```
1. Environment 객체
 1
 2
      1)Environment 객체를 이용해서 스프링 빈 설정을 할 수 있다.
 3
         Context -----> Environment -----> PropertySources
 4
                  ctx.getEnvironment()
                                                          env.getPropertySource()
                                                                                    property 추가
                   및 추출
 5
                                                                            추가:
                                                                            propertySources.addLast()
 6
                                                                            추출: env.getProperty()
 7
 8
 9
    2. Lab
10
      1)New > Spring Legacy Project > Simple Projects > Simple Spring Maven
11
            Project Name: EnvironmentDemo
12
      2)/src/main/resources/admin.properties 파일 생성
13
14
15
         admin.id=javaexpert
16
         admin.pwd=12345678
17
18
      3)/src/main/java/info.javaexpert package 생성
19
20
      4)/src/main/java/info.javaexpert.AdminConnection.java 생성
21
22
         package info.javaexpert;
23
24
         import org.springframework.beans.factory.DisposableBean;
25
         import org.springframework.beans.factory.InitializingBean;
26
         import org.springframework.context.EnvironmentAware;
27
         import org.springframework.core.env.Environment;
28
29
         public class AdminConnection implements EnvironmentAware, InitializingBean, DisposableBean{
30
            private Environment env;
31
            private String adminId;
            private String adminPwd;
32
33
34
            public void setEnv(Environment env) {
35
              this.env = env;
            }
36
37
38
            public void setAdminId(String adminId) {
39
              this.adminId = adminId;
40
41
            public void setAdminPwd(String adminPwd) {
42
43
              this.adminPwd = adminPwd;
44
45
46
            public String getAdminId() {
47
              return adminId;
48
            }
49
50
            public String getAdminPwd() {
51
              return adminPwd;
52
53
54
            @Override
55
            public void destroy() throws Exception {
56
              System.out.println("destroy()");
57
            }
58
59
            @Override
            public void afterPropertiesSet() throws Exception {
60
61
              System.out.println("afterPropertiesSet()");
              setAdminId(env.getProperty("admin.id"));
62
63
              setAdminPwd(env.getProperty("admin.pwd"));
            }
64
65
```

```
66
            //bean이 생성되기 전에 callback 으로 호출됨. 가장 먼저 호출됨.
 67
            //MainClass에서 사용하는 env 정보가 넘어옴.
 68
 69
            @Override
 70
            public void setEnvironment(Environment env) {
               System.out.println("setEnvironment()");
 71
 72
               setEnv(env);
 73
            }
          }
 74
 75
 76
       5)/src/main/resources/beans.xml 생성
 77
          <?xml version="1.0" encoding="UTF-8"?>
 78
 79
          <beans xmlns="http://www.springframework.org/schema/beans"</pre>
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 80
            xsi:schemaLocation="http://www.springframework.org/schema/beans
 81
            http://www.springframework.org/schema/beans/spring-beans.xsd">
 82
 83
             <bean id="adminConnection" class="info.javaexpert.AdminConnection" />
 84
 85
          </beans>
 86
 87
       6)/src/main/java/MainClass.java 생성
 88
 89
          package info.javaexpert;
 90
 91
          import java.io.IOException;
 92
 93
          import org.springframework.context.ConfigurableApplicationContext;
          import org.springframework.context.support.GenericXmlApplicationContext;
 94
 95
          import org.springframework.core.env.ConfigurableEnvironment;
 96
          import org.springframework.core.env.MutablePropertySources;
 97
          import org.springframework.core.io.support.ResourcePropertySource;
 98
 99
          public class MainClass {
            public static void main(String [] args){
100
101
               ConfigurableApplicationContext ctx = new GenericXmlApplicationContext();
102
               ConfigurableEnvironment env = ctx.getEnvironment();
103
104
               MutablePropertySources propertySouces = env.getPropertySources();
               //내가 원하는 정보를 얻을 때까지 모든 propertySources를 앞에서 부터 차례로 모두 검색함.
