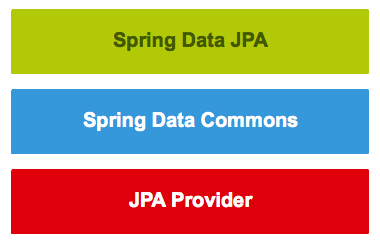
**Spring JPA**

The Spring application contains 3 layers as described below:

* Spring data JPA – provides support for creating repositories by extending Spring data repositories interfaces.
* Spring data commons – Provides infrastructure to support datastore specifications.
* Java persistence API.

[](https://www.petrikainulainen.net/wp-content/uploads/springdatajpalayers.png)

The spring data Commons projects provide following interfaces:

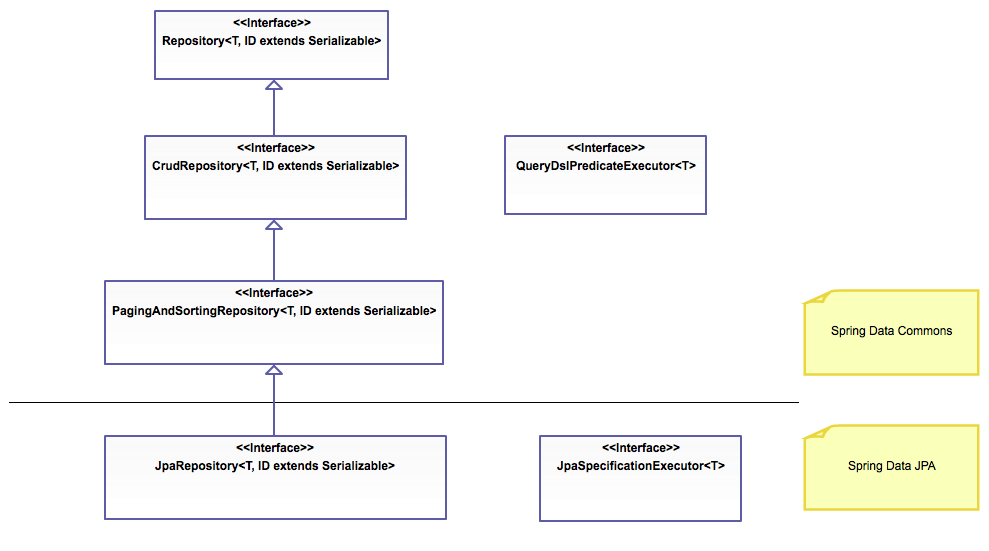
* **Repository<T,ID>** is marker interface, purpose of which is to capture the Managed Entity and type of the primary ID. And enables Spring container to discover the concrete implementation for the Repository.
* **CrudRepository<T,ID>** provides CRUD operation for the Entity.
* **PagingAndSortingRepository<T,ID>** declares method that are used to sort and paginate the entities that are retrieved from the database.
* **JPARepository<T,ID>** combines the method declared by the common repository interface behind a single interface.

Apart from above mentioned repositories we have following repositories interfaces:

* **QueryDslPredicateExecutor<T>** Declares method that are used to retrieve entities from the database by using **QueryDsl** predicate objects.
* **JpaSpecificationExecutor<T>** declares the methods to retrieve objects from database by using **Specification<T>** objects using JPA criteria API.

The implementation for JPA specifications is provided by vendors as below:

* Hibernate
* TopLink
* iBatis
* OpenJpa



**Dependencies needed for SPRING JPA :**

* Spring-data-jpa
* Hibernate-entity-manager
* Connectors.