


```

<servlet-mapping>
    <servlet-name>spring</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
</web-app>

```

8f k l / < R Java 2 ~ Z Spring\$

Spring Java 2 ? @! È @Configuration Q G < @Bean Q G " 2 C È @Bean Q G
 2 ~ TM { < , - J _ - È Spring 2 IoC | } " i ð C
 @Bean y 2 q / | <bean/> Í C v @Configuration y Q G 2 Í z _ - Í
 2 () ² 2 ! q " bean 2 ´ µ C v @Configuration 2 Í O ´ è b û , - Í 2 ²
 / @bean ~ TM " X â bean 2 Ý j ¢ £ C
 x Ø 2 @Configuration Í * i 2 Ú Û £

```

@Configuration
public class AppConfig{
    @Bean
    public MyService myService() {
        return new MyServiceImpl();
    }
}

```

R â i 2 @Beans > œ§ û 2 XML > œk ° £

```

<beans>
    <bean id="myService" class="com.somnus.services.MyServiceImpl"/>
</beans>

```

â ~ Z 2 ~ Z k ° £ ¼/ AnnotationConfigApplicationContext Í n o

```

public static void main(String[] args) {
    ApplicationContext ctx = new
    AnnotationConfigApplicationContext(AppConfig.class);
    MyService myService = ctx.getBean(MyService.class);
    myService.doStuff();
}

```

```
}
```

```
) K / É œÉ J Ò Ó / @Configuration n o Q G Õ O Ł
```

```
@Configuration  
@ComponentScan(basePackages = "com.somnus")  
public class AppConfig {  
    ...  
}
```

```
b â ĩ 2 ĩ J com.acme ĩ v J ¥ m ĩ } ² v @Component 2 ĩ J  
m _ ' ĩ V W Sring bean n o Q C
```

```
k ) b 2 web. / 3 4 ĩ Õ / â 2 2 ~ Z 2 J Ó ) /
```

```
AnnotationConfigWebApplicationContext ĩ "  
Servlet } ContextLoaderListener F Spring MVC 2 DispatcherServletC
```

```
<web-app>
```

```
<!-- Configure ContextLoaderListener to use  
AnnotationConfigWebApplicationContext  
instead of the default XmlWebApplicationContext -->
```

```
<context-param>
```

```
<param-name>contextClass</param-name>
```

```
<param-value>
```

```
org.springframework.web.context.support.AnnotationConfigWebApplicati  
onContext
```

```
</param-value>
```

```
</context-param>
```

```
<!-- Configuration locations must consist of one or more comma- or  
space-delimited
```

```
fully-qualified @Configuration classes. Fully-qualified  
packages may also be
```

```
specified for component-scanning -->
```

```
<context-param>
```

```
<param-name>contextConfigLocation</param-name>
```

```
<param-value>com.howtodoinjava.AppConfig</param-value>
```

```
</context-param>
```

```
<!-- Bootstrap the root application context as usual using  
ContextLoaderListener -->
```

```

    <listener>
      <listener-
class>org.springframework.web.context.ContextLoaderLi stener</li stener
-class>
    </li stener>

    <!-- Declare a Spring MVC DispatcherServlet as usual -->
    <servlet>
      <servlet-name>di spatcher</servlet-name>
      <servlet-
class>org.springframework.web.servlet.Di spatcherServlet</servlet-
class>
      <!-- Configure DispatcherServlet to use
AnnotationConfigWebAppl icationContext
      instead of the default XmlWebAppl icationContext -->
      <ini t-param>
        <param-name>contextCl ass</param-name>
        <param-val ue>
org.springframework.web.context.support.AnnotationConfigWebAppl icati o
nContext
        </param-val ue>
      </ini t-param>
      <!-- Again, config locations must consist of one or more comma-
or space-del imited
      and fully-qual ified @Configurati on classes -->
      <ini t-param>
        <param-name>contextConfigLocati on</param-name>
        <param-val ue>com.howtodoi nj ava.web.MvcConfig</param-
val ue>
      </ini t-param>
    </servlet>

    <!-- map all requests for /app/* to the dispatcher servlet -->
    <servlet-mappi ng>
      <servlet-name>di spatcher</servlet-name>
      <url -pattern>/app/*</url -pattern>
    </servlet-mappi ng>
  </web-app>