105
106
               try{
107
                 propertySouces.addLast(new ResourcePropertySource("classpath:admin.properties"));
                 //property 추가
108
109
                 System.out.println(env.getProperty("admin.id")); //property 추출
                 System.out.println(env.getProperty("admin.pwd"));
110
               }catch(IOException ex){}
111
112
113
               GenericXmlApplicationContext qCtx = (GenericXmlApplicationContext)ctx;
114
               gCtx.load("beans.xml");
115
               gCtx.refresh();
116
117
               AdminConnection adminConnection = qCtx.qetBean("adminConnection",
               AdminConnection.class);
118
               System.out.println("admin ID: " + adminConnection.getAdminId());
119
               System.out.println("admin PWD: " + adminConnection.getAdminPwd());
120
121
               gCtx.close();
122
               ctx.close();
123
            }
124
          }
125
126
127
     3. Property 파일을 이용한 설정
       1)환경에 따라 자주 변경되는 내용의 분리
128
129
       2)XML의 Bean 설정 메타정보는 어플리케이션 구조가 바뀌지 않으면 자주 변경되지 않는다.
```

```
3)반면에 프로퍼티 값으로 제공되는 일부 설정정보(예-DataSource Bean이 사용하는 DB 연결정보)는
130
       어플리케이션이 동작하는 환경(개발, 테스트, 스테이징, 운영)에 따라서 자주 바뀔 수 있다.
       4)변경되는 이유와 시점이 다르다면 분리하는 것이 객체지향 설계의 기본 원칙이기에 설정에도 동일한 원칙을
131
       적용할 수 있다.
132
       5)환경에 따라 자주 변경될 수 있는 내용은 properties 파일로 분리하는 것이 가장 깔끔하다.
133
       6)XML 처럼 복잡한 구성이 필요없고 키와 값의 쌍(key=value)으로 구성하면 된다.
       7)환경에 따라 자주 변경되는 내용의 분리의 예시
134
         -value속성에 설정된 값들은 환경에 따라 변경될 수 있는 내용이다.
135
136
         -자주 변경되는 값들은 properties 파일에 넣어 분리하는 것이 좋다.
137
138
            <beans.xml>
139
            <bean id="dataSource"</pre>
140
                        class="org.springframework.jdbc.datasource.SimpleDriverDataSource">
141
                 cproperty name="driverClass" value="com.mysql.jdbc.Driver" />
                 cproperty name="url" value="jdbc:mysql://localhost/testdb" />
142
                 cproperty name="username" value="spring" />
143
                 cproperty name="password" value="book" />
144
145
            </bean>
146
147
         -properties 파일로 분리한 정보는 ${}(property 치환자)을 이용하여 설정한다.
148
         -${} 값을 치환해주는 기능은 <context:property-placeholder> 태그에 의해 자동으로 등록되는
         PropertyPlaceHolderConfigurer Bean이 담당한다.
