Spring Data
Advanced Querying

Query Methods, JPQL
Advanced Repositories
Spring Configuration

SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Contents

Questions

SoftUni Foundation

sli.do

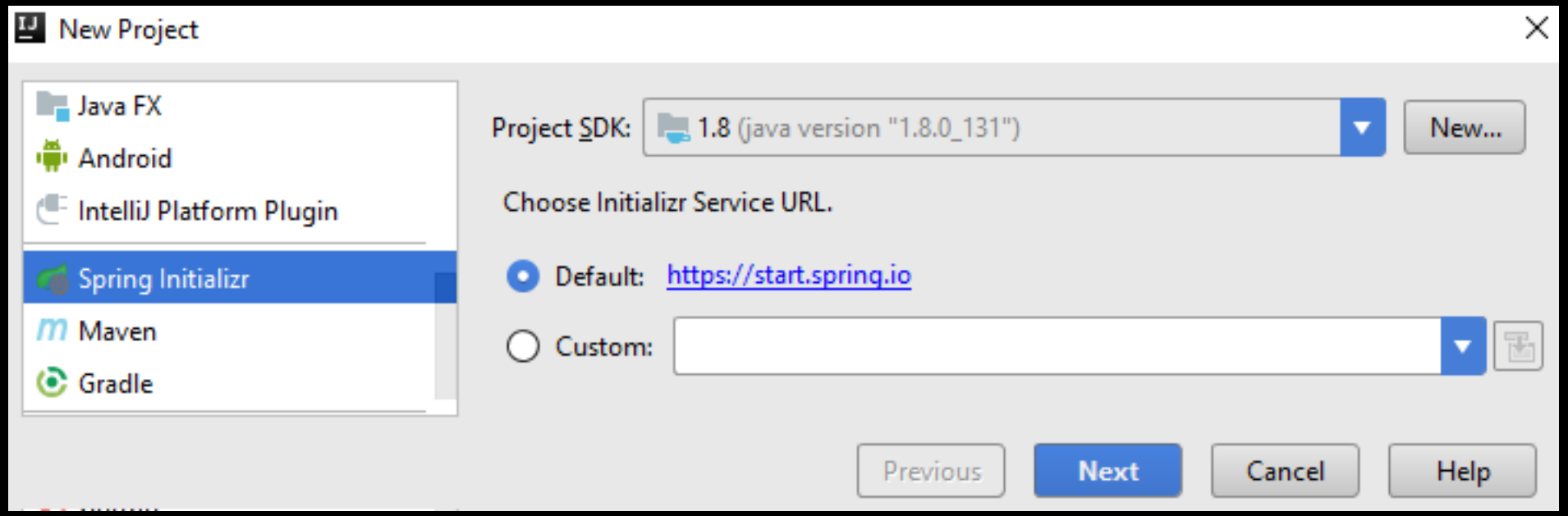
#JavaDB

SoftUni Foundation

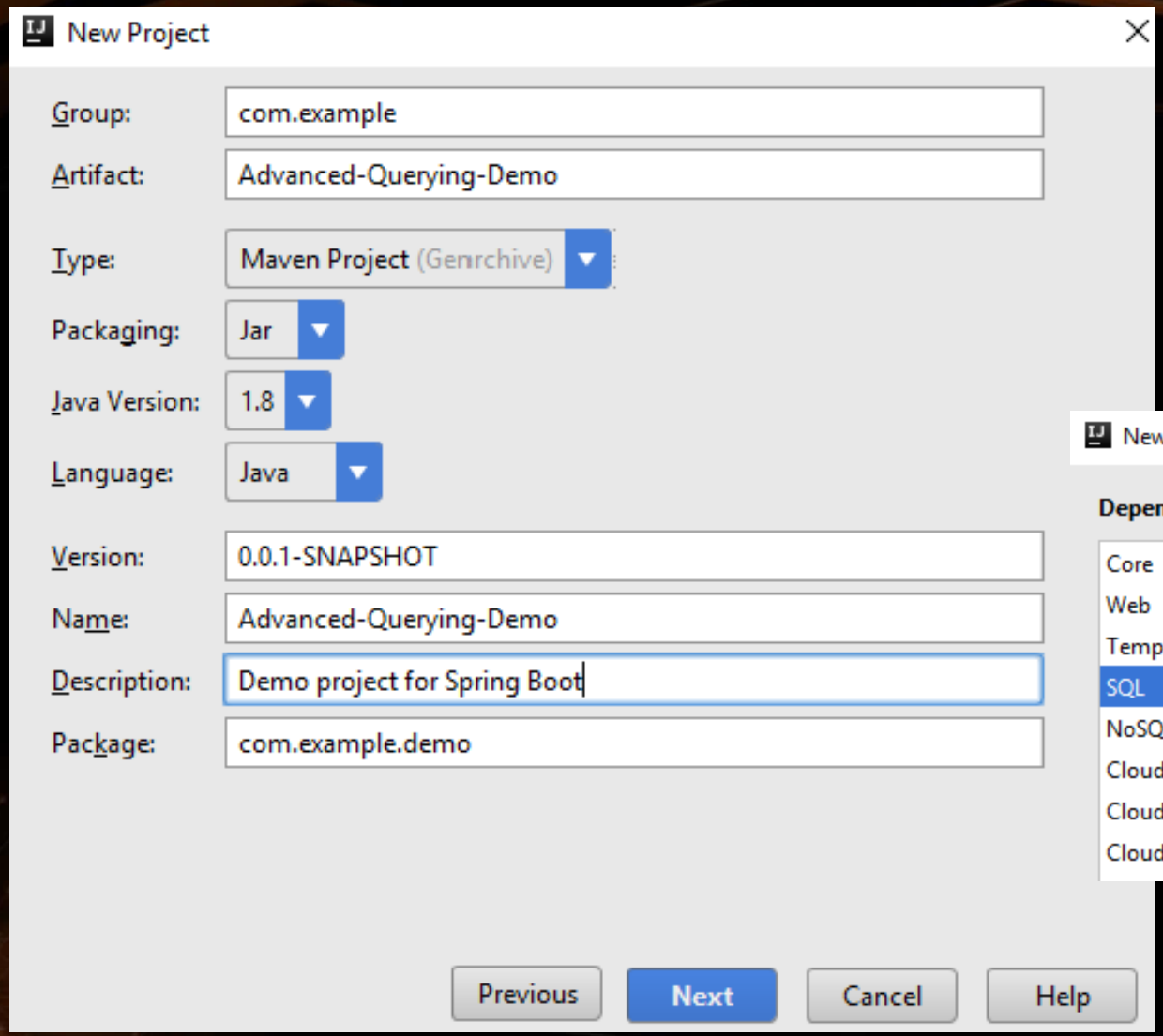
Querying

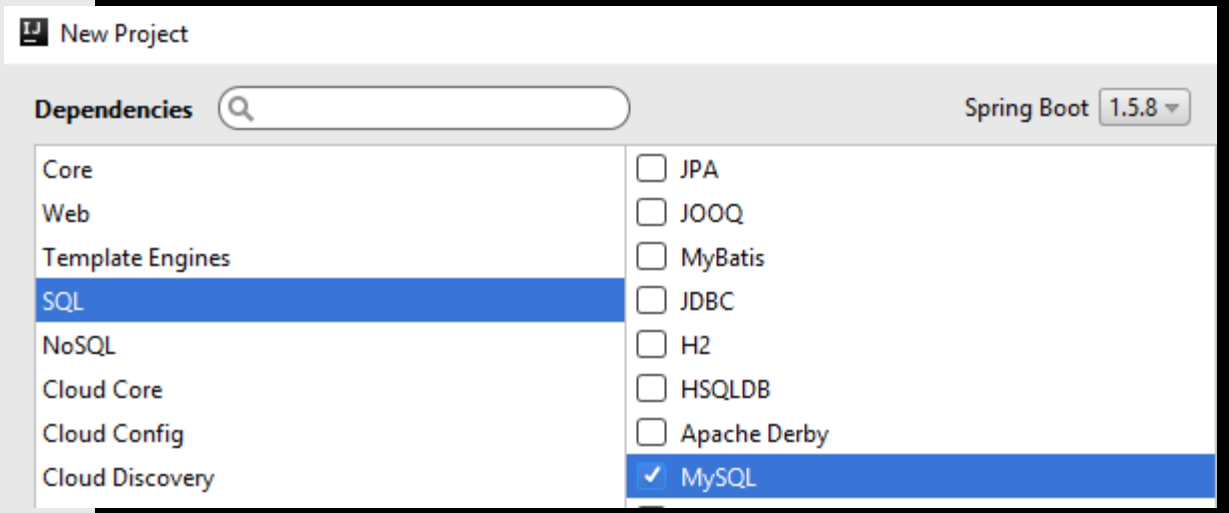
Retrieving Data by Custom Queries

Spring Project



Spring Project (2)





