# 1. Spring JPA là gì?

Ví dụ trong DB mình có table có nhiều column. Thì Spring JPA nó sẽ ánh xạ table đó với 1 object trong code của mình. Bây giờ thao tác với Database thì sẽ thao tác thông qua object, giúp việc code dễ dàng hơn.

# 2. Có mấy cách dependency injection trong java?

Có 2 cách: dựa vào constructor và dựa vào setter

Dựa vào constructor:

<bean id="dungsi" class="com.kaopiz.dungsi.DungSiThongThai">

<constructor-arg ref="nhiemvu" />

</bean>

<bean id="nhiemvu" class="com.kaopiz.nhiemvu.NhiemVuMuaRauDen">

<constructor-arg value="#{T(System).out}" />

</bean>

Dựa vào setter:

<bean id = "john-classic" class = "com.example.Person">

<property name = "name" value = "John Doe"/>

<property name = "spouse" ref = "jane"/>

</bean>

<bean name = "jane" class = "com.example.Person">

<property name = "name" value = "John Doe"/>

</bean>

# 3. Có mấy cách autowired

Có 3 cách:

Theo name

Theo type

Theo constructor, nghĩa là kết hợp cả name và type. Default là lấy name xong lấy type

# 4. Cho biết ý nghĩa của @Qualifier trong Spring

Ví dụ như ta có 1 interface là A

2 component B và C implement thằng A

Mà A được autowired thì ta phải sử dụng Qualifier để chỉ rõ là sử dụng implement B hay C

# 5. Datasource là gì? Cần khai báo gì trong datasource

Datasource là nơi để chứa thông tin kết nối DB. Cần có driverClass, url, username và password

# 6. Connection Pool là gì?

Connection pool là nơi chứa các connection để chứa các connection kết nối đến database, có thể sử dụng lại các connection này thay vì liên tục tạo mới và close.

# 7. Trong jdbc e sử dụng connection management nào để quản lý connection. Cần truyền tham số gì để thiết lập connection pool này?

JdbcTemplate. Cần truyền vào Datasource

# 8. Em sử dụng connection pool nào trong hibernate

Không quan tâm về connection pool trong hibernate vì bên trong hibernate là có sử dụng kỹ thuật connection pooling rồi. Điều quan tâm là Session trong hibernate.

# 9. Tuần tự làm việc của session?

\_ SessionFactory tạo ra 1 cái session.

\_Session begin 1 Transaction

\_ Session thực hiện query

\_ Transaction commit

\_ Session đóng connection

Tuy nhiên có thể sử dụng @Transactional là có thể giúp quản lý các transaction.

<pre>

\* Session sess = factory.openSession();

\* Transaction tx;

\* try {

\* tx = sess.beginTransaction();

\* //do some work

\* ...

\* tx.commit();

\* }

\* catch (Exception e) {

\* if (tx!=null) tx.rollback();

\* throw e;

\* }

\* finally {

\* sess.close();

\* }

\* </pre>

Ta có thể sử dụng LocalSessionFactoryBean để quản lý session.

9. Có cách nào quản lý transaction