```

9† ° / Q G 2 ~ Z Spring\$

Spring b 2.5 S ´ m3 ? @ / Q G 2 ~ Z " Y j Q â C O ´ / Q G 2 ~ Z " Ú XML ~
Z 2 bean J O ´ bean ý É œ í 2 ² J Ó) b § ¢ í á t ~ ¢ F
â K / Q G Œ O C Q G Q â v | } b XML Q â v • ð J y ´ m F F R ú ,
- Ñ > 2 • ð C

Q G b Spring I ! ¢ ¿ 2 C y ´ Ó) b Spring > œ l , ° ´ K / < R Q G 2
* Z C k) b 2 . / O 1 I K / ¢ R Q G 2 ~ ¢ 2 J k ° 2 C

```
<beans>
  <context:annotation-config/>
  <!-- bean definitions go here -->
</beans>
```

b <context:annotation-config/> g ´ m J ´ O ´ / Q G 2 ~ Z b Spring I , Ñ
> † ~ ¢ ¢ § ^ ~ ¢ | ° ® C
° I ! ¢ ©) 2 Q G í ð Ł

1. @RequiredŁ Q G . / R X ~ ¢ C
2. @AutowiredŁ Q G . / R % X ~ ¢ † X ~ ¢ † § ^ ~ ¢ ¢ ° ® C
3. @QualifierŁ Q G < @Autowired Q G K / J / R % æ bean 2 C
4. JSR-250 AnnotationsŁ Spring ? @ < R JSR-250 Q G 2 ´ ° Q G J @ResourceŁ
@PostConstruct < @PreDestroyC

10† G Spring Bean 2 \$

Spring Bean 2 × Ø C b , - bean v J Ó) o , £ - 2 q
´ O / 2 C ú ° 2 J , - bean h b v / Ó) n o § ¢ 2 § q J ½ bean |
} I æ C

Spring bean factory i ð b spring | } I v 2 bean 2 C Bean 2
É É f call back † ~ ¢ É g C

1. m / 2 ~ ¢ C
2. / 2 ~ ¢ C

Spring " # 5 6 7 ´ ° ~ Z " i ð bean 2 í œ Ł

InitializingBean < DisposableBean ñ ò

o " 2 Å Æ Aware ñ ò

Bean > œ l 2 Custom init() ~ ¢ < destroy() ~ ¢

17† G * Z 2 \$

b Spring" # l ú % 5 J ` a , w C

1. noŁ _! Spring" # 2 X J b X ° ! ¢ ĵ 2 J 3 4 F Ó) o b bean
l / 2 X Ÿ j ¢ £ C
2. byNameŁ Ô Ö O´ € bean X Ÿ j ¢ £ C „ , - bean l , - Ñ > J |
} € bean 2 b b › œl , - 2 bean C k 2 J ¹ _ - Ñ
> J k ¾ 2 ¹ C
3. byTypeŁ Ô Ö O´ € bean Í õ X Ÿ j ¢ £ C „ , - bean l , - Ñ > J | }
€ bean 2 Í õ b b › œl , - 2 bean C k 2 J ¹ _ - Ñ > J
k ¾ 2 ¹ C
4. constructorŁ ^ } 2 < byType * Z Í J ! Ô Ö ó / R | % š ^ } § û ~ 2
bean J k b | } l ¾ % | š ^ } ~ Í õ , 2 bean J ž Đ Ą C
5. autodetectŁ * Z Ą E K / š ^ } F byType C J / 9
ó 2 ´ ~ 2 š ^ } J k 2 ¹ ! / š ^ } J k b bean² ¾ % § . 2 š
^ } F ! š ^ } J | } ¹ Ô ä byTpe 2 ~ Z C