Query methods

SoftUni Foundation

ShampooRepository.java

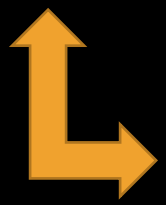
```
@Repository
public interface BasicShampooDao extends JpaRepository<BasicShampoo,
Long> {

    List<BasicShampoo> findByBrand(String brand);
}
```

Query method

Paramater

The method translates to the following query:



SQL

```
SELECT *
  FROM shampoos AS s
 WHERE s.brand = ?
```

Paramater

Query Lookup

Query methods

SoftUni Foundation

ShampooRepository.java

```
@Repository
public interface BasicShampooDao extends JpaRepository<BasicShampoo,
Long> {

    List<BasicShampoo> findByBrandAndSize(String brand, Size size);
}
```

Query method

Paramater

Paramater

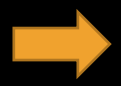
SQL

```
SELECT *
  FROM shampoos AS s
 WHERE s.brand = ?
   AND s.size = ?
```

Problem: Select Shampoos by Size

- Write a method that selects all shampoos by input size
  - Order the result by shampoo id
- Example input-output:

```
MEDIUM
```



```
Nature Moments Mediterranean Olive Oil &
Aloe Vera MEDIUM 6.50lv.
Volume & Fullness Lavender MEDIUM 5.50lv.
Rose Shine & Hydration MEDIUM 6.50lv.
Color Protection & Radiance MEDIUM 6.75lv.
…
```

Solution: Select Shampoos by Size

ShampooRepository.java

```
@Repository
public interface ShampooRepository extends
CrudRepository<BasicShampoo, Long> {
    List<BasicShampoo> getAllBySizeOrderById(Size sizeValue);
```

