## **Spring Batch Interview Questions**

1. **Question 1. Explain Spring Batch Framework.**

**Answer:**

Spring Batch frame work, a collaborative effort from Accenture and SpringSource, is a lightweight, comprehensive framework that facilitates the development of batch applications that helps the day to day activities of enterprise systems. Batch application or processing refers to automated offline systems that performs bulk data processing, periodic updates and delegated processing.

Examples include loading csv file data to database, process feed file once received and push daily transactions to the upstream or downstream systems.

1. **Question 2. List out some of the Practical Usage Scenario of Spring Batch Framework?**

**Answer:**

Reading large number of records from a database, file, queue or any other medium, process it and store the processed records into medium, for example, database.

* + Concurrent and massively parallel processing.
  + Staged, enterprise message-driven processing.
  + Sequential processing of dependent steps.
  + Whole-batch transaction.
  + Scheduled and repeated processing.

1. **Question 3. Technical Advantages of Using Spring Batch Framework from a Developer Perspective?**

**Answer:**

Batch framework leverages spring programming model thus allows developers to concentrate on the business logic or the business procedure and framework facilitates the infrastructure.

Clear separation of concerns between the infrastructure, the batch execution environment, the batch application and the different steps/proceses within a batch application.

Provides common scenario based, core execution services as interfaces that the applications can implement and in addition to that framework provides its default implementation that the developers could use or partially override based on their business logic.

Easily configurable and extendable services across different layers.

Provides a simple deployment model built using Maven.

1. **Question 4. Explain The Spring Batch Framework Architecture?**

**Answer:**

**Spring Batch exhibit a layered architecture and it comprises of three major high level components:**

Application, Core and Infrastructure.

The application layer contains all the batch job configurations, custom codes for business logic and job Meta information developed by Application developers.

The Batch Core has the core runtime classes necessary to launch and control any batch job. Some of the core runtime classes include JobLauncher, Job, and Step implementations.

The infrastructure contains API for common readers and writers, and services for retrying on failure, repeat jobs etc. The infrastructure layer are used both by application developers (ItemReader and ItemWriter) and the core framework itself for controlling the batch job such as Retry, repeat. Thus Batch Core and Application layers are built on top of Infrastructure layer.

1. **Question 5. How Do You Categorize Batch Applications Based On Input Source In Spring Batch?**

**Answer:**

Database-driven applications are driven by rows or values received from the database.

File-driven applications are driven by records or values retrieved from a file.

Message-driven applications are driven by messages retrieved from a message queue.

1. **Question 6. What Are The Typical Processing Strategies In Spring Batch?**

**Answer:**

Normal processing during offline.

Concurrent batch or online processing.

Parallel processing of many different batch or jobs at the same time.

Partitioning (processing of many instances of the same job at the same time).

1. **Question 7. How Do I Start A Spring Batch Job?**

**Answer:**

A Job Launcher can be used to execute a Spring Batch Job. Also a batch job can be launched/scheduled using a web container as well.

Execution of a job is termed as Job Instance. Each Job Instance is provided with an execution id which can be used to restart the job if required.

Job can be configured with parameters which is passed to it from the Job Launcher.

1. **Question 8. What Are The Important Features Of Spring Batch?**

**Answer:**

**Restorability:** Restart a batch program from where it failed.

**Different Readers and Writers:** Provides great support to read from text files, csv, JMS, JDBC, Hibernate, iBatis etc. It can write to JMS, JDBC, Hibernate, files and many more.

**Chunk Processing:** If we have 1 Million records to process, these can be processed in configurable chunks (1000 at a time or 10000 at a time).

Easy to implement proper transaction management even when using chunk processing.

Easy to implement parallel processing. With simple configuration, different steps can be run in parallel.

1. **Question 9. Explain Normal Processing Strategy In Spring Batch Framework?**

**Answer:**

Normal processing refers to the batch processes that runs in a separate batch window, the data being updated is not required by on-line users or other batch processes, where concurrency would not be a concern and a single commit can be done at the end of the batch run.

Single commit point may be a concern in terms of scalability and volume of data it could handle, it is always a good practice to have restart recovery options.

1. **Question 10. Explain Concurrent Batch On-line Processing In Spring Batch Framework?**

**Answer:**

Concurrent/on-line batch processing refers to the batch process that handles data being concurrently used/updated by online users so the data cannot be locked in database or file as the online users will need it. Also the data updates should be committed frequently at the end of few transactions to minimize the portion of data that is unavailable to other processes and the elapsed time the data is unavailable.

1. **Question 11. Explain Parallel Processing In Spring Batch Framework?**

**Answer:**

Parallel processing enables multiple batch runs jobs to run in parallel to reduce the total elapsed batch processing time. Parallel processing is simpler as long as the same file or database table is not shared among the processes otherwise the processes should process partitioned data.

