Problem definition

We want to make an endpoint which returns a list of employees' information. Each employee should also contain the relevant country specific information about their country. Depending on the region of the country, an employee's additional identifier might be required. The initial list of employees, without the country specific information, is already given. We need to first retrieve details for each country, and then return a list of employees' information that includes their country specific information and the additional identifier (in case it is needed).

Problem solution

We retrieve a desired list of employees in a few steps:

1. From the initial Employee list, get a distinct array of employees' country codes.
2. Make a global array that contains region names that require the additional identifier. Let it's name be *AdditionalIdentifRegions*.
3. Get country details from https://restcountries.com/ by providing an array of distinct codes from the step 1. API call: `https://restcountries.com/v3.1/alpha?codes={code},{code},{code}`. In the example, that should be `https://restcountries.com/v3.1/alpha?codes=us,gbr,ind`the Let the name of the API output be *restCountriesInfo*.
4. For each of the country codes in the array made in step 1, we do:
    a. Find the element in *restCountriesInfo* where one of the cca2/ccn3/cca3/cioc fields is the same as the country code that is in the loop. Let it's name be *restCountryInfo.*
    b. Create an object (let's call it *CountryInfo*) that has fields: fullName, currencies, languages, timezones. We get fullName as by getting field *name*, currencies by getting field *currencies*, languages by getting field *languages* and timezones by getting field *timezones* from the *restCountryInfo* object.
    c. Filter initial employee list by the country code, so that we get all the employees from the country currently in the loop.
    d. Add the *CountryInfo* object as a field to each of the filtered Employee objects

e. If there are mutual elements in *AdditionalIdentifRegions* and *restCountryInfo.region*, to each of Employee objects, add a field *AdditionalIdentifier* that is formed as a concatenation of employee's firstName,lastName,dateOfBirth. This value should be converted to lowercase and get all the special characters trimmed.

5. Return the changed employee list.

Different approaches

There are many other suitable API calls that we can make to the restCountries API. For example, we could have used the *filter response* call, to include only the specified fields. In our case, that would be https://restcountries.com/v2/all?fields=name,currencies,languages,timezones,region,cca2,cca3 This way we'd get all the world's countries information, so the filtering would need to process more data. Another possibility is to make a separate API call for every distinct country that is in the Employee List. In that case, filtering is optimal, but the process would be slower because of the need to make multiple API calls.

Instead of looping through every distinct country from the Employee list, we can loop through every employee, and add a *CountryInfo* object to each of them. That way, the number of iterations would be equal or greater than the number of iterations that is written in the proposal.

What can go wrong

Some possible use cases where this code would fail, is when, first of all, the Rest Countries API is down, or retrieves an error. The code should be implemented in a way that there are valid try/catch blocks so that the application doesn't crash.