

JancoBlog 个人博客管理系统 V1.0

源程序

```
package com.Jancoyan.domain;
import java.util.Date;
public class User {
    private Integer userId;
    private String userNickname;
    private String userName;
    private String userPwd;
    private String userEmail;
    private Integer userSex;
    private Integer userRole;
    private Date userCreateDate;
    private String userProfile;
    private Date userBirthday;
    private String userIp;
    public Integer getUserId() {
        return userId;
    }
    public void setUserId(Integer userId) {
        this.userId = userId;
    }
    public String getUserNickname() {
        return userNickname;
    }
    public void setUserNickname(String userNickname) {
        this.userNickname = userNickname == null ? null : userNickname.trim();
    }
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName == null ? null : userName.trim();
    }
    public String getUserPwd() {
        return userPwd;
    }
    public void setUserPwd(String userPwd) {
        this.userPwd = userPwd == null ? null : userPwd.trim();
    }
    public String getUserEmail() {
        return userEmail;
    }
    public void setUserEmail(String userEmail) {
        this.userEmail = userEmail == null ? null : userEmail.trim();
    }
    public Integer getUserSex() {
        return userSex;
    }
    public void setUserSex(Integer userSex) {
        this.userSex = userSex;
    }
}
```

```
public Integer getUserRole() {
    return userRole;
}
public void setUserRole(Integer userRole) {
    this.userRole = userRole;
}
public Date getUserCreateDate() {
    return userCreateDate;
}
public void setUserCreateDate(Date userCreateDate) {
    this.userCreateDate = userCreateDate;
}
public String getUserProfile() {
    return userProfile;
}
public void setUserProfile(String userProfile) {
    this.userProfile = userProfile == null ? null : userProfile.trim();
}
public Date getUserBirthday() {
    return userBirthday;
}
public void setUserBirthday(Date userBirthday) {
    this.userBirthday = userBirthday;
}
public String getUserIp() {
    return userIp;
}
public void setUserIp(String userIp) {
    this.userIp = userIp == null ? null : userIp.trim();
}
}

package com.Jancoyan.domain;
import java.util.Date;
public class Article {
    private String articleId;
    private String articleTitle;
    private Integer articleAuthorId;
    private String articleAbstract;
    private Integer articleType;
    private Date articleEditTime;
    private Date articlePostDate;
    private Integer articleViewTime;
    private Integer articleCommentCount;
    private Integer articleLikeCount;
    public String getArticleId() {
        return articleId;
    }
    public void setArticleId(String articleId) {
        this.articleId = articleId == null ? null : articleId.trim();
    }
}
```

```

public String getArticleTitle() {      return articleTitle;    }
public void setArticleTitle(String articleTitle) {
    this.articleTitle = articleTitle == null ? null : articleTitle.trim();
}
public Integer getArticleAuthorId() {      return articleAuthorId;    }
public void setArticleAuthorId(Integer articleAuthorId) {
    this.articleAuthorId = articleAuthorId;
}
public String getArticleAbstract() {      return articleAbstract;    }
public void setArticleAbstract(String articleAbstract) {
    this.articleAbstract = articleAbstract == null ? null : articleAbstract.trim();
}
public Integer getArticleType() {      return articleType;    }
public void setArticleType(Integer articleType) {      this.articleType = articleType;    }
public Date getArticleEditTime() {      return articleEditTime;    }
public void setArticleEditTime(Date articleEditTime) {
    this.articleEditTime = articleEditTime;
}
public Date getArticlePostDate() {      return articlePostDate;    }
public void setArticlePostDate(Date articlePostDate) { this.articlePostDate = articlePostDate;    }
public Integer getArticleViewTime() {      return articleViewTime;    }
public void setArticleViewTime(Integer articleViewTime) {
    this.articleViewTime = articleViewTime;
}
public Integer getArticleCommentCount() { return articleCommentCount;    }
public void setArticleCommentCount(Integer articleCommentCount) {
    this.articleCommentCount = articleCommentCount;
}
public Integer getArticleLikeCount() { return articleLikeCount;    }
public void setArticleLikeCount(Integer articleLikeCount) { this.articleLikeCount = articleLikeC
ount;    }
}
package com.Jancoyan.domain;
import java.util.Date;
public class ArticleCommentKey {
    private String articleId;
    private String authorEmail;
    private Date commentDate;
    public String getArticleId() {
        return articleId;
    }
    public void setArticleId(String articleId) {
        this.articleId = articleId == null ? null : articleId.trim();
    }
    public String getAuthorEmail() {
        return authorEmail;
    }
    public void setAuthorEmail(String authorEmail) {
        this.authorEmail = authorEmail == null ? null : authorEmail.trim();
    }
}

```

```

    public Date getCommentDate() {
        return commentDate;
    }
    public void setCommentDate(Date commentDate) {
        this.commentDate = commentDate;
    }
}

package com.Jancoyan.domain;
public class ArticleComment extends ArticleCommentKey {
    private String authorNickname;
    private String authorIp;
    private Integer likeCount;
    private String commentContent;
    private String articleTitle;
    public String getAuthorNickname() {
        return authorNickname;
    }
    public void setAuthorNickname(String authorNickname) {
        this.authorNickname = authorNickname == null ? null : authorNickname.trim();
    }
    public String getAuthorIp() { return authorIp; }
    public void setAuthorIp(String authorIp) {
        this.authorIp = authorIp == null ? null : authorIp.trim();
    }
    public Integer getLikeCount() { return likeCount; }
    public void setLikeCount(Integer likeCount) { this.likeCount = likeCount; }
    public String getCommentContent() { return commentContent; }
    public void setCommentContent(String commentContent) {
        this.commentContent = commentContent == null ? null : commentContent.trim();
    }
    public String getArticleTitle() { return articleTitle; }
    public void setArticleTitle(String articleTitle) { this.articleTitle = articleTitle; }
}

package com.Jancoyan.domain;
import java.util.List;
public class Type {
    private Integer typeId;
    private String typeName;
    private Integer preTypeId;
    private List<Type> subTypes;
    @Override
    public String toString() {
        return "Type{" +
            "typeId=" + typeId +
            ", typeName=" + typeName + "\" +
            ", preTypeId=" + preTypeId +
            ", subTypes=" + subTypes +
            '>';
    }
    public List<Type> getSubTypes() {

```

```

        return subTypes;
    }
    public void setSubTypes(List<Type> subTypes) {
        this.subTypes = subTypes;
    }
    public Integer getTypeId() {        return typeId;    }
    public void setTypeId(Integer typeId) {        this.typeId = typeId;    }
    public String getTypeName() {        return typeName;    }
    public void setTypeName(String typeName) {
        this.typeName = typeName == null ? null : typeName.trim();
    }
    public Integer getPreTypeId() {        return preTypeId;    }
    public void setPreTypeId(Integer preTypeId) {        this.preTypeId = preTypeId;    }
}

package com.Jancoyan.domain;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;
public class ArticleExample {
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;
    public ArticleExample() {
        oredCriteria = new ArrayList<Criteria>();
    }
    public void setOrderByClause(String orderByClause) {
        this.orderByClause = orderByClause;
    }
    public String getOrderByClause() {        return orderByClause;    }
    public void setDistinct(boolean distinct) {        this.distinct = distinct;    }
    public boolean isDistinct() {        return distinct;    }
    public List<Criteria> getOredCriteria() {        return oredCriteria;    }
    public void or(Criteria criteria) {        oredCriteria.add(criteria);    }
    public Criteria or() {
        Criteria criteria = createCriteriaInternal();
        oredCriteria.add(criteria);
        return criteria;
    }
    public Criteria createCriteria() {
        Criteria criteria = createCriteriaInternal();
        if (oredCriteria.size() == 0) {
            oredCriteria.add(criteria);
        }
        return criteria;
    }
    protected Criteria createCriteriaInternal() {
        Criteria criteria = new Criteria();
        return criteria;
    }
    public void clear() {

```

```

        ordCriteria.clear();
        orderByClause = null;
        distinct = false;
    }
    protected abstract static class GeneratedCriteria {
        protected List<Criterion> criteria;
        protected GeneratedCriteria() {
            super();
            criteria = new ArrayList<Criterion>();
        }
        public boolean isValid() {
            return criteria.size() > 0;
        }
        public List<Criterion> getAllCriteria() {
            return criteria;
        }
        public List<Criterion> getCriteria() {
            return criteria;
        }
        protected void addCriterion(String condition) {
            if (condition == null) {
                throw new RuntimeException("Value for condition cannot be null");
            }
            criteria.add(new Criterion(condition));
        }
        protected void addCriterion(String condition, Object value, String property) {
            if (value == null) {
                throw new RuntimeException("Value for " + property + " cannot be null");
            }
            criteria.add(new Criterion(condition, value));
        }
        protected void addCriterion(String condition, Object value1, Object value2, String property) {
            if (value1 == null || value2 == null) {
                throw new RuntimeException("Between values for "+property+" cannot be null");
            }
            criteria.add(new Criterion(condition, value1, value2));
        }
        public Criteria andArticleTitleLike(String value) {
            addCriterion("article_title like", value, "articleTitle");
            return (Criteria) this;
        }
        public Criteria andArticleAuthorIdEqualTo(Integer value) {
            addCriterion("article_author_id =", value, "articleAuthorId");
            return (Criteria) this;
        }
        public Criteria andArticleTypeEqualTo(Integer value) {
            addCriterion("article_type =", value, "articleType");
            return (Criteria) this;
        }
        public Criteria andArticleTypeIn(List<Integer> values) {

```

```
        addCriterion("article_type in", values, "articleType");
        return (Criteria) this;
    }

    public static class Criteria extends GeneratedCriteria {
    protected Criteria() {
        super();
    }
    }

    public static class Criterion {
        private String condition;
        private Object value;
        private Object secondValue;
        private boolean noValue;
        private boolean singleValue;
        private boolean betweenValue;
        private boolean listValue;
        private String typeHandler;
        public String getCondition() {
            return condition;
        }
        public Object getValue() {
            return value;
        }
        public Object getSecondValue() {
            return secondValue;
        }
        public boolean isNoValue() {
            return noValue;
        }
        public boolean isSingleValue() {
            return singleValue;
        }
        public boolean isBetweenValue() {
            return betweenValue;
        }
        public boolean isListValue() {
            return listValue;
        }
        public String getTypeHandler() {
            return typeHandler;
        }
        protected Criterion(String condition) {
            super();
            this.condition = condition;
            this.typeHandler = null;
            this.noValue = true;
        }
        protected Criterion(String condition, Object value, String typeHandler) {
            super();
            this.condition = condition;
```

```

        this.value = value;
        this.typeHandler = typeHandler;
        if (value instanceof List<?>) {
            this.listValue = true;
        } else {
            this.singleValue = true;
        }
    }
    protected Criterion(String condition, Object value) {
        this(condition, value, null);
    }
    protected Criterion(String condition, Object value, Object secondValue, String typeHandler)
{
    super();
    this.condition = condition;
    this.value = value;
    this.secondValue = secondValue;
    this.typeHandler = typeHandler;
    this.betweenValue = true;
}
    protected Criterion(String condition, Object value, Object secondValue) {
        this(condition, value, secondValue, null);
    }
}
}}
package com.Jancoyan.domain;
import java.util.ArrayList;
import java.util.List;
public class TypeExample {
    protected String orderByClause;
    protected boolean distinct;
    protected List<Criteria> oredCriteria;
    public TypeExample() {
        oredCriteria = new ArrayList<Criteria>();
    }
    public void setOrderByClause(String orderByClause) {
        this.orderByClause = orderByClause;
    }
    public String getOrderByClause() {
        return orderByClause;
    }
    public void setDistinct(boolean distinct) {
        this.distinct = distinct;
    }
    public boolean isDistinct() {
        return distinct;
    }
    public List<Criteria> getOredCriteria() {
        return oredCriteria;
    }
}

