

```
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-11.0.12' and workspace compiler preferences]
8      3)Uncheck [Create module-info.java file]
9      4)Next
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>5.3.10</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)1.18.20 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>5.3.10</version>
54              </dependency>
55              <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
56              <dependency>
```

```
57         <groupId>org.projectlombok</groupId>
58         <artifactId>lombok</artifactId>
59         <version>1.18.20</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, DisposableBean {

98 @Setter

99 private Environment env;

100 @Getter

101 @Setter

102 private String adminId;

103 @Getter

104 @Setter

105 private String adminPwd;

106
107 @Override

108 public void destroy() throws Exception {

109 System.out.println("destroy()");

110 }
111

```

112     @Override
113     public void afterPropertiesSet() throws Exception {
114         System.out.println("afterPropertiesSet()");
115         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 }

```

8. resources > right-click > New > Spring Bean Configuration File

1)File name : beans.xml

2)Finish

```

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>

```

9. com.example.MainClass Class 생성

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

2)Name : MainClass

3)Finish

```

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
161                 GenericXmlApplicationContext();
162             ConfigurableEnvironment env = ctx.getEnvironment();
163
164             MutablePropertySources propertySouces = env.getPropertySources();
165             //내가 원하는 정보를 얻을 때까지 모든 propertySources를 앞에서 부터 차례로 모두
166             검색함.

```

```

165         try{
166             propertySouces.addLast(new
                ResourcePropertySource("classpath:admin.properties")); //property
                추가
167
168             System.out.println(env.getProperty("admin.id"));    //property 추출
169             System.out.println(env.getProperty("admin.pwd"));
170         }catch(IOException ex){}
171
172         GenericXmlApplicationContext gCtx = (GenericXmlApplicationContext)ctx;
173         gCtx.load("classpath:beans.xml");
174         gCtx.refresh();
175
176         AdminConnection adminConnection =
            gCtx.getBean("adminConnection", AdminConnection.class);
177         System.out.println("admin ID : " + adminConnection.getAdminId());
178         System.out.println("admin PWD : " + adminConnection.getAdminPwd());
179
180         gCtx.close();
181         ctx.close();
182     }
183 }
184
185

```

10. 실행

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

```

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

```

Task2. Lab

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

1)Project Name : PropertyDemo

2)JRE

-Select [Use default JRE 'jdk-11.0.12' and workspace compiler preferences]

3)Uncheck [Create module-info.java file]

4)Next

5)Finish

208

209

2. src > right-click > New > Package

1)Name : com.example

2)Finish

213

214

3. POJO Class 작성

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

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

```

4. Java Project를 Spring Project로 변환

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

```

2)PropertyDemo Project > right-click > Spring > Add Spring Project Nature

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>5.3.10</version>
302         </dependency>
303     </dependencies>

```

4pom.xml > right-click > Run As > Maven install

[INFO] BUILD SUCCESS 확인

5. Lombok library 추가

1)<https://mvnrepository.com/>에서 'lombok'으로 검색

2)'Project Lombok' click

3)1.18.20 click

4)dependency copy해서 pom.xml에 붙여넣기

```

315     <dependencies>
316         <dependency>
317             <groupId>org.springframework</groupId>
318             <artifactId>spring-context</artifactId>
319             <version>5.3.10</version>
320         </dependency>
321         <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
322         <dependency>
323             <groupId>org.projectlombok</groupId>
324             <artifactId>lombok</artifactId>
325             <version>1.18.20</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

```

9. com.example.MainClass Class 생성

- 1)com.example > right-click > New > Class
- 2)Name : MainClass
- 3)Finish

```

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
409             GenericXmlApplicationContext("classpath:beans.xml");
410
411         Hello hello = (Hello)ctx.getBean("hello");
412         System.out.println(hello.sayHello());
413         hello.print();
414
415         Printer printer = ctx.getBean("printer", StringPrinter.class);
416         System.out.println(printer.toString());
417
418         hello.getNames().forEach(value -> System.out.println(value));
419     }
420 }

```

10. 실행

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

```

425     Hello Spring
426     Hello Spring
427     AOP
428     Spring
429     DI

```

11. JUnit 5 & Spring TestContext Framework으로 Test

- 1)src > right-click > New > Package
- 2)Name : com.example.test
- 3)Finish
- 4)com.example.test > right-click > New > JUnit Test Case > Select [New JUnit Jupiter test]
- 5)Name : HelloJUnitTest
- 6)Finish

