Date : 22/12/2020
Spring Boot 9AM
Mr. RAGHU

Project Lombok API

This is OpenSource java api, used to generate code for
Constrcutors, set/get methods, toString, equals and hashCode..etc

=> To use Lombok API, we should enable/activate Lombok in STS/Eclipse
then we can use Lombok API Annotations.

---Lombok Activation Steps----
#1 Open IDE(Eclipse/STS/..) > create Spring Starter Project
Name : SpringBoot2LombokTest

> Next > Search using lombok > Select checkbox [v]Lombok
> Next > Finish

#2 Write below class (or any lombok annotation)

```
package in.nareshit.raghu;

import lombok.Getter;
import lombok.Setter;

@Setter
@Getter
public class Employee {

        private Integer id;
}
```

#3(a) Update Maven Project (if you are unable to find Lombok)
> Right click On Project > Maven > Update Project
> Choose Checkbox [v] Force Update...
> Finish

#3(b). Close IDEs.

#4 Goto below location
C:\Users\<UserName>\.m2\repository\org\projectlombok\lombok\1.18.16
Ex:
C:\Users\Raghu\.m2\repository\org\projectlombok\lombok\1.18.16

S#5 Open cmd prompt in this location
cmd> java -jar lombok-1.18.16.jar
(or double click on jar name : lombok-1.18.16.jar)

S#6 Wait for few minutes (or) Click on Specify Location of IDE
Ex: E:\SoftwaresLatest\sts-4.7.1.RELEASE (select location)

S#7 Click on Install/Update Button
S#8 Check success messsage and close

==========

Lombok Annotation

@Getter   : It will generate get() method for every variable
@Setter   : It will generate set() method
@ToString   : Generates toString ()

@EqualsAndHashCode : To compare two objects of same class , we must generate
                          equals() and hashCode().

@NoArgsConstructor : Lombok Generates default constrcutor
  *** Java compiler provides default constrcutor if no constrcuor exist in code.

@AllArgsConstructor : generate a constrcutor with all variables.

*) Lombok is going to work after your coding and before compilation only.
*) Lombok Dependency/jar not required once compilation is done.
   ie This JAR not required in production/Deployment time.

Q) Who will provide default/zero param constrcutor and when?
A) Java Compiler provides default/zero param constrcutor.
   If programmer did not write any constructor.

*** If either Programmer or Lombok , defining any param constructor,
   first define default/zero args constructor, then any param constructor is
   fine.

---------code---
#1. Model class

```
package in.nareshit.raghu;

import lombok.AllArgsConstructor;
import lombok.EqualsAndHashCode;
import lombok.Getter;
import lombok.NoArgsConstructor;
//ctrl+shift+O
import lombok.Setter;
import lombok.ToString;

@Getter
@Setter
@ToString
@NoArgsConstructor
@AllArgsConstructor
@EqualsAndHashCode
public class Employee {

        private Integer empId;
        private String empName;
        private Double empSal;
```

```
}

#2. Runner class
package in.nareshit.raghu;

import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;

@Component
public class LombokTestRunner implements CommandLineRunner {

        @Override
        public void run(String... args) throws Exception {
                Employee e1 = new Employee();
                e1.setEmpId(100);
                e1.setEmpName("AA");
                e1.setEmpSal(2500.0);

                Employee e2 = new Employee(101,"AA",200.0);

                //System.out.println(emp.getEmpId());
                System.out.println(e1);//toString
                System.out.println(e1.equals(e2)); //true or false?

        }
}
============================================================
*) @RequiredArgsConstructor + @NonNull
 If we want to create any specific params constrcutor then
 use above annotatons.

*) @NonNull will select variable for constructor param.
--Examples and Generated code by Lombok-----------------------
#1.
@RequiredArgsConstructor
class Emp {
  @NonNull
  Integer eid;
  String ename;
  Double esal;
}
-Generated code-
class Emp {
  Integer eid;
  String ename;
  Double esal;

  Emp(Integer eid) {
    this.eid=eid;
  }

}
-----------
#2
@RequiredArgsConstructor
class Emp {
```

```
  @NonNull
  Integer eid;
  String ename;
  @NonNull
  Double esal;
}
-Generated code-
class Emp {
  Integer eid;
  String ename;
  Double esal;

  Emp(Integer eid,Double esal) {
    this.eid=eid;
    this.esal=esal;
  }
}
---Ex#3----
@NoArgsConstructor
@RequiredArgsConstructor
class Emp {
  @NonNull
  Integer eid;
  String ename;
  Double esal;
}

-Generated code-
class Emp {
  Integer eid;
  String ename;
  Double esal;

  Emp() {
    super();
  }

  Emp(Integer eid) {
    super();
    this.eid=eid;
  }
}

---Ex#4----
@NoArgsConstructor
@RequiredArgsConstructor
@AllArgsConstructor
class Emp {
  @NonNull
  Integer eid;
  String ename;
  Double esal;
}
-Generated code-
class Emp {
  Integer eid;
  String ename;
```

```
  Double esal;

  Emp() {
    super();
  }

  Emp(Integer eid) {
    super();
    this.eid=eid;
  }

  Emp(Integer eid,String ename,Double esal) {
    super();
    this.eid=eid;
    this.ename=ename;
    this.esal=esal;
  }
}
-----------------------------------------------------------------
```