



**Lib Service**

**도서관 관리 프로그램**

# 프로필

---



**이 상 훈**

010 2245 2048

bbubbush@naver.com

## 학력사항

강남대학교 경영학과(2016.08졸업)

## 자격사항

정보처리기사

컴퓨터활용능력 1급

운전면허 1종 보통

## 교육사항

2016. 12. 12 ~ 2017. 06. 26

KH 정보교육원 자바 과정 수료

2014. 10. 20 ~ 2015. 03. 03

한국 IT 비즈니스 진흥협회 빅데이터 과정 수료

# 개발환경

---



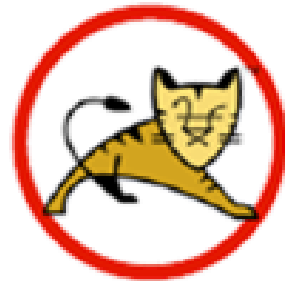
JDK 1.8.0\_121



Oracle Database 11.2.0.1.0



HTML 5



Apache Tomcat 7.0.75

# [일반도서 검색 기능]

- 상세 항목에 따른 쿼리를 작성하여 주입
- 검색어에 따라 페이징 적용

Home

도서관 소개 ▾

일반도서 검색

전자도서관 ▾

이용예약 ▾

평생교육

고객지원서비스 ▾

일반도서 검색

철학 +

종교 +

인문과학 +

자연과학 +

예술 +

언어 +

문학 +

역사 +

Search



철학 ▾

모두보기 ▾

저자

출판사



자료검색



ETS TOEIC Test 공식문제집 RC

글쓴이  
출판사

ETS  
YBM

```
37 @RequestMapping(value="/bkDetailSearch.ju")
38 public ModelAndView bkDetailSearch(@RequestParam(value="bk_lg", defaultValue="99")int bk_lg,
39     @RequestParam(value="bk_md", defaultValue="99")int bk_md,
40     @RequestParam(value="query", defaultValue="")String query,
41     @RequestParam(value="bk_writer", defaultValue="")String bk_writer,
42     @RequestParam(value="bk_publisher", defaultValue="")String bk_publisher,
43     @RequestParam(value="page", defaultValue="1")int page){
44     if( !("").equals(query)) && query.length() >=2 ){
45         log.info(query+"/");
46     }
47     boolean chk[] = {false, false, false, false, false};
48     String chkQuery[] = {"", "", "", "", ""};
49     String subQuery = "";
50     int chkCount = 0;
51     if(bk_lg!=99){
52         chkCount++;
53         chk[0] = true;
54         chkQuery[0] += "bk_lg = " + bk_lg;
55     }
56     if(bk_md!=99){
57         chkCount++;
58         chk[1] = true;
59         chkQuery[1] += "bk_md = " + bk_md;
60     }
61     if( !("").equals(bk_writer)) ){
62         chkCount++;
```



```
76
77 // 5개중 하나라도 값이 있다면
78 if(chkCount>0){
79     subQuery += "WHERE ";
80     for (int i = 0; i < chk.length; i++) {
81         if( (chkCount!=1) && chk[i] ){
82             subQuery += chkQuery[i]+ " and ";
83             chkCount--;
84         }else if( (chkCount==1) && chk[i] ){
85             subQuery += chkQuery[i];
86         }
87     }
88 }
89
90 //신쿼리
91 String sql = "SELECT X.rnum, X.bk_subject, X.bk_writer, X.bk_publisher, X.bk_url, X.bk_info FROM "
92             + "(SELECT rownum AS rnum, A.bk_subject, A.bk_writer, A.bk_publisher, A.bk_url, A.bk_info FROM "
93             + "(SELECT bk_subject, bk_writer, bk_publisher, bk_url, bk_info FROM book " + subQuery
94             + " group by bk_subject, bk_writer, bk_publisher, bk_url, bk_info ORDER BY bk_subject)A "
95             + "WHERE rownum <= " + (page*COUNTLIST) + ") X WHERE X.rnum >= " + ((page-1)*COUNTLIST+1);
96 String totalPagesSql = "SELECT sum(count(distinct bk_subject)) FROM book " + subQuery
97             + " group by bk_subject";
98 int totalCount = FindDao.totalPages(totalPagesSql);
99 int totalPage = totalCount/COUNTLIST;
100 if(totalCount % COUNTLIST > 0){
101     totalPage++;

