

```

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>010-SpringProfiles</artifactId>
16 <version>0.0.1-SNAPSHOT</version>
17 <name>010-SpringProfiles</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-web</artifactId>
26 </dependency>
27
28 <dependency>
29 <groupId>org.springframework.boot</groupId>
30 <artifactId>spring-boot-starter-test</artifactId>
31 <scope>test</scope>
32 <exclusions>
33 <exclusion>
34 <groupId>org.junit.vintage</groupId>
35 <artifactId>junit-vintage-engine</artifactId>
36 </exclusion>
37 </exclusions>
38 </dependency>
39 </dependencies>
40
41 <build>
42 <plugins>
43 <plugin>
44 <groupId>org.springframework.boot</groupId>
45 <artifactId>spring-boot-maven-plugin</artifactId>
46 </plugin>
47 </plugins>
48 </build>
49
50 </project>
51 =====application.properties=====
52 spring.profiles.active=prod
53 server.port=9003
54 spring.application.name=SpringProfiles
55 app.message=This is the primary Application Property ${spring.application.name}
56
57 =====application-dev.properties=====
58
59 app.message=This is the property file for the ${spring.application.name} specific to
60 DEV Environment
61
62 spring.datasource.driver-class-name=org.h2.Driver
63 spring.datasource.url=jdbc:h2:mem:db;DB_CLOSE_DELAY=-1
64 spring.datasource.username=sa
65 spring.datasource.password=sa
66
67 =====application-test.properties=====
68
69 app.message=This is the property file for the ${spring.application.name} specific to
70 TEST Environment
71
72 spring.datasource.driver-class-name=com.mysql.cj.Driver
73 spring.datasource.url=jdbc:mysql://localhost:3306/testDB

```

```

70 spring.datasource.username=Test_Username
71 spring.datasource.password=Test_Password
72
73 =====application-prod.properties=====
74
75 app.message=This is the property file for the ${spring.application.name} specific to
PROD Environment
76
77 spring.datasource.driver-class-name=oracle.jdbc.OracleDriver
78 spring.datasource.url=jdbc:oracle:thin:@localhost:1521:XE
79 spring.datasource.username=ora_username
80 spring.datasource.password=ora_password
81
82 =====ProfileWiseDBConfig.java=====
83 package com.ameya.config;
84
85 import org.springframework.boot.context.properties.ConfigurationProperties;
86 import org.springframework.context.annotation.Bean;
87 import org.springframework.context.annotation.Configuration;
88 import org.springframework.context.annotation.Profile;
89
90 @Configuration
91 @ConfigurationProperties("spring.datasource")
92 /*
93  Spring uses some relaxed rules for binding properties.
94  So the following variations are all bound to the property hostName:
95 mail.hostName
96 mail.hostname
97 mail.host_name
98 mail.host-name
99 mail.HOST_NAME
100 */
101 public class ProfilewiseDBConfig {
102
103     private String driverClassName;
104     private String url;
105     private String username;
106     private String password;
107     public String getDriverClassName() {
108         return driverClassName;
109     }
110     public void setDriverClassName(String driverClassName) {
111         this.driverClassName = driverClassName;
112     }
113     public String getUrl() {
114         return url;
115     }
116     public void setUrl(String url) {
117         this.url = url;
118     }
119     public String getUsername() {
120         return username;
121     }
122     public void setUsername(String username) {
123         this.username = username;
124     }
125     public String getPassword() {
126         return password;
127     }
128     public void setPassword(String password) {
129         this.password = password;
130     }
131
132     @Profile("dev")
133     @Bean
134     public String devDBConnection() {
135         System.out.println("DB For DEV - H2");
136         System.out.println(driverClassName);
137         System.out.println(url);
138         return "DB For DEV - H2";
139     }
140     @Profile("test")
141     @Bean

```

```

142     public String testDBConnection() {
143         System.out.println("DB For Test - MySQL");
144         System.out.println(driverClassName);
145         System.out.println(url);
146         return "DB For Test - MySQL";
147     }
148     @Profile("prod")
149     @Bean
150     public String prodDBConnection() {
151         System.out.println("DB For Prod - Oracle");
152         System.out.println(driverClassName);
153         System.out.println(url);
154         return "DB For Prod - Oracle";
155     }
156 }
157 =====ProfileAppController,java=====
=====
158 package com.ameya.controllers;
159
160 import org.springframework.beans.factory.annotation.Value;
161 import org.springframework.web.bind.annotation.GetMapping;
162 import org.springframework.web.bind.annotation.RestController;
163
164 @RestController
165 public class ProfileAppController {
166
167     @Value("${app.message}")
168     private String welcomeMessage;
169     @GetMapping("/welcome")
170     public String getDBConnectionDetails() {
171         return welcomeMessage;
172     }
173 }
174 =====
175 Run the app 3 times setting ports to 9001, 9002,9003 and Each time set the active
176 profile to prod dev test respectively in application.properties observe the console
177 invoke localhost:9001/welcome
178 invoke localhost:9002/welcome
179 invoke localhost:9003/welcome

```