**Difference between get and load in Hibernate?**

* **Get – sessioncache-DB-return fully intialzed object.- if not null.**
* **Load- proxy –DB but not fully initialated only with ID – on demand loading - ObjectNotFoundException**

**Employee** Employee = (Employee) session.get(Employee.**class**, EmployeeID);

**Employee** Employee = (Employee) session.load(Employee.**class**, EmployeeID);

get will hit the database if object is not found in the cache and returned completely initialized object, which may involve several database call while load() method can return proxy, if object is not found in cache and only hit database if any method other than getId() is called. This can save lot of performance in some cases.

This [lazy initialization](http://javarevisited.blogspot.sg/2011/03/10-interview-questions-on-singleton.html) can save couple of database round-trip which result in better performance.

**1. Behavior when Object is not found in Session Cache**

Apart from performance this is another difference between get and load which is worth remembering. get method of Hibernate Session class returns null if object is not found in cache as well as on database while load() method [throws](http://javarevisited.blogspot.sg/2012/02/difference-between-throw-and-throws-in.html) ObjectNotFoundException if object is not found on cache as well as on database but never return null.

**2. Database hit**

Get method always hit [database](http://javarevisited.blogspot.sg/2011/10/how-to-use-truncate-and-delete-command.html) while load() method may not always hit the database, depending upon which method is called.

**3. Proxy**

Get method never returns a proxy, it either returns null or fully initialized Object, while load() method may return proxy, which is the object with ID but without initializing other properties, which is lazily initialized. If you are just using returned object for creating relationship and only need Id then load() is the way to go.

**4. Performance**

By far most important difference between get and load in my opinion. get method will return a completely initialized object if  Object is not on the cache but exists on [Database](http://javarevisited.blogspot.sg/2011/10/selct-command-sql-query-example.html), which may involve multiple round-trips to database based upon object relational mappings while load() method of Hibernate can return a **proxy** which can be initialized on demand (lazy initialization) when a non identifier method is accessed. Due to above reason use of load method will result in slightly **better performance**, but there is a caveat that proxy object will throw **ObjectNotFoundException** later if corresponding row doesn’t exists in database, instead of failing immediately so not a [fail fast](http://javarevisited.blogspot.sg/2012/02/fail-safe-vs-fail-fast-iterator-in-java.html) behavior.

**5.** load method exists prior to get method which is added on user request.

**Difference between save and persist method in Hibernate**

In last section we saw What are difference between save and saveOrUpdate and now we will see Difference on save vs persist method.

1)First difference between save and persist is there return type. Similar to save method persist also INSERT records into database but **return type of persist is void** while return type of save is [Serializable](http://javarevisited.blogspot.sg/2012/01/serializable-externalizable-in-java.html) object.

2) Another difference between persist and save is that both methods make a [transient](http://javarevisited.blogspot.sg/2012/03/difference-between-transient-and.html) instance persistent. However, persist() method doesn't guarantee that the identifier value will be assigned to the persistent instance immediately, the assignment might happen at flush time.

3) One more thing which differentiate persist and save method in Hibernate is that is there behavior on outside of transaction boundaries. persist() method guarantees that it will not execute an INSERT statement if it is called outside of [transaction boundaries](http://javarevisited.blogspot.sg/2011/11/database-transaction-tutorial-example.html). save() method does not guarantee the same, it returns an identifier, and if an INSERT has to be executed to get the identifier (e.g. "identity" generator), this INSERT happens immediately, no matter if you are inside or outside of a transaction

4) Fourth difference between save and persist method in Hibernate is related to previous difference on save vs persist. Because of its above behavior of persist method outside transaction boundary, its useful in long-running conversations with an extended Session context. On the other hand save method is not good in a long-running conversation with an extended Session context.

**Difference between save and saveOrUpdate in Hibernate**

Main *difference between save and saveOrUpdate method* is that save() generates a **new identifier** and INSERT record into [database](http://javarevisited.blogspot.sg/2011/10/selct-command-sql-query-example.html) while saveOrUpdate can either INSERT or UPDATE based upon existence of record. Clearly saveOrUpdate is more flexible in terms of use but it involves an extra processing to find out whether record already exists in table or not. In summary save() method saves records into database by INSERT SQL query, Generates a new identifier and return the [Serializable](http://javarevisited.blogspot.sg/2011/04/top-10-java-serialization-interview.html) identifier back. On the other hand saveOrUpdate() method either INSERT or UPDATE based upon existence of object in database. If persistence object already exists in database then UPDATE SQL will [execute](http://javarevisited.blogspot.sg/2012/03/how-to-create-and-execute-jar-file-in.html) and if there is no corresponding object in database than INSERT will run.