GlobalLogic®

Spring Data and Specification

Vinicius Antonio Gai
https://www.linkedin.com/in/vinicius-antonio-gai-25361823/
https://github.com/viniciustoni

GlobalLogic

Who am I?

Who am I?

- Vinicius Antonio Gai
- Curitiba - Paraná - Brazil
- Living in krakow for almost 3 years
- 13 years coding for food(beer)
- 11 years Java environment
- Visual Basic, ASP, Java, Maven, MongoDb, EJB2, EJB3/3.1, Spring4/5, Oracle, Postgres, GCP, etc..
- From E-commerces app to Bank and ERP apps
- GLT HSBC, Ed. Abril, Netshoes, Contabilizei, Luxoft, GlobalLogic

GlobalLogic

Agenda

Agenda

- Statements/Prepared Statement
- Hibernates/JPA
- What is Spring data?
- What is Specification?
- Spring Data JPA + Specifications
- What is QueryDSL?
- Spring Data JPA + Specifications + QueryDSL
- Example - Code
- So… Why should I use it?

GlobalLogic

Statements / Prepared Statements

Statements/Prepared Statement

- Limited, for example:
  - Not possible multiple values per “?” - IN clause
  - Not possible to re-use the same “?”
  - Parameter should be in correct order of “?”
  - Hard to perform a DB migration(SQL dialect)
- Deal with resultSet and mappers(Query to Java Object);
- Native SQL;

GlobalLogic

Statements/Prepared Statement

```
PreparedStatement updateSales = null;
PreparedStatement updateTotal = null;

String updateString =
    "update " + dbName + ".COFFEES " +
    "set SALES = ? where COF_NAME = ?";

String updateStatement =
    "update " + dbName + ".COFFEES " +
    "set TOTAL = TOTAL + ? " +
    "where COF_NAME = ?";
```

```
try {
    con.setAutoCommit(false);
    updateSales = con.prepareStatement(updateString);
    updateTotal = con.prepareStatement(updateStatement);

    for (Map.Entry<String, Integer> e : salesForWeek.entrySet()) {
        updateSales.setInt(1, e.getValue().intValue());
        updateSales.setString(2, e.getKey());
        updateSales.executeUpdate();
        updateTotal.setInt(1, e.getValue().intValue());
        updateTotal.setString(2, e.getKey());
        updateTotal.executeUpdate();
        con.commit();
    }
} catch (SQLException e ) {
    JDBCTutorialUtilities.printSQLException(e);
    if (con != null) {
        try {
            System.err.print("Transaction is being rolled back");
            con.rollback();
        } catch(SQLException excep) {
            JDBCTutorialUtilities.printSQLException(excep);
        }
    }
} finally {
    if (updateSales != null) {
        updateSales.close();
    }
    if (updateTotal != null) {
        updateTotal.close();
    }
    con.setAutoCommit(true);
}
```

GlobalLogic

Statements/Prepared Statement

```
List possibleValues = ...
StringBuilder builder = new StringBuilder();

for( int i = 0 ; i < possibleValue.size(); i++ ) {
    builder.append("?,");
}

String stmt = "select * from test where field in ("
              + builder.deleteCharAt( builder.length() -1 ).toString() + ")";
PreparedStatement pstmt = ...
```

```
int index = 1;
for( Object o : possibleValue ) {
   pstmt.setObject(  index++, o ); // or whatever it applies
}
```

GlobalLogic

Hibernate / JPA

Hibernate/JPA

- Easy mapping our Java Object x Table;
- HSQL - based on entities and not on tables;
- Auto generate your database schema;
  - Actually I do not recommend to use this option;
    - Admin access;
    - Can drop your database(create-drop);
- 2 levels of caching;

Hibernate/JPA

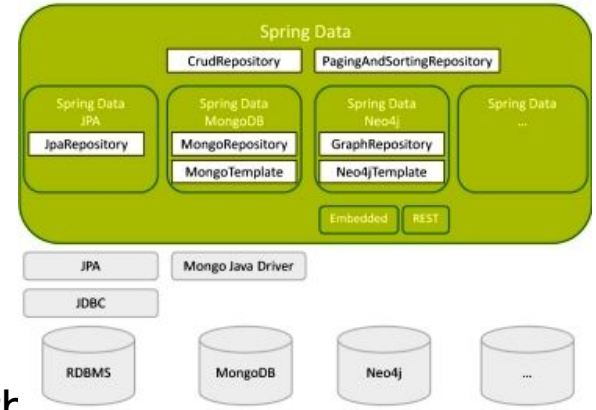
```
public Optional<Country> findByNameJPAQuery(final String name) {

    final StringBuilder sqlBuilder = new StringBuilder();
    sqlBuilder.append("FROM Country ");
    sqlBuilder.append(" where upper(name) = :name ");

    return this.entityManager
            .createQuery(sqlBuilder.toString(), Country.class)
            .setParameter("name", name)
            .getResultList()
            .stream()
            .findFirst();
}
```

GlobalLogic

What is Spring Data?

