

# 스프링 인 액션(3판, 1판1쇄) 수정사항

[1쇄에서 발견된 오타자 정보와 오류 사항 그리고 보다 매끄러운 문장을 위해 수정한 내용들입니다(2쇄 반영 내용). 불편을 끼쳐드려 죄송합니다.]

(1판1쇄)

(최종 수정일 : 2012.04.25)

연번	페이지	세부위치	1판 1쇄	1판 2쇄(수정사항)	설명	구분
1	p.16	코드 1.9	<code>quest.embark(); quest.embark(); minstrel.singAfterQuest( }</code>	<code>quest.embark(); minstrel.singAfterQuest( }</code>		
2	p.23	밑에서 두 번째 줄	애플리케이션 텍스트	애플리케이션 컨텍스트		
3	p.25	표 1.5	"BeanNameAware의 setBeanName()"가 두개	뒤에 있는 것을 "BeanFactoryAware의 setBeanFactory()"로 변경		
4	p.25	밑에서 두 번째 줄	본 애플리케이션 컨텍스트	둘러싼 애플리케이션 컨텍스트	(문맥 수정)	
5	p.26	1.3 스프링 현황 위 두 번째 줄	기능이 무엇인지 현대 스프링 현황을 조사해 보자	기능이 무엇인지 현재의 스프링 현황을 조사해 보자	(문맥 수정)	
6	p.44	코드 다음 문단 세 번째 줄	이제 스프링은 Juggler의 다른 생성자를 이용한다.	이제 스프링은 Juggler의 다른 생성자를 이용한다		
7	p.45	코드 2.2	<code>public PoeticJuggler(Poem poem) { super(); this.poem = poem; }  public PoeticJuggler(int beanBags, Poem poem) { super(beanBags); this.poem = poem; }</code>	<code>public PoeticJuggler(Poem poem) { super(); this.poem = poem; }  public PoeticJuggler(int beanBags, Poem poem) { super(beanBags); this.poem = poem; }</code>	(들어쓰기)	
8	p.45	코드 2.2	<code>System.out.println("While reciting...");</code>	<code>System.out.println("WHILE RECITING...");</code>	(대문자로 변경)	원서오류
9	p.47	마지막 문단	<b>JUGGLING 15 BEANBAGS WHILE RECITING... When, in disgrace with fortune and men's eyes, ...</b> ... ... ...  <b>with kings.</b>	<b>JUGGLING 15 BEANBAGS WHILE RECITING...</b> <b>When, in disgrace with fortune and men's eyes, I all alone bewep my outcast state</b> <b>And trouble deaf heaven with my bootless cries</b> <b>And look upon myself and curse my fate,</b> <b>Wishing me like to one more rich in hope,</b> <b>Featured like him, like him with friends possess'd,</b> <b>Desiring this man's art and that man's scope,</b> <b>With what I most enjoy contented least;</b> <b>Yet in these thoughts myself almost despising,</b> <b>Haply I think on thee, and then my state,</b> <b>Like to the lark at break of day arising</b> <b>From sullen earth, sings hymns at heaven's gate;</b> <b>For thy sweet love remember'd such wealth brings</b> <b>That then I scorn to change my state with kings.</b>	(개행 수정) System.out.println을 사용하므로 각각 새로운 줄에 출력됨	원서오류
10	p.53	코드 2.5	<code>private Instrument instrument;</code>  <code>public void setInstrument(Instrument instrument) {</code>	<code>private Instrument instrument;</code>  <code>public void setInstrument(Instrument instrument) {</code>	(들어쓰기)	
11	P.54	2.2.1 아래 세 번째 줄	세터 메소드 호출하여	세터 메소드 호출하여		
12	p.55	세 번째 문단 첫 번째 줄	잠시 동안 Instrumentalist 클래스가 int 타입의 age 프로퍼티가	잠시 동안 Instrumentalist 클래스에 int 타입의 age 프로퍼티가	(문맥수정)	
13	p.56	본문 아래에서 14번째 줄 (코드포함, 빈줄미포함)	<code>&lt;bean id="kenny2"</code>	<code>&lt;bean id="kenny"</code>		원서오류
14	p.61	코드 2.8	<code>private Collection&lt;Instrument&gt; instruments;</code>  <code>public void setInstruments(...</code>	<code>private Collection&lt;Instrument&gt; instruments;</code>  <code>public void setInstruments(...</code>	(들어쓰기)	
15	p.65	표 2.4	value-ref : 스프링 컨텍스트 내의 내에 있는	value-ref : 스프링 컨텍스트 내에 있는		
16	p.72	표 2.5	and, or, not, !	and, or, not, !		원서오류
17	p.75	표 2.7	not or !	not 또는 !		
18	p.75	밑에서 12번째 줄 (코드포함, 빈줄미포함)	<code>&lt;property name="outOfStock".../&gt;</code>	<code>&lt;property name="outOfStock".../&gt; (폰트 크기 작게)</code>	(폰트 수정)	
19	p.80	세 번째 문단	<code>^[]</code> 와 <code>.\$  </code> 도 제공한다. <code>^[]</code> 와 <code>.\$  </code> 는 ~	<code>^[]</code> 와 <code>.\$  </code> 도 제공한다. <code>^[]</code> 와 <code>.\$  </code> 는 ~	점(.) 추가	
20	p.85	위에서 첫 번째 코드의 첫 번째 줄	<code>&lt;bean id="kenny2" =</code>	<code>&lt;bean id="kenny"</code>		원서오류
21	p.86	역주1	상속 개념까지 고려한 개념이다	상속까지 고려한 개념이다	(문맥수정)	
22	p.145	위에서 세 번째 줄	<code>&lt;property name="criticisms"&gt;</code>	<code>&lt;property name="criticismPool "&gt;</code>		원서오류
23	p.155	역자한마디 (밑에서 네 번째 줄)	유일한 프레임워크는 스프링이 거의 유일하기 때문이다	프레임워크는 스프링이 거의 유일하기 때문이다	(문맥수정)	
24	p.169	역주4	<code>public void hello(String..., names){}</code>	<code>public void hello(String..., names){}</code>		