149
150
            <database.properties>
151
              db.driverClass=com.mysql.jdbc.Driver
152
              db.url=jdbc:mysql://localhost/testdb
153
              db.username=spring
154
              db.password=book
155
156
            <beens.xml>
157
              <context:property-placeholder
158
                     location="classpath:config/database.properties" />
159
              <bean id="dataSource"</pre>
160
                     class="org.springfremework.jdbc.datasource.SimpleDriverDataSource">
                   cproperty name="driverClass" value="${db.driverClass}" />
161
162
                   cproperty name="url" value="${db.url}" />
163
                   coperty name="username" value="${db.username}" />
                   coperty name="password" value="${db.password}" />
164
165
              </bean>
166
167
168
    4. Lab
169
       1)New > Spring Legacy Project > Simple Projects > Simple Spring Maven
170
            Project Name: PropertyDemo
171
172
       2)src/main/java > Create Package
173
         -info.javaexpert
174
175
       3)POJO class 작성
176
         -info.javaexpert > right-click > New > Class
177
         -Class name: Hello
178
179
            <Hello.java>
180
              package info.javaexpert;
181
182
              public class Hello{
183
                private String name;
184
                private Printer printer;
185
                private List<String> names;
186
187
                public Hello(){}
188
189
                public void setName(String name){
190
                   this.name = name;
191
                }
192
193
                public void setPrinter(Printer printer){
```

```
194
                      this.printer = printer;
195
                   }
196
197
                   public void setNames(List<String> list){
198
                      this.names = list;
199
                   }
200
                   public List<String> getNames(){
201
202
                      return this.names;
203
204
205
                   public String sayHello(){
206
                      return "Hello " + name;
207
208
209
                   public void print(){
210
                      this.printer.print(sayHello());
211
                   }
212
                }
213
           -info.javaexpert > right-click > New > Interface
214
215
             Interface name: Printer
216
217
              <Printer.java>
218
                package info.javaexpert;
219
220
                public interface Printer{
221
                   void print(String message);
222
223
224
           -info.javaexpert > right-click > New > Class
225
             Class Name: StringPrinter
226
227
              <StringPrinter.java>
228
                package info.javaexpert;
229
230
                public class StringPrinter implements Printer{
231
                   private StringBuffer buffer = new StringBuffer();
232
233
                   public void print(String message){
234
                      this.buffer.append(message);
235
236
237
                   public String toString(){
238
                      return this.buffer.toString();
239
240
                }
241
242
           -info.javaexpert > right-click > New > Class
             Class Name: ConsolePrinter
243
244
245
              <ConsolePrinter.java>
246
                package info.javaexpert;
247
                public class ConsolePrinter implements Printer{
248
249
                   public void print(String message){
250
                      System.out.println(message);
251
252
                }
253
254
        4)Bean Configuration XML 작성
255
           -/src/main/resources > right-click > New > Spring Bean Configuration File
256
             File name: beans.xml > Finish
257
                <?xml version="1.0" encoding="UTF-8"?>
258
                <beans xmlns="http://www.springframework.org/schema/beans"</pre>
259
                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
260
```

```
261
                   xsi:schemaLocation="http://www.springframework.org/schema/beans
                   http://www.springframework.org/schema/beans/spring-beans.xsd">
262
263
                   <bean id="hello" class="info.javaexpert.Hello">
                     cproperty name="name" value="Spring" />
264
                     cproperty name="printer" ref="printer" />
265
266
                     cproperty name="names">
                        t>
267
268
                           <value>AOP</value>
269
                           <value>Spring</value>
270
                           <value>DI</value>
271
                        </list>
272
                     </property>
273
                   </bean>
274
275
                   <bean id="printer" class="info.javaexpert.StringPrinter" />
276
                   <bean id="consolePrinter" class="info.javaexpert.ConsolePrinter" />
277
278
                </beans>
279
280
        5)DI Test 클래스 작성
281
          -/src/info.javaexpert > right-click > New > Class
282
             Class Name: MainClass
283
284
                package info.javaexpert;
285
286
                import java.util.List;
287
288
                import org.springframework.context.ApplicationContext;
289
