배우기

spring-batch

#spring-
batch

1:

, .

. spring-batch .

3.0.x	2016-04-22
2.2.x	2014-05-22
2.1.x	2012-08-20
2.0.x	2009-10-19
1.1.x	2009-01-29
1.0.x	2008-05-06

Examples

Maven Java Config

- Maven .
- Intellij, Eclipse NetBeans IDE .

Maven

(, ID `com.organization.app` , Artifact ID `SpringBatchExample` :

```
SpringBatchExample
|-- pom.xml
`-- src
    |-- main
    |   `-- java
    |       `-- com
    |           `-- organization
    |               `-- app
    |   `-- resources
    `-- test
        `-- java
            `-- com
                `-- organization
                    `-- app
```

Hello World

Spring Batch Spring .   Java .

`pom.xml` :

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                             http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.organization.app</groupId>
    <artifactId>SpringBatchExample</artifactId>
    <version>1.0-SNAPSHOT</version>

    <build>
        <plugins>
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.1</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
        </plugins>
    </build>

    <dependencies>
        <dependency>
            <groupId>org.springframework.batch</groupId>
            <artifactId>spring-batch-core</artifactId>
            <version>3.0.7.RELEASE</version>
        </dependency>
        <dependency>
            <groupId>commons-logging</groupId>
            <artifactId>commons-logging</artifactId>
            <version>1.1.1</version>
        </dependency>
    </dependencies>
</project>
```

`src/main/java/com/organization/app` `SpringBatchHelloWorldConfig.java` . Spring Batch `Employee` ,
, `Employee` :

```java
@Configuration
@EnableBatchProcessing
public class SpringBatchHelloWorldConfig {

    @Autowired
    private JobBuilderFactory jobBuilderFactory;

    @Autowired
```

```java
    private StepBuilderFactory stepBuilderFactory;

    @Bean
    public Step step1() {
        return stepBuilderFactory.get("step1")
                .<Employee, Employee>chunk(2)
                .reader(employeeItemReader())
                .processor(employeeItemProcessor())
                .writer(employeeItemWriter())
                .build();
    }

    @Bean
    public Job listEmployeesJob(Step step1) throws Exception {
        return jobBuilderFactory.get("listEmployeesJob")
                .start(step1)
                .build();
    }

    @Bean
    ItemReader<Employee> employeeItemReader() {

        FlatFileItemReader<Employee> reader = new FlatFileItemReader<>();
        reader.setResource(new ClassPathResource("employees.csv"));

        DefaultLineMapper defaultLineMapper = new DefaultLineMapper();
        DelimitedLineTokenizer delimitedLineTokenizer = new DelimitedLineTokenizer();
        delimitedLineTokenizer.setNames(new String[] {"firstName", "lastName", "age",
"salary"});

        BeanWrapperFieldSetMapper<Employee> fieldSetMapper = new
BeanWrapperFieldSetMapper<>();
        fieldSetMapper.setTargetType(Employee.class);

        defaultLineMapper.setLineTokenizer(delimitedLineTokenizer);
        defaultLineMapper.setFieldSetMapper(fieldSetMapper);
        reader.setLineMapper(defaultLineMapper);

        return reader;
    }

    @Bean
    ItemProcessor<Employee, Employee> employeeItemProcessor() {
        return new ItemProcessor<Employee, Employee>() {
            @Override
            public Employee process(Employee employee) throws Exception {
                employee.setFirstName(employee.getFirstName().toUpperCase());
                employee.setLastName(employee.getLastName().toUpperCase());
                return employee;
            }
        };
    }

    @Bean
    ItemWriter<Employee> employeeItemWriter() {
        return new ItemWriter<Employee>() {
            @Override
            public void write(List<? extends Employee> employeesList) throws Exception {
                for (Employee employee : employeesList) {
                    System.out.println("Name: "
                            + employee.getFirstName() + " "
```

```
                          + employee.getLastName() + "; "
                          + "Age: " + employee.getAge() + "; "
                          + "Salary: " + employee.getSalary());
            }
        }
    };
    }
}
```

Employee.java :

```java
public class Employee {
    private String firstName;
    private String lastName;
    private int age;
    private int salary;

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public int getSalary() {
        return salary;
    }

    public void setSalary(int salary) {
        this.salary = salary;
    }
}
```

resources employees.csv .

```
John,Doe,35,90000
Sue,Smith,45,95000
Joe,Brown,33,86000
Carol,Dunn,25,75000
Mike,Ward,23,70000
Lisa,Jones,22,69000
```

src/main/java/com/organization/app   .

Main.java :

```java
public class Main {
    public static void main(String[] args) {

        ApplicationContext context = new
AnnotationConfigApplicationContext(SpringBatchHelloWorldConfig.class);

        JobLauncher jobLauncher = context.getBean(JobLauncher.class);
        Job job = context.getBean("listEmployeesJob", Job.class);

        JobParameters jobParameters = new JobParametersBuilder().toJobParameters();

        try {
            JobExecution jobExecution = jobLauncher.run(job, jobParameters);
        }
        catch (JobExecutionAlreadyRunningException e) {
            e.printStackTrace();
        }
        catch (JobRestartException e) {
            e.printStackTrace();
        }
        catch (JobInstanceAlreadyCompleteException e) {
            e.printStackTrace();
        }
        catch (JobParametersInvalidException e) {
            e.printStackTrace();
        }

    }
}
```

: https://riptutorial.com/ko/spring-batch/topic/4089/--

S. No		Contributors
1		badjr, Community