```

```

public void or(Criteria criteria) {
    oredCriteria.add(criteria);
}
public Criteria or() {
    Criteria criteria = createCriteriaInternal();
    oredCriteria.add(criteria);
    return criteria;
}
public Criteria createCriteria() {
    Criteria criteria = createCriteriaInternal();
    if (oredCriteria.size() == 0) {
        oredCriteria.add(criteria);
    }
    return criteria;
}
protected Criteria createCriteriaInternal() {
    Criteria criteria = new Criteria();
    return criteria;
}
public void clear() {
    oredCriteria.clear();
    orderByClause = null;
    distinct = false;
}
protected abstract static class GeneratedCriteria {
    protected List<Criterion> criteria;
    protected GeneratedCriteria() {
        super();
        criteria = new ArrayList<Criterion>();
    }
    public boolean isValid() {      return criteria.size() > 0;      }
    public List<Criterion> getAllCriteria() {      return criteria;      }
    public List<Criterion> getCriteria() {
        return criteria;
    }
    protected void addCriterion(String condition) {
        if (condition == null) {
            throw new RuntimeException("Value for condition cannot be null");
        }
        criteria.add(new Criterion(condition));
    }
    protected void addCriterion(String condition, Object value, String property) {
        if (value == null) {
            throw new RuntimeException("Value for " + property + " cannot be null");
        }
        criteria.add(new Criterion(condition, value));
    }
    protected void addCriterion(String condition, Object value1, Object value2, String property)
{

```

```

        if (value1 == null || value2 == null) {
            throw new RuntimeException("Between values for " + property + " cannot be nul
l");
        }
        criteria.add(new Criterion(condition, value1, value2));
    }
    public Criteria andPreTypeIdIsNull() {
        addCriterion("pre_type_id is null");
        return (Criteria) this;
    }
    public Criteria andPreTypeIdEqualTo(Integer value) {
        addCriterion("pre_type_id =", value, "preTypeId");
        return (Criteria) this;
    }
    public static class Criteria extends GeneratedCriteria {
        protected Criteria() {
            super();
        }
    }
    public static class Criterion {
        private String condition;
        private Object value;
        private Object secondValue;
        private boolean noValue;
        private boolean singleValue;
        private boolean betweenValue;
        private boolean listValue;
        private String typeHandler;
        public String getCondition() {
            return condition;
        }
        public Object getValue() {
            return value;
        }
        public Object getSecondValue() {
            return secondValue;
        }
        public boolean isNoValue() {
            return noValue;
        }
        public boolean isSingleValue() {
            return singleValue;
        }
        public boolean isBetweenValue() {
            return betweenValue;
        }
        public boolean isListValue() {
            return listValue;
        }
        public String getTypeHandler() {

```

```

        return typeHandler;
    }
    protected Criterion(String condition) {
        super();
        this.condition = condition;
        this.typeHandler = null;
        this.noValue = true;
    }
    protected Criterion(String condition, Object value, String typeHandler) {
        super();
        this.condition = condition;
        this.value = value;
        this.typeHandler = typeHandler;
        if (value instanceof List<?>) {
            this.listView = true;
        } else {
            this.singleValue = true;
        }
    }
    protected Criterion(String condition, Object value) {
        this(condition, value, null);
    }
    protected Criterion(String condition, Object value, Object secondValue, String typeHandler)
{
        super();
        this.condition = condition;
        this.value = value;
        this.secondValue = secondValue;
        this.typeHandler = typeHandler;
        this.betweenValue = true;
    }
    protected Criterion(String condition, Object value, Object secondValue) {
        this(condition, value, secondValue, null);
    }
}
}
}
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme() + ":"
        request.getServerName() + ":" + request.getServerPort() +
        request.getContextPath() + "/";
%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Welcome</title>
    <script src="/static/js/jquery-1.12.js"></script>
    <link rel="stylesheet" href="/static/css/articlelist.css">

```

```
<script>
    var totalPages = 0;
    var currentPage = 1;
</script>
</head>
<body>
<div class="nav">
    <a href="./index.jsp" class="back-to-index">JancoBlog</a>
    <div class="signin">
        <%
            String aTag = null;
            User user = (User) session.getAttribute("user");
            if(user == null){
                aTag = "<a href='login.jsp'>登录</a>";
            } else {
                aTag = "<a href='./workbench'>后台</a>";
            }
        %>
        <%=aTag%>
    </div>
</div>
<div class="main clearfix">
    <div class="title-and-signature">
        <h1>JANCOYAN</h1> <br>
        <div class="quote">十年饮冰, 难凉热血</div>
    </div>
    <div class="sort-and-search">
        <ul class="types clearfix">
            <div class="search">
                <input type="text" name="search" placeholder="博文标题" id="search-text" />
                <button id="search-btn">搜索</button>
            </div>
        </ul>
    </div>
    <div class="index-content clearfix">
        <div class="left-content clearfix">
            <ul class="clearfix"></ul>
        </div>
        <div class="right-content">
            <div class="article-rank-list">
                <div class="article-rank-title">
                    热门文章
                </div>
                <div class="segment-line"></div>
                <ul class="hot-articles"></ul>
            </div>
        </div>
    </div>
</div>
<script>
```

```

$(function(){
    build_nav_bar();
    get_article_page(1);
    build_hot_rank();
});
function get_article_page(pn) {
    $.ajax({
        url: "articles",
        data : "pn=" + pn,
        type : "GET",
        success: function (result) {
            build_article_list(result);
        }
    });
}
function build_article_list(result) {
    var articleList = result.extend.pageInfo.list;
    totalPage = result.extend.pageInfo.pages;
    $.each(articleList, function (index, item) {
        var articleBoxLi = $("- </li>").addClass("article-box");
        var articlePath = "./static/p/" + item.articleId + ".html";
        var aTitle = $("</a>").attr("href", articlePath).append(item.articleTitle);
        aTitle.click(function () {
            add_article_view(item.articleId);
        });
        var segmentLine = $("

</div>").addClass("article-abstract")
            .append(item.articleAbstract);
        var likeArticles = $("

</div>").addClass("like-articles")
            .append("<span class='like-logo'>♥ </span>")
            .append("<span class='like-count'></span>").append(item.articleLikeCount);
        likeArticles.click(function(){
            add_like_count(this.childNodes[1], item.articleId);
        });
        var articleInfo = $("

</div>")
            .addClass("article-info clearfix")
            .append(likeArticles)
            .append("<div></div>").addClass("view-count").append("浏览量: "+item.articleView
Time))
            .append("<div></div>").addClass("post-time").append("发布日期: "+dateFormat(ite
m.articlePostDate));
        articleBoxLi.append(aTitle)
            .append(segmentLine)
            .append(articleAbstract)
            .append(articleInfo)
            .appendTo(".left-content ul");
    })
}
function build_nav_bar() {
    $.ajax({

```

```

url: "types",
type: "get",
success: function (result) {
    var superType = result.extend.types;
    $.each(superType, function (index, item) {
        var currTypeName =
            $("<a target='_blank' ></a>").attr("href",
                "/sortlist.jsp?type="+item.typeId).text(item.typeName);
        var currSuperTypeDrop = $("<div></div>").addClass("drop-content");
        var subTypes = item.subTypes;
        $.each(subTypes, function (index, item1) {
            currSuperTypeDrop
                .append($("<a target='_blank'></a>")
                    .attr("href", "sortlist.jsp?type=" + item1.typeId)
                    .append(item1.typeName)
                );
        });
        $("<li></li>").addClass("drop-down clearfix")
            .append(currTypeName)
            .append(currSuperTypeDrop)
            .prependTo(".types");
    });
}
});
}
function build_hot_rank() {
    $.ajax({
        url: "articles",
        type: "POST",
        success: function (result) {
            var hotArticles = result.extend.hotArticles;
            $.each(hotArticles, function (index, item) {
                var hotArticleLi = $("<li></li>").addClass("hot-article");
                var orderDiv = $("<div></div>").addClass("hot-order").append(index + 1);
                var articlePath = "/static/p/" + item.articleId + ".html";
                var titleA = $("<a></a>").attr("href", articlePath).append(item.articleTitle);
                titleA.click(function () {
                    add_article_view(item.articleId);
                });
                var viewTime =
                    $("<div></div>").addClass("hot-article-views").append(item.articleViewTim
e);

                hotArticleLi.append(orderDiv)
                    .append(titleA)
                    .append(viewTime)
                    .appendTo(".hot-articles");
            });
        }
    });
}
}

```

```

function add_like_count(element, id){
    var nowCount = element.innerHTML;
    $.ajax({
        url: "article/like",
        type: "post",
        data: {
            "count": nowCount,
            "id" : id
        },
        success: function (result) {
            var nowCount = result.extend.count;
            element.innerHTML = nowCount;
        }
    });
}

function add_article_view(articleId) {
    $.ajax({
        url: "article/view",
        type: "POST",
        data: {
            "id": articleId,
        },
        success: function () {
        }
    });
}

$("#search-btn").click(function (){
    var keyWord = $("#search-text").val();
    if(keyWord == ""){
        alert("搜索字段不能为空");
        return;
    } else {
        window.location.href = "./search.jsp?keyword=" + keyWord;
    }
});

$(window).scroll(function(){
    var scrollTop = $(this).scrollTop();
    var scrollHeight = $(document).height();
    var windowHeight = $(this).height();
    if(scrollTop + windowHeight == scrollHeight && currentPage < totalPages){
        currentPage += 1;
        get_article_page(currentPage);
    }
});

function dateFormat(date) {
    var s = new Date(date)
    var y = s.getFullYear()
    var m = (s.getMonth() + 1) < 10 ? '0' + (s.getMonth() + 1) : (s.getMonth() + 1)
    var dd = s.getDate() < 10 ? '0' + s.getDate() : s.getDate()
    var hh = s.getHours() < 10 ? '0' + s.getHours() : s.getHours()

```



```

        var mm = s.getMinutes() < 10 ? '0' + s.getMinutes() : s.getMinutes()
        var ss = s.getSeconds() < 10 ? '0' + s.getSeconds() : s.getSeconds()
        var enddate = y + '-' + m + '-' + dd + ' ' + hh + ':' + mm + ':' + ss
        return enddate
    }
</script>
</body>
</html>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme() + ":"
        request.getServerName() + ":" + request.getServerPort() +
        request.getContextPath() + "/";
%>
<!DOCTYPE html>
<html>
<head>
    <base href="<%=basePath%>" />
    <meta charset="utf-8">
    <title>文章分类</title>
    <link rel="stylesheet" href="/static/css/articlelist.css">
    <script type="text/javascript" src="/static/js/jquery-1.12.js"></script>
</head>
<script type="text/javascript">
    var typeid = <%=request.getParameter("type")%>;
    var currentPage = 1;
    $(function(){
        build_nav_bar();
        get_articles(typeid, currentPage);
        $(document).on("click", ".nav-title", function () {
            var navContent = $(this).parent(".nav-menu").children(".nav-content");
            if(navContent.css("display") != "none"){
                navContent.hide(200);
            }else{
                var sibling = navContent.parents(".nav-menu").siblings();
                sibling.each(function(){
                    $(this).children(".nav-content").hide(200);
                })
                navContent.show(200);
            }
        })
    });
</script>
<body>
<div class="nav">
    <a href="/index.jsp" class="back-to-index">JancoBlog</a>
</div>
<div class="main">
    <div class="side-nav-bar">
        <div class="nav-bar-title"><a href="javascript:;">分类查看</a></div>