439 7)New JUnit Test Case창에서, [Perform the following action: Add JUnit 5 library to
the build path]
440 8)OK
441 9)<https://mvnrepository.com>에서 'spring-test'로 검색
442 10)Spring TestContext Framework에서
443 11)5.3.10로 들어가서
444 12)복사해서 pom.xml로 붙여넣기
445 <!-- <https://mvnrepository.com/artifact/org.springframework/spring-test> -->
446 <dependency>
447 <groupId>org.springframework</groupId>
448 <artifactId>spring-test</artifactId>
449 <version>5.3.10</version>
450 <scope>test</scope>
451 </dependency>
452
453 13)pom.xml > right-click > Run As > Maven install
454 [INFO] BUILD SUCCESS 확인
455 -만일 Error 발생하면 PropertyDemo > right-click > Maven > Update Project > OK
456 -pom.xml > right-click > Run As > Maven install
457 -[INFO] BUILD SUCCESS 확인
458
459

460 12. com.example.test.HelloJUnitTest

```
461 package com.example.test;  
462  
463 import static org.junit.jupiter.api.Assertions.assertEquals;  
464 import static org.junit.jupiter.api.Assertions.assertSame;  
465  
466 import org.junit.jupiter.api.Test;  
467 import org.junit.jupiter.api.extension.ExtendWith;  
468 import org.springframework.beans.factory.annotation.Autowired;  
469 import org.springframework.context.ApplicationContext;  
470 import org.springframework.test.context.ContextConfiguration;  
471 import org.springframework.test.context.junit.jupiter.SpringExtension;  
472  
473 import com.example.Hello;  
474 import com.example.Printer;  
475  
476 @ExtendWith(SpringExtension.class)  
477 @ContextConfiguration(locations = "classpath:beans.xml")  
478 class HelloJUnitTest {  
479     @Autowired  
480     ApplicationContext ctx;  
481  
482     @Test  
483     public void test() {  
484         Hello hello = (Hello) ctx.getBean("hello");  
485         assertEquals("Hello Spring", hello.sayHello());  
486         hello.print();  
487  
488         Printer printer = (Printer) ctx.getBean("printer");  
489         assertEquals("Hello Spring", printer.toString());  
490     }  
491  
492     @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

```

1)만일 해당 객체를 찾을 수 없다는 오류가 계속 발생하면

-해당 Project > right-click > Build Path > Libraries tab

-spring-test-5.3.10.jar 선택 후 [Remove] 로 삭제

-[Add External JARs...] Click

-%M2_HOME%\repository\org\springframework\spring-test\5.3.10 선택할 것

-[Order and Export] tab에서 spring-test-5.3.10.jar 선택 후 [Up] button을 클릭

-해당 Project/src 바로 아래까지 올리고 [Apply and Close] Click

2)right-click > Run As > Junit Test

3)결과 -> Junit View에 초록색 bar

13. resources/value.properties 생성

myname=Spring

myprinter=printer

value1=HTML5

value2=CSS3

value3=JavaScript

14. /resources/beans.xml 수정하기

1)beans.xml에서 [Namespaces] tab

2)목록에서 'context-<http://www.springframework.org/schema/context>' check

3)<context:property-placeholder />를 사용하기 위해서 다음과 같이 수정

```
<context:property-placeholder
```

```
    location="classpath:value.properties" />
```

```
<bean id="hello" class="com.example.Hello">
```

```
    <property name="name" value="${myname}" />
```

```
    <property name="printer" ref="${myprinter}" />
```

```
    <property name="names">
```

```
        <list>
```

```
            <value>${value1}</value>
```

```
            <value>${value2}</value>
```

```
            <value>${value3}</value>
```

```
        </list>
```

```
    </property>
```

```
</bean>
```

```
<bean id="printer" class="com.example.StringPrinter" />
```

```
<bean id="consolePrinter" class="com.example.ConsolePrinter" />
```

```
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         <!--
581         https://mvnrepository.com/artifact/javax.annotation/javax.annotation-api -->
582         <dependency>
583             <groupId>javax.annotation</groupId>
584             <artifactId>javax.annotation-api</artifactId>
585             <version>1.3.2</version>
586         </dependency>
587
588     5)pom.xml > right-click > Run As > Maven install
589     [INFO] BUILD SUCCESS 확인
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

```

19. StringPrinter.java 수정

```

630 package com.example;
631
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

```

20. beans.xml 수정하기

```

651
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">