JPQL

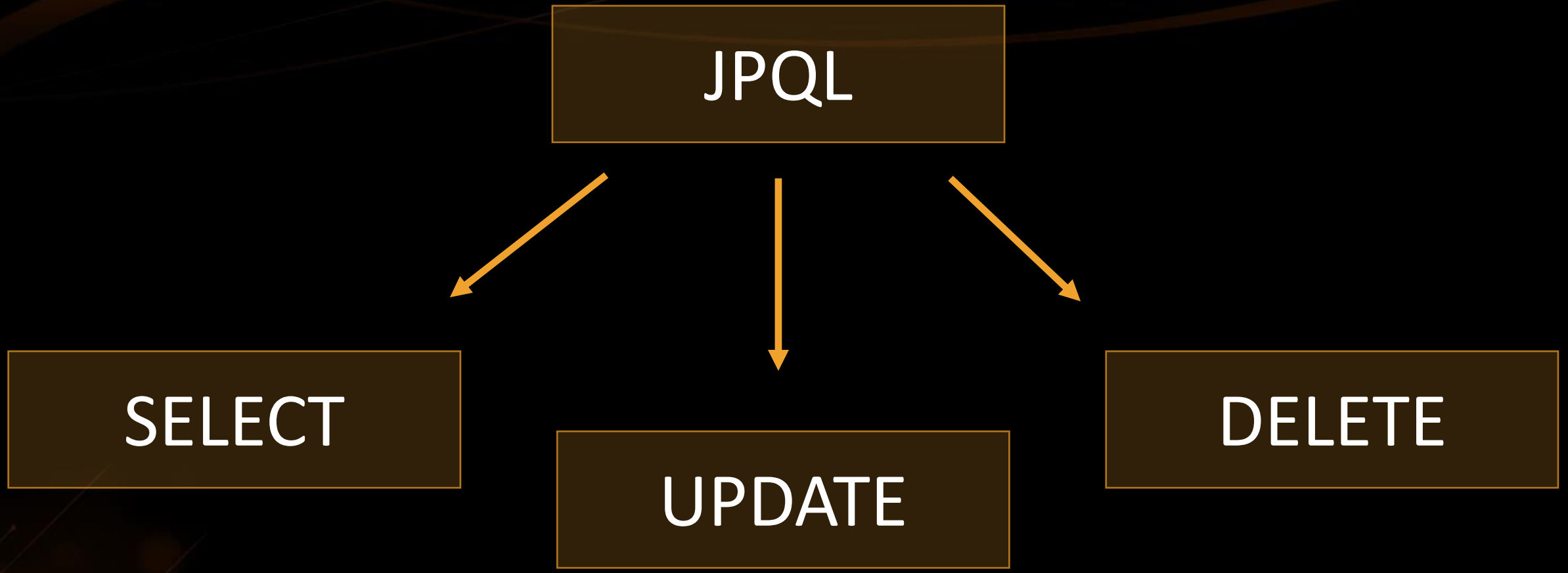
Java Persistence Query Language

JPQL

SoftUni Foundation

- Object-oriented query language
  - Part of the Java Persistence API
  - Used to make queries against entities stored in a relational database
  - SQL syntax operating with entities, not tables in the data source

JPQL Functionalities



JPQL Select Syntax

JPQL Join Syntax

JPQL Syntax

- Update:

```
"UPDATE BasicIngredient AS b
    SET b.price = b.price*1.10
  WHERE b.name IN :names"
```

Parameter

- Delete:

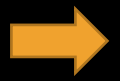
```
"DELETE FROM BasicIngredient AS b
  WHERE b.name = :name"
```

Problem: Select Shampoos by Ingredients

SoftUni Foundation

- Write a method that selects all shampoos with ingredients in a given list
- Example input-output:

```
Berry
Mineral-Colagen
```



```
Color Protection & Radiance
Fresh it Up!
Nectar Nutrition
Superfruit Nutrition
Color Protection & Radiance
Nectar Nutrition
…
```

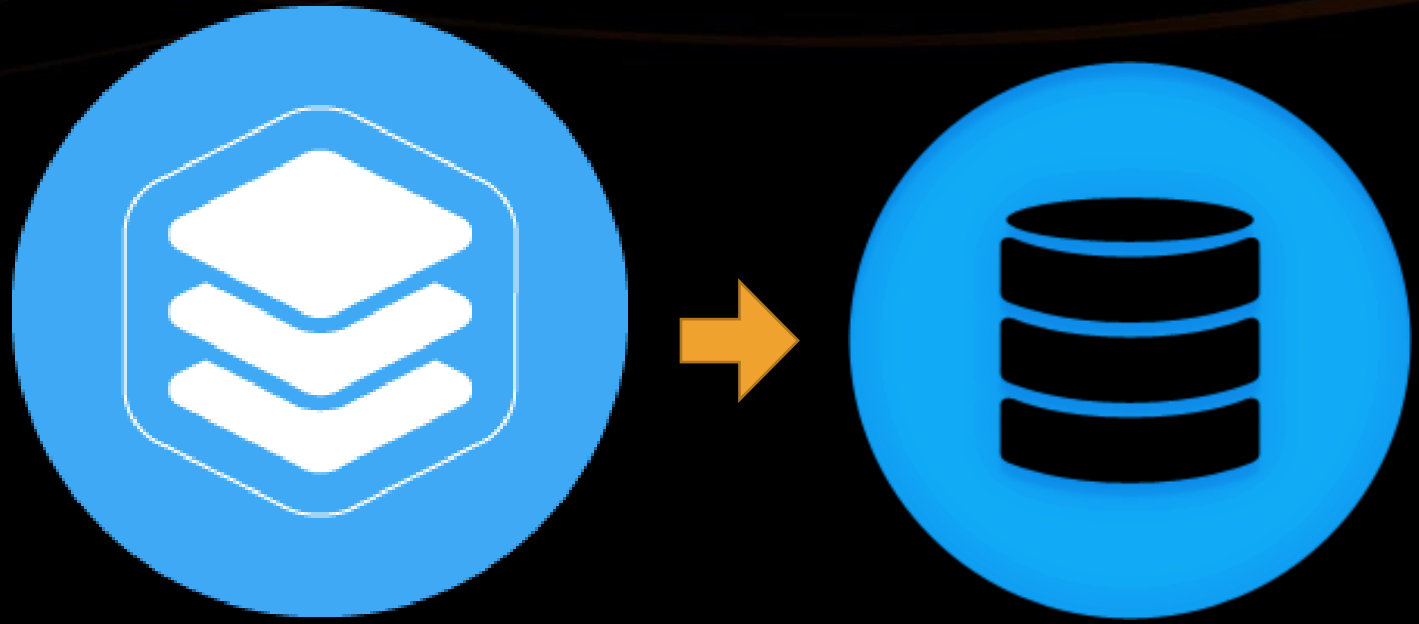
Solution: Select Shampoos by Ingredients