```

```

    <div class="segment-line"></div>
    <div class="nav-list"></div>
</div>
<div class="article-sort-list">
    <ul></ul>
</div>
</div>
<script>
function build_nav_bar() {
    $.ajax({
        url: "types",
        type: "get",
        success: function (result) {
            var superType = result.extend.types;
            $.each(superType, function (index, item) {
                var navMenu = $("<div></div>").addClass("nav-menu");
                navMenu.append($("<div></div>").addClass("nav-title").append(item.typeName));
                var navContent =
                    $("<div></div>").addClass("nav-content").css("display","none");
                var subTypes = item.subTypes;
                $.each(subTypes, function (index, item1) {
                    navContent
                        .append($("<a></a>")
                            .attr("href", "./sortlist.jsp?type=" + item1.typeId)
                            .append(item1.typeName)
                        );
                });
                navMenu.append(navContent).appendTo(".nav-list");
            });
        }
    });
}

function get_articles(typeid, pn) {
    $.ajax({
        url: "type/" + typeid,
        type: "GET",
        data:"pn=" + pn,
        success: function (result) {
            if(result.extend.pageInfo.pages === 0){
                alert("暂时没有这方面的文章...");
                return;
            }
            build_article_list(result);
        }
    });
}

function build_article_list(result) {
    var articleList = result.extend.pageInfo.list;
    $.each(articleList, function (index, item){
        var articleBoxLi = $("<li></li>").addClass("article-box");
    });
}

```

```

        var articlePath = "/static/p/" + item.articleId + ".html";
        var aTitle = $("<a></a>").attr("href", articlePath).append(item.articleTitle);
        aTitle.click(function () {
            add_article_view(item.articleId);
        });
        var segmentLine = $('<div class="segment-line"></div>');
        var articleAbstract = $('<div></div>').addClass("article-abstract")
        .append(item.articleAbstract);
        var articleInfo = $('<div></div>')
        .addClass("article-info")
        .append($('<div></div>').addClass("view-count").append("浏览量: "+item.articleViewTim
e))
        .append($('<div></div>').addClass("post-time").append("发布日期: "+dateFormat(item.arti
clePostDate)));
        articleBoxLi.append(aTitle)
        .append(segmentLine)
        .append(articleAbstract)
        .append(articleInfo)
        .appendTo(".article-sort-list ul");
    });
}
$(window).scroll(function(){
    var scrollTop = $(this).scrollTop();
    var scrollHeight = $(document).height();
    var windowHeight = $(this).height();
    if(scrollTop + windowHeight === scrollHeight){
        currentPage += 1;
        get_articles(typeid, currentPage);
    }
});
function add_article_view(articleId) {
    $.ajax({
        url: "article/view",
        type: "POST",
        data: {
            "id": articleId
        }
    });
}
function dateFormat(date) {
    var s = new Date(date)
    var y = s.getFullYear()
    var m = (s.getMonth() + 1) < 10 ? '0' + (s.getMonth() + 1) : (s.getMonth() + 1)
    var dd = s.getDate() < 10 ? '0' + s.getDate() : s.getDate()
    var hh = s.getHours() < 10 ? '0' + s.getHours() : s.getHours()
    var mm = s.getMinutes() < 10 ? '0' + s.getMinutes() : s.getMinutes()
    var ss = s.getSeconds() < 10 ? '0' + s.getSeconds() : s.getSeconds()
    var enddate = y + '-' + m + '-' + dd + ' ' + hh + ':' + mm + ":" + ss
    return enddate
}

```

```

</script>
</body>
</html>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme() + ":"
        request.getServerName() + ":" + request.getServerPort() +
        request.getContextPath() + "/";
%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>Result</title>
        <link rel="stylesheet" type="text/css" href="/static/css/articlelist.css"/>
        <script src="/static/js/jquery-1.12.js"></script>
        <base href="<%=basePath%>" />
    </head>
    <style>
        .article-search-list ul{
            width: 70%;
            margin: 0 auto;
        }
        .article-search-list ul>li{
            width: 100%;
        }
        .article-search-list{
            width: 100%;
            margin: 60px auto;
        }
    </style>
    <script>
        <%
            String keyword = request.getParameter("keyword");
        %>
        var keyword = '<%=keyword%>';
        var currentPage = 1;
        var totalPages = 999;
        $(function (){
            get_articles(currentPage);
        });
    </script>
    <body>
        <div class="nav">
            <a href="#" class="back-to-index">JancoBlog</a>
        </div>
        <div class="main clearfix">
            <h2 class="search-title"></h2>
            <div class="article-search-list">
                <ul></ul>
            </div>
        </div>
    </body>
</html>

```

```

    </div>
  </div>
<script>
  function get_articles(pn) {
    $.ajax({
      url: "search",
      type: "GET",
      data: {
        "pn": pn,
        "keyword": keyword
      },
      success: function (result){
        build_article_list(result);
      }
    });
  }
  function build_article_list(result) {
    var articleList = result.extend.pageInfo.list;
    $(".search-title").text("找到关于\"" + keyword + "\"的结果共 " +
      result.extend.pageInfo.total + " 条");
    totalPage = result.extend.pageInfo.pages;
    $.each(articleList, function (index, item) {
      var articleBoxLi = $("<li></li>").addClass("article-box");
      var articlePath = "./static/p/" + item.articleId + ".html";
      var aTitle = $("<a></a>").attr("href", articlePath).append(item.articleTitle);
      aTitle.click(function () {
        add_article_view(item.articleId);
      });
      var segmentLine = $('<div class="segment-line"></div>');
      var articleAbstract = $("<div></div>").addClass("article-abstract")
        .append(item.articleAbstract);
      var articleInfo = $("<div></div>")
        .addClass("article-info")
        .append($("<div></div>").addClass("view-count").append("浏览量: "+item.article
ViewTime))
        .append($("<div></div>").addClass("post-time").append("发布日期: "+dateForma
t(item.articlePostDate)));
      articleBoxLi.append(aTitle)
        .append(segmentLine)
        .append(articleAbstract)
        .append(articleInfo)
        .appendTo(".article-search-list ul");
    })
  }
  function add_article_view(articleId) {
    $.ajax({
      url: "article/view",
      type: "POST",
      data: {
        "id": articleId,

```

```

    }
    });
}
function dateFormat(date) {
    var s = new Date(date)
    var y = s.getFullYear()
    var m = (s.getMonth() + 1) < 10 ? '0' + (s.getMonth() + 1) : (s.getMonth() + 1)
    var dd = s.getDate() < 10 ? '0' + s.getDate() : s.getDate()
    var hh = s.getHours() < 10 ? '0' + s.getHours() : s.getHours()
    var mm = s.getMinutes() < 10 ? '0' + s.getMinutes() : s.getMinutes()
    var ss = s.getSeconds() < 10 ? '0' + s.getSeconds() : s.getSeconds()
    var enddate = y + '-' + m + '-' + dd + ' ' + hh + ':' + mm + ':' + ss
    return enddate
}
$(window).scroll(function(){
    var scrollTop = $(this).scrollTop();
    var scrollHeight = $(document).height();
    var windowHeight = $(this).height();
    if(scrollTop + windowHeight == scrollHeight){
        currentPage += 1;
        get_articles(currentPage);
    }
});
</script>
</body>
</html>
<%@ page import="com.Jancoyan.domain.User" %>
<%@ page import="com.Jancoyan.domain.Article" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme() + ":"
        request.getServerName() + ":" + request.getServerPort() +
        request.getContextPath() + "/";
%>
<!DOCTYPE html>
<html>
<head>
    <title>编辑博文</title>
    <base href="<%=basePath%>" />
    <link rel="stylesheet" href="/static/js/editor.md/css/editormd.min.css">
    <link rel="stylesheet" href="/static/css/edit_other.css">
    <script src="/static/js/jquery-1.12.js"></script>
    <script src="/static/js/editor.md/editormd.js"></script>
    <script>
        <%
            Article article = (Article) session.getAttribute("article");
            String content = (String) session.getAttribute("content");
            String title = "";
            if(article != null) title = article.getArticleTitle();
        %>
    </script>

```

```

$(function(){
    var testEditor;
    testEditor = editormd("test-editormd", {
        palceholder:'欢迎',
        htmlDecode:"style, script, iframe",
        width: "100%",
        height: 700,
        syncScrolling:"single",
        emoji: true,
        path: "./static/js/editor.md/lib/",
        saceHTMLToTextarea: true,
        tocm: true,
        tex: true,
        flowChart: true,
        imageFormats: ["jpg", "gif", "png", "jpeg"],
        imageUpload: true,
        imageUploadURL:"uploadarticlepicture",
        value: '<%=content == null ? "" : content%>'
    });

    $.ajax({
        url: "types",
        type: "get",
        success: function (result) {
            var types = result.extend.types;
            var subTypeList = $("<div></div>").addClass("sub-type-list");
            $.each(types, function (index, item){
                var currTypeA = $("<a></a>")
                    .attr("attr", "javascript:;")
                    .attr("select-id", item.typeId)
                    .append(item.typeName);
                currTypeA.click(function (){
                    $(".super-type>a").removeClass("active");
                    $(this).addClass("active");
                    $(".sub-type>a").removeClass("active");
                    $(".sub-type").css("display", "none");
                    var queryStr = ".sub-type[select-status=" + item.typeId + "]";
                    $(queryStr).css("display", "block");
                });
                currTypeA.appendTo(".super-type");
                var currSubTypes = $("<div></div>")
                    .addClass("sub-type")
                    .attr("select-status", item.typeId)
                    .css("display", "none");
                $.each(item.subTypes, function (index, item) {
                    var currSubTypeA =
                        $("<a></a>")
                            .attr("href", "javascript:;")
                            .attr("select-id", item.typeId)
                            .append(item.typeName);

```

```

        currSubTypeA.click(function (){
            $(".sub-type>a").removeClass("active");
            $(this).addClass("active");
        });
        currSubTypeA.appendTo(currSubTypes);
    });
    currSubTypes.appendTo(".sub-type-list");
});
    }
});
</script>
</head>
<body>
    <div class="nav">
        <a href="/index.jsp" class="back-to-index">JancoBlog</a>
    </div>
    <div class="main">
        <div id="title">
            <input id="title_input_text" type="text" placeholder="请输入文章标题" value="<%=titl
e ==
            null ? "" : title%>">
            <input type="button" id="submit_btn" value="发布文章">
        </div>
        <div class="clearfix">
            <div style="float: left;width:80%; border: 0px solid red; padding: 30px 10%; ">
                <div id="test-editormd">
                    <textarea name="message" id="message"></textarea>
                </div>
            </div>
        </div>
        <div id="type-select">
            <span>文章类型: </span>
            <div class="super-type"></div>
            <div class="sub-type-list"></div>
        </div>
    </div>
</script>
$(document).on("click", "#submit_btn", function () {

    var articleTitle = $("#title_input_text").val();
    var articleContent = $(".editormd-preview")[0].innerHTML;
    var mdContent = $(".editormd-markdown-textarea").val();
    var idxNumber = $(".active").length;
    var typeId;

    if("" == articleTitle){
        alert("标题不能为空");
        return;
    }