Another approach would be using a control table for maintaining interdependencies and to track each shared resource in use by any process or not.

Other key issues in parallel processing include load balancing and the availability of general system resources such as files, database buffer pools etc. Also note that the control table itself can easily become a critical resource.

1. **Question 12. Explain Partitioning In Spring Batch Framework?**

**Answer:**

Partitioning facilitates multiple large batch applications to run concurrently that minimize the elapsed time required to process long batch jobs. Processes which can be successfully partitioned are those where the input file can be split and/or the main database tables partitioned to allow the application to run against different sets of data.

Processes which are partitioned must be designed to only process their assigned data set.

1. **Question 13. What Is Tasklet In Spring Batch Framework?**

**Answer:**

The Tasklet is an interface which performs any single task such as setup resource, running a Sql update, cleaning up resources etc.

1. **Question 14. How Do I Configure A Job In Spring Batch Framework?**

**Answer:**

A Job in Spring Batch contains a sequence of one or more Steps. Each Step can be configured with the list of parameters/attribute required to execute each step.

**Next:** next step to execute

**Tasklet:** task or chunk to execute. A chunk can be configured with a Item Reader, Item Processor and Item Writer.

**Decision:** Decide which steps need to executed.

1. **Question 15. What Are Spring Batch Metadata Schema?**

**Answer:**

The Spring Batch Meta-Data tables are used to persist batch domain objects such as JobInstance, JobExecution, JobParameters, and StepExecution for internally managing the Batch Jobs.

The JobRepository is responsible for saving and storing each Java object into its correct table

1. **Question 16. Can We Create A Spring Batch With No Step?**

**Answer:**

No. There must exists at least one step or flow or split configuration within a Spring Batch job.

1. **Question 17. What Are The Different Bean Scope In Spring Batch 3.0?**

**Answer:**

Step scope- there is only one instance of such a bean per executing step.

<bean id="..." class="..." scope="step">

Job scope- there is only one instance of such a bean per executing Job.

<bean id="..." class="..." scope="job">

1. **Question 18. How Do We Track The Number Of Item Processed By The ItemReader In Spring Batch?**

**Answer:**

The item mapping bean can implement org.springframework.batch.item. ItemCountAware, a marker interface to have the item position tracked.

1. **Question 19. Define A Job In Spring Batch?**

**Answer:**

A Job is an entity that encapsulates an entire batch process.

Job will be wired together using a XML configuration file or Java based configuration. This configuration is also referred as "job configuration".

A Job is simply a container for Steps and it combines multiple steps that runs logically together in a flow.

1. **Question 20. What Is Job Launcher In Spring Batch Framework?**

**Answer:**

JobLauncher represents a simple interface for launching a Job with a given set of JobParameters.

1. **Question 21. What Is Executioncontext?**

**Answer:**

An ExecutionContext represents a collection of key/value pairs that are persisted and controlled by the framework in order to provide the developers a placeholder to store persistent state that is scoped to a StepExecution or JobExecution.

1. **Question 22. How Do You Run Spring Batch Jobs In Production Environment?**

**Answer:**

Usually The Java batch Job main class and its dependencies are passed to the java command and it is stored in a command line Batch file or shell script in terms of Linux/Unix.

These script file can be run using scheduler like Autosys at the Production environment.

1. **Question 23. What Is Commandlinejobrunner In Spring Batch?**

**Answer:**

CommandLineJobRunner is one of the ways to bootstrap your spring batch Job. The xml script launching the job needs a Java class main method as as entry point and CommandLineJobRunner helps you to start your job directly using the XML script.

**The CommandLineJobRunner performs 4 tasks:**

Load the appropriate ApplicationContext.

Parse command line arguments into JobParameters.

Locate the appropriate job based on arguments.

Use the JobLauncher provided in the application context to launch the job.

The CommandLineJobRunner arguments are jobPath, the location of the XML file that will be used to create an ApplicationContext and the jobName, the name of the job to be run.

Bash$ java CommandLineJobRunner DailyJobConfig.xml processDailyJob

These arguments must be passed in with the path first and the name second. All arguments after these are considered to be JobParameters and must be in the format of 'name=value'.

1. **Question 24. What Is Resourceaware Is Spring Batch?**

**Answer:**

ResourceAware is a marker interface which will set the current resource on any item that implement this interface.

1. **Question 25. Difference between Spring Batch and Quartz Scheduler?**

**Answer:**

Spring Batch and Quartz have different features and responsibility. Spring Batch provides functionality for processing large volumes of data while Quartz provides functionality for scheduling tasks. Thus Quartz could complement Spring Batch and a common combination would be to use Quartz as a trigger for a Spring Batch job using a Cron expression.

