1

Backend & full-stack developers Exercise : Automatic irrigation system

What is automatic irrigation system ?

As a irrigation system which helps the automatic irrigation of agricultural lands without human intervention, system has to be designed to fulfil the requirement of maintaining and configuring the plots of land by the irrigation time slots and the amount of water required for each irrigation period.
The irrigation system should have integration interface with a sensor device to direct letting the sensor to irrigate based on the configured time slots/amount of water.

Technology constraints

- At the digital factory, we currently have a microservices architecture with services mostly written in Spring Boot.
- We would like you to develop a single service using Spring Boot with either Java or Kotlin that exposes several REST endpoints (see next page)
- You can use whatever database and data access method; the application should create the required database structure.
- You should include a README file that has instructions for us to get the solution running on our machines.

What are we looking to test?

- The overall software architecture of the application
- The structure and **quality of the code itself**
- The use of well-known patterns for REST and Spring development
- Your ability to model the problem domain (data models and APIs)
- **Full-Stack developers/leads (only),** overall Angular project structure and code-quality.
- Bonus points if you include unit tests in your solution.

What are we *not* looking to test?

- We don’t expect you to implement authentication or authorization
- We don’t expect you to implement real IOT system , just simulation via mocking is enough.

External Resources

- Building a RESTful service in Spring Boot
- Testing in Spring Boot

2

Backend & full-stack developers Exercise : Automatic irrigation system

Your APIs should provide the following functionality to downstream consumers of the API (in a **RESTful** way):

Plot of land

- Add new plot of land
- Configure a plot of land
- Edit a plot of land
- List all plots and it's details

Automatic irrigation system

- Integration interface with the sensor device once a plot of land need to be irrigate
- Update the status of the slot once the request is successfully sent to the sensor device
- Retry calls to the sensor device in case the sensor not available (pre configured)

Alerting

- Alerting system to be implemented in case the sensor not available and after exceeding the retry times.

TIP #1: focus on defining a very clear data model for plots, time slots, and others. You should include the SQL create table statements etc. in your code base.

TIP #2: invest time in building some 'seed data' that covers all the use cases you want to demonstrate.

ONLY FULL-STACK DEVELOPER/LEADER

TIP #3: build web frontend pages, for some of the journeys as APIs described using Angular 9+.

BONUS POINTS:

Predict the (slots time / amount of water) based on the given type of agricultural crop / cultivated area.