

```

1  package com.study.DemoAnnotations;
2
3  public interface FortuneService {
4      public String getFortune();
5  }
6
7  ///////////////////////////////////////////////////
8
9  package com.study.DemoAnnotations;
10
11  import org.springframework.stereotype.Component;
12
13  @Component
14  public class HappyFortuneService implements FortuneService {
15
16      @Override
17      public String getFortune() {
18          return "Its a happy day today";
19      }
20  }
21
22  ///////////////////////////////////////////////////
23  package com.study.DemoAnnotations;
24
25  import java.util.Random;
26
27  import org.springframework.stereotype.Component;
28
29  @Component
30  public class RandomFortuneService implements FortuneService {
31
32      String[] messages = { "Beware of sheep in wolf clothing",
33                          "Diligence is the mother of good work",
34                          "Journey is the reward" };
35
36      private Random random = new Random();
37
38      @Override
39      public String getFortune() {
40          int index = random.nextInt(messages.length);
41          return messages[index];
42      }
43  }
44
45  ///////////////////////////////////////////////////
46
47  package com.study.DemoAnnotations;
48
49  public interface Coach {
50      public String getDailyWorkout();
51
52      public String getDailyFortune();
53  }
54
55  ///////////////////////////////////////////////////
56
57  package com.study.DemoAnnotations;
58
59  import org.springframework.beans.factory.annotation.Autowired;
60  import org.springframework.stereotype.Component;
61
62  @Component
63  public class TennisCoach implements Coach {
64
65      private FortuneService fortuneService;
66
67      // using constructor injection
68      @Autowired
69      public TennisCoach(FortuneService fortuneService) {

```

```

70         super();
71         this.fortuneService = fortuneService;
72     }
73
74     @Override
75     public String getDailyWorkout() {
76         return "Practice your backhand volley";
77     }
78
79     @Override
80     public String getDailyFortune() {
81         return fortuneService.getFortune();
82     }
83 }
84
85 ///////////////////////////////////////////////////
86
87 package com.study.DemoAnnotations;
88
89 import org.springframework.context.support.ClassPathXmlApplicationContext;
90
91 public class AnnotationsDemoApp {
92
93     public static void main(String[] args) {
94         // load the spring configuration file
95         ClassPathXmlApplicationContext context = new
96             ClassPathXmlApplicationContext("annotations-applicationContext.xml");
97
98         // retrieve the bean from the spring container
99         Coach theCoach = context.getBean("tennisCoach", Coach.class);
100        // call methods on bean
101        System.out.println(theCoach.getDailyWorkout());
102
103        System.out.println(theCoach.getDailyFortune());
104
105        // close the context
106        context.close();
107    }
108 }
109
110 ///////////////////////////////////////////////////
111
112 <?xml version="1.0" encoding="UTF-8"?>
113 <beans xmlns="http://www.springframework.org/schema/beans"
114     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
115     xmlns:context="http://www.springframework.org/schema/context"
116     xsi:schemaLocation="http://www.springframework.org/schema/beans
117         http://www.springframework.org/schema/beans/spring-beans.xsd
118         http://www.springframework.org/schema/context
119         http://www.springframework.org/schema/context/spring-context.xsd">
120
121     <!-- add entry to enable component scanning -->
122     <context:component-scan base-package="com.study.DemoAnnotations" />
123
124 </beans>
125
126 ///////////////////////////////////////////////////
127 /** Modified class for Auto Wiring by Setter methods or any method */
128
129 package com.study.DemoAnnotations;
130
131 import org.springframework.beans.factory.annotation.Autowired;
132 import org.springframework.stereotype.Component;
133
134 @Component
135 public class TennisCoach implements Coach {
136
137     private FortuneService fortuneService;

