Date : 04/09/2020

Spring Boot 7AM

Mr. RAGHU

---------------------------------------------

javabyraghu@gmail.com

https://www.facebook.com/groups/thejavatemple

Admin : Srikanth (+91-6302968665)

----------------------------------

\*)Spring Configuration:

->\*\*\* Annotation Configuration

@Component, @Value, @Autowired

->\*\* Java Based Configuration

@Configuration, @Bean

---Examples------------------------------

Spring Annotation Config Ex#1

#1 Create one Spring Starter project

> File > new > Spring Starter project

> Enter details

Name : SpringBoot2AnnotationExOne

Package Name: in.nareshit.raghu

> Next > Next > Finish

#2. Create one class under package in.nareshit.raghu

> Right click on 'in.nareshit.raghu' > new > class

> Enter class and package name

package in.nareshit.raghu;

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

import org.springframework.stereotype.Component;

//ctrl+shift+O

@Component("pob")

public class Product {

@Value("5")

private int pid;

@Value("PEN")

private String pcode;

public void display() {

System.out.println(pid + "-" + pcode);

}

}

#3. Main class(modified)

package in.nareshit.raghu;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

//ctrl+shift+O (imports)

@SpringBootApplication

public class SpringBoot2AnnotationExOneApplication {

public static void main(String[] args) {

ApplicationContext ac = SpringApplication.run(SpringBoot2AnnotationExOneApplication.class, args);

Product p = ac.getBean("pob",Product.class);

p.display();

}

}

========================================================================

HAS-A (Association):-

Using child(class/interface) as a Datatype and creating Variable(Reference Variable)

inside Parent(class) is called HAS-A Relation.

Syntax: HAS-A

Parent --------<> Child

\*) To link parent object with child object use annotation @Autowired.

=====Example=============

#1 Create one Spring Starter project

> File > new > Spring Starter project

> Enter details

Name : SpringBoot2AnnotationExThree

Package Name: in.nareshit.raghu

> Next > Next > Finish

#2. classes

package in.nareshit.raghu;

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

import org.springframework.stereotype.Component;

//ctrl+shift+O (imports)

@Component("aob")

public class Address {

@Value("5-A")

private String hno;

@Value("HYD")

private String loc;

// > Source > generate toString()

// alt+shift+S then S then Generate

@Override

public String toString() {

return "Address [hno=" + hno + ", loc=" + loc + "]";

}

}

------------

package in.nareshit.raghu;

import org.springframework.beans.factory.annotation.Autowired;

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

import org.springframework.stereotype.Component;

@Component("eob")

public class Employee {

@Value("10")

private int eid;

@Value("SAM")

private String ename;

@Autowired

private Address addr;

//alt+shift+S, S , Generate

@Override

public String toString() {

return "Employee [eid=" + eid + ", ename=" + ename + ", addr=" + addr + "]";

}

}

-----------

#3. Modified main class

package in.nareshit.raghu;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.context.ApplicationContext;

@SpringBootApplication

public class SpringBoot2AnnotationExThreeApplication {

public static void main(String[] args) {

ApplicationContext ac = SpringApplication.run(SpringBoot2AnnotationExThreeApplication.class, args);

Employee emp = ac.getBean("eob",Employee.class);

System.out.println(emp);

}

}

---------------