Date : 09/12/2020
Spring Boot 9AM
Mr. RAGHU

Srikanth (admin): +91-630 29 68 665 (whatsapp also)
https://www.facebook.com/groups/thejavatemple
email : javabyraghu@gmail.com

First 4 Demo Sessions Links:
Spring Boot Day -1: https://youtu.be/L7zUhVLgoBA
Spring Boot Day -2: https://youtu.be/oCG4w6Rkcag
Spring Boot Day -3: https://youtu.be/diWOnew5VLk
Spring Boot Day -4: https://youtu.be/tmxJZlBB7Jw

@Component--> Creating object to our class (not abstract, not be a interface).
@Value --> static data/properties file/expression(SpEL).

Chapter #1 Spring Core + Spring Boot Core

Spring Container:-
a) First detect/find/scan class
b) then create object
c) provide data
d) link objects
e) destory the objects finally.

Component Scaning:-
It is process of finding our classes by using one package name.
Done by Spring container.

=> Container will read our class from Given package and sub package classes also.
=> If our class is not under given package then object is not created.

*) We should provide @ComponentScan(basePackage="onePackageName") to find
classes by Spring container in Spring F/w. Not required in Spring Boot.

*) In Spring boot, by default starter class package is considered as
basePackage, then our classes must be provided under starter class package
or its sub package.

** If our class is not under base package or its sub package then
container will not find and no object is created,
if we try to read object, container says exception:

NoSuchBeanDefinitionException: No bean named 'product' available

==

===
you class package =
main class pagae is fine, main class + extra also fine.

Examples: DETECTED-YES, NOT DETECTED--NO

```
 main class ---  package : com.app.one

 class#A    ---  package com.app.one  (YES)
 class#B   ---   package app.test.one (NO)
 class#C   ---   package com.app      (NO)
                        (This is not same as main class package)
 class#D   ---   package com.app.one.test (YES)
```

=============Example code===================
S#1 Create Spring Starter Project
> File > New > Spring Starter Project
> Enter details

```
Name    :   SpringBoot2ComponentScanningEx
Package :   in.nareshit.raghu
```

> Next > Next > Finish

S#2 Create class under basepackage or its sub package only.

```
package in.nareshit.raghu.bean;

//ctrl+shift+O (imports)

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;

@Component
public class Product {

        @Value("AA")
        private String pcode;

        @Override
        public String toString() {
                return "Product [pcode=" + pcode + "]";
        }


}
```

#3. Modified Starter class

```
package in.nareshit.raghu;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.ApplicationContext;

import in.nareshit.raghu.bean.Product;
```

```
//ctrl+shift+O (imports)
@SpringBootApplication
//internal code=> @ComponentScan(basePackage="in.nareshit.raghu")
public class SpringBoot2ComponentScanningExApplication {

        public static void main(String[] args) {
                ApplicationContext ac =
SpringApplication.run(SpringBoot2ComponentScanningExApplication.class,
args);
                //read object from container
                Product p = ac.getBean("product",Product.class);
                //print object
                System.out.println(p);
        }

}
```

Q) Who will provide @ComponentScan(basePackage="") in Spring boot?
A) Spring boot main class contains -> @SpringBootApplication
   that takes main class package as basePackage.

Q) Can we modify default basepackage in Spring boot?
A) YES. (But not recomanded)

We can provide manually using @ComponentScan(basePackage="")
one or multiple package even.

But in this case , default package given by spring boot is overriden
(ie no starter class package is now basePackage).

Ex#1

```
@ComponentScan(basePackages = "com.app")
```

Ex#2

```
@ComponentScan(basePackages =
        {
                "com.app",
                "in.nareshit.raghu",
                "text.xyz",
                "aa.bb.cc",
                "mno.pqr.sju"
        }
)
```

** word basePackage is optional

```
@ComponentScan("com.app")

@ComponentScan({
        "com.app",
        "on.two",
        "abc.xyz"
})
```

Q) Can we give empty basePackage Externally?
A) YES. In this case, again default is considered (again starter class page)

```
@ComponentScan(basePackages = "")
```