

```
1 HOL : 외부 파일을 이용한 설정
2 -----
3 Task1. Lab
4 1. In Package Explorer > right-click > New > Java Project
5     1)Project Name : EnvironmentDemo
6     2)JRE
7         -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
8     3)Next
9     4)Uncheck [Create module-info.java file]
10    5)Finish
11
12
13 2. src > right-click > New > Package
14     1)Name : com.example
15     2)Finish
16
17
18 3. Java Project를 Spring Project로 변환
19     1)EnvironmentDemo Project > right-click > Configure > Convert to Maven Project
20         -Project : /EnvironmentDemo
21         -Group Id : EnvironmentDemo
22         -Artifact Id : EnvironmentDemo
23         -version : 0.0.1-SNAPSHOT
24         -Packaging : jar
25         -Finish
26
27     2)EnvironmentDemo Project > right-click > Spring > Add Spring Project Nature
28
29     3)pom.xml file에 Spring Context Dependency 추가하기
30         <version>0.0.1-SNAPSHOT</version>
31         <dependencies>
32             <dependency>
33                 <groupId>org.springframework</groupId>
34                 <artifactId>spring-context</artifactId>
35                 <version>5.2.5.RELEASE</version>
36             </dependency>
37         </dependencies>
38
39     4)pom.xml > right-click > Run As > Maven install
40         [INFO] BUILD SUCCESS 확인
41
42
43 4. Lombok library 추가
44     1)https://mvnrepository.com/에서 'lombok'으로 검색
45     2)'Project Lombok' click
46     3)1.18.12 click
47     4)dependency copy해서 pom.xml에 붙여넣기
48
49     <dependencies>
50         <dependency>
51             <groupId>org.springframework</groupId>
52             <artifactId>spring-context</artifactId>
53             <version>5.2.5.RELEASE</version>
54         </dependency>
55         <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
56         <dependency>
57             <groupId>org.projectlombok</groupId>
58             <artifactId>lombok</artifactId>
```

```
59         <version>1.18.12</version>
60         <scope>provided</scope>
61     </dependency>
62 </dependencies>
63
64 5)pom.xml > right-click > Run As > Maven install
65 [INFO] BUILD SUCCESS 확인
66
67
68 5. EnvironmentDemo/resources folder 생성
69 1)EnvironmentDemo project > right-click > New > Source Folder
70 2)Folder name : resources
71 3)Finish
72
73
74 6. resources > right-click > New > File
75 1)File name : admin.properties
76 2)Finish
77
78     admin.id=javaexpert
79     admin.pwd=12345678
80
81
82 7. com.example.AdminConnection 생성
83 1)com.example > right-click > New > Class
84 2)Name : AdminConnection
85 3)Finish
86
87     package com.example;
88
89     import org.springframework.beans.factory.DisposableBean;
90     import org.springframework.beans.factory.InitializingBean;
91     import org.springframework.context.EnvironmentAware;
92     import org.springframework.core.env.Environment;
93
94     import lombok.Getter;
95     import lombok.Setter;
96
97     public class AdminConnection implements EnvironmentAware, InitializingBean,
98     DisposableBean {
99         @Setter
100         private Environment env;
101         @Getter
102         @Setter
103         private String adminId;
104         @Getter
105         @Setter
106         private String adminPwd;
107
108         @Override
109         public void destroy() throws Exception {
110             System.out.println("destroy()");
111         }
112
113         @Override
114         public void afterPropertiesSet() throws Exception {
115             System.out.println("afterPropertiesSet()");
116             setAdminId(env.getProperty("admin.id"));
```

```

116         setAdminPwd(env.getProperty("admin.pwd"));
117     }
118
119     // bean이 생성되기 전에 callback 으로 호출됨. 가장 먼저 호출됨.
120     // MainClass에서 사용하는 env 정보가 넘어옴.
121     @Override
122     public void setEnvironment(Environment env) {
123         System.out.println("setEnvironment()");
124         setEnv(env);
125     }
126 }
127
128
129 8. resources > right-click > New > Spring Bean Configuration File
130 1)File name : beans.xml
131 2)Finish
132
133     <?xml version="1.0" encoding="UTF-8"?>
134     <beans xmlns="http://www.springframework.org/schema/beans"
135           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
136           xsi:schemaLocation="http://www.springframework.org/schema/beans
137                               http://www.springframework.org/schema/beans/spring-beans.xsd">
138
139         <bean id="adminConnection" class="com.example.AdminConnection" />
140
141     </beans>
142
143 9. com.example.MainClass Class 생성
144 1)com.example > right-click > New > Class
145 2)Name : MainClass
146 3)Finish
147
148     package com.example;
149
150     import java.io.IOException;
151
152     import org.springframework.context.ConfigurableApplicationContext;
153     import org.springframework.context.support.GenericXmlApplicationContext;
154     import org.springframework.core.env.ConfigurableEnvironment;
155     import org.springframework.core.env.MutablePropertySources;
156     import org.springframework.core.io.support.ResourcePropertySource;
157
158     public class MainClass {
159         public static void main(String [] args){
160             ConfigurableApplicationContext ctx = new GenericXmlApplicationContext();
161             ConfigurableEnvironment env = ctx.getEnvironment();
162
163             MutablePropertySources propertySouces = env.getPropertySources();
164             //내가 원하는 정보를 얻을 때까지 모든 propertySources를 앞에서 부터 차례로 모두 검색함.
165             try{
166                 propertySouces.addLast(new
167                     ResourcePropertySource("classpath:admin.properties")); //property 추가
168
169                 System.out.println(env.getProperty("admin.id")); //property 추출
170                 System.out.println(env.getProperty("admin.pwd"));
171             }catch(IOException ex){}