```



```
if("" == articleContent){
    alert("内容不能为空");
    return;
}
if(0 == idxNumber){
    alert("请选择文章类型");
    return;
} else if (1 == idxNumber){
    typeId = $(".active").attr("select-id");
} else {
    typeId = $(".active")[1].attributes[1].value;
}
<%
    User user = (User) session.getAttribute("user");
    if(article == null){
%>
$.ajax({
    url:"article",
    type:"PUT",
    data: {
        'innerHTML': articleContent,
        'innerMD' : mdContent,
        'userId': <%=user.getUserId()%>,
        'userNickname': '<%=user.getUserNickname()%>',
        'title': $("#title_input_text").val(),
        'type': typeId,
    },
    success: function (result) {
        alert("发布成功!");

        window.location.href="/index.jsp";
    }
});
<%
} else {
%>
$.ajax({
    url: "article/update",
    type:"POST",
    data: {
        'innerHTML': articleContent,
        'innerMD' : mdContent,
        'userNickname': '<%=user.getUserNickname()%>',
        'title': $("#title_input_text").val(),
        'type': typeId,
    },
    success: function (result) {
        alert("修改成功!");

        window.location.href="/index.jsp";
```

```

    }
    })
    <%
    }
    %>

    });
    </script>
    </body>
</html>
<%@ page import="com.Jancoyan.domain.User" %>
<%@ page import="com.Jancoyan.utils.TimeUtils" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    String basePath = request.getScheme() + ":"
        request.getServerName() + ":" + request.getServerPort() +
        request.getContextPath() + "/";
%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <base href="<%=basePath%>" />
    <script src="/static/js/jquery-1.12.js"></script>
    <link rel="stylesheet" href="/static/css/workbench.css">
</head>
<%
    User user = (User) session.getAttribute("user");
%>
<body>
<div class="nav">
    <a href="/index.jsp" class="back-to-index">JancoBlog</a>
</div>
<h1>
    后台管理
</h1>
<div class="main clearfix">
    <div class="function-list">
        <ul class="operate-list">
            <li><a href="/edit.jsp" id="write-eassy">写博文</a></li>
            <br>
            <li><a href="javascript:;" id="personal-info-item">个人信息</a></li>
            <li><a href="javascript:;" id="article-manage-item">文章管理</a></li>
            <li><a href="javascript:;" id="comment-manage-item">评论管理</a></li>
            <li><a href="javascript:;" id="change-password-item">修改密码</a></li>
        </ul>
    </div>
    <div class="personal-info">
        <table>
            <tr>

```

```

        <td>用户名</td>
        <td><input type="text" disabled="disabled" name="name" id="user-name" value="
<%=user.getUserNickname()%>" /></td>
    </tr>
    <tr>
        <td>邮箱</td>
        <td><input type="text" disabled="disabled" name="name" id="user-email"
            value="<%=user.getUserEmail()%>" /></td>
    </tr>
    <tr>
        <td>性别</td>
        <td>
            <div class="user-sex">
                <div class="male <%=user.getUserSex() == 1 ? "choose" : ""%>">男</di
v>
                <div class="famale <%=user.getUserSex() == 0 ? "choose" : ""%>">女</
div>
                <div class="none <%=user.getUserSex() == -1 ? "choose" : ""%>">未知
</div>
            </div>
        </td>
    </tr>
    <tr>
        <td>生日</td>
        <td><input type="text" disabled="disabled" name="name" id="user-create-date"
            value="<%=TimeUtils.castDateTypeToDateString(user.getUserCreateDate
())%>" placeholder="格式: 2001-12-1"/></td>
    </tr>
</table>
<button id="edit-personal-info">编辑</button>
<button id="save-personal-info">保存</button>
<button id="cancel-personal-info">取消</button>
</div>
<div class="manage-articles">
    <table class="list" id="article-list">
        <thead>
            <tr>
                <th>#</th>
                <th>文章标题</th>
                <th>发布时间</th>
                <th>修改时间</th>
                <th>浏览量</th>
                <th>评论数</th>
                <th>点赞数</th>
                <th>操作</th>
            </tr>
        </thead>
        <tbody>
        </tbody>
    </table>

```

```

    <div id="article-page-nav-bar"></div>
</div>
<!-- 评论管理 -->
<div class="manage-comment">
    <table class="list" id="comment-list">
        <thead>
            <tr>
                <th>#</th>
                <th>文章名</th>
                <th>评论摘要</th>
                <th>时间</th>
                <th>邮箱</th>
                <th>操作</th>
            </tr>
        </thead>
        <tbody></tbody>
    </table>
    <div id="comment-page-nav-bar"></div>
</div>
<div class="change-password">
    <div class="old-password">
        <input type="password" placeholder="旧密码" id="old-password" />
    </div>
    <div class="new-password">
        <input type="password" placeholder="新密码" id="new-password" />
    </div>
    <div class="repeat-new-password">
        <input type="password" placeholder="重复新密码" id="repeat-new-password" />
    </div>
    <button class="submit-password">提交修改</button>
</div>
<div id="myModal" class="modal">
    <div class="modal-content">
        <div class="modal-header">
            <div class="show-details">评论详细信息查看</div>
        </div>
        <div class="modal-body">
        </div>
    </div>
</div>
</div>
<footer>
</footer>
<script type="text/javascript">
    var currentArticlePage = 1;
    var currentCommentPage = 1;
    $(function () {
        to_article_page(1);
    });
    $("#article-manage-item").click(function () {

```

```

$(".personal-info").css("display","none");
$(".change-password").css("display", "none");
$(".manage-comment").css("display", "none");
$(".manage-articles").css("display","block");
to_article_page(1);
});
$("#personal-info-item").click(function () {

    $(".manage-articles").css("display","none");
    $(".change-password").css("display", "none");
    $(".manage-comment").css("display", "none");
    $(".personal-info").css("display","block");
});
$("#change-password-item").click(function (){
    $(".manage-articles").css("display","none");
    $(".personal-info").css("display","none");
    $(".manage-comment").css("display", "none");
    $(".change-password").css("display", "block");
})
$("#comment-manage-item").click(function () {
    $(".manage-articles").css("display","none");
    $(".personal-info").css("display","none");
    $(".change-password").css("display", "none");
    $(".manage-comment").css("display", "block");
    to_comment_page(1);
})
$("#write-eassy").click(function () {
    <%
    if(session.getAttribute("content") != null){
        session.removeAttribute("content");
    }
    if(session.getAttribute("article") != null){
        session.removeAttribute("article");
    }
    %>
});
$("#edit-personal-info").click(function () {
    $("#save-personal-info").css("display", "inline");
    $("#cancel-personal-info").css("display", "inline");
    $("#edit-personal-info").css("display", "none");
    $(".personal-info input").removeAttr("disabled");
    $.each($(".user-sex>div"), function (index, item) {
        item.onclick = function(){
            $(".user-sex>div").removeClass("choose");
            item.classList.add("choose");
        }
    });
});
$("#save-personal-info").click(function () {
    var userName = $("#user-name").val();

```

```

var email = $("#user-email").val();
var creat = $("#user-create-date").val();
var sex = -1;
if($(".male").hasClass("choose")){
    sex = 1;
} else if($(".female").hasClass("choose")) {
    sex = 0;
}
var nameReg = /^[u4E00-\u9FA5A-Za-z0-9]{2,20}$/;
if(!nameReg.test(userName)){
    alert("用户名格式不对");
    return;
}
var emailReg = /^[a-z0-9_\.~]+@([da-z\.-]+)\.([a-z]{2,6})$/;
if(!emailReg.test(email)){
    alert("用户邮箱格式不对");
    return;
}
var birthdayReg =
    /^(0-9){4}-(((0[13578]|(10|12))-(0[1-9]||[1-2][0-9]||3[0-1]))|(02-(0[1-9]||[1-2][0-9]))|((0[469]||11)
    -(0[1-9]||[1-2][0-9]||30)))$/;
if(!birthdayReg.test(creat)){
    alert("用户生日日期不对");
    return;
}
$.ajax({
    url: "user",
    type: "POST",
    data: {
        "userNickName": userName,
        "userEmail" : email,
        "userSex" : sex,
        "userBirthdayDay" : creat
    },
    success : function (result) {
        alert("更新成功");
        $("#user-name").attr("disabled", "disabled");
        $("#user-email").attr("disabled", "disabled");
        $("#user-create-date").attr("disabled", "disabled");
        $(".user-sex>div").removeAttr("click");
        $("#save-personal-info").css("display", "none");
        $("#cancel-personal-info").css("display", "none");
        $("#edit-personal-info").css("display", "block");
    }
});
$("#cancel-personal-info").click(function () {
    $("#user-name").val("<%=user.getUserNickname()%>").attr("disabled", "disabled");
    $("#user-email").val("<%=user.getUserEmail()%>").attr("disabled", "disabled");
    $("#user-create-date").val("<%=TimeUtils.castDataTypeToDateString(user.getUserCreateDate

```

```

()%)>").attr("disabled", "disabled");
    $(".user-sex>div").removeClass("choose");
    if(1 == <%=user.getUserSex()%>){
        $(".male").addClass("choose");
    }else if(0 == <%=user.getUserSex()%>){
        $(".famale").addClass("choose");
    }else {
        $(".none").addClass("choose");
    }
    $("#edit-personal-info").css("display","block");
    $("#cancel-personal-info").css("display", "none");
    $("#save-personal-info").css("display", "none");
    $.each($(".user-sex>div"), function (index, item) {
        item.onclick = null;
    })
});
$(".submit-password").click(function () {
    var oldPwd = $("#old-password").val();
    var newPwd = $("#new-password").val();
    var repeat = $("#repeat-new-password").val();
    $.ajax({
        url: "user/password",
        type: "POST",
        data: {
            "oldPwd": oldPwd,
            "newPwd": newPwd,
            "repeat": repeat
        },
        success: function (result){
            alert(result.extend.msg);
            var oldPwd = $("#old-password").val("");
            var newPwd = $("#new-password").val("");
            var repeat = $("#repeat-new-password").val("");
        }
    });
});
$(document).on("click", "#edit-article-btn", function () {
    var id = $(this).attr("edit-id");
    $.ajax({
        url: "redirect/article",
        type: "POST",
        data: "id=" + id,
        success : function (){
            window.location.href = "/edit.jsp";
        }
    });
});
$(document).on("click", "#delete-article-btn", function () {
    var articleTitle = $(this).parents("tr").find("td:eq(1)").text();
    var articleId = $(this).attr("delete-id");

```

```

        if(confirm("确认删除" + articleTitle + "吗? ")){
            $.ajax({
                url: "article/" + articleId,
                type:"DELETE",
                success: function (result) {
                    alert("删除成功");
                    to_article_page(currentArticlePage);
                }
            });
        }
    });
$(document).on("click", "#show-comment-btn", function () {
    var articleId =
        $(this).parents("tr").find("td:eq(5)").find("#delete-comment-btn").attr("delete-id");
    var authorEmail = $(this).parents("tr").find("td:eq(4)").text();
    var commentDate = $(this).parents("tr").find("td:eq(3)").text();
    var articleTitle = $(this).parents("tr").find("td:eq(1)").text();
    $.ajax({
        url: "getcomment",
        type: "GET",
        data: {
            "articleId": articleId,
            "email" : authorEmail,
            "commentDate" : commentDate
        },
        success: function (result) {
            var comment = result.extend.comment;
            $(".modal-body").empty();
            $(".modal-body")
                .append($(".modal-body").append("所在文章 id: " + comment.articleId))
                .append($(".modal-body").append("文章名称: " + articleTitle))
                .append($(".modal-body").append("评论人昵称: " + comment.authorNickname))
                .append($(".modal-body").append("内容: " + comment.commentContent))
                .append($(".modal-body").append("邮箱: " + comment.authorEmail))
                .append($(".modal-body").append("评论日期: " + dateFormat(comment.commentDate)))
            $(".modal-body").append($(".modal-body").append("评论人地址: " + comment.authorIp));
            $(".modal").css("display", "block");
            $(window).click(function(event){
                if(event.target == $(".modal")[0]){
                    $(".modal").css("display", "none");
                }
            })
        }
    })
})
$(document).on("click", "#delete-comment-btn", function () {
    var articleAbstract = $(this).parents("tr").find("td:eq(2)").text();
    var articleId =
        $(this).parents("tr").find("td:eq(5)").find("#delete-comment-btn").attr("delete-id");