```

```
99     int totalPages = totalCount/COUNTLIST;
100     if(totalCount % COUNTLIST > 0){
101         totalPages++;
102     }
103
104     if(page>totalPages && totalPages!=0){
105         page = totalPages;
106     }else if(page < 1){
107         page = 1;
108     }
109
110     int startPage = ((page-1)/10)*10+1;
111     int endPage = startPage + COUNTPAGE -1;
112     if(endPage > totalPages){
113         endPage = totalPages;
114     }
115
116     ModelAndView mav = new ModelAndView();
117     List<FindDTO> list = FindDao.bkList(sql);
118     mav.addObject("startPage", startPage);
119     mav.addObject("endPage", endPage);
120     mav.addObject("page", page);
121     mav.addObject("list", list);
122     mav.setViewName("juJson");
123     return mav;
124 }
```



# [일반도서 정보 보기]

- Ajax를 통한 페이지 처리
- 모달을 사용하여 깔끔한 UI 구현

일반도서 검색

철학

+

종교

+

인문과학

+

자연과학

+

예술

+

언어

+

문학

+

역사

+

## 도서 정보

×



### ETS TOEIC Test 공식문제집 RC

저자 ETS  
출판사 YBM  
출판일 2017-05-31  
보유권수 1  
예약가능권수 1

대출여부확인

☆예약하기

택배대출

#### 상세정보

ETS가 최신 신유형 경향을 반영한 실전문제 5세트를 개발, 독점 제공한다. 이 책의 모든 문항은 토익 출제기관인 ETS가 정기시험과 동일한 유형 및 난이도로 개발하여 정기시험과 100% 동일한 문제 퀄리티와 난이도를 자랑한다. 모든 문제에 대해 답이 되는 이유뿐만 아니라 오답 보기까지 분석한 해설을...

목록으로



출판사 YBM  
간략소개  
ETS가 최신 신유형 경...



### THE ART OF Disney 디즈니 골든 에이지 포스트카드 컬렉션 100

글쓴이 Disney Enterprises|INC.  
출판사 아르누보  
간략소개  
다양한 디즈니 명작을

서영주님 환영합니다

로그아웃

고객지원서비스

```
21 function bkView() {
22     $('#tbody>tr>td').click(function () {
23         result = 0;
24         result2 = 0;
25         var selectNum=$("#tbody>tr>td").index(this);
26         var bk_subject=$("#tbody>tr>td").eq(selectNum).data("subject");
27         $('#mbody').html("");
28         $.ajax({
29             type: "GET",
30             url: "bkView.ju",
31             data: {'bk_subject' : bk_subject},
32             dataType: "json",
33             success: function(list){
34                 var modalBody = "";
35                 var modalFooter = "";
36                 modalBody = '<div class="media-left media">'
37                 +''
39                 +'</div>'
40                 +'<div class="media-body">'
41                 +'<h4 class="media-heading" id="media-heading">'+list.list.bk_subject+'</h4>'
42                 +'<div class="row-fluid">'
43                 +'<div class="col-xs-3">저자</div>'
44                 +'<div class="col-xs-9">'+list.list.bk_writer+'</div>'
45                 +'<div class="col-xs-3">출판사</div>'
46                 +'<div class="col-xs-9">'+list.list.bk_publisher+'</div>'
```

```
54     +'    <button class="btn btn-default" id="refreshSpan_1" onClick="bkRefresh()">'
55     +'        <span class="glyphicon glyphicon-refresh"'
56     +'            aria-hidden="true"></span><span>대출여부확인</span>'
57     +'    </button>'
58     +'    <button class="btn btn-default" onClick="bkYeyak()">'
59     +'        <span class="glyphicon glyphicon-star-empty"'
60     +'            aria-hidden="true"></span><span id="yeyak_1">예약하기</span>'
61     +'</button>'
62     +'<button class="btn btn-default" onClick="bkFedex()">'
63     +'    <span class="glyphicon glyphicon-gift"'
64     +'        aria-hidden="true"></span><span id="fedex_1">택배대출</span>'
65     +'    </button>'
66     +'</div>'
67     +'</div>'
68     +'</div>';
69
70     modalFooter=list.list.bk_info
71
72     $("#modal-body").html(modalBody);
73     $("#info").html(modalFooter);
74     }
75
76     });
77     $("#myModal").modal("show");
78 })
79 }
```

# ISBN으로 도서정보얻기

- 네이버 도서 API를 통해 도서 정보 가져오기
- XML 형태로 가져온 데이터 파싱하여 사용

관리자

회원관리 +

전체도서 관리 x

일반도서 관리 x

일반도서 리스트

일반도서 등록

전자도서 관리 +

대출 관리 +

택배대출 관리

평생교육 관리 +

이용시설 관리

캘린더 관리

## 일반도서 등록

SEARCH

### 유의사항

1. 꼭 ISBN만 입력해주시기 바랍니다.
2. 바코드기를 통해 입력하는 방법이 유용합니다.