```

```
172     GenericXmlApplicationContext gCtx = (GenericXmlApplicationContext)ctx;
173     gCtx.load("classpath:beans.xml");
174     gCtx.refresh();
175
176     AdminConnection adminConnection = gCtx.getBean("adminConnection",
177         AdminConnection.class);
178     System.out.println("admin ID : " + adminConnection.getAdminId());
179     System.out.println("admin PWD : " + adminConnection.getAdminPwd());
180
181     gCtx.close();
182     ctx.close();
183 }
184 }
185
```

10. 실행

186 1)MainClass > right-click > Run As > Java Application

```
187
188
189     javaexpert
190     12345678
191     setEnvironment()
192     afterPropertiesSet()
193     admin ID : javaexpert
194     admin PWD : 12345678
195     destroy()
196
197
198
199 -----
```

Task2. Lab

200 1. In Package Explorer > right-click > New > Java Project

201 1)Project Name : PropertyDemo

202 2)JRE

203 -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]

204 3)Next

205 4)Uncheck [Create module-info.java file]

206 5)Finish

207

208

209

210 2. src > right-click > New > Package

211 1)Name : com.example

212 2)Finish

213

214

215 3. POJO Class 작성

216 1)com.example > right-click > New > Class

217 2)Name : Hello

218 3)Finish

219

220 package com.example;

221

222 import java.util.List;

223

224 public class Hello {

225 private String name;

226 private Printer printer;

227 private List<String> names;

228

```
229     public String sayHello(){
230         return "Hello " + name;
231     }
232
233     public void print(){
234         this.printer.print(sayHello());
235     }
236 }
237
238 4)com.example > right-click > New > Interface
239 5)Name : Printer
240 6)Finish
241
242     package com.example;
243
244     public interface Printer{
245         void print(String message);
246     }
247
248 7)com.example > right-click > New > Class
249 8)Name : StringPrinter
250 9)Interfaces : com.example.Printer
251 10)Finish
252
253     package com.example;
254
255     public class StringPrinter implements Printer{
256         private StringBuffer buffer = new StringBuffer();
257
258         @Override
259         public void print(String message){
260             this.buffer.append(message);
261         }
262
263         public String toString(){
264             return this.buffer.toString();
265         }
266     }
267
268 11)com.example > right-click > New > Class
269 12)Name : ConsolePrinter
270 13)Interfaces : com.example.Printer
271 14)Finish
272
273     package com.example;
274
275     public class ConsolePrinter implements Printer{
276
277         @Override
278         public void print(String message){
279             System.out.println(message);
280         }
281     }
282
283
284 4. Java Project를 Spring Project로 변환
285 1)PropertyDemo Project > right-click > Configure > Convert to Maven Project
286     -Project : /PropertyDemo
```

```
287 -Group Id : PropertyDemo
288 -Artifact Id : PropertyDemo
289 -version : 0.0.1-SNAPSHOT
290 -Packaging : jar
291 -Finish
292
293 2)PropertyDemo Project > right-click > Spring > Add Spring Project Nature
294
295 3)pom.xml file에 Spring Context Dependency 추가하기
296 <version>0.0.1-SNAPSHOT</version>
297 <dependencies>
298 <dependency>
299 <groupId>org.springframework</groupId>
300 <artifactId>spring-context</artifactId>
301 <version>5.2.5.RELEASE</version>
302 </dependency>
303 </dependencies>
304
305 4pom.xml > right-click > Run As > Maven install
306 [INFO] BUILD SUCCESS 확인
307
308
309 5. Lombok library 추가
310 1)https://mvnrepository.com/에서 'lombok'으로 검색
311 2)'Project Lombok' click
312 3)1.18.12 click
313 4)dependency copy해서 pom.xml에 붙여넣기
314
315 <dependencies>
316 <dependency>
317 <groupId>org.springframework</groupId>
318 <artifactId>spring-context</artifactId>
319 <version>5.2.5.RELEASE</version>
320 </dependency>
321 <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
322 <dependency>
323 <groupId>org.projectlombok</groupId>
324 <artifactId>lombok</artifactId>
325 <version>1.18.12</version>
326 <scope>provided</scope>
327 </dependency>
