

Criteria API

```
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?

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

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

QueryDsl

```
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

```
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));
```