# 스프링 인 액션(3판, 1판1쇄) 수정사항

[1쇄에서 발견된 오타자 정보와 오류 사항 그리고 보다 매끄러운 문장을 위해 수정한 내용들입니다(2쇄 반영 내용). 불편을 끼쳐드려 죄송합니다.]

(1판1쇄)

(최종 수정일 : 2012.04.25)

연번	페이지	세부위치	1판 1쇄	1판 2쇄(수정사항)	설명	구분
25	p.170	다섯 번째 문단 두 번째 줄	<bean id="spitterDao" class="com.habuma.....SimpleJdbcTemplateSpitterDao"> <property name="jdbcTemplate" ref="jdbcTemplate" /> </bean>	<bean id="spitterDao" class="com.habuma.spitter.persistence.JdbcSpitterDAO"> <property name="jdbcTemplate" ref="jdbcTemplate" /> </bean>		원서오류
26	p.172	밑에서 세 번째 줄	<b>선택적으로</b> 명명된 파라미터를 사용할 수 있다.	<b>원한다면</b> 명명된 파라미터를 사용할 수 있다.	(문맥수정)	
27	p.178	역주6	getCurrentSession()	getCurrentSession()	(공백 제거)	
28	p.187	네 번째 문단 세 번째 줄	EntityManagerFactory	EntityManagerFactory	(대문자로 변경)	
29	p.187	밑에서 두 번째 문단 첫 번째 줄	어떤 컨테이너나 같은 곳에서 실행되는 환경이라면	<b>동일한 컨테이너 내에서</b> 실행되는 환경이라면	(문맥수정)	
30	p.187	밑에서 첫 번째 문단 첫 번째 줄	데이터 소스의 세부정보를	데이터 소스의 세부정보를		
31	p.205	코드 6.2	txTemplate.execute	transactionTemplate.execute		원서오류
32	p.205	코드 6.2	spitterDao.saveSpittle(spittle); try{ spitterDao.saveSpittle(spittle); } catch(RuntimeException){	spitterDao.saveSpittle(spittle); try{ <b>spitterDao.saveSpittle(spittle);</b> } catch(RuntimeException){	(들어쓰기)	원서오류
33	p.209	표 6.2 PROPAGATION_NESTED	본 트랜잭션(enclosing transaction)과 독립적으로 커밋되거나 롤백될 수 있다. 만약 본 트랜잭션이 없는 상황이라면	<b>동료</b> 트랜잭션(enclosing transaction)과 독립적으로 커밋되거나 롤백될 수 있다. 만약 <b>동료</b> 트랜잭션이 없는 상황이라면	(문맥수정)	
34	p.212	두 번째 문단 두 번째 줄	PROPAGATION_REQUIRED	PROPAGATION_REQUIRED		
35	p.214	밑에서 8 번째 줄 (코드포함, 빈줄미포함)	transactionmanager 애트리뷰트에	transaction-manager 애트리뷰트에		원서오류
36	p.215	네 번째 문단 두 번째 줄	스프링 2.0	스프링 3.0		원서오류
37	p.215	밑에서 첫 번째 문단과 두 번째 문단 사이	~따로 지정하면 조금은 더 길게 만들 수 있다.  더 이상 할 게 없다. 이 단 한줄의 XML이~	~따로 지정하면 조금은 더 길게 만들 수 있다.  <b>&lt;tx:annotation-driven transaction-manager="txManager"/&gt;</b>  더 이상 할 게 없다. 이 단 한줄의 XML이~	(누락)	
38	p.236	첫 번째 문단 세 번째 줄	spittle_form.jsp	spittleForm.jsp		
