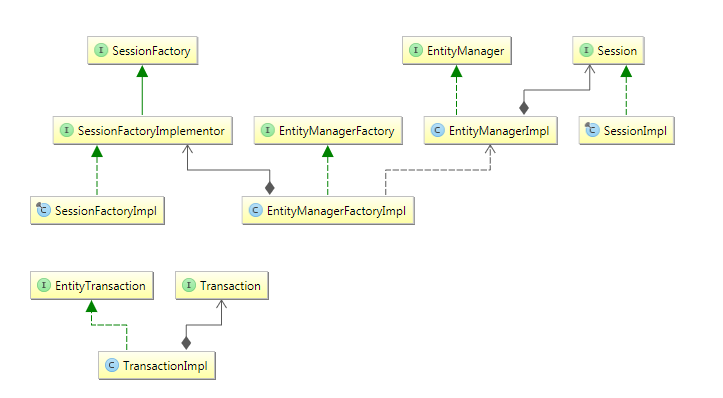
## Architecture

****

SessionFactory (org.hibernate.SessionFactory)

A thread-safe (and immutable) representation of the mapping of the application domain model to a database. Acts as a factory for org.hibernate.Session instances. The EntityManagerFactory is the JPA equivalent of a SessionFactory and basically those two converge into the same SessionFactory implementation.

A SessionFactory is very expensive to create, so, for any given database, the application should have only one associated SessionFactory. The SessionFactory maintains services that Hibernate uses across all Session(s)such as second level caches, connection pools, transaction system integrations, etc.

Session (org.hibernate.Session)

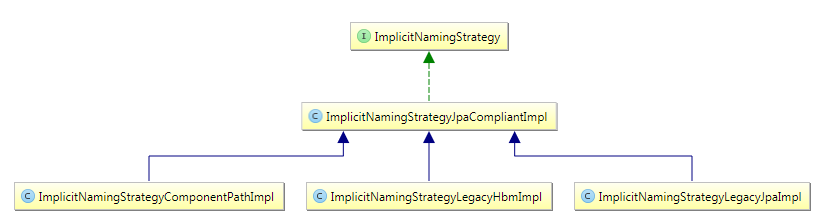
A single-threaded, short-lived object conceptually modeling a "Unit of Work" [PoEAA](file:///E:/java_OpenSource/hibernate-release-5.2.16.Final/documentation/userguide/html_single/Hibernate_User_Guide.html" \l "PoEAA). In JPA nomenclature, the Sessionis represented by an EntityManager.

Behind the scenes, the Hibernate Session wraps a JDBC java.sql.Connection and acts as a factory for org.hibernate.Transaction instances. It maintains a generally "repeatable read" persistence context (first level cache) of the application domain model.

Transaction (org.hibernate.Transaction)

A single-threaded, short-lived object used by the application to demarcate individual physical transaction boundaries.EntityTransaction is the JPA equivalent and both act as an abstraction API to isolate the application from the underlying transaction system in use (JDBC or JTA).

## Domain Model

* Mapping types  
  Hibernate understands both the Java and JDBC representations of application data. The ability to read/write this data from/to the database is the function of a Hibernate type. A type, in this usage, is an implementation of the org.hibernate.type.Type interface. This Hibernate type also describes various aspects of behavior of the Java type such as how to check for equality, how to clone values, etc.
* Naming strategies
  + ImplicitNamingStrategy  
    
  + PhysicalNamingStrategy