328 </dependencies>
329
330 5)pom.xml > right-click > Run As > Maven install
331 [INFO] BUILD SUCCESS 확인
332
333
334 6. Hello.java에 lombok Annotation 붙이기
335 1)Hello.java
336
337 package com.example;
338
339 import java.util.List;
340
341 import lombok.Getter;
342 import lombok.NoArgsConstructor;
343 import lombok.Setter;
344
```

```
345     @NoArgsConstructor
346     @Setter
347     public class Hello {
348         private String name;
349         private Printer printer;
350         @Getter private List<String> names;
351
352         public String sayHello(){
353             return "Hello " + name;
354         }
355
356         public void print(){
357             this.printer.print(sayHello());
358         }
359     }
360
361
362 7. PropertyDemo/resources folder 생성
363     1)PropertyDemo project > right-click > New > Source Folder
364     2)Folder name : resources
365     3)Finish
366
367
368 8. Bean Configuration XML 작성
369     1)PropertyDemo/resources > right-click > New > Spring Bean Configuration File
370     2)File name : beans.xml
371     3)Finish
372
373     <?xml version="1.0" encoding="UTF-8"?>
374     <beans xmlns="http://www.springframework.org/schema/beans"
375         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
376         xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">
377
378         <bean id="hello" class="com.example.Hello">
379             <property name="name" value="Spring" />
380             <property name="printer" ref="printer" />
381             <property name="names">
382                 <list>
383                     <value>AOP</value>
384                     <value>Spring</value>
385                     <value>DI</value>
386                 </list>
387             </property>
388         </bean>
389
390         <bean id="printer" class="com.example.StringPrinter" />
391         <bean id="consolePrinter" class="com.example.ConsolePrinter" />
392
393     </beans>
394
395
396 9. com.example.MainClass Class 생성
397     1)com.example > right-click > New > Class
398     2)Name : MainClass
399     3)Finish
400
401     package com.example;
```

```

402
403 import org.springframework.context.ApplicationContext;
404 import org.springframework.context.support.GenericXmlApplicationContext;
405
406 public class MainClass {
407     public static void main(String [] args){
408         ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
409
410         Hello hello = (Hello)ctx.getBean("hello");
411         System.out.println(hello.sayHello());
412         hello.print();
413
414         Printer printer = ctx.getBean("printer", StringPrinter.class);
415         System.out.println(printer.toString());
416
417         hello.getNames().forEach(value -> System.out.println(value));
418     }
419 }
420
421
422 10. 실행
423 1)MainClass > right-click > Run As > Java Application
424
425     Hello Spring
426     Hello Spring
427     AOP
428     Spring
429     DI
430
431
432 11. JUnit 5 & Spring TestContext Framework으로 Test
433 1)src > right-click > New > Package
434 2)Name : com.example.test
435 3)Finish
436 4)com.example.test > right-click > New > JUnit Test Case > Select [New JUnit Jupiter test]
437 5)Name : HelloJUnitTest
438 6)Finish
439 7)New JUnit Test Case창에서, [Perform the following action: Add JUnit 5 library to the build
440 path]
441 8)OK
442 9)https://mvnrepository.com에서 'spring-test'로 검색
443 10)Spring TestContext Framework에서
444 11)5.2.5.RELEASE로 들어가서
445 12)복사해서 pom.xml로 붙여넣기
446     <!-- https://mvnrepository.com/artifact/org.springframework/spring-test -->
447     <dependency>
448         <groupId>org.springframework</groupId>
449         <artifactId>spring-test</artifactId>
450         <version>5.2.5.RELEASE</version>
451         <scope>test</scope>
452     </dependency>
453
454 13)pom.xml > right-click > Run As > Maven install
455     [INFO] BUILD SUCCESS 확인
456     -만일 Error 발생하면 PropertyDemo > right-click > Maven > Update Project > OK
457     -pom.xml > right-click > Run As > Maven install
458     -[INFO] BUILD SUCCESS 확인

