Spring Batch Interview Questions.

In this post we will look at Spring Batch Interview questions. Examples are provided with explanation.

Q:Explain spring batch framework.
A: Spring Batch is a lightweight, comprehensive batch framework designed to enable the development of robust batch applications vital for the daily operations of enterprise systems. Spring Batch builds upon the productivity, POJO-based development approach, and general ease of use capabilities people have come to know from the Spring Framework, while making it easy for developers to access and leverage more advanced enterprise services when necessary.

Q:When to use Spring Batch?
A: Consider an environment where users have to do a lot of batch processing. This will be quite different from a typical web application which has to work 24/7. But in classic environments it's not unusual to do the heavy lifting for example during the night when there are no regular users using your system. Batch processing includes typical tasks like reading and writing to files, transforming data, reading from or writing to databases, create reports, import and export data and things like that. Often these steps have to be chained together or you have to create more complex workflows where you have to define which job steps can be run in parallel or have to be run sequentially etc. That's where a framework like Spring Batch can be very handy. Spring Boot Batch provides reusable functions that are essential in processing large volumes of records, including logging/tracing, transaction management, job processing statistics, job restart, skip, and resource management. It also provides more advanced technical services and features that will enable extremely high-volume and high performance batch jobs though optimization and partitioning techniques.Simple as well as complex, high-volume batch jobs can leverage the framework in a highly scalable manner to process significant volumes of information.
Q:Explain the Spring Batch framework architecture.
A:
This layered architecture highlights three major high level components: Application, Core, and Infrastructure.The application contains all batch jobs and custom code written by developers using Spring Batch. The BatchCore contains the core runtime classes necessary to launch and control a batch job. It includes things such as aJobLauncher,Job, andStepimplementations. Both Application and Core are built on top of a commoninfrastructure. This infrastructure contains common readers and writers, and services

such as theRetryTemplate, which are used both by application developers(ItemReaderandItemWriter) and the coreframework itself.

Q:How Spring Batch works?
A:

- **step -** A Step that delegates to a Job to do its work. This is a great tool for managing dependencies between jobs, and also to modularise complex step logic into something that is testable in isolation. The job is executed with parameters that can be extracted from the step execution, hence this step can also be usefully used as the worker in a parallel or partitioned execution.
- **ItemReader -** Strategy interface for providing the data. Implementations are expected to be stateful and will be called multiple times for each batch, with each call to read() returning a different value and finally returning null when all input data is exhausted. Implementations need not be thread-safe and clients of a ItemReader need to be aware that this is the case. A richer interface (e.g. with a look ahead or peek) is not feasible because we need to support transactions in an asynchronous batch.
- **ItemProcessor -** Interface for item transformation. Given an item as input, this interface provides an extension point which allows for the application of business logic in an item oriented processing scenario. It should be noted that while it's possible to return a different type than the one provided, it's not strictly necessary. Furthermore, returning null indicates that the item should not be continued to be processed.
- **ItemStreamWriter -** Basic interface for generic output operations. Class implementing this interface will be responsible for serializing objects as necessary. Generally, it is responsibility of implementing class to decide which technology to use for mapping and how it should be configured. The write method is responsible for making sure that any internal buffers are flushed. If a transaction is active it will also usually be necessary to discard the output on a subsequent rollback. The resource to which the writer is sending data should normally be able to handle this itself.

Q: What is difference between Step, Tasklet and Chunk in Spring Batch?
A: Spring Batch - Difference between Step, Chunk and Tasklet (/spring/batchtaskchunk)

Q: Have you implement Spring Batch Tasklet? What was the use case?
A: The Tasklet which is a simple interface with just one method execute. Using this we can perform single

tasks like executing queries, deleting files.
Spring Batch Tasklet - Hello World example (/spring/springbatchtasklet)

Q:How to configure Spring Batch with Spring boot?
A: Spring Boot Batch Simple example (/spring/bootbatch)

Q:What is Tasklet in Spring Batch?
A: Spring Batch provides a Tasklet interface, which will be called to perform a single task only, like clean or delete or set up resources before or after any step execution.
Spring Boot +Batch + FileDeleting Tasklet (/spring/springbootbatchtaskscheduler)

Q: How can we schedule a Spring Batch Job?
A: Spring Batch can be scheduled using Cron Job.
Spring Boot +Batch + Scheduler (/spring/springbootbatchtaskscheduler)