

```
1  =====Instructions=====
2  Create a new Spring starter project
3  dependencies web,lombok
4
5  To work with lombok
6  Exit/Shutdown STS
7  Find lombok Jar in ~/.m2/repository/org/projectlombok/lombok/version.x
8  From Command Prompt/Shell java -jar lombok-1.x.y.jar
9  start STS
10 right-click on project -> Maven -> Update Project
11
12 Note that if you're using IntelliJ, you'll want to install the Lombok plugin
13 (available from IDE settings)
14 and also enable annotation processing.
15
16 =====pom.xml=====
17 <?xml version="1.0" encoding="UTF-8"?>
18 <project xmlns="http://maven.apache.org/POM/4.0.0"
19 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
20 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
21 http://maven.apache.org/xsd/maven-4.0.0.xsd">
22 <modelVersion>4.0.0</modelVersion>
23 <parent>
24 <groupId>org.springframework.boot</groupId>
25 <artifactId>spring-boot-starter-parent</artifactId>
26 <version>2.3.0.RELEASE</version>
27 <relativePath/> <!-- lookup parent from repository -->
28 </parent>
29 <groupId>com.ameya</groupId>
30 <artifactId>008-SpringBootLombok</artifactId>
31 <version>0.0.1-SNAPSHOT</version>
32 <name>008-SpringBootLombok</name>
33 <description>Demo project for Spring Boot</description>
34 <properties>
35 <java.version>1.8</java.version>
36 </properties>
37 <dependencies>
38 <dependency>
39 <groupId>org.springframework.boot</groupId>
40 <artifactId>spring-boot-starter-web</artifactId>
41 </dependency>
42
43 <dependency>
44 <groupId>org.projectlombok</groupId>
45 <artifactId>lombok</artifactId>
46 <optional>true</optional>
47 </dependency>
48 <dependency>
49 <groupId>org.springframework.boot</groupId>
50 <artifactId>spring-boot-starter-test</artifactId>
51 <scope>test</scope>
52 <exclusions>
53 <exclusion>
54 <groupId>org.junit.vintage</groupId>
55 <artifactId>junit-vintage-engine</artifactId>
56 </exclusion>
57 </exclusions>
58 </dependency>
59 </dependencies>
60
61 <build>
62 <plugins>
63 <plugin>
64 <groupId>org.springframework.boot</groupId>
65 <artifactId>spring-boot-maven-plugin</artifactId>
66 <configuration>
67 <excludes>
68 <exclude>
69 <groupId>org.projectlombok</groupId>
70 <artifactId>lombok</artifactId>
71 </exclude>
72 </excludes>
73 </configuration>
74 </plugin>
75 </plugins>
76 </build>
77 </project>
```

```

71         </plugin>
72     </plugins>
73 </build>
74
75 </project>
76 =====Message.java=====
77 package com.ameya.models;
78
79 import lombok.Data;
80
81 @Data
82 /*
83 Generates getters for all fields, a useful toString method, and hashCode and equals
84 implementations that check all non-transient fields.
85 Will also generate setters for all non-final fields, as well as a constructor.
86 Equivalent to @Getter @Setter
87 @RequiredArgsConstructor (Generates a constructor with required arguments.
88 Required arguments are final fields and fields with constraints such as @NonNull.)
89 @ToString @EqualsAndHashCode.
90 */
91 public class Message {
92
93     private Long id;
94     private String message;
95 }
96 =====MessageService.java=====
97 package com.ameya.services;
98
99 import org.springframework.stereotype.Service;
100
101 import com.ameya.models.Message;
102
103 import lombok.NonNull;
104
105 @Service
106 public class MessageService {
107
108     public Message createMessage(@NonNull String user) {
109         Message message=new Message();
110         message.setId(Long.valueOf(1));
111         message.setMessage("Hello "+user.toUpperCase()+" Welcome To Lombok !!");
112         return message;
113     }
114 }
115 =====MessageController.java=====
116 package com.ameya.controllers;
117
118 import org.springframework.beans.factory.annotation.Autowired;
119 import org.springframework.web.bind.annotation.GetMapping;
120 import org.springframework.web.bind.annotation.RequestParam;
121 import org.springframework.web.bind.annotation.RestController;
122
123 import com.ameya.models.Message;
124 import com.ameya.services.MessageService;
125
126 import lombok.extern.slf4j.Slf4j;
127
128 @RestController
129 @Slf4j
130 public class MessageController {
131
132     @Autowired
133     private MessageService messageService;
134     @GetMapping(path="/welcome")
135     public Message greetUser(@RequestParam String user) {
136         log.info("USER =>" +user);
137         Message message=messageService.createMessage(user);
138         log.info("MESSAGE =>" +message.toString());
139         return message;
140     }
141 }
142 =====
143 http://localhost:9001/welcome?user=Ameya

```