```



```
459
460 12. com.example.test.HelloJUnitTest
461
462 package com.example.test;
463
464 import static org.junit.jupiter.api.Assertions.assertEquals;
465 import static org.junit.jupiter.api.Assertions.assertSame;
466
467 import org.junit.jupiter.api.Test;
468 import org.junit.jupiter.api.extension.ExtendWith;
469 import org.springframework.beans.factory.annotation.Autowired;
470 import org.springframework.context.ApplicationContext;
471 import org.springframework.test.context.ContextConfiguration;
472 import org.springframework.test.context.junit.jupiter.SpringExtension;
473
474 import com.example.Hello;
475 import com.example.Printer;
476
477 @ExtendWith(SpringExtension.class)
478 @ContextConfiguration(locations = "classpath:beans.xml")
479 class HelloJUnitTest {
480     @Autowired
481     ApplicationContext ctx;
482
483     @Test
484     public void test() {
485         Hello hello = (Hello) ctx.getBean("hello");
486         assertEquals("Hello Spring", hello.sayHello());
487         hello.print();
488
489         Printer printer = (Printer) ctx.getBean("printer");
490         assertEquals("Hello Spring", printer.toString());
491     }
492
493     @Test
494     public void test2() {
495         Hello hello = (Hello) ctx.getBean("hello");
496
497         Hello hello2 = ctx.getBean("hello", Hello.class);
498         assertSame(hello, hello2);
499
500         assertEquals(3, hello2.getNames().size());
501     }
502 }
503
504
505 1)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면
506     -해당 Project > right-click > Build Path > Libraries tab
507     -spring-test-5.2.5.RELEASE.jar 선택 후 [Remove] 로 삭제
508     -[Add External JARs...] Click
509     -%M2_HOME%\repository\org\springframework\spring-test\5.2.5.RELEASE 선택할 것
510     -[Order and Export] tab에서 spring-test-5.2.5.RELEASE.jar 선택 후 [Up] button을 클릭
511     -해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click
512
513 2)right-click > Run As > Junit Test
514 3)결과 -> Junit View에 초록색 bar
515
516
```

```
517 13. resources/value.properties 생성
518
519 myname=Spring
520 myprinter=printer
521 value1=HTML5
522 value2=CSS3
523 value3=JavaScript
524
525
526 14. /resources/beans.xml 수정하기
527 1)beans.xml에서 [Namespaces] tab
528 2)목록에서 'context-http://www.springframework.org/schema/context' check
529 3)<context:property-placeholder />를 사용하기 위해서 다음과 같이 수정
530
531     <context:property-placeholder
532         location="classpath:value.properties" />
533
534     <bean id="hello" class="com.example.Hello">
535         <property name="name" value="${myname}" />
536         <property name="printer" ref="${myprinter}" />
537         <property name="names">
538             <list>
539                 <value>${value1}</value>
540                 <value>${value2}</value>
541                 <value>${value3}</value>
542             </list>
543         </property>
544     </bean>
545
546     <bean id="printer" class="com.example.StringPrinter" />
547     <bean id="consolePrinter" class="com.example.ConsolePrinter" />
548
549
550 15. Test
551 1)com.example.MainClass.java
552 2)right-click > Run As > Java Application
553
554     Hello Spring
555     Hello Spring
556     HTML5
557     CSS3
558     JavaScript
559
560 3)/src/test/java/HelloJUnitTest.java
561 -right-click > Run As > JUnit Test
562 -Green Bar
563
564
565 16. resources/value.properties 수정
566 myname=Spring
567 myprinter=printer
568 value1=HTML5
569 value2=CSS3
570 value3=JavaScript
571 printer1=stringPrinter
572 printer2=consolePrinter
573
574
```

```
575 17. pom.xml에 javax.annotation API dependency 추가하기
576 1)https://mvnrepository.com에서 'javax annotation'로 검색
577 2)Javax Annotation API click
578 3)1.3.2. click
579 4)dependency 복사해서 pom.xml로 붙여넣기
580 <!-- https://mvnrepository.com/artifact/javax.annotation/javax.annotation-api -->
581 <dependency>
582     <groupId>javax.annotation</groupId>
583     <artifactId>javax.annotation-api</artifactId>
584     <version>1.3.2</version>
585 </dependency>
586
587 5)pom.xml > right-click > Run As > Maven install
588 [INFO] BUILD SUCCESS 확인
589
590
591 18. Hello.java code 수정
592 1)com.example/Hello.java
593
594     package com.example;
595
596     import java.util.List;
597
598     import javax.annotation.Resource;
599
600     import org.springframework.beans.factory.annotation.Value;
601     import org.springframework.stereotype.Component;
602
603     import lombok.Getter;
604     import lombok.NoArgsConstructor;
605     import lombok.Setter;
606
607     @NoArgsConstructor
608     @Setter
609     @Component
610     public class Hello {
611         @Value("${myname}")
612         private String name;
613
614         @Resource(name="${printer1}")
615         private Printer printer;
616
617         @Value("${value1},${value2},${value3}")
618         @Getter private List<String> names;
619
620         public String sayHello(){
621             return "Hello " + name;
622         }
623
624         public void print(){
625             this.printer.print(sayHello());
626         }
627     }
628
629
630 19. StringPrinter.java 수정
631     package com.example;
632
```

```
633 import org.springframework.stereotype.Component;
634
635 @Component("stringPrinter")
636 public class StringPrinter implements Printer {
637     private StringBuffer buffer = new StringBuffer();
638
639     @Override
640     public void print(String message) {
641         this.buffer.append(message);
642     }
643
644     @Override
645     public String toString(){
646         return this.buffer.toString();
647     }
648 }
649
650
651 20. beans.xml 수정하기
652
653 <?xml version="1.0" encoding="UTF-8"?>
654 <beans xmlns="http://www.springframework.org/schema/beans"
655     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
656     xmlns:context="http://www.springframework.org/schema/context"
657     xsi:schemaLocation="http://www.springframework.org/schema/beans
658         http://www.springframework.org/schema/beans/spring-beans.xsd
659         http://www.springframework.org/schema/context
660         http://www.springframework.org/schema/context/spring-context-3.2.xsd">
661
662     <context:property-placeholder location="classpath:value.properties"/>
663     <context:component-scan base-package="com.example" /> <!--추가하기
664     <!-- 나머지 코드 삭제 -->
