getOne()

returns a reference to the entity with the given identifier. getOne internally invokes EntityManager.getReference() method. As per docs, this method will always return a proxy without hitting the database (lazily fetched). This method will throw EntityNotFoundException at the time of actual access if the requested entity does not exist in the database.

Hibernate: select user0\_.id as id1\_3\_0\_, user0\_.name as name2\_3\_0\_, user0\_.national\_id as national3\_3\_0\_, user0\_.phone\_number as phone\_nu4\_3\_0\_, user0\_.surname as surname5\_3\_0\_, user0\_.age as age6\_3\_0\_, user0\_.email as email7\_3\_0\_, user0\_.project\_id as project\_8\_3\_0\_ from usertbl user0\_ where user0\_.id=?

Hibernate: select project0\_.id as id1\_2\_0\_, project0\_.name as name2\_2\_0\_ from project project0\_ where project0\_.id=?

findById()

This method will actually hit the database and return the real object mapping to a row in the database. It is EAGER loaded operation that returns null if no record exists in database.

Hibernate: select user0\_.id as id1\_3\_0\_, user0\_.name as name2\_3\_0\_, user0\_.national\_id as national3\_3\_0\_, user0\_.phone\_number as phone\_nu4\_3\_0\_, user0\_.surname as surname5\_3\_0\_, user0\_.age as age6\_3\_0\_, user0\_.email as email7\_3\_0\_, user0\_.project\_id as project\_8\_3\_0\_ from usertbl user0\_ where user0\_.id=?

Which one to choose?

The only real difference between these methods is about the performance. Lazily loaded getOne() method avoids database roundtrip from the JVM as it never hits the database until the properties of returned proxy object are actually accessed.

There are scenarios when you just want to get retrieve an entity from database and assign it as a reference to another object, just to maintain the relationship (OneToOne or ManyToOne).

Notice => if you want to use getOne() for returning a reference to the entity you must use fetch = FetchType.Lazy