Exercises

a)

We have to create a new api to handle positions, the client provides us with an endpoint https://jobs.github.com/positions.json to consume all the positions that they have.

They are paginating by 50 elements and starting on page = 1, by default page is set on 1
So https://jobs.github.com/positions.json means https://jobs.github.com/positions.json?page=1

Each position has the following structure:

```
{
  "id": "fa6469df-0093-4cee-a666-716d98f4aea2",
  "type": "Contract",
  "url": "https://jobs.github.com/positions/fa6469df-0093-4cee-a666-716d98f4aea2",
  "created_at": "Wed May 15 21:58:24 UTC 2019",
  "company": "Infinity Consulting Solutions (ICS)",
  "company_url": "http://www.infinity-cs.com",
  "location": "Wilmington",
  "title": "Java Software Engineer - Contract to Hire, Large Financial Tech Company",
  "description": "<p>In this role, you will develop an ....",
  "how_to_apply": "<p>If interested and qualified, please e-mail your resume to ...",
  "Company_logo": "..."
}
```

First we want to store the first 100 positions in our system as open positions, we usually receive different apis to get positions.
Our goal in the future is to have the ability to import positions from any external api with the option to collect the count of registers that we want. So we should expect two variables, a url for the request and the count of positions to store.
We need to store a position with at least these fields:

- ExternalId, the id in the external api, (ie: fa6469df-0093-4cee-a666-716d98f4aea2)
- Name, the type of position, (ie: Java Software Engineer - Contract to Hire, Large Financial Tech Company)
- Current company, (ie: Infinity Consulting Solutions (ICS))
- Type, type of position, (ie: Contract|Full Time|Part Time)
- Location (ie: Wilmington)

Apart from that we need to retrieve positions from our data base filtering by type of positions, location and description for our new clients.

We have to deliver in the next sprint the following endpoints

- An endpoint for our client greenFuture: all the Java positions on San Francisco
- An endpoint for our client vonMises: all the data scientist position
- An endpoint for our client freeToChoose: all the full time positions on New York

Task
Create an application from scratch to support all the functionality with the best practices that you know

Extra comments
We suggest use java with spring boot framework
We suggest to use a sql database
Each client is going to have more functionality in a short term
Be aware of the scalability to reuse this api for new third external apis
The design is very important for us, be aware of using clean code

Bonus
If you want to add security to the api it is a plus
If you want to add swagger to document the api it is a plus

b)

When writing a class to split input lines into fields, which of the following signatures is more orthogonal?

```
class Split1 {
  public Split1(InputStreamReader rdr) { …
  public void readNextLine() throws IOException { …
  public int numFields() { …
  public String getField(int fieldNo) { ...
}

class Split2 {
  public Split1(String line) { …
  public int numFields() { …
  public String getField(int fieldNo) { ...
}
```