搜索功能详解

搜索功能主要是采用了模糊搜索的模式，用户输入关键字，后端去数据库中找到含有关键字的相关文章，并返回。

先从前端讲起。

页面样式采用，按下回车后进行搜索。
所以js代码如下：

```
document.onkeydown = function(e){
            if(!e) e = window.event;//火狐中是 window.event
            if((e.keyCode || e.which) == 13){
                window.event ? window.event.returnValue = false : e.preventDefault();
                var key=document.getElementById("searchKey").value;
                if(key!=''){
                    locationUrl('/search/'+key,"");
                }
            }
        }
```

key是输入的关键字，我们将这个关键字发送到后端进行请求。

```
@RequestMapping("search/{key}")
    //根据关键字查询
    public ModelAndView search(@PathVariable("key") String key,HttpServletRequest request,@RequestParam(value = "page", defaultValue = "0") Integer page,
                               @RequestParam(value = "size", defaultValue = "25") Integer size){
                               //同样采用分页的方式，这个规定搜索只给出与关键字最相似的25项。
        Page<Topic> pages = topicServiceImp.getTopicBySearch(page,size,key);
        User user = getSessionUser(request);
        ModelAndView mv = new ModelAndView();
        mv.setViewName("user/search");
        mv.addObject("user",user);
        List<Topic> topics2 = new ArrayList<>();
        for(Topic topic:pages) {
            topic.setUser(userServiceImp.getUserById(topic.getUserId()));
            topic.setBoard(boardServiceImp.findBoardById(topic.getBoardId()));
            topic.setUrl("/userReadDetail/"+topic.getTopicId());
            topics2.add(topic);
        }
        mv.addObject("topics",topics2);
        return  mv;
    }
```

后端映射其实不是重点，重点是介绍如何进行模糊查询。

```
@Override
    public Page<Topic> getTopicBySearch(int pageNum, int PageSize, String key) {
        key = "%"+key+"%";
        //利用占位符 "%" 即可实现
        Pageable pageable = PageRequest.of(pageNum,PageSize,Sort.by(Sort.Direction.DESC,"createTime"));
        return topicDao.findByTopicTitleLike(key,pageable);
    }


Page<Topic> findByTopicTitleLike(String topicTitle,Pageable pageable);
//Dao层需要用like
```