**Enumeration constants and summary**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr | Enumeration | Values | Description |
| 1 | AccessType | FIELD | Field based access is used |
|  |  | PROPERTY | Property based access is used |
|  |  |  |  |
| 2 | CascadeType | ALL | Cascade all operations |
|  |  | DETACH | Cascade Detach operations |
|  |  | MERGE | Cascade merge operations |
|  |  | PERSIST | Cascade persist operations |
|  |  | REFRESH | Cascade refresh operations |
|  |  | REMOVE | Cascade remove operations |
|  |  |  |  |
| 3 | DiscriminatorType | CHAR | Single character as discriminator type |
|  |  | INTEGER | Integer as discriminator type |
|  |  | STRING | String as discriminator type |
|  |  |  |  |
| 4 | EnumType | ORDINAL | Persist enumerated type property or field as an integer |
|  |  | STRING | Persist enumerated type property of field as a string |
|  |  |  |  |
| 5 | FetchType | EAGER | Defines that data must be eagerly fetched |
|  |  | LAZY | Defines that data must be lazily fetched |
|  |  |  |  |
| 6 | FlushModeType | AUTO | Flushing to occur at query execution |
|  |  | COMMIT | Flushing to occur at transaction commit |
|  |  |  |  |
| 7 | GenerationType | AUTO | Indicates that persistence provider should pick an appropriate strategy for the particular database |
|  |  | IDENTITY | Indicates that the persistence provider must assign primary keys for the entity using a database identity column |
|  |  | SEQUENCE | Indicates that the persistence provider must assign primary keys for the entity using a database sequence |
|  |  | TABLE | Indicates that the persistence provider must assign primary keys for the entity using an underlying database table to ensure uniqueness |
|  |  |  |  |
| 8 | InheritanceType | JOINED | A strategy in which fields that are specific to subclass are mapped to a separate table than the fields that are common to parent class and a join is performed to instantiate the subclass |
|  |  | SINGLE\_TABLE | A single table per class hierarchy |
|  |  | TABLE\_PER\_CLASS | A table per concrete entity class |
|  |  |  |  |
| 9 | LockModeType | NONE | No lock |
|  |  | OPTIMISTIC | Optimistic lock |
|  |  | OPTIMISTIC\_FORCE\_INCREMENT | Optimistic lock with version update |
|  |  | PESSIMISTIC\_FORCE\_INCREMENT | Pessimistic write lock with version update |
|  |  | PESSIMISTIC\_READ | Pessimistic read lock |
|  |  | PESSIMISTIC\_WRITE | Pessimistic write lock |
|  |  | READ | Synonymous with OPTIMISTIC |
|  |  | WRITE | Synonymous with OPTIMISTIC\_FORCE\_INCREMENT |
|  |  |  |  |
| 10 | PersistenceContextType | EXTENDED | Extended persistence context |
|  |  | TRANSACTION | Transaction scoped persistence context |
|  |  |  |  |
| 11 | PessimisticLockScope | EXTENDED | In addition to behavior for PessimisticLockScope.NORMAL, element collections and relationships owned by the entity that are contained in join tables will be locked if javax.persistence.lock.scope property is specified with the value of PessimisticLockScope.EXTENDED |
|  |  | NORMAL | This value defines the default behavior for pessimistic locking |
|  |  |  |  |
| 12 | SharedCacheMode | ALL | All entities and entity related state and data are cached |
|  |  | DISABLE\_SELECTIVE | Caching is enabled for all entities except those for which Cacheable(false) is specified |
|  |  | NONE | Caching is disabled for the persistence unit |
|  |  | UNSPECIFIED | Caching behavior is undefined; provider specific defaults may apply |

**Exception types and summary**

|  |  |  |
| --- | --- | --- |
| Sr. | Exception | Summary |
| 1 | EntityExistsException | Thrown by persistence provider when EntityManager.persist(Object) is called and entity already exists |
| 2 | EntityNotFoundException | Thrown by persistence provider when an entity reference obtained by EntityManager.getReference is accessed but the entity does not exist. |
| 3 | LockTimeoutException | Thrown by persistence provider when a pessimistic locking conflict occurs that does not result in transaction roll back |
| 4 | NonUniqueResultException | Thrown by persistence provider when Query.getSingleResult() or TypedQuery.getSingleResult() is executed on a query and there is more than one result from the query |
| 5 | NoResultException | Thrown by the persistence provider when Query.getSingleResult() or TypedQuery.getSingleResult() is executed on a query and there is no result to return |
| 6 | OptimisticLockException | Thrown by the persistence provider when an optimistic locking conflict occurs |
| 7 | PersistenceException | Thrown by the persistence provider when a problem occurs |
| 8 | PessimisticLockException | Thrown by the persistence provider when an pessimistic locking conflict occurs |
| 9 | QueryTimeoutException | Thrown by the persistence provider when a query times out and only the statement is rolled back |
| 10 | RollbackException | Thrown by persistence provider when EntityTransaction.commit() fails |
| 11 | TransactionRequiredException | Thrown by persistence provider when a transaction is required but is not active |
|  |  |  |