```

```
659
660     <context:property-placeholder location="classpath:value.properties" />
661     <context:component-scan base-package="com.example" /> <!--추가하기
662     <!-- 나머지 코드 삭제 -->
663 </beans>
664
665
```

21. MainClass.java 수정하기

```
666
667
668 package com.example;
669
670 import org.springframework.context.ApplicationContext;
671 import org.springframework.context.support.GenericXmlApplicationContext;
672
673 public class MainClass {
674     public static void main(String[] args) {
675         ApplicationContext ctx = new
676             GenericXmlApplicationContext("classpath:beans.xml");
677
678         Hello hello = (Hello) ctx.getBean("hello");
679         System.out.println(hello.sayHello());
680         hello.print();
681
682         Printer printer = ctx.getBean("stringPrinter", StringPrinter.class);
683         System.out.println(printer.toString());
684
685         hello.getNames().forEach(value -> System.out.println(value));
686     }
687 }
688
```

22. 실행

689 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-11.0.12' and workspace compiler preferences

704 3)Uncheck [Create module-info.java file]

705 4)Next

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>5.3.10</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)1.18.20 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>5.3.10</version>

765 </dependency>

766 <!-- <https://mvnrepository.com/artifact/org.projectlombok/lombok> -->

767 <dependency>

768 <groupId>org.projectlombok</groupId>

769 <artifactId>lombok</artifactId>

```
770         <version>1.18.20</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=javasoft
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
            http://www.springframework.org/schema/beans/spring-beans.xsd
```

```

824      http://www.springframework.org/schema/context
      http://www.springframework.org/schema/context/spring-context-3.2.xsd
      ">
825
826      <context:property-placeholder location="classpath:admin.properties,
      classpath:sub.admin.properties" />
827
828      <bean id="adminConnection" class="com.example.AdminConnection">
829          <property name="adminId">
830              <value>${admin.id}</value>
831          </property>
832          <property name="adminPwd">
833              <value>${admin.pwd}</value>
834          </property>
835          <property name="subAdminId">
836              <value>${sub.admin.id}</value>
837          </property>
838          <property name="subAdminPwd">
839              <value>${sub.admin.pwd}</value>
840          </property>
841      </bean>
842  </beans>
843
844
845  10. /src/com.example.MainClass.java 생성
846      1)com.example > right-click > New > Class
847      2)Name : MainClass
848      3)Finish
849
850      package com.example;
851
852      import org.springframework.context.support.AbstractApplicationContext;
853      import org.springframework.context.support.GenericXmlApplicationContext;
854
855      public class MainClass {
856          public static void main(String[] args) {
857              AbstractApplicationContext ctx = new
              GenericXmlApplicationContext("classpath:beans.xml");
858              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      }
867
868
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-11.0.12' and workspace compiler preferences

884 3)Uncheck [Create module-info.java file]

885 4)Next

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>5.3.10</version>

927 </dependency>

928 </dependencies>

929

930 4pom.xml > right-click > Run As > Maven install

931 [INFO] BUILD SUCCESS 확인

932

933

934 5. Lombok library 추가

935 1) <https://mvnrepository.com/>에서 'lombok'으로 검색

936 2)'Project Lombok' click

937 3)1.18.20 click

938 4)dependency copy해서 pom.xml에 붙여넣기

939

940 <dependencies>

941 <dependency>

942 <groupId>org.springframework</groupId>

943 <artifactId>spring-context</artifactId>

944 <version>5.3.10</version>

945 </dependency>

946 <!-- <https://mvnrepository.com/artifact/org.projectlombok/lombok> -->

947 <dependency>

948 <groupId>org.projectlombok</groupId>

949 <artifactId>lombok</artifactId>

950 <version>1.18.20</version>

951 <scope>provided</scope>

952 </dependency>

953 </dependencies>

954

955 5)pom.xml > right-click > Run As > Maven install

956 [INFO] BUILD SUCCESS 확인

957

958

959 6. AdminConnection.java 수정

960

961 package com.example;

962

963 import lombok.Getter;

964 import lombok.Setter;

965

966 @Setter

967 @Getter

968 public class AdminConnection {

969 private String adminId;

970 private String adminPwd;

971 private String subAdminId;

972 private String subAdminPwd;

973 }

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. AppConfig Class 생성
994     1)com.example > right-click > New > Class
995     2)Name : AppConfig
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
1004     org.springframework.context.support.PropertySourcesPlaceholderConfigurer;
1005     import org.springframework.core.io.ClassPathResource;
1006
1007     @Configuration
1008     @ComponentScan(basePackages = "com.example")
1009     public class AppConfig {
1010         @Bean
1011         public static PropertySourcesPlaceholderConfigurer properties() {
1012             PropertySourcesPlaceholderConfigurer configurator = new
1013             PropertySourcesPlaceholderConfigurer();
1014             configurator.setLocations(new ClassPathResource("admin.properties"),
1015             new ClassPathResource("sub.admin.properties"));
1016             return configurator;
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 {
1032         @Value("${admin.id}")
1033         private String adminId;
1034         @Value("${admin.pwd}")
1035         private String adminPwd;
1036         @Value("${sub.admin.id}")
1037         private String subAdminId;
1038         @Value("${sub.admin.pwd}")
1039         private String subAdminPwd;
1040     }

