

# Technical test: Backend

Welcome  
to the Jungle

## APPENDICES

You will find enclosed to this document, 2 files:

- 1/ A list of a few thousands of job offers:

```
profession_id,contract_type,name,office_latitude,office_longitude  
2,FULL_TIME,Dev Full Stack,48.8768868,2.3091203
```

- 2/ A list of “professions” (cf. “profession\_id” above):

```
id,name,category_name  
16,Développement Fullstack,Tech
```

## EXERCISES / QUESTIONS

### 01 / 01 . Exercise: Continents grouping

Using the previous data, your goal is to develop a script which will return the count of job offers per profession category per continent.

Exemple :

	TOTAL	TECH
TOTAL	1	1
EUROPE	1	1
ASIE	0	0

### 02 / 02 . Question: Scaling ?

Now, let's imagine we have 100 000 000 job offers in our database, and 1000 new job offers per second (yeah, it's a lot coming in!). What do you implement if we want the same output than in the previous exercise in real-time?

*NB: no code necessary for this question. We expect a thorough thinking around this subject.*

---

## FAQ

### Which languages should I use?

To help us review your test, **Elixir** or **Ruby** are the go to languages. If you are not familiar with those languages, don't worry, we will review your test with this in mind. The goal here is to have a base to discuss and a peek of your reflection.

### How much time should the test take?

You have 7 days to complete the test. But, you don't have to allow all that time for it. Few hours should be enough. In general, candidates spend 5 to 10 hours on their tests. Of course, you are free to allocate the time you judge necessary to finish your test.

### What is the expected output?

- A nice git repository with commits history that will help us follow your flow.
- A readme to help us install and launch your application.
- You can add some lines to explain your train of thoughts on the ex01.
- Anything you judge is valuable for your test :)