```

```

var authorEmail = $(this).parents("tr").find("td:eq(4)").text();
var commentDate = $(this).parents("tr").find("td:eq(3)").text();
if(confirm("确认删除 " + articleAbstract + " 吗? ")){
    $.ajax({
        url: "comment",
        type: "PUT",
        data: {
            "articleId" : articleId,
            "email" : authorEmail,
            "commentDate" : commentDate
        },
        success: function (result) {
            alert("删除成功");
            to_comment_page(currentCommentPage);
        }
    });
}
})
function to_article_page(pn) {
    $.ajax({
        url:"articles/" + "<%=user.getUserId()%>",
        type:"GET",
        data: "pn=" + pn,
        success: function (result) {
            currentArticlePage = result.extend.pageInfo.pageNum;
            build_article_table(result);
            build_article_nav_bar(result);
        }
    });
}
function build_article_table(result) {
    $("#article-list tbody").empty();
    var articles = result.extend.pageInfo.list;
    $.each(articles, function (index, item) {
        var no = $("<td></td>").append((currentArticlePage - 1) * 10 + index + 1);
        var titleTd = $("<td></td>").append(item.articleTitle);
        var createTd = $("<td></td>").append(dateFormat(item.articlePostDate));
        var updateTd = $("<td></td>").append(dateFormat(item.articleEditTime));
        var viewtimeTd = $("<td></td>").append(item.articleViewTime);
        var commentTd = $("<td></td>").append(item.articleCommentCount);
        var likeTd = $("<td></td>").append(item.articleLikeCount);
        var editButton = $("<button id='edit-article-btn'></button>").addClass("edit-btn").append("
编辑");
        editButton.attr("edit-id", item.articleId);
        var deleteButton = $("<button id='delete-article-btn'></button>").addClass("delete-btn").ap
pend("删除");
        deleteButton.attr("delete-id", item.articleId);
        var operateTd = $("<td></td>").append(editButton).append(deleteButton);
        $("<tr></tr>").append(no)
        .append(titleTd)

```

```

        .append(createTd)
        .append(updateTd)
        .append(viewtimeTd)
        .append(commentTd)
        .append(likeTd)
        .append(operateTd)
        .appendTo("#article-list tbody")
    });
}
function build_article_nav_bar(result) {
    $("#article-page-nav-bar").empty();
    var ul = $("

</ul>").addClass("pagination");
    var pageInfo = result.extend.pageInfo;
    var firstPageLi =
        $("<li></li>").append($("<a></a>").append("首页").attr("href", "javascript:;"));
    var prePageLi = $("<li></li>").append($("<a></a>").append("&laquo;"));
    if(pageInfo.hasPreviousPage == false){
        firstPageLi.addClass("disabled");
        prePageLi.addClass("disabled");
    }else{
        firstPageLi.click(function(){
            to_article_page(1);
        });
        prePageLi.click(function(){
            to_article_page(pageInfo.pageNum - 1);
        });
    }
    var nextPageLi = $("<li></li>").append($("<a></a>").append("&raquo;").attr("href", "javascript:;"));
    var lastPageLi = $("<li></li>").append($("<a></a>").append("末页").attr("href", "javascript:;"));
    if(pageInfo.hasNextPage == false){
        nextPageLi.addClass("disabled");
        lastPageLi.addClass("disabled");
    }else{
        nextPageLi.click(function(){
            to_article_page(pageInfo.pageNum + 1);
        });
        lastPageLi.click(function(){
            to_article_page(pageInfo.pages);
        });
    }
    ul.append(firstPageLi).append(prePageLi);
    $.each(pageInfo.navigatepageNums, function(index, item){
        var numLi = $("<li></li>").append($("<a></a>").attr("href", "javascript:;").append(item));
        if(pageInfo.pageNum == item){
            numLi.addClass("active");
        }
        numLi.click(function(){
            to_article_page(item);
        });
    });
}

```

```

        ul.append(numLi);
    });
    ul.append(nextPageLi).append(lastPageLi);
    var navEle = $("<nav></nav>").append(ul);
    navEle.appendTo("#article-page-nav-bar");
}
function to_comment_page(pn) {
    $.ajax({
        url:"comment",
        type:"GET",
        data: "pn=" + pn,
        success: function (result) {
            currentCommentPage = result.extend.pageInfo.pageNum;
            build_comment_table(result);
            build_comment_nav_bar(result);
        }
    });
}
function build_comment_table(result) {
    $("#comment-list tbody").empty();
    var comments = result.extend.pageInfo.list;
    $.each(comments, function (index, item) {
        var no = $("<td></td>").append((currentCommentPage - 1) * 15 + index + 1);
        var titleTd = $("<td></td>").append(item.articleTitle);
        var commentAbstract = $("<td></td>").append(get_comment_abstract(item.commentContent));

        var postDate = $("<td></td>").append(dateFormat(item.commentDate));
        var authorEmail = $("<td></td>").append(item.authorEmail);
        var showButton =
            $("<button id='show-comment-btn'></button>").addClass("show-btn").append("查看");
        showButton.attr("show-id", item.articleId);
        var deleteButton =
            $("<button id='delete-comment-btn'></button>").addClass("delete-btn").append("删除");

        deleteButton.attr("delete-id", item.articleId);
        var operateTd = $("<td></td>").append(showButton).append(deleteButton);
        $("<tr></tr>").append(no)
            .append(titleTd)
            .append(commentAbstract)
            .append(postDate)
            .append(authorEmail)
            .append(operateTd)
            .appendTo("#comment-list tbody")
    })
}
function build_comment_nav_bar(result) {
    $("#comment-page-nav-bar").empty();
    var ul = $("<ul></ul>").addClass("pagination");
    var pageInfo = result.extend.pageInfo;
    var firstPageLi =

```

```

        $("<li></li>").append($("<a></a>").append("首页").attr("href", "javascript:;"));
var prePageLi = $("<li></li>").append($("<a></a>").append("&laquo;"));
if(pageInfo.hasPreviousPage == false){
    firstPageLi.addClass("disabled");
    prePageLi.addClass("disabled");
}else{
    firstPageLi.click(function(){
        to_comment_page(1);
    });
    prePageLi.click(function(){
        to_comment_page(pageInfo.pageNum - 1);
    });
}
var nextPageLi = $("<li></li>").append($("<a></a>").append("&raquo;").attr("href", "javascript:;"));
var lastPageLi = $("<li></li>").append($("<a></a>").append("末页").attr("href", "javascript:;"));
if(pageInfo.hasNextPage == false){
    nextPageLi.addClass("disabled");
    lastPageLi.addClass("disabled");
}else{
    nextPageLi.click(function(){
        to_comment_page(pageInfo.pageNum + 1);
    });
    lastPageLi.click(function(){
        to_comment_page(pageInfo.pages);
    });
}
ul.append(firstPageLi).append(prePageLi);
$.each(pageInfo.navigatepageNums,function(index,item){
    var numLi = $("<li></li>").append($("<a></a>").attr("href", "javascript:;").append(item));
    if(pageInfo.pageNum == item){
        numLi.addClass("active");
    }
    numLi.click(function(){
        to_comment_page(item);
    });
    ul.append(numLi);
});
ul.append(nextPageLi).append(lastPageLi);
var navEle = $("<nav></nav>").append(ul);
navEle.appendTo("#comment-page-nav-bar");
}

function dateFormat(date) {
    var s = new Date(date)
    var y = s.getFullYear()
    var m = (s.getMonth() + 1) < 10 ? '0' + (s.getMonth() + 1) : (s.getMonth() + 1)
    var dd = s.getDate() < 10 ? '0' + s.getDate() : s.getDate()
    var hh = s.getHours() < 10 ? '0' + s.getHours() : s.getHours()
    var mm = s.getMinutes() < 10 ? '0' + s.getMinutes() : s.getMinutes()
    var ss = s.getSeconds() < 10 ? '0' + s.getSeconds() : s.getSeconds()

```

```

        var enddate = y + '-' + m + '-' + dd + ' ' + hh + ':' + mm + ":" + ss
        return enddate
    }
    function get_comment_abstract(comment) {
        var str = "";
        if(comment.length > 10){
            str = comment.substring(0, 5) + "....." + comment.substr(-5);
        } else {
            str = comment;
        }
        return str;
    }
</script>
</body>
</html>
var article;
var subType = null;
var superType = null;
$(function(){
    get_article();
    get_sub_types();
    get_super_types()
    build_page();
    get_comments();
    $(".add-comment-btn").click(function (){
        let nickName = $("#edit-comment-author").val();
        let email = $("#edit-comment-email").val();
        let content = $("#edit-comment-content").val();
        var nameReg = /^[u4E00-\u9FA5A-Za-z0-9]{2,20}$/;
        if(!nameReg.test(nickName)){
            alert("用户名格式不对");
            return;
        }
        var emailReg = /^[a-z0-9_\.-]+@([\da-z\.-]+)\.([a-z]{2,6})$/;
        if(!emailReg.test(email)){
            alert("用户邮箱格式不对");
            return;
        }
        if("" === content){
            alert("内容不能为空");
            return;
        }
        $.ajax({
            url: "../comment",
            type: "POST",
            data: {
                "id" : article.articleId,
                "nickName": nickName,
                "email": email,
                "content" : content
            }
        })
    })
})

```

```

    },
    success: function (result) {
        let comment = result.extend.comment;
        $(".comment-list").append(create_comment_box(comment));
        add_article_comment_count();
        $("#edit-comment-author").val("");
        $("#edit-comment-email").val("");
        $("#edit-comment-content").val("");
    }
});
});
});
function get_comments(){
    $.ajax({
        url: "../comments",
        data: "id=" + article.articleId,
        type: "GET",
        success: function (result) {
            var comments = result.extend.comments;
            var list = $(".comment-list");
            $.each(comments, function (index, item) {
                list.append(create_comment_box(item));
            });
        }
    })
}
function create_comment_box(comment) {
    var commentBox = $("

</div>").addClass("comment-box clearfix");
    var commentAuthorInfo = $("

</div>").addClass("comment-author-info")
        .append($("<div class='comment-author'></div>").append("昵称: " + comment.authorNicknam
e))
        .append($("<div class='comment-ip'></div>").append("地址: " + comment.authorIp));
    var commentContent = $("

</div>").addClass("comment-conetnt").append(comment.commentC
ontent);
    var otherInfo = $("


