Date : 11/12/2020
Spring Boot 9AM
Mr. RAGHU

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

Watch these videos:-
https://www.youtube.com/c/NareshIT/search?query=maven%20raghu

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

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

Q) Can we write and load our own properties file with any name in Spring Boot?
A) YES, But not recomanded.

S#1 Define your properties file with any name (extension ____.properties)

> Right click on 'src/main/resources' > new > file > Enter name : mydata.properties
> Enter content also

--mydata.properties---
app.title-one=my title
app.version=888.88

S#2 At Main class/Startre class write
@PropertySource("classpath:mydata.properties")

---------Example----------
#1. Create Sping Starter Project
Name : SpringBoot2PropertiesExTwo

#2 create new properties file under src/main/resources

--myinfo.properties--
my.prod.id=6666
my.prod.code=TTTT

--application.propreties--
my.prod.id=990
#my.prod.code=ABC

#3. Class
package in.nareshit.raghu.bean;

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

@Component
public class Product {
        @Value("${my.prod.id}")
        private Integer pid;

        @Value("${my.prod.code}")
        private String pcode;

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


}

#4. Modified Starter class
package in.nareshit.raghu;

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

import in.nareshit.raghu.bean.Product;

@SpringBootApplication
@PropertySource("classpath:myinfo.properties")
public class SpringBoot2PropertiesExTwoApplication {

        public static void main(String[] args) {
                ApplicationContext ac =
SpringApplication.run(SpringBoot2PropertiesExTwoApplication.class,
args);
                Product pob = ac.getBean("product",Product.class);
                System.out.println(pob);
        }

}
==========================================================================
==
Q) If a key is present in both appication.properties and our properties
file
  (abcd.properties)? then which one is selected ?

A) Key is selected from 'appication.properties' only.

Q) Can we create and load multiple properties files in application?
A) YES.

Crate multiple properties files and load using

@PropertySource({
        "classpath:mydata.properties",
```

```
        "classpath:myinfo.properties",
        "....",
        "....",
        "...."
})
```

Q) If we define same key with different value in different properties files
   which one is selected?

A) Last loaded proeprties file key is considered based on ovrriding rule.

```
Ex:
@PropertySource({
        "classpath:sample.properties",       //loading order 1st
        "classpath:process.properties"       // loadng order 2nd
})
```

=> First check in process.properties for given key, if not read from sample.properties
=> if both places not exist then priority is : application.properties

==========

Creating Project with out STS (or) its plugin

Step#1 Goto
   https://start.spring.io/

Step#2 Fill All details
  Project Type: Maven
  Name : SpringBoot2FirstApp
  Package : in.nareshit.raghu

Step#3 Click on Generate button
   and Extract to a folder

** Current Spring Boot version: 2.4

Step#4 Download and Extract Eclipse S/w

Speific one:
https://www.eclipse.org/downloads/packages/release/kepler/sr2/eclipse-ide-java-ee-developers

Choose one manually:
https://www.eclipse.org/downloads/packages/release

Step#5 open Eclipse with one workspace(folder name)

Step#6 Copy Folder location of Project where pom.xml exist
ex: F:\StarterApps\FirstApp

Step#7*** Follow below steps in Eclipse

> File Menu

> Import
> Search using Maven
> Select Existing Maven Project
> Next
> Enter location (or browse for same)
> Press Enter key
> Finish

Step#8 Open project starter class and run after coding.

==========
*)Adding STS 4 Plugin to Eclipse:- (one time stup)

> open Eclipse
> Help Menu
> Eclips Market Place
> Click RED COLOR stop button
> Search using Spring tools 4
> Click on Install button
> Next > Next > Finish

> Close and Open Eclipse
> File > New > Spring Starter project
..(same process)...
==========
Spring Java Configuration

*) If a class is pre-defined (we have only .jar or .class files) then
   in this case we can't use @Component.

*) Use, Spring java Configuration concept @Configuration and @Bean.

Step#1 Define one public class with any name
       and apply @Configuration

```
@Configuration
public class <anyName> {

}
```

Step#2 Write one method for One object using below syntax

```
    public  <ClassName>  <objectName>() {
         //object code
        return obj;
    }
```

Step#3 Apply @Bean over method.

```
--pre-defined class--
class MyDataSource {
  String driver;
}
---------------------
```

a) Spring Boot Java Configuration

```
@Configuration
public class AppConfig {

   @Bean
   public  MyDataSource  dsObj() {
      MyDataSource d = new MyDataSource();
      d.setDriver("Oracle");
      return d;
   }

}
```

```
public class PdfExport {
   private String module;
}

@Configuration
public class AppConfig {
   @Bean
   public  PdfExport  peObj() {
      PdfExport  p = new PdfExport();
      p.setModule("ABC");
      return p;
   }
}
```

Srikanth(admin)
+91- 630 296 8665
3 PM

int -->primitive   defualt ->zero

Integer --> class /method ->null
 int->String-> Hexa Decimal

ctrl+M x2