관리자

회원관리 +

전체도서 관리 ×

일반도서 관리 ×

일반도서 리스트

일반도서 등록

전자도서 관리 +

대출 관리 +

택배대출 관리

평생교육 관리 +

이용시설 관리

캘린더 관리

## 책 등록 페이지



ISBN 9788994120966

도서명 지적 대화를 위한 넓고 얇은 지식

저자명 채사장

출판사명 한빛비즈

출판일

분류 전체 ▼

**책 설명** 힘 있는 지식인이 되기 위한 필수 기초 교양! 어김없이 오늘도 우리는 사람들과 대화를 나눈다. 어제 본 드라마부터 시작되는 대화는 늘 같은 패턴이지만, 오늘따라 왜인지 시시한 기분이 든다. 곧 색다른 주제의 대화를 이어가려 하지만 금방 지식에 한계가 오는 듯하다. 문득 자신의 부족한 지식수준을 채우기...

책 등록

```
212 // 책 검색해서 정보 뽑아오기
213 @RequestMapping("/bookAddForm.ju")
214 public ModelAndView mookAddForm(String query) throws Exception{
215
216     ModelAndView mav = new ModelAndView();
217     // 전달할 변수
218     String title = null;
219     String image = null;
220     String description = null;
221     String author = null;
222     String publisher = null;
223     String isbn = null;
224
225     //유니코드를 쿼리로 검색
226     isbn = query;
227     //java코드로 특정 url에 요청보내고 응답받기
228     String clientID="qkgP2gHuMg16GcvTN_aB"; //네이버 개발자 센터에서 발급받은 clientID입력
229     String clientSecret = "MqzAe52lXP"; //네이버 개발자 센터에서 발급받은 clientSecret입력
230     URL url = new URL("https://openapi.naver.com/v1/search/book.xml?query="+query); //API 기본정보의 요청 url
231
232     URLConnection urlConn=url.openConnection(); //openConnection 해당 요청에 대해서 쓸 수 있는 connection 객체
233
234     urlConn.setRequestProperty("X-Naver-Client-ID", clientID);
235     urlConn.setRequestProperty("X-Naver-Client-Secret", clientSecret);
236
237     BufferedReader br = new BufferedReader(new InputStreamReader(urlConn.getInputStream()));
```

```
239     String msg = null;
240     if((msg = br.readLine()) != null) {
241
242         /* data parsing */
243         // title
244         String regex = "<item><title>(\\w|\\W)+(?=<\\/title>)";
245         Pattern pattern = Pattern.compile(regex);
246         Matcher match = pattern.matcher(msg);
247         if(match.find()){
248             String m = match.group(0);
249             String ms[] = m.split("<item><title>");
250             title = ms[1];
251         }
252         String[] result = title.split("\\(");
253         for(int i=0; i<result.length; i++){
254             title = result[0];
255         }
256         // image
257         regex = "<image>(\\w|\\W)+.jpg";
258         pattern = Pattern.compile(regex);
259         match = pattern.matcher(msg);
260         if(match.find()){
261             String m = match.group(0);
262             String ms[] = m.split("<image>");
263             image = ms[1];
264         }
```

```
281         String ms[] = m.split("<publisher>");
282         publisher = ms[1];
283     }
284
285     //description
286     regex = "<\\//isbn><description>(\\w|\\W)+(?=<\\//description><\\//item>)";
287     pattern = Pattern.compile(regex);
288     match = pattern.matcher(msg);
289     if(match.find()){
290         String m = match.group(0);
291         String ms[] = m.split("</isbn><description>");
292         description = ms[1];
293     }
294
295     mav.addObject("title",title);
296     mav.addObject("image",image);
297     mav.addObject("author",author);
298     mav.addObject("publisher",publisher);
299     mav.addObject("isbn",isbn);
300     mav.addObject("description",description);
301
302     mav.setViewName("admin/bookManage/bookAddForm");
303
304     }// end if
305     return mav;
306 }//end main
```

