

Lombok

=====

steps to use the lombok in STS :

step1: download the lombok.jar file

step 2: execute this jar file

>java -jar lombok.jar

**specify the location of STS (STS main launcher file)
install and quit**

step 3:

add the lombok maven dependency inside pom.xml

Step 4: in order to reflect the lombok in our project update the maven project. and restart the STS

Annotations of Lombok:

@Data

@Setter

@Getter

@NoArgsConstructor

@ToString

@AllArgsConstructor

@EqualsAndHashCode

Swagger :-

=====

--after developing the api endpoints, we need to inform the consumer of the api, like ,structure of our api endpoints(including structure/type of the payload(req-resp body),http status code,exceptions,etc..)

--to provide those information to the consumer we need to prepare a document and share with the consumer.

in SOAP based WS we use WSDL file (xml file)

--in REST we can generate this documentation in autogenerated format.

--there is no any kind of restriction to generate the api documentation of REST api.

--in spring Boot we use swagger tool to generate the documentation for the REST api.

--Swagger is an open specification to develop a REST api documentation tool.

--Springfox is the implementation of this Swagger specification

steps the add swagger in our spring boot application:-

step 1:- we need to add "springfox boot starter" dependency

<!-- https://mvnrepository.com/artifact/io.springfox/springfox-boot-starter -->

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-boot-starter</artifactId>

<version>3.0.0</version>

</dependency>

--It will generate json documentaion of our REST api at /v2/api-docs GET

--and can generate REST api documentation in human readable html at : /swagger-ui/
GET

step 2:- enable the swagger in our spring boot application

apply `@EnableSwagger2` annotation on the top of main class / `@SpringBootApplication`
annotated class.

step 3:- configuring the swagger to provide some extra info about our api.

step 4:- add the following entry inside application.properties file:-

`spring.mvc.pathmatch.matching-strategy = ANT_PATH_MATCHER`