REFCARDZ RESEARCH WEBINARS ZONES V

Spring, Spring Boot, and Component Scan

Component scanning can make your life easier in Spring, and Spring Boot brings an additional level of power. In this post, we take a closer look at component scanning.

by Ranga Karanam ⋒MVB <>CORE · Jan. 02, 18 · Java Zone · Tutorial **Like** (9) Comment (1) ☆ Save 184.83K Views Join the DZone community and get the full member experience. **JOIN FOR FREE**

Component Scan. Let's understand that in this article. You Will Learn

This guide will help you understand the most important concept in Spring — Component Scan. Spring Boot does some magic around

• What is Component Scan?

- Why is Component Scan important?
- Which packages does Spring Boot scan automatically?
- How do you define Component Scan with Spring Boot?
- How do you resolve problems involving Component Scan?
- @ComponentScan

If you understand Component Scan, you understand Spring.

Spring is a dependency injection framework. It is all about beans and wiring in dependencies.

The first step of defining Spring Beans is by adding the right annotation — @Component or @Service or @Repository.

However, Spring does not know about the bean unless it knows where to search for it.

This part of "telling Spring where to search" is called a Component Scan.

You define the packages that have to be scanned. Once you define a Component Scan for a package, Spring would search the package and all its sub packages for components/beans.

Defining a Component Scan • If you are using Spring Boot, check the configuration in Approach 1.

• If you are doing a JSP/Servlet or a Spring MVC application without using Spring Boot, use Approach 2.

Approach 1: Component Scan in a Spring Boot Project

1 package com.in28minutes.springboot.basics.springbootin10steps;

4 import org.springframework.boot.autoconfigure.SpringBootApplication;

6 import org.springframework.context.ConfigurableApplicationContext;

com.in28minutes.springboot.basics.springbootin10steps.

3 import org.springframework.boot.SpringApplication;

5 import org.springframework.context.ApplicationContext;

• If your other package hierarchies are below your main app with the @SpringBootApplication annotation, you're covered by the implicit Component Scan.

• If there are beans/components in other packages that are not sub-packages of the main package, you should manually add them as @ComponentScan

Consider the class below:

8 @SpringBootApplication 9 public class SpringbootIn10StepsApplication { 11 public static void main(String[] args) { @SpringBootApplication is defined in the SpringbootIn10StepsApplication class which is in the package com.in28minutes.springboot.basics.springbootin10steps @SpringBootApplication defines an automatic Component Scan on the package

In this case, you would need to add the new package into Component Scan.

You are fine if all your components are defined in the above package or a sub-package of it.

You have two options:

However, let's say one of the components is defined in package com.in28minutes.springboot.somethingelse

• This would scan the entire parent tree of com.in28minutes.springboot.

• Or define two specific Component Scans by using an array. • @ComponentScan({"com.in28minutes.springboot.basics.springbootin10steps","com.in28minutes.springboot.somethingels

e"})

• Define @ComponentScan("com.in28minutes.springboot")

- 1 @ComponentScan("com.in28minutes.springboot") 2 @SpringBootApplication
- 3 public class SpringbootIn10StepsApplication {

1 @ComponentScan({"com.in28minutes.springboot.basics.springbootin10steps","com.in28minutes.springboot.somethingelse"}) 2 @SpringBootApplication 3 public class SpringbootIn10StepsApplication {

Option 1:

Application Context.

Option 2:

Option 1:

Approach 2: Non-Spring Boot Project In a non-Spring Boot Project, we would typically define the component scan explicitly in an XML application context or a Java

1 @ComponentScan("com.in28minutes) 2 @Configuration 3 public class SpringConfiguration {

Option 2: 1 @ComponentScan({"com.in28minutes.package1","com.in28minutes.package2"})

XML application context:

2 @Configuration

1 <context:component-scan base-package="com.in28minutes" /> Specific multiple packages:

1 <context:component-scan base-package="com.in28minutes.package1, com.in28minutes.package2" />

URL Not Working

The server starts up fine, but: My URL is not working

1 #### No qualifying bean of type found

3 public class SpringConfiguration {

DispatcherServlet with name 'dispatcher' WARNING: No mapping found for HTTP request with URI [/login] in DispatcherServlet with name 'dispatcher' WARNING: No mapping found for HTTP request with URI [/list-todos] in DispatcherServlet with name 'dispatcher'

My login URL is not working

No qualifying bean of type [com.in28minutes.springboot.jpa.UserRepository] found for dependency [com.in28minutes.springboot.jpa.UserRepository]: expected at least 1 bean which qualifies as autowire candidate for this dependency. $Dependency\ annotations:\ \{ @org.springframework.beans.factory.annotation. Autowired (required=true) \}$

• My todo URL is not working ``` WARNING: No mapping found for HTTP request with URI [/spring-mvc/login] in

- @Component vs. @ComponentScan
 - @Component indicates that a class might be a candidate for creating a bean. It's like putting a hand up. • @ComponentScan is searching packages for Components. Trying to find out who all put their hands up.

@Component and @ComponentScan are for different purposes.

Topics: JAVA, COMPONENT SCANNING, SPRING BOOT, TUTORIAL

Published at DZone with permission of Ranga Karanam, DZone MVB. See the original article here.

· Agile Software Development: Principles, Team Structure, and Frameworks

Popular on DZone A Guide to Component Driven Development (CDD)

How to Set Up PostgreSQL High Availability With Patroni

Understanding Kubernetes Basics

Opinions expressed by DZone contributors are their own.

Java Partner Resources

ABOUT US

About DZone

MVB Program

CONTRIBUTE ON DZONE

Become a Contributor Visit the Writers' Zone

Article Submission Guidelines

Send feedback Careers Sitemap

> LEGAL **Terms of Service Privacy Policy**

+1 (919) 678-0300

ADVERTISE

Advertise with DZone

600 Park Offices Drive

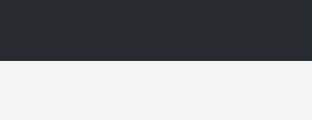
Durham, NC 27709

support@dzone.com

+1 (919) 238-7100

CONTACT US

Suite 300



AnswerHub...

Let's be friends:

DZone.com is powered by

Same root cause for both above problems — the component is not being picked up. There are three possible things you would need to look at: • You have not added the right annotation — @Controller, @Repository, or @Controller • You have not added a Component Scan. • The package of your component is not defined in Component Scan. You have two options: 1. Add the annotation or component scan 2. Move the component to a package already under Component Scan