```

```

        data: "id=" + articleId,
        success: function(result){
            article = result.extend.article;
        }
    });
}
function get_sub_types() {
    $.ajax({
        url: "../types",
        type: "POST",
        async: false,
        data: "id=" + article.articleType,
        success: function(result){
            subType = result.extend.type;
        }
    })
}
function get_super_types() {
    $.ajax({
        url: "../types",
        type: "POST",
        async: false,
        data: "id=" + subType.preTypeId,
        success: function(result){
            superType = result.extend.type;
        }
    })
}
function build_page(){
    $(".article-view-time").append("浏览量: " + article.articleViewTime);
    if(superType != null){
        var superPath = "../sortlist.jsp?type=" + superType.typeId;
        var superTypeA = $("
```

```

var s = new Date(date)
var y = s.getFullYear()
var m = (s.getMonth() + 1) < 10 ? '0' + (s.getMonth() + 1) : (s.getMonth() + 1)
var dd = s.getDate() < 10 ? '0' + s.getDate() : s.getDate()
var hh = s.getHours() < 10 ? '0' + s.getHours() : s.getHours()
var mm = s.getMinutes() < 10 ? '0' + s.getMinutes() : s.getMinutes()
var ss = s.getSeconds() < 10 ? '0' + s.getSeconds() : s.getSeconds()
var enddate = y + '-' + m + '-' + dd + ' ' + hh + ':' + mm + ":" + ss
return enddate
}
package com.Jancoyan.utils;
public class ArticleUtils {
    public static String NextLineToText(String str){
        String content = str.replaceAll("\n", "\\n");
        return content;
    }
    public static String getArticleAbstract(String innerHTML){
        String rst = innerHTML.replaceAll("<.*?>", "");
        rst = rst.replaceAll("\s", " ");
        if(rst.length() > 150){
            rst = rst.substring(0, 150) + ".....";
        }
        return rst;
    }
    public static String createArticle(String title, String userNickname, String innerHTML){
        String content = "<!DOCTYPE html>" +
            "<html>" +
            "<head>" +
            "<meta charset='utf-8'>" +
            "<title>" + title + "</title>" +
            "    <link rel='stylesheet' href='\" + ConstUtils.LINK_EDITOR_PATH + "\">" +
            "    <script src='\" + ConstUtils.SCRIPT_JQUERY_PATH + "\" type='text/javascri
pt" +
            "\" " +
            "charset='utf-8'></script>\n" +
            "    <script src='\" + ConstUtils.SCRIPT_PRELOAD_PATH + "\" type='text/javas
cript" +
            "\" " +
            "charset='utf-8'></script>\n" +
            "<link rel='stylesheet' href='\" + ConstUtils.LINK_ARTICLE_PATH + "\">" +
            "</head>" +
            "<body><!-- 导航条 -->" +
            "<div class='nav'>" +
            "    <!-- 回主页的图标 -->" +
            "    <a href='../index.jsp' class='back-to-index'>JancoBlog</a>" +
            "</div>" +
            "" +
            "<!-- 所属分类 -->" +
            "<div class='article-sort'>" +
            "</div>" +

```



```

        "<!-- 标题栏 -->" +
        "<h1 class=\"article-title\">" + title + "</h1>" +
        "<!-- 作者栏 -->" +
        "<div class=\"article-info\">" +
        "<div class=\"article-author\">作者: " + userNickname + "</div>" +
        "<div class=\"article-submit-time\">发布时间: " + TimeUtils.getCurrentTimeString()
+ "</div>" +
        "<div class=\"article-view-time\"></div>" +
        "</div><div class=\"article-content\">" + innerHTML + "</div>" +
        "<div class=\"article-comment\">" +
        "<div class=\"comment-title\">添加笔记</div> <hr>" +
        "<div class=\"comment-list\"></div>" +
        "<div class=\"add-comment\">" +
        "<textarea maxlength=\"20\" placeholder=\"昵称\" id=\"edit-comment-author\" rows=
\"1\" " +
        "cols=\"110\"></textarea>" +
        "<textarea maxlength=\"50\" placeholder=\"邮箱\" id=\"edit-comment-email\" " +
        "rows=\"1\" " +
        "cols=\"110\"></textarea>" +
        "<textarea maxlength=\"1000\" placeholder=\"笔记\" id=\"edit-comment-content\" "
+
        "rows=\"5\" " +
        "cols=\"110\"></textarea>" +
        "<button class=\"add-comment-btn\">添加</button>" +
        "</div>" +
        "</div>" +
        "</body>" +
        "</html>";
    return content;
}
}
package com.Jancoyan.utils;
import java.io.File;
public class ConstUtils {
    public static final String CURRENTPATH = System.getProperty("user.dir").replace("bin",
        "webapps") + File.separator + "JancoBlog";
    public static final String HTMLPATH = System.getProperty("user.dir").replace("bin",
        "webapps") + File.separator + "JancoBlog" + File.separator + "static" +
        File.separator + "p" + File.separator;
    public static final String MDPATH = System.getProperty("user.dir").replace("bin",
        "webapps") + File.separator + "JancoBlog" + File.separator + "static" +
        File.separator + "md" + File.separator;
    public static final String LINK_EDITOR_PATH = ".." + File.separator + "js" + File.separator +
        "editor.md" + File.separator + "css" + File.separator + "editormd.min.css";
    public static final String SCRIPT_JQUERY_PATH = ".." + File.separator + "js" + File.separator
+ "jquery-1.12.js";
    public static final String SCRIPT_PRELOAD_PATH = ".." + File.separator + "js" + File.separat
or + "article-preload.js";
    public static final String LINK_ARTICLE_PATH = ".." + File.separator + "css" + File.separator
+ "article.css";

```

```
}
package com.Jancoyan.utils;
import org.springframework.web.multipart.MultipartFile;
import java.io.*;
import java.util.ArrayList;
import java.util.List;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
public class FileIo {
    public static boolean deleteImagesInHtmlFile(String path){
        List<String> tmp = new ArrayList<>();
        String content = readFile(path);
        String regex = "<img (.*)>";
        Pattern pattern = Pattern.compile(regex);
        Matcher matcher = pattern.matcher(content);
        String prefix = path.substring(0, path.lastIndexOf("static"));
        while (matcher.find()){
            String temp = matcher.group(1).replace('/', '\\');
            temp = prefix + temp.substring(temp.indexOf("static"),temp.lastIndexOf("\" alt\""));
            temp = temp.replaceAll("[\\\\\\\\]", Matcher.quoteReplacement(File.separator));
            tmp.add(temp);
        }
        for (String s : tmp){
            deleteFileIfExists(s);
        }
        return true;
    }
    public static void deleteFileIfExists(String path){
        File file = new File(path);
        if(file.exists()){
            file.delete();
        }
    }
    public static void uploadPicture(MultipartFile multipartFile, String directory,String fileName){
        createDirectoryIfNotExists(directory);
        FileOutputStream fileOutputStream = null;
        String fullPath = directory + "/" + fileName;
        try {
            fileOutputStream = new FileOutputStream(fullPath);
            fileOutputStream.write(multipartFile.getBytes());
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                try {
                    fileOutputStream.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

```
    }
}

public static void createDirectoryIfNotExists(String path){
    File file = new File(path);
    if(!file.isDirectory()){
        file.mkdirs();
    }
}

public static void writeFile(String path, String content) {
    FileOutputStream fos = null;
    try {
        fos = new FileOutputStream(path, true);
        byte[] bytes = content.getBytes();
        fos.write(bytes);
        fos.flush();
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (fos != null) {
            try {
                fos.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}

public static boolean deleteFile(String path){
    File file = new File(path);
    if(!file.exists()){
        return false;
    }else{
        file.delete();
    }
    return true;
}

public static String readFile(String path) {
    FileReader reader = null;
    String content = "";
    try {
        reader = new FileReader(path);
        char[] chars = new char[40];
        int readCount = 0;
        while((readCount = reader.read(chars)) != -1) {
            content += new String(chars,0,readCount);
        }
    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        if (reader != null) {
```

```

        try {
            reader.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
return content;
}
}
package com.Jancoyan.utils;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class MD5Util {
    public static String getMD5(String password) {
        try {
            MessageDigest digest = MessageDigest.getInstance("md5");
            byte[] result = digest.digest(password.getBytes());
            StringBuffer buffer = new StringBuffer();
            for (byte b : result) {
                int number = b & 0xff;
                String str = Integer.toHexString(number);
                if (str.length() == 1) {
                    buffer.append("0");
                }
                buffer.append(str);
            }
            return buffer.toString();
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
            return "";
        }
    }
}
package com.Jancoyan.utils;
import java.util.HashMap;
import java.util.Map;
public class Msg {
    private int code;
    private String msg;
    private Map<String, Object> extend = new HashMap<String, Object>();
    public static Msg success(){
        Msg result = new Msg();
        result.setCode(100);
        result.setMsg("成功! ");
        return result;
    }
    public static Msg fail(){
        Msg result = new Msg();
        result.setCode(200);

```

```
        result.setMsg("失败! ");
        return result;
    }
    public Msg add(String key,Object value){
        this.getExtend().put(key, value);
        return this;
    }
    public int getCode() {
        return code;
    }
    public void setCode(int code) {
        this.code = code;
    }
    public String getMsg() {
        return msg;
    }
    public void setMsg(String msg) {
        this.msg = msg;
    }
    public Map<String, Object> getExtend() {
        return extend;
    }
    public void setExtend(Map<String, Object> extend) {
        this.extend = extend;
    }
}

package com.Jancoyan.utils;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
public class TimeUtils {
    public static Date castDateStrToDateTypeYMD(String dateStr){
        Date date = null;
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        try {
            date = sdf.parse(dateStr);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        return date;
    }
    public static Date caseDateStrToDateTypeYMDHMS(String dateStr){
        Date date = null;
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        try {
            date = sdf.parse(dateStr);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        return date;
    }
}
```

```

    }
    public static String getCurrentTimeString(){
        SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
        return df.format(new Date());
    }
    public static String getCurrentTimestamp(){
        return String.valueOf(System.currentTimeMillis());
    }
    public static String castDateTypeToDateString(Date date){
        SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd");
        return simpleDateFormat.format(date);
    }
}

package com.Jancoyan.test;
import org.mybatis.generator.api.MyBatisGenerator;
import org.mybatis.generator.config.Configuration;
import org.mybatis.generator.config.xml.ConfigurationParser;
import org.mybatis.generator.internal.DefaultShellCallback;
import java.io.File;
import java.util.ArrayList;
import java.util.List;
public class MBGTest {
    public static void main(String[] args) throws Exception {
        List<String> warnings = new ArrayList<String>();
        boolean overwrite = true;
        File configFile = new File("mbg.xml");
        ConfigurationParser cp = new ConfigurationParser(warnings);
        Configuration config = cp.parseConfiguration(configFile);
        DefaultShellCallback callback = new DefaultShellCallback(overwrite);
        MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config, callback, warnings);
        myBatisGenerator.generate(null);
        System.out.println("Success!");
    }
}

package com.Jancoyan.test;
import com.Jancoyan.utils.TimeUtils;
import org.junit.Test;
import java.util.Date;
public class TimeUtilsTest {
    @Test
    public void testCastTimeStringToDateType(){
        System.out.println(TimeUtils.castDateStringToDateTypeYMD("2021-11-23"));
    }
    @Test
    public void testCurrentTime(){
        String currentTime = TimeUtils.getCurrentTimeString();
        System.out.println(currentTime);
    }
    @Test
    public void testCurrentTimeMillions(){

```

```

        String timeMillion = TimeUtils.getCurrentTimestamp();
        System.out.println(timeMillion);
    }
}
package com.Jancoyan.test;
import com.Jancoyan.utils.FileIo;
import org.junit.Test;
public class FileIOTest {
    @Test
    public void readFileTest(){
        String path = "R:\\0.txt";
        String content = FileIo.readFile(path);
        System.out.println(content);
    }
}
package com.Jancoyan.test;
import com.Jancoyan.utils.ArticleUtils;
import org.junit.Test;
public class ArticleUtilsTest {
    @Test
    public void testNextLineToText(){
        String content = ArticleUtils.NextLineToText("\n\nasd\n");
        System.out.println(content);
    }
    @Test
    public void testArticleAbstract(){
        String testStr = "<div class=\"markdown-body editormd-preview-container\" previewcontainer
        =\"true\" style=\"padding: 20px;\"><h1 id=\"h1-ajax\"><a name=\"Ajax\" class=\"reference-link\"></a><
        span class=\"header-link octicon octicon-link\"></span>Ajax</h1><blockquote>\n" +
            "\t\t<p>实际开发中使用 JQuery + Ajax 较多，这部分内容作为了解即可。 </p>\n" +
            "\t\t<p>重点内容是 JQuery 的 【## JQuery + Ajax 中的普通查询和级联查询】<
        /p>\n" +
            "\t\t</blockquote>\n" +
            "\t\t<p>ajax:Asynchronous JavaScript and XML（异步的 JavaScript 和 XML）。</
        p>\n" +
            "\t\t<ul>\n";
        String rst = ArticleUtils.getArticleAbstract(testStr);
        System.out.println(rst);
    }
}
package com.Jancoyan.dao;
import com.Jancoyan.domain.User;
import com.Jancoyan.domain.UserExample;
import java.util.List;
import org.apache.ibatis.annotations.Param;
public interface UserMapper {
    long countByExample(UserExample example);
    int deleteByExample(UserExample example);
    int deleteByPrimaryKey(Integer userId);

```

