Date : 31-Aug-20

Spring Boot 7AM

Mr. RAGHU

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

hr@nareshit.com

Admin : Srikanth (+91-6302 968 665)

email: javabyraghu@gmail.com

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

Spring Boot Demo:

https://www.youtube.com/watch?v=xuMyjgqWWVs

https://www.youtube.com/watch?v=pcpaxSRN1tg

Youtube Playlist:

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

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

**Introduction Spring Boot (RAD)**

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

RAD = Rapid Application Development.

Faster Application development (reduce work done by programmer)

--3 Setup files----

a. Starter class | boot strap class | main() class

Java Class :

Create Spring Container and Start application

b. Input files (key-val)

|- application.properties

(or)

|- application.yml [YAML]

c. Build Information file

Maven -- pom.xml

Gradle -- build.gradle

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

a. Starter class | boot strap class | main() class:-

Spring Container:- [pre-defined application in Spring]

-> Create object to detected classes [non abstract class, not a interface]

-> Provide data to objects

-> Link one object with another object (HAS-A Relation)

-> Destroy the object (while stopping application/server).

--code---

@SpringBootApplication

public class DemoApp {

public static void main(String[] args) {

SpringApplication.run(DemoApp.class, args);

}

}

-------

\*) Two types of Container :

a. BeanFactory (I) [Legacy-old container] - works with XML Configuration

b. ApplicationContext (I) [New container] - works with XML/Java/Annotation

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

b. Input files (key-val)

|- application.properties

(or)

|- application.yml [YAML]

=> These are used to provide data to Spring container objects (key=val)

=> keys are two types

i. Pre-defined keys (95%)

https://docs.spring.io/spring-boot/docs/current/reference/html/appendix-application-properties.html

ii. Programmer defined keys(5%)

---Example: application.properties -------------

spring.datasource.driver-class-name=com.mysq.jdbc.driver.Driver

spring.datasource.url=jdbc:mysql://localhost:3306/test

spring.datasource.username=root

spring.datasource.password=root

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

---application.yml--------

spring:

datasource:

driver-class-name: com.mysq.jdbc.driver.Driver

url: jdbc:mysql://localhost:3306/test

username: root

password: root

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

YAML is format used to store data in key-val (without duplicate words).

Uses symbol : (colon). In Spring Boot , Snake YAML API is used , that converts

yml file to proprties file (Java understand only Properties).

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

c. Build Information file

Maven -- pom.xml

Gradle -- build.gradle

-> Jars of Hibernare, Servlets, Spring, Spring boot...

-> Folder System

-> Plugins (Jar, War, compiler..etc)

-> Phases and Goals

code -> compiled -> tested -> packed (.jar/.war) -> deployed

-- pom.xml------

a. Parent Project details

b. Child Project Details

c. Properties (jdk version, cloud version..etc)

d. Dependencies (JARS required for Project)

e. Dependeny Management (JARs with version Management)

f. Build plugins

...etc

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