

```

1  =====pom.xml=====
2  <?xml version="1.0" encoding="UTF-8"?>
3  <project xmlns="http://maven.apache.org/POM/4.0.0"
4  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
6  https://maven.apache.org/xsd/maven-4.0.0.xsd">
7  <modelVersion>4.0.0</modelVersion>
8  <parent>
9  <groupId>org.springframework.boot</groupId>
10 <artifactId>spring-boot-starter-parent</artifactId>
11 <version>2.3.0.RELEASE</version>
12 <relativePath/> <!-- lookup parent from repository -->
13 </parent>
14 <groupId>com.ameya</groupId>
15 <artifactId>002-AOP</artifactId>
16 <version>0.0.1-SNAPSHOT</version>
17 <name>002-AOP</name>
18 <description>Demo project for Spring Boot</description>
19 <properties>
20 <java.version>1.8</java.version>
21 </properties>
22 <dependencies>
23 <dependency>
24 <groupId>org.springframework.boot</groupId>
25 <artifactId>spring-boot-starter-aop</artifactId>
26 </dependency>
27 <dependency>
28 <groupId>org.springframework.boot</groupId>
29 <artifactId>spring-boot-starter-web</artifactId>
30 </dependency>
31 <dependency>
32 <groupId>org.springframework.boot</groupId>
33 <artifactId>spring-boot-starter-test</artifactId>
34 <scope>test</scope>
35 <exclusions>
36 <exclusion>
37 <groupId>org.junit.vintage</groupId>
38 <artifactId>junit-vintage-engine</artifactId>
39 </exclusion>
40 </exclusions>
41 </dependency>
42 </dependencies>
43 <build>
44 <plugins>
45 <plugin>
46 <groupId>org.springframework.boot</groupId>
47 <artifactId>spring-boot-maven-plugin</artifactId>
48 </plugin>
49 </plugins>
50 </build>
51
52 </project>
53 =====application.properties=====
54 =====
55 server.port=9001
56 logging.level.web=trace
57 =====Employee.java=====
58 =====
59 package com.ameya.models;
60
61 public class Employee {
62     private int age;
63     private String firstName;
64     private String lastName;
65     private double salary;
66     public Employee() {}
67     public Employee(int age, String firstName, String lastName, double salary) {
68         super();
69         this.age = age;
70         this.firstName = firstName;
71         this.lastName = lastName;

```

```

70         this.salary = salary;
71     }
72     public int getAge() {
73         return age;
74     }
75     public void setAge(int age) {
76         this.age = age;
77     }
78     public String getFirstName() {
79         return firstName;
80     }
81     public void setFirstName(String firstName) {
82         this.firstName = firstName;
83     }
84     public String getLastName() {
85         return lastName;
86     }
87     public void setLastName(String lastName) {
88         this.lastName = lastName;
89     }
90     public double getSalary() {
91         return salary;
92     }
93     public void setSalary(double salary) {
94         this.salary = salary;
95     }
96     @Override
97     public String toString() {
98         return "Employee [age=" + age + ", firstName=" + firstName + ", lastName=" +
99             lastName + ", salary=" + salary
100             + "]";
101     }
102
103 }
104 =====EmployeeService.java=====
105
106 package com.ameya.services;
107
108 import org.springframework.stereotype.Service;
109
110 import com.ameya.models.Employee;
111
112 @Service
113 public class EmployeeService {
114
115     public Employee createEmployee(int age,String firstName,
116         String lastName,double salary) {
117         Employee emp=new Employee(age, firstName, lastName, salary);
118         System.out.println("++++ EmployeeService.createEmployee ++++"+emp);
119
120         return emp;
121     }
122     public void updateEmployee(Employee employee) {
123         System.out.println("++++ EmployeeService.updateEmployee "
124             + "Employee Updated ++++");
125     }
126     public void deleteEmployee() {
127         System.out.println("++++ EmployeeService.deleteEmployee "
128             + "Employee Deleted ++++");
129     }
130 }
131 =====EmployeeController.java=====
132
133 package com.ameya.controllers;
134
135 import org.springframework.beans.factory.annotation.Autowired;
136 import org.springframework.web.bind.annotation.GetMapping;
137 import org.springframework.web.bind.annotation.RequestMapping;
138 import org.springframework.web.bind.annotation.RestController;
139
140 import com.ameya.models.Employee;
141 import com.ameya.services.EmployeeService;