```

```

139
140     /*
141     * public TennisCoach(FortuneService fortuneService) { super();
142     * this.fortuneService = fortuneService; }
143     */
144
145     public TennisCoach() {
146         super();
147         System.out.println("Inside the default constructor of TennisCoach");
148     }
149
150     // setter function for Autowiring ..... actually it can be any method name
151     @Autowired
152     public void setFortuneService(FortuneService fortuneService) {
153         System.out.println("Inside the set method setFortuneService of TennisCoach ");
154         this.fortuneService = fortuneService;
155     }
156
157     @Override
158     public String getDailyWorkout() {
159         return "Practice your backhand volley";
160     }
161
162     @Override
163     public String getDailyFortune() {
164         return fortuneService.getFortune();
165     }
166 }
167
168 ///////////////////////////////////////////////////
169
170 package com.study.DemoAnnotations;
171
172 import org.springframework.beans.factory.annotation.Autowired;
173 import org.springframework.stereotype.Component;
174
175 @Component
176 public class TennisCoach implements Coach {
177
178     @Autowired
179     private FortuneService fortuneService;
180
181     /*
182     * public TennisCoach(FortuneService fortuneService) { super();
183     * this.fortuneService = fortuneService; }
184     */
185
186     public TennisCoach() {
187         super();
188         System.out.println("Inside the default constructor of TennisCoach");
189     }
190
191     /*
192     * // setter function for Autowiring ..... actually it can be any method name
193     *
194     * @Autowired public void setFortuneService(FortuneService fortuneService) {
195     * System.out.println("Inside the set method setFortuneService of TennisCoach "
196     * ); this.fortuneService = fortuneService; }
197     */
198
199     @Override
200     public String getDailyWorkout() {
201         return "Practice your backhand volley";
202     }
203
204     @Override
205     public String getDailyFortune() {
206         return fortuneService.getFortune();
207     }

```

```

208 }
209
210 //////////////////////////////////////////////////
211
212 @Autowired
213 public TennisCoach(@Qualifier("randomFortuneService") FortuneService theFortuneService) {
214
215     System.out.println(">> TennisCoach: inside constructor using @autowired and
216     @qualifier");
217
218     fortuneService = theFortuneService;
219 }
220
221 //////////////////////////////////////////////////
222 // add support to resolve ${...} properties
223 @Bean
224 public static PropertySourcesPlaceholderConfigurer
225     propertySourcesPlaceHolderConfigurer() {
226
227     return new PropertySourcesPlaceholderConfigurer();
228 }
229
230
231 package com.study.config;
232
233 import org.springframework.context.annotation.Bean;
234 /*import org.springframework.context.annotation.ComponentScan;*/
235 import org.springframework.context.annotation.Configuration;
236 import org.springframework.context.annotation.PropertySource;
237
238 @Configuration
239 /* @ComponentScan("com.study.config") */
240 @PropertySource("classpath:sport.properties")
241 public class SportConfig {
242
243     // define bean for a SadFortuneService, method name is the actual bean id
244     @Bean
245     public FortuneService sadFortuneService() {
246         return new SadFortuneService();
247     }
248
249     @Bean
250     public Coach swimCoach() {
251         return new SwimCoach(sadFortuneService());
252     }
253 }
254
255
256 <?xml version="1.0" encoding="UTF-8"?>
257 <beans xmlns="http://www.springframework.org/schema/beans"
258     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
259     xmlns:context="http://www.springframework.org/schema/context"
260     xsi:schemaLocation="http://www.springframework.org/schema/beans
261     http://www.springframework.org/schema/beans/spring-beans.xsd
262     http://www.springframework.org/schema/context
263     http://www.springframework.org/schema/context/spring-context.xsd">
264
265     <!-- add entry to enable component scanning -->
266     <context:component-scan base-package="com.study.springdemo" />
267
268     <!-- Load the properties file -->
269     <context:property-placeholder location="classpath:sport.properties"/>
270
271     <bean id="myLoggerConfig" class="com.study.springdemo.MyLoggerConfig"
272     init-method="initLogger">
273         <property name="rootLoggerLevel" value="FINE" />
274         <property name="printedLoggerLevel" value="FINE"/>
275     </bean>

```

```
275
276 <!-- define the dependency -->
277 <bean id="myFortuneService"
278     <class="com.study.springdemo.HappyFortuneService">
279 </bean>
280
281 <bean id="myCoach"
282     <class="com.study.springdemo.TrackCoach">
283
284     <!-- setup constructor injection -->
285     <constructor-arg ref="myFortuneService"></constructor-arg>
286 </bean>
287
288 <bean id="myCricketCoach"
289     <class="com.study.springdemo.CircketCoach" scope="singleton">
290
291     <!-- setup setter injection -->
292     <property name="fortuneService" ref="myFortuneService" />
293
294     <!-- setup setter injection for literal values -->
295     <property name="emailAddress" value="${foo.email}" />
296     <property name="team" value="${foo.team}" />
297
298 </bean>
299 </beans>
300
301
302
303
304
```