18† k l 3 < R Q G 2 \$

) K / @Autowired J Ó) Q AutowiredAnnotationBeanPostProcessor J O´
%´ ° ~ Z " Ł

1† â › œl 2 <bean>° â <context: annotation-config>

```
<beans>
  <context: annotation-config />
</beans>
```

2† b bean › œl ñ â AutowiredAnnotationBeanPostProcessor

```
<beans>
  <bean
class="org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor"/>
</beans>
```

```

191  — G @Required Q G $
b 2. / I J IoC | } O ' 7 ~ » 7 beanJ bean | bean % ø ù 2 Ÿ j ¢
£ C X Q G ~ ™ 2 , ' ! y % 2 Ñ > ! v Q G ! , Ö » w 2 q C O ' è b
<bean>I X dependency-check ½ " G H _ - fi fl C
b . / O 1 2 I J O ' h b y % bean 2 Ñ > ! V W å ° > > œ
C F O - bean 2 Ñ > ! v 2 X C Ö K ! / dependency-
check Ñ > h ' à " 2 G H _ - fi fl J b _ ¶ . ° J Ó ) K / @Required Q G C
Ó ) / k ° 2 ~ Z K / " bean 2 X ~ ™ C

```

```

public class EmployeeFactoryBean extends AbstractFactoryBean<Object>{
    private String designation;
    public String getDesignation() {
        return designation;
    }
    @Required
    public void setDesignation(String designation) {
        this.designation = designation;
    }
    //more code here
}

```

```

RequiredAnnotationBeanPostProcessor! SpringI 2 m • ð / " v
@Required Q G 2 bean Ñ > ! v 2 X 7 C b K /
RequiredAnnotationBeanPostProcess " bean Ñ > J ) b IoC | } I
Å n o Q ¤

```

```

<bean
class="org.springframework.beans.factory.annotation.RequiredAnnotati
onBeanPostProcessor" />

```

```
! k ¾%Ñ> v / @Required Q G è 2 J m • ð } Ð , -
BeanInitializationException . c
```

```
20† — G @Autowired Q G $
```

```
@Autowired Q G I I • v 5 6 7 ª u é 2 ü ý c @Autowired Q G O
' @Required Q G † š ^ } , ° v / R b bean 2 X ~ ºå bean
2 Ñ> J , - ~ F ' % ' % u - ~ 2 ~ ºC
" k J O ' b X ~ ºå K / @Autowired Q G " Ú > œl 2 <property>
C Spring | } b setter ~ ºå @Autowired Q G J // byType
C
¥ ` a O ' b š ^ ~ ºå K / @Autowired Q G C ' % @Autowired Q G 2 š ^ ~ º
b , - bean v J Œ á b > œl K / <constructor-arg> c
```

```
public class TextEditor {
    private SpellChecker spellChecker;
    @Autowired
    public TextEditor(SpellChecker spellChecker){
        System.out.println("Inside TextEditor constructor." );
        this.spellChecker = spellChecker;
    }
    public void spellCheck(){
        spellChecker.checkSpelling();
    }
}
```

```
° I ! ¾%š ^ ~ 2 ~ Z Ł
```

```
<beans>
```

```
<context:annotation-config/>
```

```
<!-- Definition for textEditor bean without constructor-arg -->
<bean id="textEditor" class="com.howtodoinjava.TextEditor"/>
```

```
<!-- Definition for spellChecker bean -->
<bean id="spellChecker" class="com.howtodoinjava.SpellChecker"/>
```



```
</beans>
```

211 — @Qualifier Q G \$

@Qualifier Q G O ' b v Q bean 2 å O ' C Qualifier Q
G O ' / " %Spring h ' %2 bean . / C

* I 2 z b Customer 2 person Ñ > I person 2 C

```
public class Customer{  
    @Autowired  
    private Person person;  
}
```

* I ` a) b > œl " Person f C

```
<bean id="customer" class="com.somnus.common.Customer" />
```

```
<bean id="personA" class="com.somnus.common.Person" >  
    <property name="name" value="lokes" />  
</bean>
```

```
<bean id="personB" class="com.somnus.common.Person" >  
    <property name="name" value="alex" />  
</bean>
```