SoftUni Foundation

ShampooRepository.java

```
@Repository
public interface BasicIngredientDao extends JpaRepository<BasicIngredient,
Long>{

     @Query(value = "select s from BasicShampoo s " +
      "join s.ingredients i where i in :ingredients")
     List<BasicShampoo> findByIngredientsIn(@Param(value = "ingredients")
Set<BasicIngredient> ingredients);
}
```



Advanced Repositories

Repository Inheritance

Repository Inheritance

SoftUni Foundation

- In bigger applications we have similar entities extending an abstract class
  - Their base attributes and actions towards them are the same regardless differences
- We can set up a base repository to reduce query and code duplication
  - It can be inherited to clear up specifics

Example: Repository Inheritance

Not a repository

IngredientRepository.java

```
@NoRepositoryBean
public interface IngredientRepository<T extends Ingredient> extends
JpaRepository<T, Long>{
        //…
}
```

ChemicalIngredientRepository.java

```
@Repository
public interface ChemicalIngredientRepository extends IngredientRepository
<BasicChemicalIngredient> {
    List<ChemicalIngredient> findByChemicalFormula(String chemicalFormula);
}
```

Example: Repository Inheritance

CustomShampooRepository.java

```
public interface CustomShampooRepository {

    void create(BasicShampoo basicShampoo);
}
```

CustomShampooRepositoryImpl.java

```
public class CustomShampooDaoImpl implements
CustomShampooRepository {
    @PersistenceContext
    private EntityManager entityManager;

    @Transactional
    public void create(BasicShampoo basicSh
        entityManager.persist(basicShampoo);
    }
}
```

Inject Entity Manager

Single Transaction

Spring Custom Configuration

Java-Based Setup

Application Properties

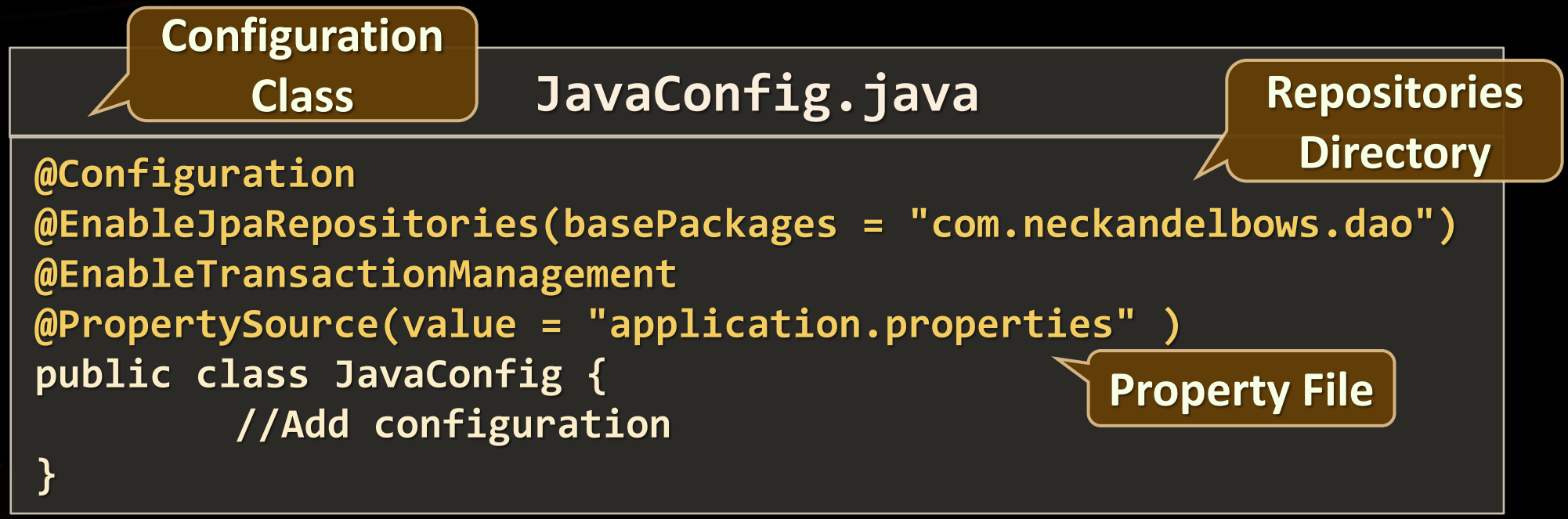
- So far we've configured our project with a spring properties file:

