JPA Entity Lifecycle

1. Transient
2. Managed
3. Detached
4. Removed

As a reminder the purpose of the EntityManager is to the relationship between the JPA entity and the underlying datasource.

The above diagram shows the 5 key stages of JPA entity management –

Object Doesnt Exist – This is a null object

|  |  |
| --- | --- |
| 1 | MyObject myObject = null; |

New Object – Not associated with the EntityManager, and doesnt exist on database

|  |  |
| --- | --- |
| 1 | MyObject myObject = new MyObject(); |

Managed – This is the stage were the object becomes persisted and managed by the EntityManager. To do this we need to call the persist method from within a transaction. The object is then persisted to the database when the commit method is called

|  |  |
| --- | --- |
| 1  2  3  4 | entityManager.getTransaction().begin();                             MyObject myObject = new MyObject();                             entityManager.persist(myObject);                             entityManager.getTransaction().commit(); |

Detached – This state removes the object from the EntityManager, but the object **still** exists on the database. Some EntityManager methods on a detached object will result in an IllegalArgumentException. The object can be reattached to the EntityManager through the merge method

|  |  |
| --- | --- |
| 1 | entityManager.detach(myObject); |

* Removed – Deletes the object from the database. Like persist this also needs to take place inside a transaction.

|  |  |
| --- | --- |
| 1  2  3 | entityManager.getTransaction().begin();                             entityManager.removed(myObject);                             entityManager.getTransaction().commit(); |