                import org.springframework.context.support.GenericXmlApplicationContext;
290
291
                public class MainClass {
292
                   public static void main(String [] args){
293
                     ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
294
295
                     Hello hello = (Hello)ctx.getBean("hello");
296
                     System.out.println(hello.sayHello());
297
                     hello.print();
298
299
                     Printer printer = ctx.getBean("printer", StringPrinter.class);
300
                     System.out.println(printer.toString());
301
302
                     List<String> list = hello.getNames();
303
                     for(String value : list){
304
                        System.out.println(value);
305
306
                  }
                }
307
308
309
        6)Test
310
          -/src/info.javaexpert.test/HelloBeanTest.java > right-click > Run As > Java Application
311
312
                Hello Spring
313
                Hello Spring
314
                AOP
315
                Spring
316
                DΙ
317
318
        7)jUnit으로 테스트
319
           -/src/test/java > right-click > New > JUnit Test Case > HelloTest > Finish
320
321
                import static org.junit.Assert.assertEquals;
322
                import static org.junit.Assert.assertSame;
323
324
                import java.util.List;
325
326
                import org.junit.Test;
```

```
327
                import org.junit.runner.RunWith;
328
                import org.springframework.beans.factory.annotation.Autowired;
329
                import org.springframework.context.ApplicationContext;
330
                import org.springframework.test.context.ContextConfiguration;
331
                import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;
332
333
                import info.javaexpert.Hello;
334
                import info.javaexpert.Printer;
335
336
                @RunWith(SpringJUnit4ClassRunner.class)
337
                @ContextConfiguration(locations="classpath:beans.xml")
338
                public class HelloTest {
339
                  @Autowired
340
                  ApplicationContext ctx;
341
342
                  @Test
                  public void test() {
343
344
                     Hello hello = (Hello)ctx.getBean("hello");
345
                     assertEquals("Hello Spring", hello.sayHello());
346
                     hello.print();
347
348
                     Printer printer = (Printer)ctx.getBean("printer");
349
                     assertEquals("Hello Spring", printer.toString());
350
                  }
351
352
                  @Test
353
                  public void test2(){
354
                     Hello hello = (Hello)ctx.getBean("hello");
355
                     Hello hello2 = ctx.getBean("hello", Hello.class);
356
357
                     assertSame(hello, hello2);
358
359
                     assertEquals(3, hello2.getNames().size());
360
                  }
                }
361
362
363
          -right-click > Run As > Junit Test
364
          -결과 -> Junit View에 초록색 bar
365
366
        8)/src/main/resources/value.properties 생성
367
           <value.properties>
368
369
             myname=Spring
370
             myprinter=printer
371
             value1=HTML5
372
             value2=CSS3
373
             value3=JavaScript
374
375
        9)/src/config/beans.xml 에서 [Namespaces] tab
376
          -목록에서 'context-http://www.springframework.org/schema/context' check
377
          -<context:property-placeholder />를 사용하기 위해서
378
379
             <beans.xml>
380
                <context:property-placeholder
381
                        location="classpath:value.properties" />
382
383
                <bean id="hello" class="info.javaexpert.Hello">
384
                   cproperty name="name" value="${myname}" />
385
                   cproperty name="printer" ref="${myprinter}" />
386
                   property name="names">
387
                     t>
388
                        <value>${value1}</value>
389
                        <value>${value2}</value>
390
                        <value>${value3}</value>
391
                     </list>
392
                   </property>
393
                </bean>
```

```
395
        10)Test
396
           -/src/main/java/info.javaexpert.MainClass.java
397
             --right-click > Run As > Java Application
398
                Hello Spring
399
                Hello Spring
                HTML5
400
401
                CSS3
402
                JavaScript
403
           -/src/test/java/HelloTest.java
404
405
             --right-click > Run As > JUnit Test
406
             --Green Bar
407
        11)/src/main/resources/value.properties 수정
408
409
           <value.properties>
410
             myname=Spring
411
             myprinter=printer
412
             value1=JUnit
413
             value2=AOP
414
             value3=DI
415
             printer1=stringPrinter
416
             printer2=consolePrinter
417
418
        12)Hello.java 코드 수정
419
           -/src/info.javaexpert/Hello.java
420
421
             package info.javaexpert;
422
423
             import java.util.List;
424
425