Spring) & - person bean \$ h 2 J ! o å i 2 z
J Ð ° i 2 , Ł

Caused by:

org.springframework.beans.factory.NoSuchBeanDefinitionException:
No unique bean of type [com.howtodoinjava.common.Person] is defined:
expected single matching bean but found 2: [personA, personB]

) G H å i 2 fi fl J Ó) K / @Qual i fi er Q G " Spring | }) & - beanŁ

```
public class Customer{  
    @Autowired  
    @Qual i fi er("personA")  
    private Person person;  
}
```

22ł š ^ ~ ™Q â < X Q â %Ž ž \$
Q ° 2 Ł

1. b X Q â ~ ™? @ w2 Ÿ i Q â J k ` a Ò Ó) Q â intł string< long õ 2 ° @ J ` a h) / X 2 ~ ™Q â C R < S í õ J k ` a ¾ %Q â 2 J O ´ " < S í õ X C
b š ^ ~ ™ Q â h ? @ w2 Ÿ i Q â J Û " b / š ^ ~ ™ l i â 2 š ^ ~ J
2 " C

2. X Q â h ã š ^ ~ ™2 C k ` a û , - ° ® û K / 7 š ^ ~ ™Q â K / 7 X ~ ™Q â 2 J ž š ^ ~ ™ h ' Ě X ~ ™Q â 2 C à J Û " š ^ ~ ™ b v / C

3. b K / X Q â %O ´ h ' Ÿ i ! t Ÿ v Q â J ´ ! _ 2 Ÿ i ¢ £ %O ´ ! h 2 C b , ¶ · ° J š ^ } Q â h g Ÿ i ¢ £ h 2 C

4. b X Q â k A < B \$ Ÿ i J b A Spring Ð sObjectCurrentlyInCreationException , J Û " b B v A ! h ´ v 2 J p U ¥ C y ´ Spring / X Q â 2 ~ ™G H 7 c Ÿ i 2 fi fl J Û 2 X ~ ™! b v v / 2 C
23ł Spring" # l % & ' h ú í õ 2 í œ \$

```

Spring 2 ApplicationContext 5 6 7 ? @í œœ Ú Ū I } 2 ' C
` a O` bean / " b ApplicationContext I 4 , 2 í œœC ApplicationEvent
í < b ApplicationContext ñ ò I • ò 2 í œœJ k , - bean 7
ApplicationListener ñ ò J , - ApplicationEvent v 4 , ' mJ bean v
C

```

```

public class AllApplicationEventListener implements ApplicationListener
< ApplicationEvent >{
    @Override
    public void onApplicationEvent(ApplicationEvent applicationEvent)
    {
        //process event
    }
}

```

Spring 5 6 7 ' ° 5 I 2 í œœŁ

1. â ° > â í œœf ContextRefreshedEvent‡ Ł í œœ b ApplicationContext v F
â 4 , C O` b / ConfigurableApplicationContext ñ ò I 2 refresh()™ v
4 C
2. â ° > 3 í œœf ContextStartedEvent‡ Ł | } / ConfigurableApplicationContext 2
Start()™ 3 / 3 | } 4 í œœC
3. â ° > í œœf ContextStoppedEvent‡ Ł | } / ConfigurableApplicationContext 2
Stop()™ | } 4 í œœC
4. â ° > ¢ ĵ í œœf ContextClosedEvent‡ Ł ApplicationContext v ¢ ĵ 4 í œœC | }
v ¢ ĵ J Å ï ò 2 y %Ø Bean Ū v C
5. • ò í œœf RequestHandledEvent‡ Ł b Web . / I J , - http f request‡
4 í œœC

```

æ7 å I 2 í œœ' ê J O` è ApplicationEvent í " 3 4 2 í œœC
public class CustomApplicationEvent extends ApplicationEvent{
    public CustomApplicationEvent ( Object source, final String msg ){
        super(source);
        System.out.println("Created a Custom event");
    }
}

```


