**SBATCH RETRY NOTE:**

**SCENARIO:**

**Transient Exception:**

Una eccezione transient e’ un ’eccezione legata ad un ostacolo transitorio che potrebbe essere risolto con una nuova esecuzione.

**Esempi:**- Lock su tabella  
- Risorsa di rete non disponibile  
- Time out su una chiamata remota

**Deterministic Exception:**

Eccezione che non puo’ essere rimossa in una nuova esecuzione del job.  
Esempi:  
Dati formalmente incorretti o inconsistenti che impediscono l’elaborazione (da DB o da file)

Tra le Deterministic Exception si possono identificare quelle **Skippable**.  
La singola operazione (il singolo record) viene saltato e si continua col prossimo.

**CHUNK PROCESSING:**

**ItemReader:**

Le eccezioni **retryable** devono essere anche Skippable se vengono lanciate su ItemReader.  
Anche perche’ se la lettura fallisce si puo’ solo pensare di saltarla e riprovare.

**ItemProcessor, ItemWriter:**

Le eccezioni **retryable** comportano la ripetizione della singola operazione (non dello step).  
Per cui se il reader ha prodotto dei dati, la ripetizione del processor o del writer si basera’ sugli stessi dati.  
La transitorieta’ dell’ eccezione deve essere legata alla operazione di **<<process>>** o **<<write>>**(Lock record che si cerca di aggiornare, servizio per update non disponibile)

**TASKLET:**In base a:   
http://stackoverflow.com/questions/27509585/spring-batch-skippable-exception-in-a-taskletstep

Within a Tasklet, the responsibility for exception handling is on the implementation of the Tasklet. The skip logic available in chunk oriented processing is due to the exception handling provided by the ChunkOrientedTasklet. If you want to skip exceptions in your own Tasklet implementation, you need to write the code to do so in within your own implementation.

**TUTORIAL**Classe abstract da cui ereditano i Chunk Item, gestisce conf per eccezioni.  
Commentare/Decommentare retriable e Skippable exceptions in onguna delle classi**.**

**CHUNK ORIENTED:**

**1) Solo reader e writer:**Eccezioni lanciate read  
Ris Atteso:  
  
**2) Lanciare retriable exception in processor**Eccezioni lanciate in Processor (decommentarle)  
Ris Atteso: Ripetizioni inutili perche’ dato in input non cambia piu.  
Prova cache item readers.  
  
**3) Lanciare retriable exception in Writer**:  
Eccezioni lanciate in Writer (simula db con lock o servizio momentaneamente non disp)

**3) Listeners:**  
Implementare interfaccia  
extendere Classi Support (adapter con metodi vuoti)  
Annotations su classe Item<<Operation>>

**TASKLET:**

**3) RetryTemplate  
 4) SkipTemplate**