```

    int insert(User record);
    int insertSelective(User record);
    List<User> selectByExample(UserExample example);
    User selectByPrimaryKey(Integer userId);
    int updateByExampleSelective(@Param("record") User record, @Param("example") UserExample
example);
    int updateByExample(@Param("record") User record, @Param("example") UserExample exampl
e);
    int updateByPrimaryKeySelective(User record);
    int updateByPrimaryKey(User record);
}
package com.Jancoyan.dao;
import com.Jancoyan.domain.Type;
import com.Jancoyan.domain.TypeExample;
import java.util.List;
import org.apache.ibatis.annotations.Param;
public interface TypeMapper {
    long countByExample(TypeExample example);
    int deleteByExample(TypeExample example);
    int deleteByPrimaryKey(Integer typeId);
    int insert(Type record);
    int insertSelective(Type record);
    List<Type> selectByExample(TypeExample example);
    Type selectByPrimaryKey(Integer typeId);
    int updateByExampleSelective(@Param("record") Type record, @Param("example") TypeExample
example);
    int updateByExample(@Param("record") Type record, @Param("example") TypeExample exampl
e);
    int updateByPrimaryKeySelective(Type record);
    int updateByPrimaryKey(Type record);
}
package com.Jancoyan.dao;
import com.Jancoyan.domain.Article;
import com.Jancoyan.domain.ArticleExample;
import java.util.List;
import org.apache.ibatis.annotations.Param;
public interface ArticleMapper {
    long countByExample(ArticleExample example);
    int deleteByExample(ArticleExample example);
    int deleteByPrimaryKey(String articleId);
    int insert(Article record);
    int insertSelective(Article record);
    List<Article> selectByExample(ArticleExample example);
    Article selectByPrimaryKey(String articleId);
    int updateByExampleSelective(@Param("record") Article record, @Param("example") ArticleExam
ple example);
    int updateByExample(@Param("record") Article record, @Param("example") ArticleExample exam
ple);
    int updateByPrimaryKeySelective(Article record);
    int updateByPrimaryKey(Article record);
}

```



```

}
package com.Jancoyan.controller;
import com.Jancoyan.domain.Article;
import com.Jancoyan.service.ArticleService;
import com.Jancoyan.service.TypeService;
import com.Jancoyan.utils.*;
import com.github.pagehelper.PageHelper;
import com.github.pagehelper.PageInfo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import java.io.File;
import java.io.UnsupportedEncodingException;
import java.util.Date;
import java.util.List;
@Controller
public class ArticleController {
    @Autowired
    ArticleService articleService;
    @Autowired
    TypeService typeService;

    @ResponseBody
    @RequestMapping(value = "/article/like", method = RequestMethod.POST)
    public Msg addLikeCount(String id, String count, HttpServletRequest request) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        Article article = new Article();
        article.setArticleId(id);
        Integer cnt = Integer.valueOf(count);
        cnt += 1;
        article.setArticleLikeCount(cnt);
        articleService.updateByPrimaryKeySelective(article);
        return Msg.success().add("count", cnt);
    }

    @ResponseBody
    @RequestMapping(value = "/articles", method = RequestMethod.PUT)
    public Msg getArticleByPrimaryKey(String id, HttpServletRequest request) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        Article article = articleService.selectByPrimaryKey(id);
        return Msg.success().add("article", article);
    }

    @ResponseBody
    @RequestMapping(value = "/articles", method = RequestMethod.POST)
    public Msg getHotArticles(HttpServletRequest request) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        List<Article> articles = articleService.getArticlesLimit();
    }

```

```

        return Msg.success().add("hotArticles", articles);
    }
    @ResponseBody
    @RequestMapping(value = "/article/update", method = RequestMethod.POST)
    public Msg updateArticle(
        String innerHTML,
        String innerMD,
        String userNickname,
        String title,
        Integer type,
        HttpSession session,
        HttpServletRequest request
    ) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        Article article = (Article) session.getAttribute("article");
        Article newArticle = new Article();
        String content = ArticleUtils.createArticle(title, userNickname, innerHTML);
        String fileName = article.getArticleId();
        String path = ConstUtils.CURRENTPATH + File.separator +
            "static" + File.separator +
            "p" + File.separator + fileName + ".html";
        FileIo.deleteFileIfExists(path);
        FileIo.writeFile(path, content);
        FileIo.createDirectoryIfNotExists(ConstUtils.CURRENTPATH + File.separator + "static" + File
e.separator + "md");
        String mdPath =
            ConstUtils.CURRENTPATH + File.separator + "static" +
                File.separator + "md" + File.separator
                    + fileName + ".md";
        FileIo.writeFile(mdPath, innerMD);
        newArticle.setArticleId(article.getArticleId());
        newArticle.setArticleType(type);
        newArticle.setArticleAbstract(ArticleUtils.getArticleAbstract(innerHTML));
        newArticle.setArticleTitle(title);
        newArticle.setArticleEditTime(new Date());
        articleService.updateByPrimaryKeySelective(newArticle);
        session.removeAttribute("article");
        session.removeAttribute("content");
        return Msg.success();
    }
    @ResponseBody
    @RequestMapping(value = "/article", method = RequestMethod.PUT)
    public Msg submitArticle(
        String innerHTML,
        String innerMD,
        String userId,
        String userNickname,
        String title,
        String type,
        HttpServletRequest request
    )

```

```

    ) throws UnsupportedOperationException {
        request.setCharacterEncoding("utf-8");
        String content = ArticleUtils.createArticle(title, userNickname, innerHTML);
        String fileName = userId + TimeUtils.getCurrentTimestamp();
        FileIo.createDirectoryIfNotExists(ConstUtils.CURRENTPATH + File.separator +
            "static"+File.separator + "p");
        String path = ConstUtils.HTMLPATH + fileName + ".html";
        FileIo.writeFile(path, content);
        FileIo.createDirectoryIfNotExists(ConstUtils.CURRENTPATH + File.separator + "static"
            + File.separator + "md");
        String mdPath = ConstUtils.MDPATH + fileName + ".md";
        FileIo.writeFile(mdPath, innerMD);
        Article article = new Article();
        article.setArticleId(fileName);
        article.setArticleAuthorId(Integer.parseInt(userId));
        article.setArticleAbstract(ArticleUtils.getArticleAbstract(innerHTML));
        article.setArticleTitle(title);
        article.setArticleType(Integer.parseInt(type));
        article.setArticleLikeCount(0);
        article.setArticleViewTime(0);
        article.setArticleCommentCount(0);
        Date now = new Date();
        article.setArticleEditTime(now);
        article.setArticlePostDate(now);
        articleService.submitArticle(article);
        return Msg.success();
    }
    @ResponseBody
    @RequestMapping(value = "/redirect/article", method = RequestMethod.POST)
    public void updateArticle(String id, HttpSession session, HttpServletRequest request) throws Uns
upportedEncodingException {
        request.setCharacterEncoding("utf-8");
        Article article = articleService.getArticleByPrimaryKey(id);
        String path = ConstUtils.MDPATH + id + ".md";
        String articleContent = FileIo.readFile(path);
        session.setAttribute("article", article);
        session.setAttribute("content", ArticleUtils.NextLineToText(articleContent));
    }
    @ResponseBody
    @RequestMapping(value = "/article/{id}", method = RequestMethod.DELETE)
    public Msg deleteArticle(
        @PathVariable("id") String id,
        HttpServletRequest request
    ) throws UnsupportedOperationException {
        request.setCharacterEncoding("utf-8");
        articleService.deleteByPrimaryKey(id);
        String path = ConstUtils.HTMLPATH + id + ".html";
        String mdPath = ConstUtils.MDPATH + id + ".md";
        boolean success =
            FileIo.deleteImagesInHtmlFile(path) &&

```

```

        FileIo.deleteFile(path) &&
        FileIo.deleteFile(mdPath);
        return Msg.success().add("success", success);
    }
    @ResponseBody
    @RequestMapping(value = "/articles", method = RequestMethod.GET, produces = "application/js
on; charset=utf-8")
    public Msg getIndexArticles(Integer pn, HttpServletRequest request) throws UnsupportedEncoding
Exception {
        request.setCharacterEncoding("utf-8");
        PageHelper.startPage(pn, 10);
        List<Article> articles = articleService.getAll();
        PageInfo<Article> pageInfo = new PageInfo<>(articles, 5);
        return Msg.success().add("pageInfo", pageInfo);
    }
    @ResponseBody
    @RequestMapping(value = "/articles/{id}", method = RequestMethod.GET)
    public Msg getArticlesById(
        @RequestParam(value = "pn", defaultValue = "1") Integer pn,
        @PathVariable("id") Integer id,
        HttpServletRequest request
    ) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        PageHelper.startPage(pn, 10);
        List<Article> articles = articleService.getArticlesByUserId(id);
        PageInfo<Article> pageInfo = new PageInfo<>(articles, 5);
        return Msg.success().add("pageInfo", pageInfo);
    }
    @ResponseBody
    @RequestMapping(value = "/type/{id}", method = RequestMethod.GET)
    public Msg getArticlesByType(
        @PathVariable("id") Integer id,
        @RequestParam(value = "pn", defaultValue = "1") Integer pn,
        HttpServletRequest request
    ) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        PageHelper.startPage(pn, 5);
        List<Article> articles;
        if(id < 100){
            articles = articleService.getArticlesBySuperId(id);
        }else{
            articles = articleService.getArticlesByTypeId(id);
        }
        PageInfo<Article> pageInfo = new PageInfo<>(articles, 5);
        return Msg.success().add("pageInfo", pageInfo);
    }
    @ResponseBody
    @RequestMapping(value = "/search", method = RequestMethod.GET)
    public Msg searchArticle(
        @RequestParam(value = "keyword") String name,

```

```

        @RequestParam(value = "pn", defaultValue = "1") Integer pn,
        HttpServletRequest request
    ) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        PageHelper.startPage(pn, 10);
        List<Article> articles = articleService.getArticlesLikeName(name);
        PageInfo<Article> pageInfo = new PageInfo<>(articles, 5);
        return Msg.success().add("pageInfo", pageInfo);
    }
    @ResponseBody
    @RequestMapping(value = "/article/view", method = RequestMethod.POST)
    public Msg addViewCount(String id, HttpServletRequest request) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        Article article = new Article();
        article.setArticleId(id);
        article.setArticleViewTime(articleService.selectByPrimaryKey(id).getArticleViewTime() + 1);
        articleService.updateByPrimaryKeySelective(article);
        return Msg.success();
    }
    @ResponseBody
    @RequestMapping(value = "/article/comment", method = RequestMethod.POST)
    public Msg addCommentCount(String id, HttpServletRequest request) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        Article article = new Article();
        article.setArticleId(id);
        article.setArticleCommentCount(articleService.selectByPrimaryKey(id).getArticleCommentCount() + 1);
        articleService.updateByPrimaryKeySelective(article);
        return Msg.success();
    }
}

package com.Jancoyan.controller;
import com.Jancoyan.domain.ArticleComment;
import com.Jancoyan.domain.ArticleCommentKey;
import com.Jancoyan.service.CommentService;
import com.Jancoyan.utils.Msg;
import com.Jancoyan.utils.TimeUtils;
import com.github.pagehelper.PageHelper;
import com.github.pagehelper.PageInfo;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import javax.servlet.http.HttpServletRequest;
import java.io.UnsupportedEncodingException;
import java.util.Date;
import java.util.List;

```

