API.

A live-like version of the *PreferenceProvider* is provided here to use for this task - the documentation for this service is provided as part of the API.

```
https://preferenceprovider.herokuapp.com/
```

Unfortunately the *PreferenceProvider* service is **NOT** reliable, problems include: network delays, partial responses and error responses. The data behind it changes unpredictably as it is updated in real time based on patient preferences. It is due to be improved in the future, but not before the new *DataCleaner* service needs to be launched.

Your code should appropriately manage the nature of the PreferenceProvider service.

The *DataCleaner* API, accepts a json document containing personal data, the first iteration only accepts a Identifier of a person `id` and any other data will be ignored and returned unmodified.

Please note that the identifier `0001` is a test value which is always set to obfuscate - return a masked value for that identifier.

```
 POST /cleanse HTTP/1.1
Host: <your host>
Accept: application/json
```

For the following example:

```
curl --header "Content-Type: application/json" \
    --request POST \
    --data '{"rowId": "1", "id":"0001","favouriteColour":"red"}' \
    http://<your host>/cleanse
```

The following data is posted to the API:

```
{
   "rowId": 1,
   "id": "0001",
   "favouriteColour": "red"
}
```

The data cleaner service needs to retrieve the preferences from the*PreferenceProvider* API and act accordingly. In this example the data that should be returned looks like:

```
{
   "rowId": 1,
   "id": "896bfac35e",
   "favouriteColour": "red"
}
```

The `id` has been masked with the value being returned from the*PreferenceProvider*, and the other values remain unchanged.

If you get a request that fails for any reason - the service should retry up to 3 times. Ultimately if the request cannot be met, the service should return the following HTTP status code:

```
504 Gateway Timeout
```

Thankfully, a tester has found that the*PreferenceProvider* service always fails if the identifier `0003` is provided - this helps to build up test cases.

Example Identifiers

Identifiers are string values, valid identifiers are from "0000" to "9999".

- `0001` - will always obfuscate
- `0002` - will never obfuscate
- `0003` - will always produce an error

In addition, there are around 1000 other values in the range "0004" to "9999" which will obfuscate: these change regularly.

Instructions

The DataCleaner API must be hosted as a docker container.

1. Build the new DataCleaner API to meet the requirements above.
2. Provide a `Makefile` to run the DataCleaner API with the following commands:
    - `init`: this will setup the docker container(s)

    - `test`: this will run your integration tests
    - `serve`: this will run the API as a daemon
3. There is no need to provide a README or any other documentation - as `make init`, `make test` and `make serve` are the only commands that will be run as part of your submission.
4. Ensure that the DataCleaner API has suitable tests
5. Ensure that the DataCleaner API generates useful logging information.
6. Zip up your source code and upload according to the instructions in your email.