```

```
1040
1041 11. /src/com.example.MainClass.java
1042     1)com.example > right-click > New > Class
1043     2)Name : MainClass
1044     3)Finish
1045
1046     package com.example;
1047
1048     import org.springframework.context.ApplicationContext;
1049     import
1050     org.springframework.context.annotation.AnnotationConfigApplicationContext;
1051
1052     public class MainClass {
1053         public static void main(String[] args) {
1054             ApplicationContext context = new
1055             AnnotationConfigApplicationContext(ApplicationConfig.class);
1056             AdminConnection connection = (AdminConnection)
1057             context.getBean("adminConnection");
1058             System.out.println("admin ID : " + connection.getAdminId());
1059             System.out.println("admin PWD : " + connection.getAdminPwd());
1060             System.out.println("sub admin ID : " + connection.getSubAdminId());
1061             System.out.println("sub admin PWD : " +
1062             connection.getSubAdminPwd());
1063         }
1064     }
```

```
1061
1062
1063 12. 결과
1064     1)MainClass.java > right-click > Run As > Java Application
1065         admin ID : javaexpert
1066         admin PWD : 12345678
1067         sub admin ID : javasoft
1068         sub admin PWD : 987654321
1069
1070
1071
```

```
1072 -----
```

1073 Task 5. Lab

```
1074 1. In Package Explorer > right-click > New > Java Project
1075     1)Project Name : ProfileDemo
1076     2)JRE
1077         -Select [Use default JRE 'jdk-11.0.12' and workspace compiler preferences]
1078     3)Uncheck [Create module-info.java file]
1079     4)Next
1080     5)Finish
1081
1082
1083 2. Package 생성
1084     1)/src/ > right-click > New > Package
1085     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>5.3.10</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)1.18.20 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>5.3.10</version>  
1125 </dependency>  
1126 <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->  
1127 <dependency>  
1128 <groupId>org.projectlombok</groupId>  
1129 <artifactId>lombok</artifactId>  
1130 <version>1.18.20</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
1172         http://www.springframework.org/schema/beans/spring-beans.xsd"
1173     profile="run"> <---이것이 핵심
1174
1175     <bean id="serverInfo" class="com.example.ServerInfo">
1176         <property name="ipNum" value="192.168.56.5" />
1177         <property name="portNum" value="80" />
1178     </bean>
1179 </beans>
1180
```

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
1188         http://www.springframework.org/schema/beans/spring-beans.xsd"
1189     profile="dev"> <---이것이 핵심
1190
1191     <bean id="serverInfo" class="com.example.ServerInfo">
1192         <property name="ipNum" value="localhost" />
1193         <property name="portNum" value="8080" />
1194     </bean>
1195 </beans>
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 }

```

9. 결과

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

Select dev or run :

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

Task 6. Lab

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

1)Project Name : ProfileDemo1

2)JRE

-Select [Use default JRE 'jdk-11.0.12' and workspace compiler preferences]

3)Uncheck [Create module-info.java file]

4)Next

5)Finish

2. Package 생성

1)/src/ > right-click > New > Package

2)Package name : com.example

3)Finish

3. Java Project를 Spring Project로 변환

1)ProfileDemo1 Project > right-click > Configure > Convert to Maven Project

-Project : /ProfileDemo1

-Group Id : ProfileDemo1

-Artifact Id : ProfileDemo1

-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>5.3.10</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)https://mvnrepository.com/에서 'lombok'으로 검색
1278     2)'Project Lombok' click
1279     3)1.18.20 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>5.3.10</version>
1287         </dependency>
1288         <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
1289         <dependency>
1290             <groupId>org.projectlombok</groupId>
1291             <artifactId>lombok</artifactId>
1292             <version>1.18.20</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

```

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.

```