

Using Additional Spring Data Query Options



Dan Bunker

CONSULTANT | SAAS BUILDER | MENTOR

linkedin.com/in/bunkerdan

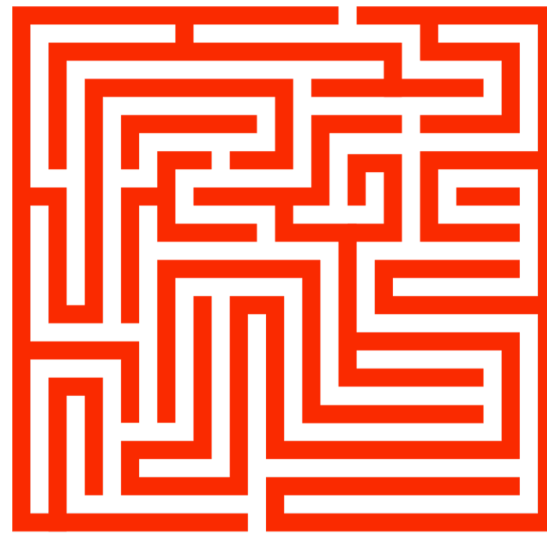
Clip 2 - Overview Placeholder

Demo: @Query Annotation

Understanding the @Query Annotation



Reuse existing JPQL



Advanced queries



Eager loading control

```
@Query("select tp from TicketPrice tp where tp.basePrice < ?1 " +  
      "and tp.ticketType.includesWorkshop = true")  
List<TicketPrice> getTicketsUnderPriceWithWorkshops(BigDecimal maxPrice);
```

@Query Options

Named Parameters

```
@Query("select tp from TicketPrice tp where tp.basePrice < :maxprice " +  
      "and tp.ticketType.includesWorkshop = true")  
List<TicketPrice> getTicketsUnderPriceWithWorkshops(@Param("maxprice") BigDecimal maxPrice);
```

Enhanced JPQL Syntax

```
@Query("select s from Session s where s.sessionName like %?1")  
List<Session> getSessionsWithName(String name);
```

@Query Options

Native SQL Queries

```
@Query(value = "select * from sessions s where s.session_name = ?0", nativeQuery = true)  
List<Session> getSessionWithName(String name);
```

Modifiable Queries

```
@Modifying  
@Query("update Session s set s.sessionName = ?1")  
int updateSessionName(String name);
```

Using JPA @NamedQueries

Queries validated at app startup

```
@Entity
@NamedQuery(
    name = "TicketPrice.namedFindTicketsByPricingCategoryName",
    query = "select tp from TicketPrice tp where tp.pricingCategory.pricingCategoryName = :name"
)
public class TicketPrice {...}
```



```
public interface TicketPriceJpaRepository extends JpaRepository<TicketPrice, Long> {
    List<TicketPrice> namedFindTicketsByPricingCategoryName(@Param("name") String name);
}
```

```
@Query(name = "TicketPrice.namedFindTicketsByPricingCategoryName")
List<TicketPrice> getTicketsByPricingCategoryName(@Param("name") String name);
```

Clip 6 - Overview Placeholder

Demo: NamedQueries

Querying with Native SQL

Native SQL Queries

```
@Query(value = "select * from sessions s where s.session_name = ?0", nativeQuery = true)  
List<Session> getSessionWithName(String name);
```

@NamedNativeQuery

```
@NamedNativeQuery(  
    name = "TicketPrice.nativeFindTicketsByCategoryWithWorkshop",  
    query = "select * from ticket_prices tp " +  
        "left join ticket_types tt on tp.ticket_type_code = tt.ticket_type_code " +  
        "left join pricing_categories pc on tp.pricing_category_code = pc.pricing_category_code " +  
        "where tt.includes_workshop = true and pc.pricing_category_name = :name"  
)  
public class TicketPrice {...}
```

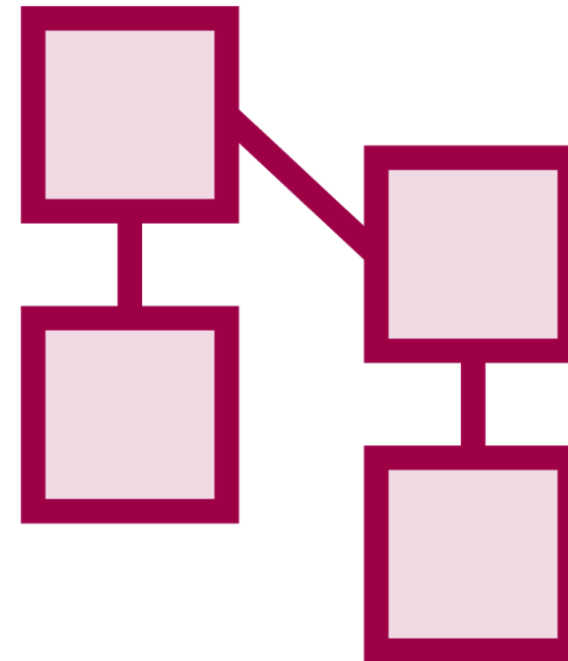

Clip 8 - Overview Placeholder

Demo: NamedNativeQueries

Tip: Where Should I Put My Queries?

SQL
JPQL
Query DSL

JPA Entities



@NamedQuery

@NamedNativeQuery

@NamedQuery

@NamedNativeQuery

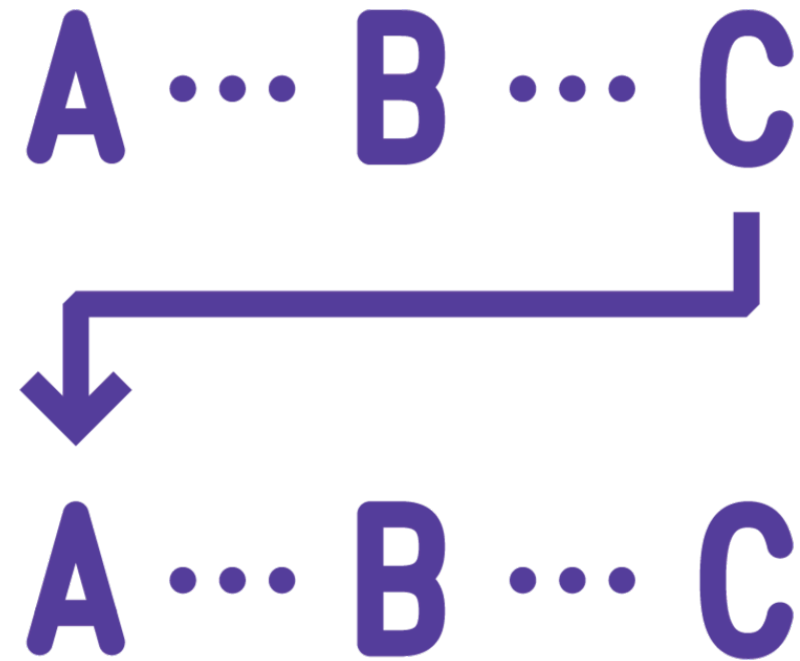
@Query

Query DSL



Repositories

JpaRepository Query Precedence



Methods with @Query annotation take highest precedence

Methods that match a named or native named query

Methods that follow the query DSL keyword naming structure

Summary

@Query annotation

JPA NamedQueries

SQL NamedNativeQueries