## 표지 앞날개

|  |  |
| --- | --- |
| 기존 | (제목)남가람북스 자료실 [www.namgarambooks.co.kr](http://www.namgarambooks.co.kr) |
| 수정 | 자료실  책예제 코드 https://github.com/zerockcode  남가람북스 www.namgarambooks.co.kr |

## P29

기존: 만일 위의 질문에 ‘yes’라고 대답한다면 ‘Spring Project’

수정: 만일 위의 질문에 ‘yes’라고 대답한다면 ‘Spring Legacy Project’

## P30

기존: 도표에 Spring Project

수정: Spring Legacy Project

## P31

P33처럼 NOTE추가

책의 모든 예제는 Maven이라는 빌드 도구를 이용해서 필요한 jar파일들과 프로젝트를 구성하게 됩니다. 프로젝트가 최초로 구성되면 필요한 라이브러리들을 다운로드 받는데, 몇 분의 시간이 필요합니다. 가끔은 네트워크상의 문제로 인해 구성시에 에러가 나는 경우가 있습니다.

이럴 경우에는 P35의 그림처럼 STS종료후에 ‘.m2’폴더의 하위 폴더들을 삭제한 후 재시작 하는 방법이나 프로젝트의 삭제 후 다시 생성하는 방법 등을 이용하여 다시 다운로드를 시도하는 것이 좋습니다.

## P41

하단의 그림 바로 위의 문장

|  |  |
| --- | --- |
| 기존 | (2015년 7월 현재 공식 버전은 4.1.7버전입니다.) |
| 수정 | (2017년 1월 현재 공식 버전은 4.3.5버전입니다. 책의 예제는 4.3.3버전을 기준으로 작성되었습니다.) |

## P 43

1.4.5 문장 중

(2016년 11월 현재 Tomcat은 9버전까지 배포되었습니다.)

대신에

(책의 예제는 Tomcat8 혹은 Tomcat9 버전을 이용하도록 합니다)

## P48

|  |  |
| --- | --- |
| 기존 | Tomcat8을 서버 설정에 추가하였습니다. |
| 수정 | Tomcat9 버전의 서버를 설정하였습니다. |

## P 59 추가될 내용

2017년 1월 현재 MySQL은 5.7버전을 다운로드 받을 수 있으므로 책의 예제 역시 이를 기준으로 작성합니다(만일 MySQL 5.6이하를 이용하는 경우는 약간의 설정에 차이가 있을 수 있습니다.).

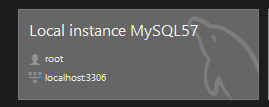
## P63

하단의 문장 내용

|  |  |
| --- | --- |
| 기존 | 이 단계를 지나면 모든 설치가 완료되고, 윈도우에 ‘MySQL56’ … |
| 수정 | 이 단계를 거치면 모든 설치가 완료되고, 윈도우에 ‘MySQL57’ … |

