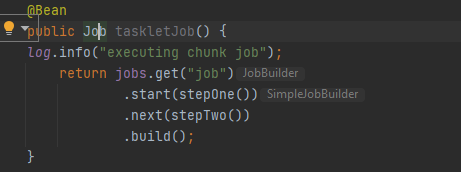
**SPRING-25 –Tasklet job.**

Create a spring batch to fetch rows from an excel/csv file, save into DB. Then read from DB and copy the contents into a new excel file.

1. Create TaskletInstance(TaskletOne) which will implement the Tasklet Interface. Override the execute method. Logic: read from the excel file provided and write into the DB.
2. Create TaskletInstance(TaskletTwo) which will implement the Tasklet Interface. Override the execute method. Logic: fecth the records from DB and write into a new excel file.
3. Create job which will execute the above tasklets one by one.

**Code:**

**Job parameters:**



**Configure Steps:**



**TaskletOne logic:**

public class TaskletOne implements Tasklet {  
  
 ExcelUtil excelUtil = new ExcelUtil();  
  
 @Value("${excel.file.location}")  
 String fileLocation;  
  
 @Autowired  
 TaskletService taskletService;  
  
 @Override  
 public RepeatStatus execute(StepContribution contribution, ChunkContext chunkContext) throws Exception {  
 *log*.info("tasklet 1 start");  
  
 List<Employee> employeeList = excelUtil.readExcelFile(fileLocation);  
 *log*.info("employee list size: " + employeeList.size());  
 taskletService.insertInDb(employeeList);  
  
 *log*.info("tasklet 1 end");  
 return RepeatStatus.*FINISHED*;  
 }  
}

**TaskletTwo logic:**

public class TaskletTwo implements Tasklet {  
 @Autowired  
 TaskletService service;  
 @Value("${write.excel.file.path}")  
 String filepath;  
  
 ExcelUtil excelUtil = new ExcelUtil();  
  
 @Override  
 public RepeatStatus execute(StepContribution contribution, ChunkContext chunkContext) throws Exception {  
 *log*.info("tasklet 2 start");  
 List<Employee> employeeList = service.getEmployeesFromDB();  
 *log*.info("employees got from DB : " + employeeList.size());  
 *log*.info("writing into excel file now");  
 excelUtil.writeIntoExcelFile(filepath, employeeList);  
 *log*.info("tasklet 2 end");  
 return RepeatStatus.*FINISHED*;  
 }  
}

**properties:**