39	p.236	첫 번째 문단 세 번째 줄	signingsignup.jsp	signinsignup.jsp	g 제거	
40	p.241	코드 7.6	@RequestMapping("/spitter")	@RequestMapping("/spitters")	s 추가	원서오류
41	p.241	코드 7.6	/spitter/spittles	/spitters/spittles	s 추가	원서오류
42	p.249	세 번째문단 두 번째 줄	유요성 감사에 대해선 7.3.4 절	유요성 감사에 대해선 7.4.3 절		
43	p.252	코드 7.11	privateString password;	private String password;	(띄어쓰기)	
44	p.257	밑에서 세번째 줄	addSpitterFormForm	addSpitterFromForm		
45	p.267	밑에서 13 번째 줄 (코드포함, 빈줄미포함)	FlowHandlerMapping은 스프링 웹 플로우에 대한 흐름 요청을 직접 처리하는 반면, FlowHandlerAdapter는 해당 흐름 요청에 응답을 처리한다.	FlowHandlerMapping은 스프링 웹 플로우에 대한 흐름 요청을 <b>안내어</b> 는 반면, FlowHandlerAdapter는 <b>호출에 응답</b> 한다.		
46	p.276	그림 8.2	시작 -> 고객 식별-> 주문 -> 지불-> 주문 저장 -> 고객에 감사 -> 종료	Start-> identifyCustomer -> buildOrder -> takePayment-> saveOrder -> thankCustomer -> endState  <b>(그림과 메소드명을 일치시키는 것이 이해도가 높아 보임)</b>		
47	p.281	그림 8.3	환영 -> 고객 검색-> 등록 양식->배달 구역 확인 -> 고객추가, 배달 경로 고객 대기, 취소	welcome -> lookupCustomer -> registrationForm -> checkDeliveryArea -> addCustomer , deliveryWarning customerReady, cancel  <b>(그림과 메소드명을 일치시키는 것이 이해도가 높아 보임)</b>		
48	p.285	세 번째 문단 세 번째 줄	CustomerNotFoundException	CustomerNotFoundException이	(공백 제거)	
49	p.305	코드9.4	<spring:url var="authUrl"	<s:url var="authUrl"		원서오류
50	p.308	밑에서 두 번째 줄	<intercept- url>	<intercept-url>	(공백 제거)	
51	p.327	두 번째 문단 첫 번째 줄	@Secured 애너테이션의 단점 중 하나는	@Secured 애너테이션의 단점 중 하나는		
52	p.330	두 번째 문단 세 번째 줄	AccessDeniedException	AccessDeniedException이	(공백 제거)	
53	p.351	밑에서 다섯번째 줄	~ 빈의 ID와 클래스다. <b>그렇지 않으면</b> Burlap 서비스를 구성하는~	~ 빈의 ID와 클래스다. <b>이름 제외하면</b> Burlap 서비스를 구성하는~	(문맥수정)	
54	p.355	그림 10.8(그림)	DispatcherServlet, HttpInvokerExporter	DispatcherServlet, HttpInvokerServiceExporter		원서오류
55	p.355	그림 10.8(설명)	스프링 MVC DispatcherServlet	스프링 MVC DispatcherServlet	(공백 제거)	
56	p.358	박스 설명 위에서 다섯 번째 줄	JaxRpcPortProxyFactoryBean	JaxRpcPortProxyFactoryBean	(공백 제거)	