             import org.springframework.beans.factory.annotation.Value;
426
             import org.springframework.stereotype.Component;
427
             import javax.annotation.Resource;
428
429
             @Component("hello")
430
             public class Hello {
431
                @Value("${myname}")
432
                private String name;
433
434
                @Resource(name="${printer1}")
435
                private Printer printer;
436
437
                @Value("${value1}, ${value2}, ${value3}")
438
                private List<String> names;
439
440
                public List<String> getNames(){
441
                  return names;
442
443
                public String sayHello(){
444
                   return "Hello " + name;
445
                }
446
447
                public void print(){
448
                   this.printer.print(sayHello());
449
                }
             }
450
451
452
        13)StringPrinter.java 수정
453
             package info.javaexpert;
454
455
             import org.springframework.stereotype.Component;
456
457
             @Component("stringPrinter")
             public class StringPrinter implements Printer {
458
459
                private StringBuffer buffer = new StringBuffer();
460
```

```
461
                @Override
462
                public void print(String message) {
463
                  this.buffer.append(message);
464
465
466
               @Override
               public String toString(){
467
468
                  return this.buffer.toString();
469
               }
470
             }
471
472
        14)beans.xml 수정하기
473
474
             <?xml version="1.0" encoding="UTF-8"?>
475
             <beans xmlns="http://www.springframework.org/schema/beans"</pre>
476
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
477
               xmlns:context="http://www.springframework.org/schema/context"
478
                xsi:schemaLocation="http://www.springframework.org/schema/beans
               http://www.springframework.org/schema/beans/spring-beans.xsd
479
                  http://www.springframework.org/schema/context
                  http://www.springframework.org/schema/context/spring-context-3.2.xsd">
480
                <context:property-placeholder location="classpath:value.properties"/>
481
482
                <context:component-scan base-package="info.javaexpert" />
483
             </beans>
484
485
        15)MainClass.java 수정하기
486
487
             package info.javaexpert;
488
489
             import java.util.List;
490
491
             import org.springframework.context.ApplicationContext;
492
             import org.springframework.context.support.GenericXmlApplicationContext;
493
494
             public class MainClass {
495
               public static void main(String [] args){
496
                  ApplicationContext ctx = new GenericXmlApplicationContext("classpath:beans.xml");
497
498
                  Hello hello = (Hello)ctx.getBean("hello");
499
                  System.out.println(hello.sayHello());
500
                  hello.print();
501
502
                  Printer printer = ctx.getBean("stringPrinter", StringPrinter.class);
503
                  System.out.println(printer.toString());
504
505
                  List<String> list = hello.getNames();
506
                  for(String value : list){
507
                     System.out.println(value);
508
                  }
509
               }
510
             }
511
512
     5. Lab
513
514
          ApplicationContext.xml --> XML 파일을 이용하는 방법
515
          -Spring 설정 XML 파일에 property 파일에 대해 명시한다.
516
          -admin.properties
517
          -sub_admin.properties
518
519
          ApplicationConfig --> Java 파일을 이용하는 방법
520
          -Spring 설정 Java 파일에 property 파일을 명시한다.
521
          -admin.properties
522
          -sub_admin.properties
523
524
        1)/src/main/java/info.javaexper package 생성
525
          -/src/main/java > right-click > New > Package
```

```
527
528
       2)/src/main/java/info.javaexpert.AdminConnection.java 생성
529
530
             <AdminConnection.java>
               package info.javaexpert;
531
532
533
               public class AdminConnection {
534
                  private String adminId;
535
                  private String adminPwd;
536
                  private String subAdminId;
537
                  private String subAdminPwd;
538
539
                  public String getAdminId() {
                    return adminId;
540
541
542
543
                  public void setAdminId(String adminId) {
544
                    this.adminId = adminId;
545
                  }
546
547
                  public String getAdminPwd() {
548
                    return adminPwd;
549
550
551
                  public void setAdminPwd(String adminPwd) {
552
                    this.adminPwd = adminPwd;
553
554
555
                  public String getSubAdminId() {
556
                    return subAdminId;
557
558
559
                  public void setSubAdminId(String subAdminId) {
560
                    this.subAdminId = subAdminId;
561
562
563
                  public String getSubAdminPwd() {
564
                    return subAdminPwd;
565
566
567
