REMOTE BACKEND ENGINEER

Use Case

Noddus

This is a coding use case, not a systems design, nor an algorithms one

This use case is comprised of building two small applications, one in Python and one in Java/Clojure or any other JVM language. These are the languages and systems we use on a daily basis and we would require applicants to be able to program useful applications in them.

Python

Small command line app that:

- Reads a csv input file given from the command line.
- Connects to a SQLite db.
- Inserts the csv records there.
- Prints a summary as output e.g N records inserted, total records are N.
- Should be launchable as "./<myscript> <my-csv-file>".
- The csv format can be fixed with the tables pre created e.g create table person (time timestamp, name varchar, age integer).

Delivery:

- As a bitbucket/github repo either private or public.
- Tar.gz file.

Java/JVM Language

Small uncomplicated app:

- Google protobuff format for {"name", "id"} records.
- Simple Http server that receives a JSON object as body {"name:" "<name>", "id": <number>}.
- Saves the data to disk using google protobuff, and rollover after timeout (just simple writing and roll over the file by closing it and opening a new one).
    - This should be threadsafe
    - Simple close file, open new file
    - No transactional roll back or update is required
    - The file should stay open till roll over
- Integrated with docker compose.
- Launched from the command line using a bash script.
- Any framework for webapps can be used, Vanilla Java, Jetty Connectors, Newer Java Frameworks, etc.
- Must have a build system either Maven, Leiningen, Boot, Gradle, Make or other.

Delivery:

- As a bitbucket/github repo either private or public.
- Tar.gz file.

EVALUATION:

Feel free to structure the use case as you prefer; these are some of the things we are expecting to see:

- The applications should meet the requirements outlined above.
- If for any reason you could not complete a particular part please state so and comment on how you would solve it.
- Programming is an endless task, if you have reached a stage where its working but you would like to refactor or would improve on please state so and comment your observations.
- Adequate and useful tests.
- Both projects should come with instructions on how to install, test and run, and both should be provided with at least a minimum build script/system/conf file, e.g pom.xml, project.clj, build.gradle, makefile, build.sh are all acceptable.

There is no single correct answer and structure, the program should be clear, buildable and testable, and the problem area solved.
You can use either OO or Functional or mix, that is up to your discretion.

At Noddus you will have a tremendous amount of freedom and we will give you complete ownership of open-ended problems. We need people we can trust to do the right thing without a lot of supervision.