665 </beans>
666
667
668 21. MainClass.java 수정하기
669
670 package com.example;
671
672 import org.springframework.context.ApplicationContext;
673 import org.springframework.context.support.GenericXmlApplicationContext;
674
675 public class MainClass {
676     public static void main(String[] args) {
677         ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
678
679         Hello hello = (Hello) ctx.getBean("hello");
680         System.out.println(hello.sayHello());
681         hello.print();
682
683         Printer printer = ctx.getBean("stringPrinter", StringPrinter.class);
684         System.out.println(printer.toString());
685
686         hello.getNames().forEach(value -> System.out.println(value));
687     }
688 }
```

```
689 22. 실행
690 1)MainClass > right-click > Run As > Java Application
691
692 2)결과
693 Hello Spring
694 Hello Spring
695 HTML5,CSS3,JavaScript
696
697
698
699 -----
700 Task 3. Lab
701 1. In Package Explorer > right-click > New > Java Project
702 1)Project Name : PropertyDemo1
703 2)JRE : Use default JRE 'jdk-13.0.2' and workspace compiler preferences
704 3)Next
705 4)Uncheck [Create module-info.java file]
706 5)Finish
707
708
709 2. /src/ right-click > New > Package
710 1)Package name : com.example
711 2)Finish
712
713
714 3. /src/com.example.AdminConnection.java 생성
715 1)com.example > right-click > New > Class
716 2)Name : AdminConnection
717 3)Finish
718
719 package com.example;
720
721 public class AdminConnection {
722     private String adminId;
723     private String adminPwd;
724     private String subAdminId;
725     private String subAdminPwd;
726 }
727
728
729 4. Java Project를 Spring Project로 변환
730 1)PropertyDemo1 Project > right-click > Configure > Convert to Maven Project
731 -Project : /PropertyDemo1
732 -Group Id : PropertyDemo1
733 -Artifact Id : PropertyDemo1
734 -version : 0.0.1-SNAPSHOT
735 -Packaging : jar
736 -Finish
737
738 2)PropertyDemo1 Project > right-click > Spring > Add Spring Project Nature
739
740 3)pom.xml file에 Spring Context Dependency 추가하기
741 <version>0.0.1-SNAPSHOT</version>
742 <dependencies>
743 <dependency>
744 <groupId>org.springframework</groupId>
745 <artifactId>spring-context</artifactId>
746 <version>5.2.5.RELEASE</version>
```

```
747     </dependency>
748 </dependencies>
749
750 4)pom.xml > right-click > Run As > Maven install
751 [INFO] BUILD SUCCESS 확인
752
753
754 5. Lombok library 추가
755 1)https://mvnrepository.com/에서 'lombok'으로 검색
756 2)'Project Lombok' click
757 3)1.18.12 click
758 4)dependency copy해서 pom.xml에 붙여넣기
759
760 <dependencies>
761 <dependency>
762 <groupId>org.springframework</groupId>
763 <artifactId>spring-context</artifactId>
764 <version>5.2.5.RELEASE</version>
765 </dependency>
766 <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
767 <dependency>
768 <groupId>org.projectlombok</groupId>
769 <artifactId>lombok</artifactId>
770 <version>1.18.12</version>
771 <scope>provided</scope>
772 </dependency>
773 </dependencies>
774
775 5)pom.xml > right-click > Run As > Maven install
776 [INFO] BUILD SUCCESS 확인
777
778
779 6. AdminConnection.java 수정
780
781 package com.example;
782
783 import lombok.Getter;
784 import lombok.Setter;
785
786 @Setter
787 @Getter
788 public class AdminConnection {
789     private String adminId;
790     private String adminPwd;
791     private String subAdminId;
792     private String subAdminPwd;
793 }
794
795
796 7. PropertyDemo/resources folder 생성
797 1)PropertyDemo1 project > right-click > New > Source Folder
798 2)Folder name : resources
799 3)Finish
800
801
802 8./resources 두 개의 properties file 생성
803
804 <admin.properties>
```

```
805     admin.id=javaexpert
806     admin.pwd=12345678
807
808     <sub.admin.properties>
809         sub.admin.id=jasasoft
810         sub.admin.pwd=987654321
811
812
813 9. Bean Configuration XML 작성
814 1)PropertyDemo1/resources > right-click > New > Spring Bean Configuration File
815 2)File name : beans.xml
816 3)Finish
817 4)Namespace tab에서 context - http://www.springframework.org/schema/context check
818
819     <?xml version="1.0" encoding="UTF-8"?>
820     <beans xmlns="http://www.springframework.org/schema/beans"
821         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
822         xmlns:context="http://www.springframework.org/schema/context"
823         xsi:schemaLocation="http://www.springframework.org/schema/beans
824             http://www.springframework.org/schema/beans/spring-beans.xsd
825             http://www.springframework.org/schema/context
826             http://www.springframework.org/schema/context/spring-context-3.2.xsd">
827
828         <context:property-placeholder location="classpath:admin.properties,
829             classpath:sub.admin.properties" />
830
831         <bean id="adminConnection" class="com.example.AdminConnection">
832             <property name="adminId">
833                 <value>${admin.id}</value>
834             </property>
835             <property name="adminPwd">
836                 <value>${admin.pwd}</value>
837             </property>
838             <property name="subAdminId">
839                 <value>${sub.admin.id}</value>
840             </property>
841             <property name="subAdminPwd">
842                 <value>${sub.admin.pwd}</value>
843             </property>
844         </bean>
845     </beans>
846
847 10. /src/com.example.MainClass.java 생성
848 1)com.example > right-click > New > Class
849 2)Name : MainClass
850 3)Finish
851
852 package com.example;
853
854 import org.springframework.context.support.AbstractApplicationContext;
855 import org.springframework.context.support.GenericXmlApplicationContext;
856
857 public class MainClass {
858     public static void main(String[] args) {
859         AbstractApplicationContext ctx = new
860             GenericXmlApplicationContext("classpath:beans.xml");
861         AdminConnection connection = ctx.getBean("adminConnection",
```

```
AdminConnection.class);
859 System.out.println("admin ID : " + connection.getAdminId());
860 System.out.println("admin PWD : " + connection.getAdminPwd());
861 System.out.println("sub admin ID : " + connection.getSubAdminId());
862 System.out.println("sub admin PWD : " + connection.getSubAdminPwd());
863
864 ctx.close();
865 }
866 }
```

