

Title: Including Spring context from an external jar does not find services

Post Body:

I am currently using Spring Boot and annotation-based configuration to manage my spring applications. The project is composed of a registration and an external library of services.

Service is composed of classes annotated with `@Service` and have a `@Repository` injected into them.

Registration injects the `@Service`.

Project Structure (4 maven projects):

```
xelamitchell (Parent POM)  +- domain    +- service (Spring JPA)  registration (separate project with Spring Boot)
```

Registration has a maven dependency to service.

PersonService:

```
package org.xelamitchell.service.person; import org.xelamitchell.domain.person.Person; import java.util.List; /** * Manages
```

PersonRepository:

```
package org.xelamitchell.service.person; import org.xelamitchell.domain.person.Person; import org.springframework.data.jpa.re
```

Service Library configuration:

```
@Configuration public class ServiceContext { }
```

Application configuration:

```
@Configuration @ComponentScan(basePackages = {'org.xelamitchell'}) @EnableAutoConfiguration //@Import({WebContext.class, Servi
```

Spring boot is defined in the POM for Application.

The exception I keep getting is that the `@Repository` classes were not able to be injected into their respective `@Service`:

```
org.springframework.beans.factory.BeanCreationException: Error creating bean with name 'personController': Injection of autowir
```

Accepted Answer:

Edited after the discussion...

Add the `'@EnableJpaRepositories'` and `'@EntityScan'` on top of your Application class.

```
@EnableJpaRepositories('org.xelamitchell') @EntityScan('org.xelamitchell')
```

HTH...

Highest Rated Answer:

Where's your implementation of interface: `PersonRepository`? find it, add `@Repository` on it, and remove this anotation from `PersonRepository` interface.