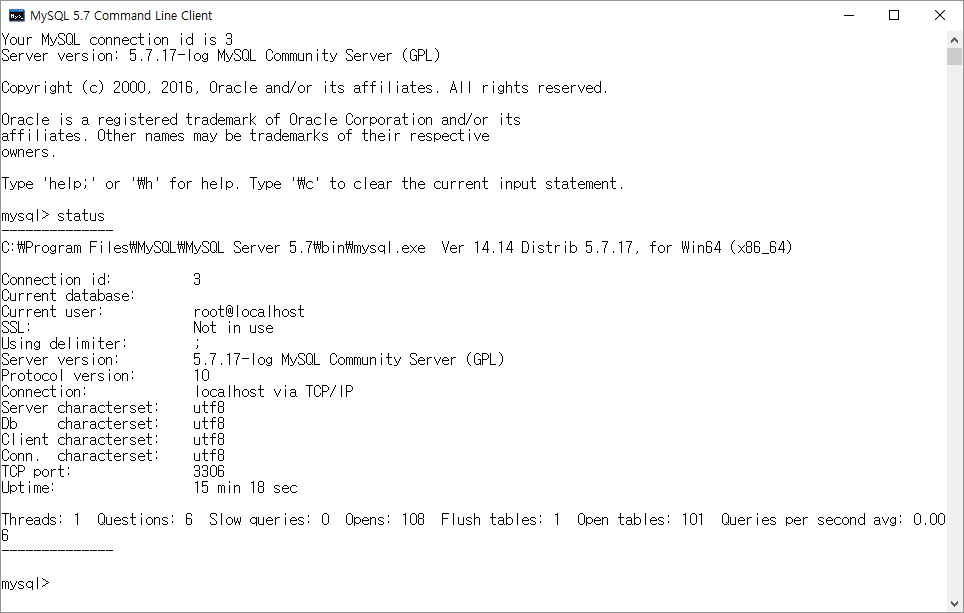
## P65

상단의 그림 변경



## P70

상단 그림 변경



## p72

상단 pom.xml 추가 박스 아래쪽에 추가할 내용

|  |  |
| --- | --- |
| 기존 | 라이브러리의 다운로드가 진행된 후에는 JDBC코딩을 통해서 실제 Java와의 연동 작업을 테스트해 봅니다. |
| 수정 | 2017년 1월 현재 MySQL사이트의 공식 버전은 5.1.40이지만, 책의 예제에는 Development Release버전인 6.0.5버전을 이용할 수 있습니다.  현재 사용하는 MySQL의 버전이 5.6이하의 경우는 5.x의 버전만을 이용할 수 있으므로 주의해야 합니다.  라이브러리의 다운로드가 진행된 후에는 JDBC코딩을 통해서 실제 Java와의 연동 작업을 테스트해 봅니다. |

## P74

하단의 코드에서 생략된 부분이 있었으므로 대체

|  |
| --- |
| package org.zerock.web;  import java.sql.Connection;  import java.sql.DriverManager;  import org.junit.Test;  public class MySQLConnectionTest {  private static final String DRIVER =  //MySQL Driver 5 이하  //"com.mysql.jdbc.Driver"  //MySQL Driver 6 이상  "com.mysql.cj.jdbc.Driver";  private static final String URL =  //MySQL 5.6 이하  //"jdbc:mysql://127.0.0.1:3306/book\_ex";  //MySQL 5.7 이상  "jdbc:mysql://127.0.0.1:3306/book\_ex?useSSL=false&serverTimezone=UTC";    private static final String USER =  "zerock";  private static final String PW =  "zerock"; |

## P75

코드 아래쪽의 설명 추가

|  |  |
| --- | --- |
| 기존 | URL 설정 부분은 뒤에 생성된 스키마를 이용합니다. |
| 수정 | URL 설정 부분은 뒤에 생성된 스키마를 이용합니다. MySQL 6.x버전의 경우 시간 설정과 SSL연결 설정을 추가합니다. 위의 코드의 경우 UTC는 세계 표준 시간 설정이고, 서울 시간을 이용할 경우에는 UTC대신 Asia/Seoul을 지정합니다. |

## P86

표의 내용 수정

|  |
| --- |
| <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">  <property name="driverClassName" value="com.mysql.cj.jdbc.Driver"></property>  <property name="url" value="jdbc:mysql://127.0.0.1:3306/book\_ex?useSSL=false&amp;serverTimezone=Asia/Seoul"></property>  <property name="username" value="zerock"></property>  <property name="password" value="zerock"></property>  </bean> |

## P126

tbl\_member 테이블 SQL 아래쪽에 추가

|  |
| --- |
| 위의 SQL문에서 default now( )의 경우 MySQL5.6이상의 버전부터 지원되므로, 이전 버전의 MySQL의 경우에는 사용할 수 없으니 주의해야 합니다. |

## P138

6.7 아래 문장

......로그를 설정해 주는 것이 좋습니다.