1. **Question 26. How Do I Schedule A Job With Spring Batch?**

**Answer:**

Use a scheduling tool such as Quartz, Control-M or Autosys. Quartz is light weight, doesn't have all the features of Control-M or Autosys. Even the OS based Task scheduler, CRON jobs could be used to schedule spring batch jobs.

1. **Question 27. How Can I Make An Item Reader Thread Safe In Spring Batch?**

**Answer:**

You can synchronize the read() method. Remember that you will lose restartability, so best practice is to mark the step as not restartable and to be safe (and efficient) you can also set saveState=false on the reader.

1. **Question 28. What Is The Latest Version Of Spring Batch?**

**Answer:**

The available latest version is 3.0.7.

1. **Question 29. What Is ItemReader In Spring Batch Framework?**

**Answer:**

ItemReader is an abstraction that represents the retrieval of input for a Step, one item/row/record at a time. When the ItemReader has exhausted the items it can provide, it will indicate this by returning null.

1. **Question 30. What Is ItemWriter In Spring Batch Framework?**

**Answer:**

ItemWriter is an abstraction that represents the output of a Step, one batch or chunk of items at a time. Generally, an item writer has no knowledge of the input it will receive next, only the item that was passed in its current invocation.

1. **Question 31. What Is Itemprocessor?**

**Answer:**

ItemProcessor is an abstraction that represents the business processing of an item. While the ItemReader reads one item, and the ItemWriter writes them, the ItemProcessor provides access to transform or apply other business processing. If, while processing the item, it is determined that the item is not valid, returning null indicates that the item should not be written out.

1. **Question 32. Mention The Different Itemreader And Itemwriter Implementations Available In Spring Batch?**

**Answer:**

There are many implementations including the ones that allow read and write operations on,

* + Flat File.
  + Xml.
  + Hibernate Cursor.
  + JDBC.
  + JMS.
  + Hibernate Paging.
  + Stored Procedure.

1. **Question 33. Name Few Of The Domain Buzzwords In Spring Batch?**

**Answer:**

* + Job.
  + JobLauncher.
  + JobRepository.
  + JobInstance.
  + JobExecution.
  + JobParameters.

1. **Question 34. What Job Configuration Consists Of?**

**Answer:**

The job configuration contains,

* + The simple name of the job.
  + Definition and order of the Steps.
  + Configurable global (to all steps) properties such as restartability.

1. **Question 35. What Is JobInstance?**

**Answer:**

A JobInstance represents the concept of a logical job run.

1. **Question 36. Difference between Step and StepExecution?**

**Answer:**

A Step is a domain object that encapsulates an independent and sequential phase of a batch job while a StepExecution represents a single attempt to execute a step.

1. **Question 37. Define Executioncontext?**

**Answer:**

An ExecutionContext represents a collection of key-value pairs that are persisted and controlled by the framework in order to allow developers a place to store persistent state that is scoped to a StepExecution or JobExecution.

1. **Question 38. What Are The Required Dependencies For Configuring A Job?**

**Answer:**

**There are 3 required dependencies:**

Job name,

JobRepository,

And one or more steps.

1. **Question 39. How Do I Setup Spring Batch Job Without Using Xml?**

**Answer:**

Spring 3 enables the ability to configure applications using java instead of XML and from Spring Batch 2.2.0, batch jobs can be configured using the same java config.

**There are 2 components for the java based configuration:**

The @EnableBatchConfiguration annotation and two builders.

@EnableBatchProcessing provides a base configuration for building batch jobs.

The core interface for this configuration is the BatchConfigurer. The default implementation provides the beans to be autowired such as JobRepository, JobLauncher.

1. **Question 40. Explain The Role Of JobRepository In Spring Batch?**

**Answer:**

The JobRepository is used for basic CRUD operations of the various persisted domain objects within Spring Batch, such as JobExecution and StepExecution. It is required by many of the major framework features, such as the JobLauncher, Job, and Step.

1. **Question 41. What Is The Default Isolation Level Of Spring Batch Transactions?**

**Answer:**

It is SERIALIZABLE by default to prevent the same job instance being executed concurrently.

1. **Question 42. What Is A Cron Job?**

**Answer:**

A Cron job is a Linux command for scheduling script on your server to execute repetitive tasks automatically. Scripts executed as a Cron job are typically used to modify files, databases and manage caching.

1. **Question 43. How Cron Job Works In Linux?**

**Answer:**

Cron is a daemon that executes scheduled commands. Cron is started automatically from /etc/init.d on entering multi-user runlevels. Cron searches its spool area (/var/spool/cron/crontabs) for crontab files (which are named after accounts in /etc/passwd); crontabs found are loaded into memory.

Cron wakes up every minute, examining all stored crontabs, checking each command to see if it should be run in the current minute. When executing commands, any output is mailed to the owner of the crontab (or to the user named in the MAILTO environment variable in the crontab, if such exists).

