Spring-Hibernate Training Program
Session 11

BoTree Consulting

Hibernate Caching

- First Level Cache
  - Transaction Level Cache
- Second Level Cache
  - Session Factory Level Cache

    The second-level cache holds on to the 'data' for all properties and associations (and collections if requested) for individual entities that are marked to be cached.
- Query Cache
  - Disabled by default
  - Limited benefits, depends on the usage pattern.

First Level Cache

- Enabled by default
- Handled by Hibernate, little roleof the user.
- save(), update(), saveOrUpdate(), load(), get(), list(), iterate(), scroll() adds object to internal Session cache.
- flush() synchronizes with database.
- evict() removes object from first level cache.
- contains() helps determine if an instance belongs to session cache.

Second Level Cache I

- As caches are not aware of the changes made to the persistent store, its utility depends heavily on the usage pattern.
- Recommended for data that changes infrequently and is fetched frequently.
- By default entities are not a part, recommended.
- Shared-cache-mode - @Cache
  - ENABLE_SELECTIVE
  - DISABLE_SELECTIVE
  - ALL
  - NONE

Second Level Cache II

- Cache Concurrency Strategy - dafault_cache_concurency_strategy
  - read-only
  - read-write
  - nonstrict-read-write
  - Transactional
- @Cacheable – javax.persistence - boolean
- @Cache – org.hibernate.annotations
- SessionFactory.evict()
- SessionFacatory.evictCollection()

Second Level Cache III

- Cache Mode controls how a particular *session* interacts with the second-level cache
  - CahceMode.NORMAL - read/write
  - CacheMode.GET - read
  - CacheMode.PUT - write

Query Cache

- Enabling Query Cache creates two regions
  - org.hibernate.cache.StandardQueryCache – holds cached query results
  - org.hibernate.cache.UpdateTimestampsCache – holds timestamps of the most recent updates to queryable tables. These are used to validate the results as they are served from the query cache. Do not set expiry.
- Setting a collection as cacheable creates a cache region with the name of the collection. 'books' in case of Author class.
- It is possible to set our own names for cache regions using setCacheRegion() for specific queries.
- Set cache expiry or timeouts.

Query Cache

- Does not cache the state of actual Entities, instead caches identifier values and results of value type. Therefore, always use the query cache in conjunction with the second-level cache for those entities which should be cached as part of a query result cache.

Statictics

- getStatistics()

End of
Spring – Hibernate Training Program
Session 11