What is Spring Data?

- High level API to access different databases;
- Simplify your access data level;
- Compatible with most of popular DBs:
  - Spring data JPA;
  - Spring data MongoDb;
  - Spring data REDIS;
- Dynamic query derivations from repository method names;
- Contains a quick start, when used together with spring-boot;
- Native support for pagination;
- Focus on business logic not technical complexity;



GlobalLogic

What is Spring Data?

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

```
import java.util.Optional;

@Repository
public interface CountryRepository extends QuerydslPredicateExecutor<Country>, CountryDao {

    Optional<Country> findByNameIgnoreCase(String name);

}
```

GlobalLogic

What is Specifications?

What is Specification?

- It is a pattern;
- Introduced by Domain Driven Design book from Eric Evans and Martin Fowler;
- Based on business rules conditions, so return boolean, always;
  - If condition is satisfied, true;
  - If not, false;
- Predicate;
- Combine conditions;

GlobalLogic

Spring Data JPA + Specifications

Spring Data JPA + Specifications

- By default on SpringData:
  - JpaSpecificationExecutor<T>;
- Used for complex queries;

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
```

Spring Data JPA + Specifications

```java
public static Specification<Address> findByAddressType(final AddressType addressType) {
    return (root, query, criteriaBuilder) -> {
        return criteriaBuilder.equal(root.get("type"), addressType);
    };
}

public static Specification<Address> findByCountryName(final String countryName) {
    return (root, query, criteriaBuilder) -> {
        return criteriaBuilder.equal(criteriaBuilder.upper(root.join("country").get("name")),
                upperCase(countryName));
    };
}
```

```java
  @Override
  public List<AddressDto> findByTypeAndCountryName(
          final AddressType addressType,
          final String countryName) {

      final Specification<Address> specification = AddressSpecification.findByAddressType(addressType)
              .and(AddressSpecification.findByCountryName(countryName));

      return addressJpaSpecificationRepository
              .findAll(specification)
              .stream()
              .map(addressMapper::addressToAddressDto)
              .collect(toList());
  }
```

GlobalLogic

QueryDSL

QueryDSL

- Helping to create type-safe queries - Based on “Q-Class”;
- You can use together with JPA;
- It’s not from spring framework, but has support;
- Prevent runtimes errors, in case of field name changes;
  - Errors will be at compile time;

GlobalLogic

Spring Data JPA + Specifications + QueryDSL

GlobalLogic

Spring Data + Specification + QueryDSL

```
<dependency>
    <groupId>com.querydsl</groupId>
    <artifactId>querydsl-core</artifactId>
</dependency>
<dependency>
    <groupId>com.querydsl</groupId>
    <artifactId>querydsl-apt</artifactId>
</dependency>
<dependency>
    <groupId>com.querydsl</groupId>
    <artifactId>querydsl-jpa</artifactId>
</dependency>
```