                  public void setSubAdminPwd(String subAdminPwd) {
568
                    this.subAdminPwd = subAdminPwd;
569
570
               }
571
572
       3)/src/main/resources 두 개의 properties 파일 생성
573
574
          <admin.properties>
575
             admin.id=javaexpert
576
             admin.pwd=12345678
577
578
          <sub.admin.properties>
579
             sub.admin.id=javasoft
580
             sub.admin.pwd=987654321
581
582
       4)/src/main/resources/beans.xml 생성
583
584
          <beans.xml>
             <?xml version="1.0" encoding="UTF-8"?>
585
586
             <beans xmlns="http://www.springframework.org/schema/beans"</pre>
587
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xmlns:context="http://www.springframework.org/schema/context"
588
589
               xsi:schemaLocation="http://www.springframework.org/schema/beans
               http://www.springframework.org/schema/beans/spring-beans.xsd
590
                  http://www.springframework.org/schema/context
                  http://www.springframework.org/schema/context/spring-context-3.2.xsd">
```

-Package name: info.javaexpert

```
591
592
                <context:property-placeholder location="classpath:admin.properties,</pre>
               classpath:sub.admin.properties" />
593
594
                <bean id="adminConnection" class="info.javaexpert.AdminConnection">
                  cproperty name="adminId">
595
596
                     <value>${admin.id}</value>
597
                  </property>
                  cproperty name="adminPwd">
598
599
                     <value>${admin.pwd}</value>
600
                  </property>
601
                  property name="subAdminId">
602
                     <value>${sub.admin.id}</value>
603
                  </property>
                  cproperty name="subAdminPwd">
604
605
                     <value>${sub.admin.pwd}</value>
606
                  </property>
607
                </bean>
608
             </beans>
609
610
        5)/src/main/java/MainClass.java 생성
611
          <MainClass.java>
612
             package info.javaexpert;
613
614
             import org.springframework.context.support.AbstractApplicationContext;
615
             import org.springframework.context.support.GenericXmlApplicationContext;
616
617
             public class MainClass {
618
               public static void main(String [] args){
                  AbstractApplicationContext ctx =
619
620
                       new GenericXmlApplicationContext("classpath:beans.xml");
                  AdminConnection connection = ctx.getBean("adminConnection", AdminConnection.class);
621
622
                  System.out.println("admin ID: " + connection.getAdminId());
623
                  System.out.println("admin PWD: " + connection.getAdminPwd());
                  System.out.println("sub admin ID: " + connection.getSubAdminId());
624
                  System.out.println("sub admin PWD: " + connection.getSubAdminPwd());
625
626
627
                  ctx.close();
628
               }
629
             }
630
631
632
     6. Lab
633
        1)/src/main/resources 두 개의 properties 파일 생성
634
635
          <admin.properties>
636
             admin.id=javaexpert
             admin.pwd=12345678
637
638
639
          <sub.admin.properties>
640
             sub.admin.id=iavasoft
641
             sub.admin.pwd=987654321
642
643
       2)/src/main/java/info.javaexpert.ApplicationConfig.java>
644
645
          <ApplicationConfig.java>
646
             package info.javaexpert;
647
648
             import org.springframework.beans.factory.annotation.Value;
649
             import org.springframework.context.annotation.Bean;
650
             import org.springframework.context.annotation.Configuration;
651
             import org.springframework.context.support.PropertySourcesPlaceholderConfigurer;
652
             import org.springframework.core.io.ClassPathResource;
653
             import org.springframework.core.io.Resource;
654
655
             @Configuration
656
             public class ApplicationConfig {
```

```
@Value("${admin.id}")
658
               private String adminId;
659
               @Value("${admin.pwd}")
660
               private String adminPwd;
661
               @Value("${sub.admin.id}")
662
               private String subAdminId;
               @Value("${sub.admin.pwd}")
663
               private String subAdminPwd;
664
665
666
               @Bean
667
               public static PropertySourcesPlaceholderConfigurer Properties(){
668
                  PropertySourcesPlaceholderConfigurer configurer =
669
                       new PropertySourcesPlaceholderConfigurer();
670
671
                  Resource [] locations = new Resource[2];
672
                  locations[0] = new ClassPathResource("admin.properties");
                  locations[1] = new ClassPathResource("sub.admin.properties");
673
674
                  configurer.setLocations(locations);
675
676
                  return configurer;
               }
677
678
679
             @Bean
             public AdminConnection adminConfig(){
680
681
               AdminConnection adminConnection = new AdminConnection();
682
               adminConnection.setAdminId(adminId);
