

On Boarding Java Tasks:

1) Create a spring boot application with three rest endpoints namely create retrieval and a health check endpoint to check if the backend call is getting success or not. Use h2 db as backend.

- Implement proper loggers in all the layers.
- Use JDBC Template or JPA Repository
- Layer Structure-controller->service->BO->EO->DAO
- Note-all classes should follow interfaces and its implementation class.
- All constants should be in util package in constants class.
- Use mapstruct to map pojo classes to new vo in service layer.
- Do basic validations like not null, min and max value for retrieval endpoint using annotations.
- Use gradle as build tool.
- Use Git and implement branching strategy and raise pull request to main/master branch

Note: **Use SpringBoot 3**

2) Create a second spring boot application and add a rest end point to call first spring boot application retrieval end point using spring **rest template** and another rest endpoint to call same retrieval endpoint using **feign client**.

- Implement proper loggers in all the layers.
- Layer Structure-controller->service->BO->EO->Rest Helper
- Note-all classes should follow interfaces and its implementation class.
- Do basic validations like not null, min and max value for retrieval endpoint using annotations.
- All constants should be in util package in constants class.
- Use mapstruct to map pojo classes.
- Use gradle as build tool.
- Use Git and implement branching strategy and raise pull request to main/master branch

3) Create a spring cloud config server to externalize the configuration to git.

- 1 & 2 should implement cloud config server and all constants should be externalized to git.

4) Implement cloud gateway design pattern using spring cloud gateway.

- Calls to 1 and 2 spring boot applications should be routed through this gateway.

5) debugging code in STS IDE

6) Setup Mockito and write Junit test cases

7) Setup BDD using Cucumber

8) Setup wiremock and use wiremock

9) Circuit Breaker Design pattern

10) Use JAXB context and convert POJO object to JSON & vice versa

11) JMeter and Gatling (Java and Scala) performance testing for application 1 and application 2

12) Implement Spring Batch by reading input from H2 DB and write to CSV file

- Use Item Reader, Item Processor, and Item Writer

13) Implement Spring Batch by reading input from CSV file and write to H2 DB

- Use Item Reader, Item Processor, and Item Writer

14) Implement spring Batch job with tasklet

- Each step should perform a particular task like itemReader as one step itemprocessor as another and 3rd step as itemWriter

15) Implement Spring Batch job with chunk size of 5 input and Data base should have at at least 10 records

16) implement Contract Testing from App2 to APP1 by using PACT to ensure compatibility

17) implement simple login application by using Spirng MVC