869 11. 결과

870 1)MainClass.java > right-click > Run As > Java Application

```
871
872 admin ID : javaexpert
873 admin PWD : 12345678
874 sub admin ID : javasoft
875 sub admin PWD : 987654321
876
```

877
878

879 -----
880 Task 4. Lab

881 1. In Package Explorer > right-click > New > Java Project

```
882 1)Project Name : PropertyDemo2
883 2)JRE : Use default JRE 'jdk-13.0.2' and workspace compiler preferences
884 3)Next
885 4)Uncheck [Create module-info.java file]
886 5)Finish
```

887
888

889 2. /src/ right-click > New > Package

```
890 1)Package name : com.example
891 2)Finish
```

892
893

894 3. /src/com.example.AdminConnection.java 생성

```
895 1)com.example > right-click > New > Class
896 2)Name : AdminConnection
897 3)Finish
```

898
899

```
package com.example;
```

900
901

```
public class AdminConnection {
902     private String adminId;
903     private String adminPwd;
904     private String subAdminId;
905     private String subAdminPwd;
906 }
```

907
908

909 4. Java Project를 Spring Project로 변환

```
910 1)PropertyDemo1 Project > right-click > Configure > Convert to Maven Project
911 -Project : /PropertyDemo2
912 -Group Id : PropertyDemo2
913 -Artifact Id : PropertyDemo2
914 -version : 0.0.1-SNAPSHOT
915 -Packaging : jar
```



```
916     -Finish
917
918 2)PropertyDemo2 Project > right-click > Spring > Add Spring Project Nature
919
920 3)pom.xml 파일에 Spring Context Dependency 추가하기
921     <version>0.0.1-SNAPSHOT</version>
922     <dependencies>
923         <dependency>
924             <groupId>org.springframework</groupId>
925             <artifactId>spring-context</artifactId>
926             <version>5.2.5.RELEASE</version>
927         </dependency>
928     </dependencies>
929
930 4)pom.xml > right-click > Run As > Maven install
931     [INFO] BUILD SUCCESS 확인
932
933 5. Lombok library 추가
934 1)<a href="https://mvnrepository.com/">https://mvnrepository.com/에서 'lombok'으로 검색
935 2)'Project Lombok' click
936 3)1.18.12 click
937 4)dependency copy해서 pom.xml에 붙여넣기
938
939     <dependencies>
940         <dependency>
941             <groupId>org.springframework</groupId>
942             <artifactId>spring-context</artifactId>
943             <version>5.2.5.RELEASE</version>
944         </dependency>
945         <!-- <a href="https://mvnrepository.com/artifact/org.projectlombok/lombok">https://mvnrepository.com/artifact/org.projectlombok/lombok -->
946         <dependency>
947             <groupId>org.projectlombok</groupId>
948             <artifactId>lombok</artifactId>
949             <version>1.18.12</version>
950             <scope>provided</scope>
951         </dependency>
952     </dependencies>
953
954 5)pom.xml > right-click > Run As > Maven install
955     [INFO] BUILD SUCCESS 확인
956
957 6. AdminConnection.java 수정
958
959     package com.example;
960
961     import lombok.Getter;
962     import lombok.Setter;
963
964     @Setter
965     @Getter
966     public class AdminConnection {
967         private String adminId;
968         private String adminPwd;
969         private String subAdminId;
970         private String subAdminPwd;
971     }
```

```
974
975
976 7. PropertyDemo/resources folder 생성
977     1)PropertyDemo1 project > right-click > New > Source Folder
978     2)Folder name : resources
979     3)Finish
980
981
982 8. ./resources 두 개의 properties file 생성
983
984     <admin.properties>
985         admin.id=javaexpert
986         admin.pwd=12345678
987
988     <sub.admin.properties>
989         sub.admin.id=javasoft
990         sub.admin.pwd=987654321
991
992
993 9. ApplicationConfig Class 생성
994     1)com.example > right-click > New > Class
995     2)Name : ApplicationConfig
996     3)Finish
997
998     package com.example;
999
1000     import org.springframework.context.annotation.Bean;
1001     import org.springframework.context.annotation.ComponentScan;
1002     import org.springframework.context.annotation.Configuration;
1003     import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;
1004     import org.springframework.core.io.ClassPathResource;
1005
1006     @Configuration
1007     @ComponentScan(basePackages = "com.example")
1008     public class ApplicationConfig {
1009         @Bean
1010         public static PropertySourcesPlaceholderConfigurer properties() {
1011             PropertySourcesPlaceholderConfigurer configurator = new
1012             PropertySourcesPlaceholderConfigurer();
1013             configurator.setLocations(new ClassPathResource("admin.properties"), new
1014             ClassPathResource("sub.admin.properties"));
1015             return configurator;
1016         }
1017     }
1018
1019
1020 10. com.example.AdminConnection.java 수정
1021
1022     package com.example;
1023
1024     import org.springframework.beans.factory.annotation.Value;
1025     import org.springframework.stereotype.Component;
1026
1027     import lombok.Getter;
1028
1029     @Getter
1030     @Component
1031     public class AdminConnection {
```

```
1030     @Value("${admin.id}")
1031     private String adminId;
1032     @Value("${admin.pwd}")
1033     private String adminPwd;
1034     @Value("${sub.admin.id}")
1035     private String subAdminId;
1036     @Value("${sub.admin.pwd}")
1037     private String subAdminPwd;
1038 }
1039
1040
1041 11. /src/com.example.MainClass.java
1042 1)com.example > right-click > New > Class
1043 2)Name : MainClass1
1044 3)Finish
1045
1046     package com.example;
1047
1048     import org.springframework.context.ApplicationContext;
