



Бесплатная электронная книга

УЧУСЬ

spring-batch

Free unaffiliated eBook created from
Stack Overflow contributors.

#spring-
batch

.....	1
1:	2
.....	2
.....	2
Examples.....	2
Maven Java Config.....	2
.....	2
Maven	2
Hello World	3
.....	8

Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [spring-batch](#)

It is an unofficial and free spring-batch ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official spring-batch.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

глава 1: Начало работы с весенней периодичностью

замечания

В этом разделе представлен обзор того, что такое Spring-пакет, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в рамках весенней партии и ссылки на связанные темы. Поскольку документация для весеннего пакета является новой, вам может потребоваться создать начальные версии этих связанных тем.

Версии

Версия	Дата выхода
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

Настройка с помощью Maven и Java Config

Предпосылки

- Установили [Maven](#) .
- Установите IDE, например IntelliJ, Eclipse или NetBeans.

Создание проекта Maven

Создайте проект Maven со стандартной структурой проекта (например, ID группы как com.organization.app и Артефакт ID как SpringBatchExample :

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

Создание программы весеннего пакета Hello World

Весеннее пакетное задание требует настройки Spring. В этом примере будет использоваться конфигурация Java.

Настройте зависимости в pom.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>

```

В пакете `src/main/java/com/organization/app` создайте `SpringBatchHelloWorldConfig.java`. Эта программа Spring Batch будет читать список `Employee` с из плоского файла, конвертировать имена в верхний регистр и распечатывать всю информацию `Employee`:

```

@Configuration
@EnableBatchProcessing
public class SpringBatchHelloWorldConfig {

    @Autowired
    private JobBuilderFactory jobBuilderFactory;

    @Autowired
    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 :

```

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;
    }
}

```

В разделе `resources` создайте `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

```

В пакете `src/main/java/com/organization/app` создайте основной класс:

Main.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();
        }
    }
}

```


Прочитайте Начало работы с весенней периодичностью онлайн:

<https://riptutorial.com/ru/spring-batch/topic/4089/начало-работы-с-весенней-периодичностью>

кредиты

S. No	Главы	Contributors
1	Начало работы с весенней периодичностью	badjr , Community