Date : 10/12/2020
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

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

## Properties File ( application.properties )

=> This file is used to Store data in key=val format in application.

=> Setup data is provided using properties file
  a) Database Connection like
      driver-class=oracle, url=jdbc:orcl, username=system, password=nit
  b) ORM/JPA Properties
      dialect=MySQLDialect, show-sql=true, ddlauto=create ..etc
  c) MVC(web)
      view.prefix=FolderName, suffix=.jsp
  e) Email
     host=gmail, un=jbr@gm.com, pwd=abcd ..etc

=> In Spring boot properties file has a pre-defined name
     'application.properties'  (we can modify even).
   This file exist in src/main/resource folder.

=> In Spring Boot, we can use two annotation to read data from Proprties file
   a) @Value  (To read one key data)
   b) @ConfigurationProperties (to read bulk/multiple values).

=> Here, we have two types of keys.
  a) Pre-defined keys (given by Spring boot)
   Link:
https://docs.spring.io/spring-boot/docs/current/reference/html/appendix-application-properties.html

  b) Programmer-defined keys (given by programmer).

=> Key can have symbols like : dot(.)  dash(-) and underscore(_) only.
=> *** Keys are case-sensitive ( DRIVER=___, Driver=___, driver=___ are different ).
=> If we use Symbol # that indicates comment.

=> If we provide duplicate key=val (ie same key with different values),

then last combination is considered.
(it is like value overriding).

ex:

```
 sid=10
 sid=20
 sid=30
```

=> now, sid is 30 value taken in application.

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

Q) How can we load properties file into Spring Application using Annotation?

A) @PropertySource("fileName+location")

ex:

```
 ProjectName
  |-src/main/resource                          <--- classpath
      |-abcd.properties

  @PropertySource("classpath:abcd.properties")

 ProjectName
  |-myfiles                        <--- file
      |-abcd.properties

 @PropertySource("file:./myfiles/abcd.properties")
```

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

```
Syntax:  @Value("${key}")         => $ - Reading data from properties

Ex:     @Value("${app.title}")
```

*) If we provide a key in code, which is not present in properties file
then spring boot throws exception:

```
  IllegalArgumentException: Could not resolve placeholder 'app.ttles' in value
                                 "${app.ttles}"
```

It means app.ttles key is not exist in properties file.

***** In case of Spring Boot, @SpringBootApplication annotation loads by default application.properties into Spring container.
No need of writing :
@PropertySource("classpath:application.properties")
annotation.

---------code------------------------------------------

S#1 Create Spring Boot Starter Project
> File > New > Spring Starter Project
> Enter details
 Name : SpringBoot2PropertiesEx

Package : in.nareshit.raghu

> Next > Next > Finish

S#2 in application.properties

```
app.title-one=NIT
app.title-one=ABC
#Last combination is taken for duplicate keys
app.title-one=MNO
app.version=3.3
```

S#3 Define your own class

```
package in.narshit.raghu.bean;

import org.springframework.beans.factory.annotation.Value;
import org.springframework.stereotype.Component;
//ctrl+shift+O (imports)

@Component("ai")
public class AppInfo {

        @Value("${app.title-one}")
        private String ttl;

        @Value("${app.version}")
        private Double ver;

        //Source Menu > Generate toString() > Generate
        public String toString() {
                return "AppInfo [ttl=" + ttl + ", ver=" + ver + "]";
        }

}
```

S#4 Modify Starter class

```
package in.narshit.raghu;

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

import in.narshit.raghu.bean.AppInfo;

//ctrl+shift+O

@SpringBootApplication
public class SpringBoot2PropertiesExApplication {

        public static void main(String[] args) {
                ApplicationContext ac =
SpringApplication.run(SpringBoot2PropertiesExApplication.class, args);
                AppInfo a = ac.getBean("ai", AppInfo.class);
                System.out.println(a);
        }
}

======================================================================
==
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")

==================================================================
```