```

```

140
141 @RestController
142 public class EmployeeController {
143
144     @Autowired
145     private EmployeeService employeeService;
146     @GetMapping(path="/add/employee")
147     public Employee addEmployee(@RequestParam("age") int age,
148                                @RequestParam("fname") String firstName,
149                                @RequestParam("lname") String lastName,
150                                @RequestParam("sal") double salary) {
151         return employeeService.createEmployee(age, firstName, lastName, salary);
152     }
153     @GetMapping(path="/delete/employee")
154     public String delete() {
155         employeeService.deleteEmployee();
156         return "Employee Deleted !!";
157     }
158     @GetMapping(path="/update/employee")
159     public String update(Employee employee) {
160         employeeService.updateEmployee(employee);
161         return "Employee Updated !!";
162     }
163 }
164 =====LoggingAspectJ.java=====
=====
165 package com.ameya.aspects;
166
167 import org.aspectj.lang.JoinPoint;
168 import org.aspectj.lang.ProceedingJoinPoint;
169 import org.aspectj.lang.annotation.After;
170 import org.aspectj.lang.annotation.Around;
171 import org.aspectj.lang.annotation.Aspect;
172 import org.aspectj.lang.annotation.Before;
173 import org.aspectj.lang.annotation.Pointcut;
174 import org.slf4j.Logger;
175 import org.slf4j.LoggerFactory;
176 import org.springframework.stereotype.Component;
177
178 @Component
179 @Aspect
180 public class LoggingAspectJ {
181
182     private static final Logger logger=LoggerFactory.
183         getLogger(LoggingAspectJ.class);
184     @Pointcut("execution(* com.ameya.services.EmployeeService.cre*(..))")
185     public void myPointcut() {}
186
187     @Before(value="myPointcut()")
188     public void beforeAdvice(JoinPoint joinPoint) {
189         logger.info("***** Before Advice ---> Before Method =>" +
190             joinPoint.getSignature());
191     }
192     @After(value="myPointcut()")
193     public void afterAdvice(JoinPoint joinPoint) {
194         logger.info("***** After Advice ---> After Method =>" +
195             joinPoint.getSignature());
196     }
197     @Around(value="myPointcut()")
198     public Object aroundAdvice(ProceedingJoinPoint joinPoint)
199         throws Throwable {
200         Object object=null;
201         logger.info("***** Around Advice ---> Before Method =>" +
202             joinPoint.getSignature());
203         try {
204             object=joinPoint.proceed();
205             logger.info("***** Around Advice ---> After Method =>" +
206                 joinPoint.getSignature());
207         }catch(Throwable t) {
208             throw t;
209         }
210         return object;
211     }

```

```
212 }
213 =====Application.java=====
214 package com.ameya;
215
216 import org.springframework.boot.SpringApplication;
217 import org.springframework.boot.autoconfigure.SpringBootApplication;
218 import org.springframework.context.annotation.EnableAspectJAutoProxy;
219
220 @SpringBootApplication
221 @EnableAspectJAutoProxy(proxyTargetClass = true)
222 public class Application {
223
224     public static void main(String[] args) {
225         SpringApplication.run(Application.class, args);
226     }
227
228 }
229 =====
230
231 http://localhost:9001/add/employee?age=45&fname=Ameya&lname=Joshi&sal=45000
232 http://localhost:9001/delete/employee
```