Students will be introduced using Spring Batch for processing batch jobs.

Introduction

- Introduction to **Batch processing**
- This **use-cases** for batch
- Introduction to **Spring Batch**
- Its relation with **jsr-352**
- Discuss batch concepts (**job**, **step**, etc)
- Understand job execution (**instance**, **execution**)
- Understand the role of **job parameters**
- Define **steps** (**Tasklets** and **Chunks**)
- Understand **step execution**

Your first project

- Set up a Spring Batch **project** (Spring Boot)
- Overview of **XML** and **Java DSL** to define jobs
- Understand the **Tasklet** interface
- Introduction to **StepScope** and **JobScope**
- Understand the **Job Repository**
- Define and use **job parameters**

Steps and Flows

- Define **sequential** flows
- Define **conditional** flows
- Understand the difference between **BatchStatus** and **ExitStatus**
- Using **deciders**
- Define **reusable** flows
- Splitting processing using **split**
- Stopping and interrupting processes (**end**, **stop** and **fail**)
- Define and register various **listeners**
- Comprehend **job execution context** and **step execution context**
- Saving state between job executions and step executions
- Sharing state **between steps**

Chunk-based Processing

- Understand the **components** that make up a chunk
- Understand a chunk step's **life-cycle**
- Define **Item Readers**
- Understand and define **Item Processors**
- Define **Writers**
- Define **Item Stream** implementations

Phoca PDF

- Work with **flat files** (csv, xml, json, …)
- Read **multiple files**
- Write multiple files (**composite** and **classified**)
- Read **database data**
- Understand the different between **cursor** and **paged** processing
- **Writing to databases** (batch)
- Discuss various processing idioms (**transform**, **filter**)
- **Combining** processors
- Data **validation** (including jsr-303)

Application Development

- **Restarting** jobs
- Handling **exceptions**
- Resilience with **retry** logic
- **Skipping** items
- Managing jobs **programmatically** using JobLauncher and JobOperator
- Discuss starting jobs (**rest**, **schedulers** etc.)
- Discuss writing unit and integration **testing**

Scalability

- Improving **throughput** options (chunk-size, commit interval, page size)
- Discuss Scalability options
- Use **Multi-threaded** steps (and understand its caveats)
- Define **asynchronous** processors
- Local and remote **partitioning**
- Discuss **remote chunking**

Phoca PDF