문장 뒤에 약간의 내용 추가

로그를 설정해 주는 것이 좋습니다(log4jdbc를 이용하는 경우 속도가 기존보다 저하될 수 있고, 데이터베이스에 따라서 지원되지 않는 경우도 있으므로 설정 후에 정상적으로 동작하는지 반드시 테스트를 해 주어야 합니다.).

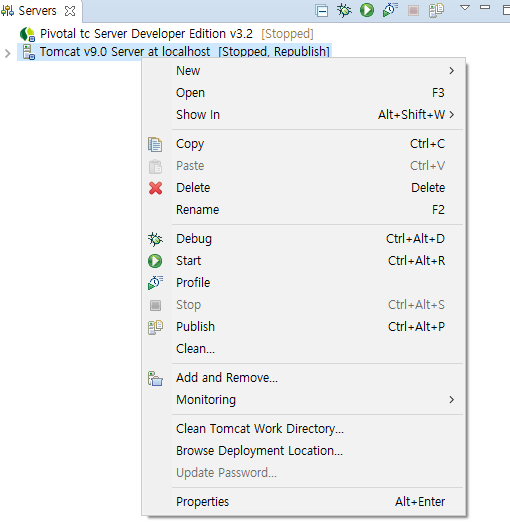
## P182

상단 코드에 내용 변경

|  |  |
| --- | --- |
| 기존 |  |
| 수정 | @RunWith(SpringJUnit4ClassRunner.**class**)  @ContextConfiguration(locations = { "file:src/main/webapp/WEB-INF/spring/\*\*/root-context.xml" })  **public** **class** BoardDAOTest {  @Inject  **private** BoardDAO dao;  **private** **static** Logger *logger* = LoggerFactory.*getLogger*(BoardDAOTest.**class**); |

## P197

이미지 교체



## P 210

예제 코드 내에서 오타 (s가 하나 더 들어갔음)

|  |  |
| --- | --- |
| 기존 | rttr.addFlashAttributes(“msg”, “SUCCESS”); |
| 수정 | rttr.addFlashAttribute(“msg”, “SUCCESS”); |

## P268

|  |  |
| --- | --- |
| 기존 | 13/10(perPageNum) = 1.3 |
| 수정 | 13/10(displayPageNum) = 1.3 |

## P 289

상단에 설명 추가

|  |  |
| --- | --- |
| 기존 | 페이지 내에서... (중간 생략).. 처리할 수 있다는 것입니다. |
| 수정 | 페이지 내에서 링크를 처리하는 또 다른 팁은 javaScript를 이용해서 페이지 번호를 클릭했을 때의 이벤트를 제어하는 방식입니다.  이를 위해서는 이전 페이지의 코드를 적용하여 아래의 그림처럼 페이지 번호를 의미하는 <a>태그의 href 속성들이 단순히 페이지 번호만을 의미하도록 만들어 주어야 합니다. |

## P 312 ~ 313

P312하단 ~ P313페이지 소스코드

|  |
| --- |
| public String makeSearch(int page) {  UriComponents uriComponents =  UriComponentsBuilder.newInstance().queryParam("page", page)  .queryParam("perPageNum", cri.getPerPageNum())  .queryParam("searchType", ((SearchCriteria) cri).getSearchType())  .queryParam("keyword", encoding(((SearchCriteria) cri).getKeyword())).build();  return uriComponents.toUriString();  }  private String encoding(String keyword) {  if (keyword == null || keyword.trim().length() == 0) {  return "";  }  try {  return URLEncoder.encode(keyword, "UTF-8");  } catch (UnsupportedEncodingException e) {  return "";  }  } |

## P 319

소스 코드 encodeURIComponent( ) 추가

|  |
| --- |
| <script>  $(document).ready(  function() {  $('#searchBtn').on(  "click",  function(event) {  self.location = "list"  + '${pageMaker.makeQuery(1)}'  + "&searchType="  + $("select option:selected").val()  + "&keyword=" + encodeURIComponent($('#keywordInput').val());  });  $('#newBtn').on("click", function(evt) {  self.location = "register";  });  });  </script> |