Criteria

QueryDsl

Spring Data

```java
public interface UserRepository extends JpaRepository<User, long> {


    // JPQL
    @Query("SELECT u FROM User u WHERE u.status = 1")
    Collection<User> findAllActiveUsers();

    // Native
    @Query(value = "SELECT * FROM USERS u WHERE u.status = 1",nativeQuery = true)
    Collection<User> findAllActiveUsersNative();

    // Query creation from method names
    List<User> findByFirstnameContainingAndLastnameIsEndingWithAndAgeGreaterThan(...);

}
```

Main purpose

Provide a **type-safe** way to express a query

Criteria API / QueryDsl

Criteria API

Defined by JPA 2.0, alternative way of defining a JPQL query

Criteria API

```java
LocalDate today = new LocalDate();

CriteriaBuilder builder = entityManager.getCriteriaBuilder();
CriteriaQuery<Customer> query = builder.createQuery(Customer.class);
Root<Customer> customer = query.from(Customer.class);

Predicate hasBirthday = builder.equal(customer.get(Customer_.birthday), today);
Predicate isLongTermCustomer = builder.lessThan(customer.get(Customer_.createdAt), today.minusYears(2);

query.where(builder.and(hasBirthday, isLongTermCustomer));

entityManager.createQuery(query.select(customer)).getResultList();
```

Specifications

```
public interface Specification<T> {
  Predicate toPredicate(Root<T> root, CriteriaQuery query, CriteriaBuilder cb);
}
```

Define reusable Predicates

```
public CustomerSpecifications {

  public static Specification<Customer> customerHasBirthday() {

    return new Specification<Customer> {

      public Predicate toPredicate(Root<T> customer, CriteriaQuery query, CriteriaBuilder cb) {
        return cb.equal(customer.get(Customer_.birthday), today);
      }

    };

  }

  ...
}
```

How will we execute these specifications?

```java
public interface CustomerRepository extends JpaRepository<Customer>, JpaSpecificationExecutor {
  // Your query methods here
}
```

```java
customerRepository.findAll(where(customerHasBirthday()).and(isLongTermCustomer()));
```

QueryDsl

Write cleaner and more concise persistence code and domain logic

QueryDsl

```java
LocalDate today = new LocalDate();

JPAQueryFactory query = new JPAQueryFactory(entityManager);
QCustomer customer = QCustomer.customer;

BooleanExpression customerHasBirthday = customer.birthday.eq(today);
BooleanExpression isLongTermCustomer = customer.createdAt.lt(today.minusYears(2));

query.selectFrom(customer).where(customerHasBirthday.and(isLongTermCustomer)).fetch();
```

How will we execute Querydsl predicates ?

```
public interface CustomerRepository extends JpaRepository<Customer>, QueryDslPredicateExecutor {
  // Your query methods here
}
```

```
customerRepository.findAll(customerHasBirthday.and(isLongTermCustomer));
```

Final result

```java
LocalDate today = new LocalDate();

QCustomer customer = QCustomer.customer;

BooleanExpression customerHasBirthday = customer.birthday.eq(today);
BooleanExpression isLongTermCustomer = customer.createdAt.lt(today.minusYears(2));

customerRepository.findAll(customerHasBirthday.and(isLongTermCustomer));
```