

Spring Batch Cheat Sheet



create a detailed cheat sheet on spring batch with all examples and 10 most asked interview questions on spring batch. I was also asked about cron jobs



Spring Batch Cheat Sheet

Overview of Spring Batch

- **Spring Batch** is a framework for batch processing in Java applications. It provides reusable functions essential for processing large volumes of data, such as reading, transforming, and writing data.

Key Concepts in Spring Batch

1. Job:

- Represents the batch process; composed of steps.
- Defined using `@EnableBatchProcessing` and `JobBuilderFactory`.

2. Step:

- Represents a single phase in a job, e.g., read, process, and write.

3. ItemReader:

- Reads data from a source (file, database, etc.).
- Examples: `FlatFileItemReader`, `JdbcCursorItemReader`.

4. ItemProcessor:

- Processes the data (e.g., transforming, filtering).
- Example: Custom processor by implementing `ItemProcessor`.

5. ItemWriter:

- Writes data to a destination (file, database, etc.).
- Examples: `FlatFileItemWriter`, `JdbcBatchItemWriter`.

6. Execution Context:

- Shared storage for storing and accessing metadata.

Annotations in Spring Batch

1. `@EnableBatchProcessing`: Enables Spring Batch features in a Spring Boot application.
2. `@JobScope`: Scopes a bean to the lifecycle of a job.
3. `@StepScope`: Scopes a bean to the lifecycle of a step.

Spring Batch Configuration Example

```
java
```

```
@Configuration
@EnableBatchProcessing
public class BatchConfig {

    @Autowired
    private JobBuilderFactory jobBuilderFactory;

    @Autowired
    private StepBuilderFactory stepBuilderFactory;

    @Bean
    public Job exampleJob(Step exampleStep) {
        return jobBuilderFactory.get("exampleJob")
            .start(exampleStep)
            .build();
    }

    @Bean
    public Step exampleStep(ItemReader<String> reader,
                           ItemProcessor<String, String> processor,
                           ItemWriter<String> writer) {
        return stepBuilderFactory.get("exampleStep")
            .<String, String>chunk(10) // Defines commit interval
            .reader(reader)
            .processor(processor)
            .writer(writer)
            .build();
    }

    @Bean
    public ItemReader<String> reader() {
        return new ListItemReader<>(List.of("item1", "item2", "item3"));
    }

    @Bean
    public ItemProcessor<String, String> processor() {
        return item -> "Processed " + item;
    }

    @Bean
    public ItemWriter<String> writer() {
        return items -> items.forEach(System.out::println);
    }
}
```

Cron Jobs in Spring Batch

Cron expressions define the schedule for executing a task.

- **Annotation:** `@Scheduled`
- **Example of cron format:** `@Scheduled(cron = "0 0 12 * * ?")` runs daily at noon.

Example of a Cron Job:

```
java
```

```
@Component
public class CronJobExample {

    @Scheduled(cron = "0 0/5 * * * ?") // Executes every 5 minutes
    public void performBatchJob() {
        System.out.println("Batch job executed at: " + LocalDateTime.now());
    }
}
```

```
}  
}
```

10 Most Asked Spring Batch Interview Questions

1. What is Spring Batch, and why is it used?

- A framework for batch processing large data volumes efficiently. It handles reading, processing, and writing data.

2. Explain the structure of a Spring Batch job.

- A job is composed of multiple steps. Each step has an ItemReader, ItemProcessor, and ItemWriter.

3. What are the common components of Spring Batch?

- Job, Step, JobLauncher, JobRepository, ItemReader, ItemProcessor, ItemWriter.

4. How do you implement a retry mechanism in Spring Batch?

- Use the `RetryTemplate` or configure retry in the `Step` using `RetryPolicy`.

5. What is a chunk in Spring Batch?

- A chunk is a set of items processed together. Defined by `chunk(size)` in a step.

6. How does Spring Batch handle transaction management?

- Transactions are managed at the chunk level. If an exception occurs, the entire chunk is rolled back.

7. What is the difference between JobScope and StepScope?

- `@JobScope`: Bean scoped to a job.
- `@StepScope`: Bean scoped to a step.

8. How to schedule a Spring Batch job?

- Use `@Scheduled` or integrate with Quartz Scheduler for more advanced scheduling.

9. How can you restart a failed Spring Batch job?

- Use `JobOperator` to restart the job, leveraging job metadata for resumption.

10. How do you use an external configuration file for Spring Batch?

- Use `JobParameters` to pass parameters from the command line or a configuration file.

Advanced Examples

File-to-Database Example

1. Reading from a File:

```
java  
  
@Bean  
public FlatFileItemReader<Person> reader() {  
    return new FlatFileItemReaderBuilder<Person>()  
        .name("personReader")  
        .resource(new ClassPathResource("data.csv"))  
        .delimited()  
        .names("name", "age")  
        .fieldSetter(new BeanWrapperFieldSetter<>() {{  
            setTargetType(Person.class);  
        }});  
}
```

```

    })
    .build();
}

```

2. Writing to a Database:

java

```

@Bean
public JdbcBatchItemWriter<Person> writer(DataSource dataSource) {
    return new JdbcBatchItemWriterBuilder<Person>()
        .itemSqlParameterSourceProvider(new BeanPropertyItemSqlParameterSourceProvider<>())
        .sql("INSERT INTO person (name, age) VALUES (:name, :age)")
        .dataSource(dataSource)
        .build();
}

```

This cheat sheet provides foundational and advanced concepts with examples to cover your interview preparation on **Spring Batch** and **cron jobs** comprehensively. Let me know if you'd like to delve deeper into any topic!