683
               adminConnection.setAdminPwd(adminPwd);
684
               adminConnection.setSubAdminId(subAdminId);
685
               adminConnection.setSubAdminPwd(subAdminPwd);
686
               return adminConnection;
687
            }
          }
688
689
690
       3)/src/main/java/info.javaexpert.MainClass1.java
691
692
          <MainClass1.java>
693
             package info.javaexpert;
694
695
             import org.springframework.context.annotation.AnnotationConfigApplicationContext;
696
697
             public class MainClass1 {
698
               public static void main(String[] args) {
699
                  AnnotationConfigApplicationContext ctx =
700
                       new AnnotationConfigApplicationContext(ApplicationConfig.class);
                  AdminConnection conn = ctx.getBean("adminConfig", AdminConnection.class);
701
702
                  System.out.println("admin ID: " + conn.getAdminId());
703
704
                  System.out.println("admin PWD: " + conn.getAdminPwd());
705
                  System.out.println("sub admin ID: " + conn.getSubAdminId());
                  System.out.println("sub admin PWD: " + conn.getSubAdminPwd());
706
707
708
             }
709
710
711
     7. Profile 속성을 이용한 설정
712
       -동일한 Spring Bean을 여러개 만들어 놓고 상황(환경)에 따라서 적절한 스프링 빈을 사용할 수 있다.
713
       -profile 속성을 사용한다.
714
       -역시 Java 파일을 이용하는 방법과 XML 설정 파일을 이용하는 방법이 있다.
715
716
717
     8. Lab
       1)Package 생성
718
719
          -/src/main/java > right-click > New > Package
720
          -Package name : info.javaexpert
721
722
       2)XML 설정 파일 2개 생성
723
          -/src/main/resource > right-click > New > Spring Bean Configuration File
```

```
725
726
                <?xml version="1.0" encoding="UTF-8"?>
727
                <beans xmlns="http://www.springframework.org/schema/beans"</pre>
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
728
                  xsi:schemaLocation="http://www.springframework.org/schema/beans
729
                  http://www.springframework.org/schema/beans/spring-beans.xsd"
730
                  profile="run">
                                 <---이것이 핵심
731
732
                  <bean id="serverInfo" class="info.javaexpert.ServerInfo">
733
                     roperty name="ipNum" value="192.168.56.5" />
734
                     cproperty name="portNum" value="80" />
                  </bean>
735
736
                </beans>
737
738
          -/src/main/resource > right-click > New > Spring Bean Configuration File
739
          -File name : dev.xml
740
741
                <?xml version="1.0" encoding="UTF-8"?>
742
                <beans xmlns="http://www.springframework.org/schema/beans"</pre>
743
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
744
                  xsi:schemaLocation="http://www.springframework.org/schema/beans
                  http://www.springframework.org/schema/beans/spring-beans.xsd"
745
                  profile="dev">
                                 <---이것이 핵심
746
747
                  <bean id="serverInfo" class="info.javaexpert.ServerInfo">
748
                     roperty name="ipNum" value="192.168.56.5" />
749
                     cproperty name="portNum" value="80" />
750
                  </bean>
751
                </beans>
752
753
        3)ServerInfo.java 생성
          -/src/main/java/info.javaexpert.ServerInfo.java
754
755
756
             package info.javaexpert;
757
758
             public class ServerInfo {
759
               private String ipNum;
760
               private String portNum;
761
               public String getIpNum() {
762
                  return ipNum;
763
764
               public void setIpNum(String ipNum) {
765
                  this.ipNum = ipNum;
766
767
               public String getPortNum() {
768
                  return portNum;
769
770
               public void setPortNum(String portNum) {
771
                  this.portNum = portNum;
772
             }
773
774
775
        4)MainClass 생성
776
          -/src/main/java/info.javaexpert.MainClass.java
777
778
             package info.javaexpert;
779
             import java.util.Scanner;
780
781
             import org.springframework.context.support.GenericXmlApplicationContext;
782
783
             public class MainClass {
784
               public static void main(String[] args) {
785
                  Scanner scan = new Scanner(System.in);
786
                  System.out.print("Select dev or run: ");
                  String config = scan.next(); //"dev" or "run"
787
788
```

-File name: run.xml

```
789
                   GenericXmlApplicationContext ctx = new GenericXmlApplicationContext();
790
                   ctx.getEnvironment().setActiveProfiles(config);
                   ctx.load("dev.xml", "run.xml");
791
792
                   ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
793
                   System.out.println("IP: " + info.getIpNum());
794
                   System.out.println("Port : " + info.getPortNum());
795
796
                   ctx.close();
797
                }
798
             }
799
800
        5)결과
801
           -입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이
           나온다.