1049     import org.springframework.context.annotation.AnnotationConfigApplicationContext;
1050
1051     public class MainClass1 {
1052         public static void main(String[] args) {
1053             ApplicationContext context = new
1054                 AnnotationConfigApplicationContext(ApplicationConfig.class);
1055             AdminConnection connection = (AdminConnection)
1056                 context.getBean("adminConnection");
1057             System.out.println("admin ID : " + connection.getAdminId());
1058             System.out.println("admin PWD : " + connection.getAdminPwd());
1059             System.out.println("sub admin ID : " + connection.getSubAdminId());
1060             System.out.println("sub admin PWD : " + connection.getSubAdminPwd());
1061         }
1062     }
1063
1064 12. 결과
1065 1)MainClass.java > right-click > Run As > Java Application
1066     admin ID : javaexpert
1067     admin PWD : 12345678
1068     sub admin ID : javasoft
1069     sub admin PWD : 987654321
1070
1071 -----
1072 Task 5. Lab
1073 1. In Package Explorer > right-click > New > Java Project
1074 1)Project Name : ProfileDemo
1075 2)JRE
1076     -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
1077 3)Next
1078 4)Uncheck [Create module-info.java file]
1079 5)Finish
1080
1081
1082 2. Package 생성
1083 1)/src/ > right-click > New > Package
1084 2)Package name : com.example
```

```
1086 3)Finish
1087
1088
1089 3. Java Project를 Spring Project로 변환
1090 1)ProfileDemo Project > right-click > Configure > Convert to Maven Project
1091 -Project : /ProfileDemo
1092 -Group Id : ProfileDemo
1093 -Artifact Id : ProfileDemo
1094 -version : 0.0.1-SNAPSHOT
1095 -Packaging : jar
1096 -Finish
1097
1098 2)ProfileDemo Project > right-click > Spring > Add Spring Project Nature
1099
1100 3)pom.xml file에 Spring Context Dependency 추가하기
1101 <version>0.0.1-SNAPSHOT</version>
1102 <dependencies>
1103 <dependency>
1104 <groupId>org.springframework</groupId>
1105 <artifactId>spring-context</artifactId>
1106 <version>5.2.5.RELEASE</version>
1107 </dependency>
1108 </dependencies>
1109
1110 4)pom.xml > right-click > Run As > Maven install
1111 [INFO] BUILD SUCCESS 확인
1112
1113
1114 4. Lombok library 추가
1115 1)https://mvnrepository.com/에서 'lombok'으로 검색
1116 2)'Project Lombok' click
1117 3)1.18.12 click
1118 4)dependency copy해서 pom.xml에 붙여넣기
1119
1120 <dependencies>
1121 <dependency>
1122 <groupId>org.springframework</groupId>
1123 <artifactId>spring-context</artifactId>
1124 <version>5.2.5.RELEASE</version>
1125 </dependency>
1126 <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
1127 <dependency>
1128 <groupId>org.projectlombok</groupId>
1129 <artifactId>lombok</artifactId>
1130 <version>1.18.12</version>
1131 <scope>provided</scope>
1132 </dependency>
1133 </dependencies>
1134
1135 5)pom.xml > right-click > Run As > Maven install
1136 [INFO] BUILD SUCCESS 확인
1137
1138
1139 5. ProfileDemo/resources folder 생성
1140 1)ProfileDemo project > New > Source Folder
1141 2)Folder name : resources
1142 3)Finish
1143
```

```
1144
1145 6. ServerInfo.java 생성
1146 1)com.example > right-click > New > Class
1147 2)Name : ServerInfo
1148 3)Finish
1149
1150     package com.example;
1151
1152     import lombok.Getter;
1153     import lombok.Setter;
1154
1155     @Getter
1156     @Setter
1157     public class ServerInfo {
1158         private String ipNum;
1159         private String portNum;
1160     }
1161
1162
1163 7. XML 설정 file 2개 생성
1164 1)resource > right-click > New > Spring Bean Configuration File
1165 2)File name : run.xml
1166 3)Finish
1167
1168     <?xml version="1.0" encoding="UTF-8"?>
1169     <beans xmlns="http://www.springframework.org/schema/beans"
1170         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1171         xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd"
1172         profile="run"> <---이것이 핵심
1173
1174         <bean id="serverInfo" class="com.example.ServerInfo">
1175             <property name="ipNum" value="192.168.56.5" />
1176             <property name="portNum" value="80" />
1177         </bean>
1178     </beans>
1179
1180 3)/resource > right-click > New > Spring Bean Configuration File
1181 4)File name : dev.xml
1182 5)Finish
1183
1184     <?xml version="1.0" encoding="UTF-8"?>
1185     <beans xmlns="http://www.springframework.org/schema/beans"
1186         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1187         xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd"
1188         profile="dev"> <---이것이 핵심
1189
1190         <bean id="serverInfo" class="com.example.ServerInfo">
1191             <property name="ipNum" value="localhost" />
1192             <property name="portNum" value="8080" />
1193         </bean>
1194     </beans>
1195
1196
1197 8. MainClass 생성
1198 1)com.example > right-click > New > Class
1199 2)Name : MainClass.java
```

```
1200 3)Finish
1201
1202 package com.example;
1203
1204 import java.util.Scanner;
1205
1206 import org.springframework.context.support.GenericXmlApplicationContext;
1207
1208 public class MainClass {
1209     public static void main(String[] args) {
1210         Scanner scan = new Scanner(System.in);
1211         System.out.print("Select dev or run : ");
1212         String config = scan.next(); //"dev" or "run"
1213
1214         GenericXmlApplicationContext ctx = new GenericXmlApplicationContext();
1215         ctx.getEnvironment().setActiveProfiles(config);
1216         ctx.load("dev.xml", "run.xml");
1217         ctx.refresh();
1218
1219         ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
