Date : 23/09/2020

Spring Boot 7AM

Mr. RAGHU

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

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

hr@nareshit.com

javabyraghu@gmail.com

**Maven Sessions:**

https://www.youtube.com/c/NareshIT/search?query=maven+raghu

Other videos:

https://www.youtube.com/watch?v=EA43S5R8LSc&list=PLVlQHNRLflP9XSWeY4x4FLwnL3UOIxnTr

Maven : Dependency Management Tool

=> Dependency: JAR Required for a Project

=> JAR-> Java Archive (similer to .zip/.rar -> Group of files)

It is a collection of .class files.

=> Developer#1 -> .java -> .class -> .jar

A.java

class A{

void m1(){}

}

my.jar [A.class]

=> Developer#2

+ added to project : my.jar

class B{

void m2(){

A a= new A(); //Jar not added? then ClassNotFoundException.

a.m1();

}

}

=> Dependency Hierarchy : Maven is adding actual jar and all its child/link jars to project ie called as Dependency Hierarchy.

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.2.8.RELEASE</version>

</dependency>

\*) Above dependency if we add in project then it gets downloaded 5 extra jars.

Q) How can we remove a child from a dependency Hierarchy?

A) using <exclusions> concept.

Ex:

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>5.2.8.RELEASE</version>

<exclusions>

<exclusion>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

</exclusion>

</exclusions>

</dependency>

Q) Why exclusions used in Realtime?

A) To avoid legacy APIs and add/migrate to new APIs (new concepts in place of old).

Ex: Remove junit-vintage-engine = JUnit 3/JUnit 4 support from Spring boot and add

only JUnit5 support.

---Ex:----

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

<exclusions>

<exclusion>

<groupId>org.junit.vintage</groupId>

<artifactId>junit-vintage-engine</artifactId>

</exclusion>

</exclusions>

</dependency>

Q) What are different Scopes in Maven ? Why they used?

A) Scope - When a Jar is loaded/used.

compile - Jar is used from compile time onwards

test - Jar used only for UnitTesting

runtime - Jar used only at runtime, not a compile time

provided - Jar is handled by Servers/Container/F/w

system - Jar loaded from System Drives (local folders)

import - Only for <dependencyManagement>

Q)\*\*\* How can we convert our project into JAR file/WAR file?

A) build : maven clean maven package

(or)

maven install

build: Convert Project into final format (.jar/.war)

need not to write: default <packaging>jar</packaging>

for web applications: <packaging>war</packaging>

Q) \*\*\* How Spring boot is handling jar/dependency versions?

A) By using Spring Boot Parent Project as : dependencyManagement tag

Q)\*\*\* How can we link our project with Spring Boot parent project?

A)By using a tag called : <parent> in pom.xml

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>2.3.4.RELEASE</version>

</parent>

Main class/Starter Class/Bootstrap class

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

Q) How can we load a properties file in Spring/Spring Boot manually?

A)

@PropertySource("classpath:abcd.properties")

@PropertySource("file:./abcd.properties") [inside project folder]

@PropertySource("file:d:/data/abcd.properties") [inside D:/data folder]

\*\* By using this annotation YAML files can not be loaded, only properties files.

YAML can be read using Snake YAML API only.

\*\* By default application.properties is loaded with priority.

Q) How spring container will scan/find classes in project?

A) By using: @ComponentScan("") by default value is starter class package name.

We can even modify default value, then starter class package rule never works.

Multiple packages can also be provided:

@ComponentScan({"pack1","pack2","pack3","pack4",.....})

Q)\*\*\* For which annotations Spring container creates object?

A)

a. For StereoType Annotations [ @Component]

b. Java Config Annotations [ @Bean ]

Some other types are sub type of @Component only :

@Repository, @Service, @Controller, @RestController..etc

Q) \*\*\* What is the difference between @Component and @Configuration?

A)

@Component : Used for creating object to our classes only.[never works for pre-defined]

@Configuration: Is used to indicate Input file to Spring container that holds object (@Bean)

details. It is called as Java based configuration file.

@Component is used for general purpose autodetection of component. Where as @Configurtaion is used to define the beans and its dependencies.

Q)\*\*\* What is the difference between @Bean and @Component?

A)

@Component : Create object for Programmer defined classes.

@Bean : Create object For both pre and Programmer defined classes. But used mostly

for pre-defined only.

Q)\*\*\* What is the Type of Spring container used in Spring Boot? What are other types even?

A) In Spring Boot Container type is : ApplicationContext

other type is : BeanFactory(legacy) that supports only XML Configuration.

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

Q) What are scopes in Spring? Which one is default? Why they are used?

A)

Scope - in programming is life time/accessability of data(variable/object..)

singleton (default)

prototype

request

session

application

Q)\*\* What is difference between @Inject, @Resource, @Value("#{obj}") and @Autowired?

A)

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