802
803
804
     9. Lab
        1)Package 생성
805
806
           -/src/main/java > right-click > New > Package
807
           -Package name : info.javaexpert
808
809
        2) Java 설정 파일 2개 생성
810
           -/src/main/java > right-click > New > Class
811
           -Class name : ApplicationConfigDev
812
813
                package info.javaexpert;
814
815
                import org.springframework.context.annotation.Bean;
816
                import org.springframework.context.annotation.Configuration;
817
                import org.springframework.context.annotation.Profile;
818
                @Configuration
819
820
                @Profile("dev")
821
                public class ApplicationConfigDev {
822
823
824
                   public ServerInfo serverInfo(){
825
                     ServerInfo info = new ServerInfo();
                     info.setIpNum("localhost");
826
827
                     info.setPortNum("8080");
828
                     return info;
829
                   }
830
                }
831
832
           -/src/main/java > right-click > New > Class
833
           -Class name : ApplicationConfigRun
834
835
                package info.javaexpert;
836
837
                import org.springframework.context.annotation.Bean;
838
                import org.springframework.context.annotation.Configuration;
839
                import org.springframework.context.annotation.Profile;
840
                @Configuration
841
                @Profile("run")
842
843
                public class ApplicationConfigRun {
844
845
                   @Bean
846
                   public ServerInfo serverInfo(){
847
                     ServerInfo info = new ServerInfo();
                     info.setIpNum("192.168.56.5");
848
849
                     info.setPortNum("80");
850
                     return info;
851
                }
852
853
854
        3)ServerInfo.java 생성
```

```
855
          -/src/main/java/info.javaexpert.ServerInfo.java
856
857
             package info.javaexpert;
858
859
             public class ServerInfo {
860
                private String ipNum;
                private String portNum;
861
862
                public String getIpNum() {
863
                  return ipNum;
864
865
                public void setIpNum(String ipNum) {
866
                  this.ipNum = ipNum;
867
                public String getPortNum() {
868
869
                  return portNum;
870
871
                public void setPortNum(String portNum) {
872
                  this.portNum = portNum;
873
                }
874
             }
875
876
        4)MainClass 생성
877
          -/src/main/java/info.javaexpert.MainClass.java
878
879
             package info.javaexpert;
880
             import java.util.Scanner;
881
882
             import org.springframework.context.annotation.AnnotationConfigApplicationContext;
883
             public class MainClass {
884
885
                public static void main(String[] args) {
886
                  Scanner scan = new Scanner(System.in);
887
                  System.out.print("Select dev or run : ");
888
                  String config = scan.next(); //"dev" or "run"
889
                  AnnotationConfigApplicationContext ctx = new AnnotationConfigApplicationContext();
890
891
                  ctx.getEnvironment().setActiveProfiles(config);
892
                  ctx.register(ApplicationConfigDev.class, ApplicationConfigRun.class);
893
                  ctx.refresh();
894
895
                  ServerInfo info = ctx.getBean("serverInfo", ServerInfo.class);
896
                  System.out.println("IP: " + info.getIpNum());
897
                  System.out.println("Port : " + info.getPortNum());
898
                  ctx.close();
899
                }
900
             }
901
902
903
          -입력시 dev를 넣으면 dev환경인 localhost/8080이 나오고, 만일 run이라고 넣으면 192.168.56.5/80이
          나온다.
```