3 - 6.

6 .

- JobLauncherCommandLineRunner JobLauncher
- ApplicationContext CommandLineRunner spring-boot-starter-batch JobLauncherCommandLineRunner
- spring.batch.job.names

REST

- SimpleJobLauncher
- TaskExecutor .

```
@Autowired
private ApplicationContext context;

@PostMapping("/run")
public ExitStatus run(@RequestBody JobLaunchRequest request) {
        Job job =  this.context.getBean(request.getName(), Job.class);
        return this.jobLauncher.run(job, request.getJobRarameters()).getExitStatus();
}
```

-
- JobParmetersIncrementer

(quartz)

- , , .
- SchedulerFactory JobDetails .
- .
- .
-
- JobDetail .

```
public class BatchScheduledJob extends QuartzJobBean {
                // job
}

@Configuration
public class QuartzConfiguration {

        @Bean
        public JobDetail quartzJobDetail() {
                return JobBuilder.newJob(BatchScheduledJob.class) //
                                .storeDurably()
                                .build();
        }

        @Bean
        public Trigger jobTrigger() {
                SimpleScheduleBuilder scheduleBuilder = SimpleScheduleBuilder.simpleSchedule()
                                .withIntervalInSeconds(5).withRepeatCount(4); //5         4  ( 5)

                return TriggerBuilder.newTrigger()
                                .forJob(quartzJobDetail())
                                .withSchedule(scheduleBuilder)
                                .build();
        }
}
```

-
  - Step ExitStatus.STOPPED;

```
return this.jobBuilderFactory.get("transactionJob")
                                .start(importTransactionFileStep())
                                .on("STOPPED").stopAndRestart(importTransactionFileStep())
                                .from(importTransactionFileStep()).on("*").to
(applyTransactionsStep())
                                .from(applyTransactionsStep()).next(generateAccountSummaryStep())
                                .end()
                                .build();
```

- StepExecution

```
private Transaction process(FieldSet fieldSet) {
                Transaction result = fieldSet.getTrainsaction();

                if(fieldSet == null) {
                        //  e
                        this.stepExecution.setTerminateOnly();
                }

                return result;
        }
```

- .
- .
-
-
- preventRestart()

```
@Bean
public Job transactionJob() {
                return this.jobBuilderFactory.get("transactionJob")
                                .preventRestart() //
                                .start(importTransactionFileStep())
                                .next(applyTransactionsStep())
                                .next(generateAccountSummaryStep())
                                .build();
}
```

- startLimit(2)

```
@Bean
public Job transactionJob() {
		return this.jobBuilderFactory.get("transactionJob")
				.startLimit(2) //
				.start(importTransactionFileStep())
				.next(applyTransactionsStep())
				.next(generateAccountSummaryStep())
				.build();
}
```

- allowStartIfComplete

```
	@Bean
	public Step importTransactionFileStep() {
		return this.stepBuilderFactory.get("importTransactionFileStep")
				.<Transaction, Transaction>chunk(100)
				.reader(transactionReader())
				.writer(transactionWriter(null))
				.allowStartIfComplete(true) //
				.listener(transactionReader())
				.build();
	}
```