# **Log를 활용한 검색어 통계**

- **Log4j로 얻은 검색어 기록을 토대로 실시간 분석**
- **데이터 시각화를 위해 버블차트 사용**

[도서관 소개](#)[도서관장 인사말](#)[도서관 연혁](#)[총별소개](#)[오시는길](#)[장서현황](#)[이용안내](#)[도서관통계](#)

## 도서관 통계

[검색어통계](#)[이용자통계](#)



```
17 @RequestMapping(value="/analSearch.ju")
18 public ModelAndView analysis() {
19     // 검색어 빈도
20     File f = new File("F:/LOG/book/searchList.log");
21     Map<String, Integer> map = new HashMap<String, Integer>();
22     ValueComparator bvc = new ValueComparator(map);
23     TreeMap<String,Integer> sorted_map = new TreeMap<String,Integer>(bvc);
24
25     if(f.exists()){
26         try {
27             BufferedReader in = new BufferedReader(new FileReader(f));
28             String s;
29             while ((s = in.readLine()) != null) {
30                 String arr[] = s.split("/");
31                 for (int i = 0; i < arr.length; i++) {
32                     if(map.get(arr[i])==null){
33                         map.put(arr[i], 1);
34                     }else{
35                         int count = map.get(arr[i])+1;
36                         map.put(arr[i], count);
37                     }
38                 }
39             }
40             in.close();
41         } catch (Exception e) {
42             e.printStackTrace();
43         }
44     }
```

```
40         in.close();
41     } catch (Exception e) {
42         e.printStackTrace();
43     }
44 }
45 sorted_map.putAll(map);
46
47 String json = "[";
48 int count = 0;
49 for (Map.Entry<String,Integer> entry : sorted_map.entrySet()) {
50     //정렬한 리스트에서 순번을 배열번호로 변경하여 원본 리스트에서 추출
51     count++;
52     if(count>7){
53         break;
54     }else{
55         json += "{text:'" + entry.getKey() + "', count:'" + map.get(entry.getKey()) + "'},";
56     }
57 }
58 json = json.substring(0, json.length()-1);
59 json += "]";
60
61 ModelAndView mav = new ModelAndView("/anal/analMain");
62 mav.addObject("list", json);
63 return mav;
64 }
65
```