```
<plugin>
    <groupId>com.mysema.maven</groupId>
    <artifactId>apt-maven-plugin</artifactId>
    <version>1.1.3</version>
    <executions>
        <execution>
            <goals>
                <goal>process</goal>
            </goals>
            <configuration>
                <outputDirectory>target/generated-sources/java</outputDirectory>
                <processor>com.querydsl.apt.jpa.JPAAnnotationProcessor</processor>
            </configuration>
        </execution>
    </executions>
    <dependencies>
        <dependency>
            <groupId>com.querydsl</groupId>
            <artifactId>querydsl-apt</artifactId>
            <version>${querydsl.version}</version>
        </dependency>
        <dependency>
            <groupId>com.querydsl</groupId>
            <artifactId>querydsl-jpa</artifactId>
            <version>${querydsl.version}</version>
        </dependency>
    </dependencies>
</plugin>
```

Example - Code

Example - Code

- https://github.com/viniciustoni/customers-techtalk

So…. Why should I use it?

So… Why should I use it?

- Reduce your code in the repository class;
- Reduce NamedQueries;
- Helping to make complex queries more readable;
- Reduce code duplication on queries;
- Business logic and queries more “readable”

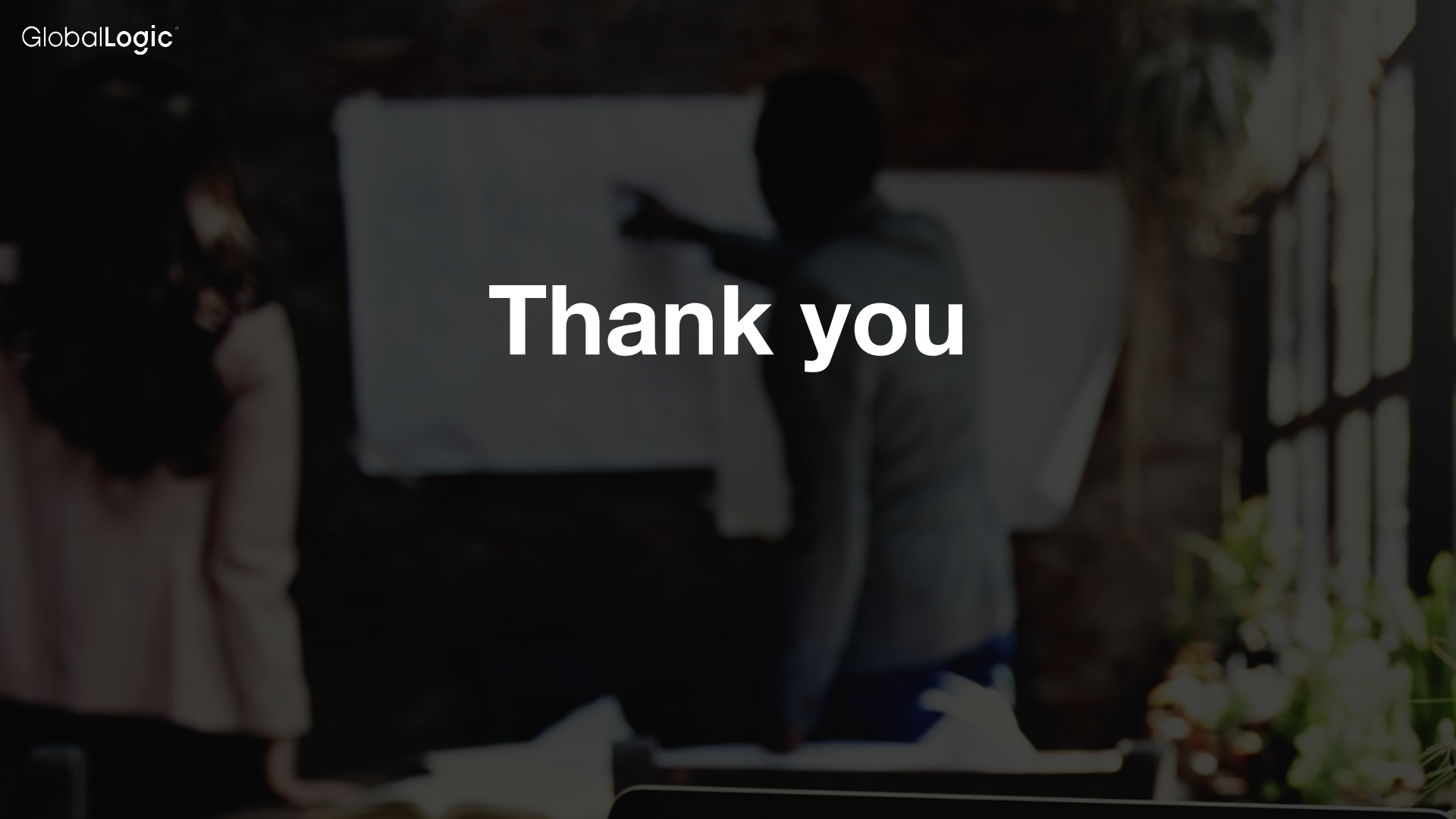
GlobalLogic

Questions

References

- https://stackoverflow.com/questions/3107044/preparedstatement-with-list-of-parameters-in-a-in-clause
- https://docs.oracle.com/javase/tutorial/jdbc/basics/prepared.html
- https://spring.io/projects/spring-data-jpa
- https://martinfowler.com/apsupp/spec.pdf
- https://java-design-patterns.com/patterns/specification/

GlobalLogic



Thank you