Spring Boot Batch

**--code files in order----**

1. ProductReader

2. ProductProcessor

3. ProductWriter

4. ProductListener

5. BatchConfig

Step <- StepBuilderFactory

Job <- JobBuilderFactory

---------------------------------

1. ProductReader

**package** org.st.reader;

**import** org.springframework.batch.item.ItemReader;

**import** org.springframework.batch.item.NonTransientResourceException;

**import** org.springframework.batch.item.ParseException;

**import** org.springframework.batch.item.UnexpectedInputException;

**public** **class** ProductReader

**implements** ItemReader<String>

{

@Override

**public** String read()

**throws** Exception,

UnexpectedInputException,

ParseException,

NonTransientResourceException {

**return** **null**;

}

}

2. ProductProcessor

**package** org.st.processor;

**import** org.springframework.batch.item.ItemProcessor;

**public** **class** ProductProcessor

**implements** ItemProcessor<String, String>

{

@Override

**public** String process(String item)

**throws** Exception {

**return** **null**;

}

}

3. ProductWriter

**package** org.st.reader;

**import** java.util.List;

**import** org.springframework.batch.item.ItemWriter;

**public** **class** ProductWriter

**implements** ItemWriter<String>

{

@Override

**public** **void** write(List<? **extends** String> items) **throws** Exception {

}

}

4. ProductListener

**package** org.st.listener;

**import** org.springframework.batch.core.JobExecution;

**import** org.springframework.batch.core.JobExecutionListener;

**public** **class** ProductListener

**implements** JobExecutionListener

{

@Override

**public** **void** beforeJob(JobExecution je) {

System.***out***.println(je.getStatus());

}

@Override

**public** **void** afterJob(JobExecution je) {

System.***out***.println(je.getStatus());

}

}

5. BatchConfig

**package** org.st.config;

**import** org.springframework.batch.core.Job;

**import** org.springframework.batch.core.JobExecutionListener;

**import** org.springframework.batch.core.Step;

**import** org.springframework.batch.core.configuration.annotation.EnableBatchProcessing;

**import** org.springframework.batch.core.configuration.annotation.JobBuilderFactory;

**import** org.springframework.batch.core.configuration.annotation.StepBuilderFactory;

**import** org.springframework.batch.core.launch.support.RunIdIncrementer;

**import** org.springframework.batch.item.ItemProcessor;

**import** org.springframework.batch.item.ItemReader;

**import** org.springframework.batch.item.ItemWriter;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.context.annotation.Bean;

**import** org.springframework.context.annotation.Configuration;

**import** org.st.listener.ProductListener;

**import** org.st.processor.ProductProcessor;

**import** org.st.reader.ProductReader;

**import** org.st.reader.ProductWriter;

@Configuration

@EnableBatchProcessing

**public** **class** BatchConfig {

//7. Autowire JobBuilderFactory

@Autowired

**private** JobBuilderFactory jf;

//8. Configure Job

@Bean

**public** Job jobA() {

**return** jf.get("jobA")

.incrementer(**new** RunIdIncrementer())

.listener(**new** ProductListener())

.start(stepA())

//.next(stepB())

.build();

}

//5. Autowire StepBuilderFactory

@Autowired

**private** StepBuilderFactory sf; //HAS-A

//6. Configure Step

@Bean

**public** Step stepA() {

**return** sf.get("stepA")

.<String,String>chunk(5)

.reader(reader())

.processor(processor())

.writer(writer())

.build();

}

//--Objects--

//1. Reader

@Bean

**public** ItemReader<String> reader(){

**return** **new** ProductReader();

}

//2. Processor

@Bean

**public** ItemProcessor<String, String> processor(){

**return** **new** ProductProcessor();

}

//3. Writer

@Bean

**public** ItemWriter<String> writer(){

**return** **new** ProductWriter();

}

//4. Listener

@Bean

**public** JobExecutionListener listener() {

**return** **new** ProductListener();

}

}