## 스프링 인 액션(3판, 1판1쇄) 수정사항

[1쇄에서 발견된 오타자 정보와 오류 사항 그리고 보다 매끄러운 문장을 위해 수정한 내용들입니다(2쇄 반영 내용). 불편을 끼쳐드려 죄송합니다.]

(1판1쇄)

(최종 수정일 : 2012.04.25)

연번	페이지	세부위치	1판 1쇄	1판 2쇄(수정사항)	설명	구분
57	p.383	위에서 11번째 줄 (빈줄미포함)	ContentnegotiatingViewResolver	ContentNegotiatingViewResolver	(대문자로 변경)	
58	p.392	위에서 9번째 줄 (빈줄미포함)	retrieveSpittlesForSpitter()	retrieveSpittlesForSpitter()	(공백 제거)	
59	p.400	위에서 8번째 줄 (빈줄미포함)	postForLocation()는 POST 요청의 바디에서	postForLocation()은 POST 요청의 바디에서		
60	p.420	코드 12.2 (밑에서 세번째 줄)	<pre>                     TextMessage textMessage = (TextMessage) message;                     System.out.println("GOT A MESSAGE: " + textMessage.getText());                     conn.start();                 } catch (JMSEException e) {             </pre>	<pre>                     TextMessage textMessage = (TextMessage) message;                     System.out.println("GOT A MESSAGE: " + textMessage.getText());                 } catch (JMSEException e) {             </pre>	(삭제)	원서오류
61	p.439	첫 번째 문단 세 번째 줄	바꾸고 싶은 경우가 JMX(Java Management Extensions)가 필요하다.	바꾸고 싶은 경우에 JMX(Java Management Extensions)가 필요하다.		
62	p.443	위에서 세번째 줄	Mbean(코드체)	Mbean(일반명문체)	(폰트 수정)	
63	p.444	두 번째 문단 다섯 번째 행	MethodNameBasedMBeanInfoAssemble	MethodNameBasedMBeanInfoAssembl <b>er</b>		
64	p.449	첫 번째 문단 두 번째 줄	관리 빈으로 애트리뷰트로	관리 빈의 애트리뷰트로		
65	p.465	다섯 번째 문단 첫 번째 줄	defaultConfiguration	defaultConfiguration		
66	p.465	여섯 번째 문단 두 번째 줄	<component:property-placeholder>	<context:property-placeholder>		원서오류
67	p.465	밑에서 첫 번째 문단 첫 번째 줄	<component:property-placeholder>	<context:property-placeholder>		원서오류
68	p.466	두 번째 문단 첫 번째 줄	<component:property-placeholder>	<context:property-placeholder>		원서오류
69	p.466	밑에서 두 번째 문단 두 번째 줄	<component:property-placeholder>	<context:property-placeholder>		원서오류
70	p.466	밑에서 두 번째 문단 두 번째 줄	<component:property-override>	<context:property-override>		원서오류
71	p.468	두 번째 문단 네 번째 줄	Property OverrideConfigurer를	PropertyOverrideConfigurer를	(공백 제거)	
72	p.486	네 번째 문단	@Autowired VelocityEngine velocityEngine;	@Autowired VelocityEngine velocityEngine;	(내어쓰기)	원서오류
73	p.490	세 번째 문단 세 번째 줄	fixed Delay	fixedDelay	(공백 제거)	

### <찾아보기>

1	p.495	(C) 위에서 여섯 번째 줄	<component:property-override>	<context:property-override>		원서오류
2	p.503	(T) 밑에서 여덟 번째 줄	transactionmanager 애트리뷰트	transaction-manager 애트리뷰트		원서오류