# Log와 ip 대역으로 접속국가 파악

- 로그인시 기록된 ip를 저장하여 국가별 ip 대역이 기록된 csv파일로 비교
- 해외 IP는 ip우회를 통해 기록



마이크 ●

서영주님 환영합니다

로그아웃

Home

도서관 소개 ▾

일반도서 검색

전자도서관 ▾

이용예약 ▾

평생교육

고객지원서비스 ▾

마이 페이지

도서관 카드

대출/예약

전자도서

회원정보 수정

나의 문의

로그인정보

회원 탈퇴

## 로그인 정보 확인

접속일	접속IP주소	접속국가
2017-06-22 10:12:39	5.62.56.160	북한
2017-06-23 08:44:39	92.223.73.0	대한민국
2017-06-24 20:36:39	5.62.56.162	북한

```
533 @RequestMapping(value="/loginLog.ju")
534 public ModelAndView loginLog(HttpSession session){
535     List<String> totalIp = new ArrayList<String>();
536     List<String> totalDate = new ArrayList<String>();
537     List<String> totalCty = new ArrayList<String>();
538     if(session.getAttribute("sid") != null){
539         String sid = (String)session.getAttribute("sid");
540
541         File f = new File("F:/LOG/member/memberInfo.log");
542         String country = "";
543         if(f.exists()){
544             try {
545                 BufferedReader in = new BufferedReader(new FileReader(f));
546                 String s;
547                 while ((s = in.readLine()) != null) {
548                     if(s.contains("login") && s.contains(sid)){
549                         String arr[] = s.split("login");
550                         String dayArr[] = arr[0].split(",");
551                         //접속 일시
552                         String day = dayArr[0];
553                         arr = arr[1].split(",");
554                         //접속 IP
555                         String ip = arr[0].substring(4);
556                         //IP 국가
557                         country = getGlobalIp(ip);
```

```
552     String day = dayArr[0];
553     arr = arr[1].split(",");
554     //접속 IP
555     String ip = arr[0].substring(4);
556     //IP 국가
557     country = getGlobalIp(ip);
558
559     if( "KR".equals(country) ){
560         country = "대한민국";
561     }else if( "KP".equals(country) ){
562         country = "북한";
563     }else{
564         country = "그 외 국가";
565     }
566
567     totalDate.add(day);
568     totalIp.add(ip);
569     totalCty.add(country);
570 }
571 }
572 in.close();
573 } catch (Exception e) {
574     e.printStackTrace();
575 }
576 } // end if
577 } // End loginLog
```



```
587 public String getGlobalIp(String ipAddress){
588     try {
589         File f = new File("F:/LOG/ip/ip.csv");
590
591         BufferedReader in = new BufferedReader(new FileReader(f));
592         String s;
593         while( (s=in.readLine()) != null ){
594             String arr[] = s.split(",");
595             long min = 0;
596             long max = 0;
597             long ipAddr = 0;
598             String minIpArr[] = arr[0].split("\\.");
599             String maxIpArr[] = arr[1].split("\\.");
600             String ipArr[] = ipAddress.split("\\.");
601
602             for (int i = 0; i < ipArr.length; i++) {
603                 int power = 3-i;
604                 int ip = Integer.parseInt(ipArr[i]);
605                 ipAddr += ip * Math.pow(256, power);
606             }
607             for (int i = 0; i < minIpArr.length; i++) {
608                 int power = 3-i;
609                 int ip = Integer.parseInt(minIpArr[i]);
610                 min += ip * Math.pow(256, power);
611             }
612             for (int i = 0; i < maxIpArr.length; i++) {
```

```
603         int power = 3-i;
604         int ip = Integer.parseInt(ipArr[i]);
605         ipAddr += ip * Math.pow(256, power);
606     }
607     for (int i = 0; i < minIpArr.length; i++) {
608         int power = 3-i;
609         int ip = Integer.parseInt(minIpArr[i]);
610         min += ip * Math.pow(256, power);
611     }
612     for (int i = 0; i < maxIpArr.length; i++) {
613         int power = 3-i;
614         int ip = Integer.parseInt(maxIpArr[i]);
615         max += ip * Math.pow(256, power);
616     }
617     for (long i = min; i <= max; i++) {
618         if(i==ipAddr){
619             return arr[2];
620         }
621     }
622     } //end while
623 in.close();
624 } catch (Exception e) {
625     e.printStackTrace();
626 }
627 return "Etc";
628 } // end method
```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	1.0.0.0	1.0.0.255	16777216	16777471	AU	Australia							
2	1.0.1.0	1.0.3.255	16777472	16778239	CN	China							
3	1.0.4.0	1.0.7.255	16778240	16779263	AU	Australia							
4	1.0.8.0	1.0.15.255	16779264	16781311	CN	China							
5	1.0.16.0	1.0.31.255	16781312	16785407	JP	Japan							
6	1.0.32.0	1.0.63.255	16785408	16793599	CN	China							
7	1.0.64.0	1.0.127.255	16793600	16809983	JP	Japan							
8	1.0.128.0	1.0.255.255	16809984	16842751	TH	Thailand							
9	1.1.0.0	1.1.0.255	16842752	16843007	CN	China							
10	1.1.1.0	1.1.1.255	16843008	16843263	AU	Australia							
11	1.1.2.0	1.1.63.255	16843264	16859135	CN	China							
12	1.1.64.0	1.1.127.255	16859136	16875519	JP	Japan							
13	1.1.128.0	1.1.255.255	16875520	16908287	TH	Thailand							
14	1.2.0.0	1.2.2.255	16908288	16909055	CN	China							
15	1.2.3.0	1.2.3.255	16909056	16909311	US	United States							
16	1.2.4.0	1.2.127.255	16909312	16941055	CN	China							
17	1.2.128.0	1.2.255.255	16941056	16973823	TH	Thailand							
18	1.3.0.0	1.3.255.255	16973824	17039359	CN	China							
19	1.4.0.0	1.4.0.255	17039360	17039615	AU	Australia							
20	1.4.1.0	1.4.127.255	17039616	17072127	CN	China							
21	1.4.128.0	1.4.255.255	17072128	17104895	TH	Thailand							
22	1.5.0.0	1.5.255.255	17104896	17170431	JP	Japan							
23	1.6.0.0	1.7.255.255	17170432	17301503	IN	India							
24	1.8.0.0	1.8.255.255	17301504	17367039	CN	China							

해외 국가 IP 대역을

기록한 CSV파일