1220         System.out.println("IP : " + info.getIpNum());
1221         System.out.println("Port : " + info.getPortNum());
1222         ctx.close();
1223     }
1224 }
1225
1226
1227 9. 결과
1228 1)MainClass.java > right-click > Run As > Java Application
1229     Select dev or run :
1230 2)입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이 나
1231     온다.
1232
1233
1234 -----
1235 Task 6. Lab
1236 1. In Package Explorer > right-click > New > Java Project
1237     1)Project Name : ProfileDemo1
1238     2)JRE
1239         -Select [Use default JRE 'jdk-13.0.2' and workspace compiler preferences]
1240     3)Next
1241     4)Uncheck [Create module-info.java file]
1242     5)Finish
1243
1244
1245 2. Package 생성
1246     1)/src/ > right-click > New > Package
1247     2)Package name : com.example
1248     3)Finish
1249
1250
1251 3. Java Project를 Spring Project로 변환
1252     1)ProfileDemo1 Project > right-click > Configure > Convert to Maven Project
1253         -Project : /ProfileDemo1
1254         -Group Id : ProfileDemo1
1255         -Artifact Id : ProfileDemo1
1256         -version : 0.0.1-SNAPSHOT
```

```
1257 -Packaging : jar
1258 -Finish
1259
1260 2)ProfileDemo1 Project > right-click > Spring > Add Spring Project Nature
1261
1262 3)pom.xml file에 Spring Context Dependency 추가하기
1263 <version>0.0.1-SNAPSHOT</version>
1264 <dependencies>
1265 <dependency>
1266 <groupId>org.springframework</groupId>
1267 <artifactId>spring-context</artifactId>
1268 <version>5.2.5.RELEASE</version>
1269 </dependency>
1270 </dependencies>
1271
1272 4)pom.xml > right-click > Run As > Maven install
1273 [INFO] BUILD SUCCESS 확인
1274
1275
1276 4. Lombok library 추가
1277 1)<a href="https://mvnrepository.com/">https://mvnrepository.com/에서 'lombok'으로 검색
1278 2)'Project Lombok' click
1279 3)1.18.12 click
1280 4)dependency copy해서 pom.xml에 붙여넣기
1281
1282 <dependencies>
1283 <dependency>
1284 <groupId>org.springframework</groupId>
1285 <artifactId>spring-context</artifactId>
1286 <version>5.2.5.RELEASE</version>
1287 </dependency>
1288 <!-- <a href="https://mvnrepository.com/artifact/org.projectlombok/lombok">https://mvnrepository.com/artifact/org.projectlombok/lombok -->
1289 <dependency>
1290 <groupId>org.projectlombok</groupId>
1291 <artifactId>lombok</artifactId>
1292 <version>1.18.12</version>
1293 <scope>provided</scope>
1294 </dependency>
1295 </dependencies>
1296
1297 5)pom.xml > right-click > Run As > Maven install
1298 [INFO] BUILD SUCCESS 확인
1299
1300
1301 5. ServerInfo.java 생성
1302 1)com.example > right-click > New > Class
1303 2)Name : ServerInfo
1304 3)Finish
1305
1306 package com.example;
1307
1308 import lombok.Getter;
1309 import lombok.Setter;
1310
1311 @Getter
1312 @Setter
1313 public class ServerInfo {
1314     private String ipNum;
```

```
1315     private String portNum;
1316 }
1317
1318
1319 6. Java 설정 file 2개 생성
1320 1)com.example > right-click > New > Class
1321 2)Name : ApplicationConfigDev
1322 3)Finish
1323
1324     package com.example;
1325
1326     import org.springframework.context.annotation.Bean;
1327     import org.springframework.context.annotation.Configuration;
1328     import org.springframework.context.annotation.Profile;
1329
1330     @Configuration
1331     @Profile("dev")
1332     public class ApplicationConfigDev {
1333
1334         @Bean
1335         public ServerInfo serverInfo(){
1336             ServerInfo info = new ServerInfo();
1337             info.setIpNum("localhost");
1338             info.setPortNum("8080");
1339             return info;
1340         }
1341     }
1342
1343 4)com.example > right-click > New > Class
1344 5)Name : ApplicationConfigRun.java
1345 6)Finish
1346
1347     package com.example;
1348
1349     import org.springframework.context.annotation.Bean;
1350     import org.springframework.context.annotation.Configuration;
1351     import org.springframework.context.annotation.Profile;
1352
1353     @Configuration
1354     @Profile("run")
1355     public class ApplicationConfigRun {
1356
1357         @Bean
1358         public ServerInfo serverInfo(){
1359             ServerInfo info = new ServerInfo();
1360             info.setIpNum("192.168.56.5");
1361             info.setPortNum("80");
1362             return info;
1363         }
1364     }
1365
1366
1367 7. MainClass 생성
1368 1)com.example > right-click > New > Class
1369 2)Name : MainClass.java
1370
1371     package com.example;
1372     import java.util.Scanner;
```



```
1373
1374 import org.springframework.context.annotation.AnnotationConfigApplicationContext;
1375
1376 public class MainClass {
1377     public static void main(String[] args) {
1378         Scanner scan = new Scanner(System.in);
1379         System.out.print("Select dev or run : ");
1380         String config = scan.next(); //"dev" or "run"
1381
1382         AnnotationConfigApplicationContext ctx = new AnnotationConfigApplicationContext();
1383         ctx.getEnvironment().setActiveProfiles(config);
1384         ctx.register(ApplicationConfigDev.class, ApplicationConfigRun.class);
1385         ctx.refresh();
1386
1387         ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
1388         System.out.println("IP : " + info.getIpNum());
1389         System.out.println("Port : " + info.getPortNum());
1390         ctx.close();
1391     }
1392 }
```

1393

1394

1395 8. 결과

1396 1)MainClass.java > right-click > Run As > Java Application

1397 Select dev or run :

1398 2)입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이 나온다.