application.properties

```
#Data Source Properties
spring.datasource.driverClassName = com.mysql.jdbc.Driver
spring.datasource.url =
jdbc:mysql://localhost:3306/neck_and_elbow?useSSL=false&createDatabaseIfNotEx
ist=true
spring.datasource.username = root
spring.datasource.password = 1234
```

Connection properties

Java-Based Configuration



Java-Based Configuration

SoftUni Foundation

JavaConfig.java

```
    @Autowired
    private Environment environment;

    @Bean
    public DataSource dataSource() {

    DriverManagerDataSource driverManagerDataSource = new DriverManagerDataSource();
driverManagerDataSource.setDriverClassName(environment.getProperty("spring.datasource.driverClass
Name"));
    driverManagerDataSource.setUrl(environment.getProperty("spring.datasource.url"));
    driverManagerDataSource.setUsername(environment.getProperty("spring.datasource.username"));
    driverManagerDataSource.setPassword(environment.getProperty("spring.datasource.password"));
    return driverManagerDataSource;
}
```

Data Source Connection

Java-Based Configuration

SoftUni Foundation

JavaConfig.java

```
    @Bean
    public EntityManagerFactory entityManagerFactory() {

        HibernateJpaVendorAdapter vendorAdapter = new HibernateJpaVendorAdapter();
        vendorAdapter.setDatabase(Database.MYSQL);
        vendorAdapter.setGenerateDdl(true);
        vendorAdapter.setShowSql(true);
        LocalContainerEntityManagerFactoryBean factory = new
LocalContainerEntityManagerFactoryBean();
        factory.setJpaVendorAdapter(vendorAdapter);
        factory.setPackagesToScan("com.neckandelbows.domain");
        factory.setDataSource(dataSource());
        Properties jpaProperties = new Properties();
        jpaProperties.setProperty("hibernate.hbm2ddl.auto","validate");
        jpaProperties.setProperty("hibernate.format_sql", "true");
        factory.setJpaProperties(jpaProperties);
        factory.afterPropertiesSet();
        return factory.getObject();
    }
```

JPA Configuration

Models Package

Java-Based Configuration

JavaConfig.java

```
    @Bean
    public PlatformTransactionManager transactionManager() {

        JpaTransactionManager txManager = new
JpaTransactionManager();

txManager.setEntityManagerFactory(entityManagerFactory());
        return txManager;
    }
```

Transaction Manager Configuration

Java-Based Configuration

SoftUni Foundation

JavaConfig.java

```
@Configuration
@EnableJpaRepositories(basePackages = "com.neckandelbows.dao")
@EnableTransactionManagement
@PropertySource(value = "application.properties" )
public class JavaConfig {
        //Add configuration

    @Bean
    public CustomShampooDaoImpl shampooDaoImpl(){
        return new CustomShampooDaoImpl();
    }
}
```

Bean Definition

Summary

- Spring Data translates methods to SQL Queries
- We can write custom queries
  - JPQL syntax on entity classes
- Repositories can be inherited
  - Reduces code duplication for inherited entities

Spring Data Advanced Querying

SoftUni Foundation

Questions?

https://softuni.bg/courses/databases-advanced-hibernate

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license

- Attribution: this work may contain portions from
  - "Databases" course by Telerik Academy under CC-BY-NC-SA license

Free Trainings @ Software University

- Software University Foundation – softuni.org
- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg