

Advanced Topics in JPA/Spring

Martin Ledvinka, Miroslav Blaško

KBSS

Winter Term 2024



Contents

1 Spring

2 Querydsl



Spring



Spring Data Repositories

```
interface UserRepository extends JpaRepository<User, Long> {  
  
    User findByUsername(String username);  
  
    List<User> findByLastName(String lastName, Sort sort);  
  
    Page<User> findByFirstName(String firstName, Pageable pageable);  
  
    @Query("SELECT u FROM User u WHERE u.role = 'ADMIN'")  
    List<User> findAllAdmins();  
}
```

Enable

```
@EnableJpaRepositories("cz.cvut.kbss.ear.eshop.dao")  
public class AppConfig {  
    ...  
}
```



Specifications – Reusable Predicates

Specification interface

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

Implementation of specifications

```
public static Specification<Product>
    hasNameLike(String name) {
    return (root, query, cb) ->
        cb.like(root.get(Product_.name),
            name);
    }
};
```

Definition of a repository

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

Usage of repository

```
dao.findAll(hasBirthday());
dao.findAll(isLongTermCustomer());
dao.findAll(hasBirthday(),
    isLongTermCustomer());
```



Querydsl



Querydsl

- Unified query framework for Java
- Support for SQL, JPA, JDO, Mongo etc.
- Nice and succinct syntax
- Static metamodel for type safety and readability

```
List<Person> persons = queryFactory.selectFrom(person)
    .where(
        person.firstName.eq("John"),
        person.lastName.eq("Doe"))
    .orderBy(person.lastName.asc())
    .fetch();
```



Resources

- <https://docs.oracle.com/javaee/6/tutorial/doc/gjrij.html>
- <https://www.baeldung.com/spring-data-criteria-queries>
- <https://spring.io/blog/2011/04/26/advanced-spring-data-jpa-specifications-and-querydsl>
- <http://www.querydsl.com/>