```

@Controller
public class CommentController {
    @Autowired
    CommentService commentService;
    @ResponseBody
    @RequestMapping(value = "/getcomment", method = RequestMethod.GET)
    public Msg getCommentByPrimaryKey(
        String articleId,
        String email,
        String commentDate,
        HttpServletRequest request
    ) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        ArticleCommentKey articleCommentKey = new ArticleCommentKey();
        articleCommentKey.setAuthorEmail(email);
        articleCommentKey.setArticleId(articleId);
        articleCommentKey.setCommentDate(TimeUtils.caseDateStringToDateTypeYMDHMS(comment
Date));
        ArticleComment articleComment = commentService.getCommentByPrimaryKey(articleComme
ntKey);
        return Msg.success().add("comment", articleComment);
    }
    @ResponseBody
    @RequestMapping(value = "/comment", method = RequestMethod.PUT)
    public Msg deleteComment(
        String articleId,
        String email,
        String commentDate,
        HttpServletRequest request
    ) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        ArticleCommentKey articleCommentKey = new ArticleCommentKey();
        articleCommentKey.setArticleId(articleId);
        articleCommentKey.setAuthorEmail(email);
        articleCommentKey.setCommentDate(TimeUtils.caseDateStringToDateTypeYMDHMS(comment
Date));
        commentService.deleteByPrimaryKey(articleCommentKey);
        return Msg.success();
    }
    @ResponseBody
    @RequestMapping(value = "/comment", method = RequestMethod.GET)
    public Msg getAll(
        Integer pn,
        HttpServletRequest request
    ) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        PageHelper.startPage(pn, 15);
        List<ArticleComment> comments = commentService.getAll();
        PageInfo<ArticleComment> pageInfo = new PageInfo<>(comments, 5);
        return Msg.success().add("pageInfo", pageInfo);
    }
}

```

```

    }
    @ResponseBody
    @RequestMapping(value = "/comment", method = RequestMethod.POST)
    public Msg insertComment(
        String id,
        String nickName,
        String email,
        String content,
        HttpServletRequest request) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        ArticleComment articleComment = new ArticleComment();
        articleComment.setCommentContent(content);
        articleComment.setArticleId(id);
        articleComment.setCommentDate(new Date());
        articleComment.setAuthorEmail(email);
        articleComment.setAuthorNickname(nickName);
        articleComment.setAuthorIp(request.getRemoteAddr());
        commentService.insertComment(articleComment);
        return Msg.success().add("comment", articleComment);
    }
    @ResponseBody
    @RequestMapping(value = "/comments", method = RequestMethod.GET)
    public Msg getCommentByArticleId(String id,
                                     HttpServletRequest request) throws UnsupportedEncodingE
xception {
        request.setCharacterEncoding("utf-8");
        List<ArticleComment> articleComments =
            commentService.getCommentByArticleId(id);
        return Msg.success().add("comments", articleComments);
    }
}

package com.Jancoyan.controller;
import com.Jancoyan.domain.User;
import com.Jancoyan.utils.FileIo;
import com.Jancoyan.utils.TimeUtils;
import com.alibaba.fastjson.JSONObject;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.multipart.MultipartFile;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import java.io.UnsupportedEncodingException;
@Controller
public class PictureController {
    @ResponseBody
    @RequestMapping(value = "/uploadarticlepicture", method = RequestMethod.POST)
    public JSONObject uploadArticlePicture(@RequestParam(value = "editormd-image-file",

```

```

        required = true)MultipartFile multipartFile,
        HttpServletRequest request,
        HttpSession session) throws UnsupportedEncodingException
    {
        request.setCharacterEncoding("utf-8");
        User user = (User) session.getAttribute("user");
        String userId = String.valueOf(user.getUserId());
        String trueFileName = multipartFile.getOriginalFilename();
        String suffix = trueFileName.substring(trueFileName.lastIndexOf("."));
        String now = TimeUtils.getCurrentTimeString();
        String fileName = now.replaceAll("[:]", "").replace(' ', '-') + userId + suffix;
        String directory = now.substring(0, 7);
        String path =
            request.getSession().getServletContext().getRealPath(
                "/static/image/" + directory
            );
        FileIo.uploadPicture(multipartFile, path, fileName);
        String backUrl = request.getScheme() + "://" +
            request.getServerName() + ":" + request.getServerPort() +
            request.getContextPath() +
            "/static/image/" + directory + "/" + fileName;
        JSONObject rst = new JSONObject();
        rst.put("url", backUrl);
        rst.put("success", 1);
        rst.put("message", "success");
        return rst;
    }
}

package com.Jancoyan.controller;
import com.Jancoyan.domain.Type;
import com.Jancoyan.service.TypeService;
import com.Jancoyan.utils.Msg;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import javax.servlet.http.HttpServletRequest;
import java.io.UnsupportedEncodingException;
import java.util.List;

@Controller
public class TypeController {
    @Autowired
    TypeService typeService;
    @ResponseBody
    @RequestMapping(value = "/types", method = RequestMethod.GET)
    public Msg getAll(HttpServletRequest request) throws UnsupportedEncodingException {
        request.setCharacterEncoding("utf-8");
        List<Type> typeList = typeService.getAll();
        return Msg.success().add("types", typeList);
    }
}

```

```

    }
    @ResponseBody
    @RequestMapping(value = "/types", method = RequestMethod.POST)
    public Msg getTYPEByPrimaryKEY(String id, HttpServletRequest request) throws UnsupportedEnc
odingException {
        request.setCharacterEncoding("utf-8");
        Type type = typeService.getTYPEByPrimaryKEY(Integer.parseInt(id));
        return Msg.success().add("type", type);
    }
}

package com.Jancoyan.controller;
import com.Jancoyan.domain.User;
import com.Jancoyan.service.UserService;
import com.Jancoyan.utils.MD5Util;
import com.Jancoyan.utils.Msg;
import com.Jancoyan.utils.TimeUtils;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.ResponseBody;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;
import java.io.UnsupportedEncodingException;
@Controller
public class UserController {
    @Autowired
    UserService userService;
    @ResponseBody
    @RequestMapping(value = "/user/password", method = RequestMethod.POST)
    public Msg changePassword(String oldPwd,
                               String newPwd,
                               String repeat,
                               HttpSession session,
                               HttpServletRequest request) throws UnsupportedEncodingException
    {
        request.setCharacterEncoding("utf-8");
        User user = (User) session.getAttribute("user");
        if (!user.getUserPwd().equals(MD5Util.getMD5(oldPwd))){
            return Msg.success().add("msg", "旧密码错误");
        } else if (!newPwd.equals(repeat)){
            return Msg.success().add("msg", "两次密码不一致");
        }
        user.setUserPwd(MD5Util.getMD5(newPwd));
        userService.updateUserSelective(user);
        session.setAttribute("user", user);
        return Msg.success().add("msg", "修改成功");
    }
}

@ResponseBody
@RequestMapping(value = "/user", method = RequestMethod.POST)

```

```

public Msg updateUser(String userNickName,
                      String userEmail,
                      String userSex,
                      String userBirthdayDay,
                      HttpSession session,
                      HttpServletRequest request) throws UnsupportedEncodingException {
    request.setCharacterEncoding("utf-8");
    User user = (User) session.getAttribute("user");
    user.setUserNickname(userNickName);
    user.setUserEmail(userEmail);
    user.setUserSex(Integer.valueOf(userSex));
    user.setUserBirthday(TimeUtils.castDateToStringToDateTypeYMD(userBirthdayDay));
    userService.updateUserSelective(user);
    session.setAttribute("user", user);
    return Msg.success();
}

@ResponseBody
@RequestMapping(value = "/login", method = RequestMethod.POST)
public Msg login(User user, HttpSession session, HttpServletRequest request) throws UnsupportedEncodingException {
    request.setCharacterEncoding("utf-8");
    user.setUserPwd(MD5Util.getMD5(user.getUserPwd()));
    user = userService.login(user);
    if(user != null){
        session.setAttribute("user", user);
        return Msg.success().add("user", user);
    }else{
        return Msg.fail().add("user", null);
    }
}
}

package com.Jancoyan.service;
import com.Jancoyan.dao.ArticleMapper;
import com.Jancoyan.domain.Article;
import com.Jancoyan.domain.ArticleExample;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.ArrayList;
import java.util.List;

@Service
public class ArticleService {
    @Autowired
    ArticleMapper articleMapper;
    public List<Article> getAll() {
        ArticleExample example = new ArticleExample();
        example.setOrderByClause("article_post_date DESC");
        return articleMapper.selectByExample(example);
    }

    public Article selectByPrimaryKey(String id){
        return articleMapper.selectByPrimaryKey(id);
    }
}

```

```

    }
    public List<Article> getArticlesByUserId(Integer id) {
        ArticleExample example = new ArticleExample();
        ArticleExample.Criteria criteria = example.createCriteria();
        criteria.andArticleAuthorIdEqualTo(id);
        List<Article> articles = articleMapper.selectByExample(example);
        return articles;
    }
    public List<Article> getArticlesLikeName(String articleName) {
        ArticleExample articleExample = new ArticleExample();
        ArticleExample.Criteria criteria = articleExample.createCriteria();
        String pattern = "%" + articleName + "%";
        criteria.andArticleTitleLike(pattern);
        List<Article> articles = articleMapper.selectByExample(articleExample);
        return articles;
    }
    public void submitArticle(Article article) {
        articleMapper.insert(article);
    }
    public void deleteByPrimaryKey(String id) {
        articleMapper.deleteByPrimaryKey(id);
    }
    public void updateByPrimaryKeySelective(Article article) {
        articleMapper.updateByPrimaryKeySelective(article);
    }
    public Article getArticleByPrimaryKey(String id) {
        return articleMapper.selectByPrimaryKey(id);
    }
    public List<Article> getArticlesLimit() {
        ArticleExample example = new ArticleExample();
        example.setOrderByClause("article_view_time desc limit 10");
        List<Article> articles = articleMapper.selectByExample(example);
        return articles;
    }
    public List<Article> getArticlesBySuperId(Integer id) {
        ArticleExample example = new ArticleExample();
        ArticleExample.Criteria criteria = example.createCriteria();
        List<Integer> ids = new ArrayList<>(100);
        ids.add(id);
        for(int i = id * 100 + 1; i < 100*(id + 1); i++)
            ids.add(i);
        criteria.andArticleTypeIn(ids);
        List<Article> articles = articleMapper.selectByExample(example);
        return articles;
    }
    public List<Article> getArticlesByTypeId(Integer id) {
        ArticleExample example = new ArticleExample();
        ArticleExample.Criteria criteria = example.createCriteria();
        criteria.andArticleTypeEqualTo(id);
        List<Article> articles = articleMapper.selectByExample(example);
    }

```

```

        return articles;
    }
}

package com.Jancoyan.service;
import com.Jancoyan.dao.ArticleCommentMapper;
import com.Jancoyan.domain.ArticleComment;
import com.Jancoyan.domain.ArticleCommentExample;
import com.Jancoyan.domain.ArticleCommentKey;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;
@Service
public class CommentService {
    @Autowired
    ArticleCommentMapper articleCommentMapper;
    public List<ArticleComment> getCommmentByArticleId(String id) {
        ArticleCommentExample example = new ArticleCommentExample();
        example.setOrderByClause("comment_date desc");
        ArticleCommentExample.Criteria criteria = example.createCriteria();
        criteria.andArticleIdEqualTo(id);
        List<ArticleComment> comments = articleCommentMapper.selectByExampleWithBLOBs(exa
mple);
        return comments;
    }
    public void insertComment(ArticleComment articleComment) {
        articleCommentMapper.insert(articleComment);
    }
    public List<ArticleComment> getAll() {
        return articleCommentMapper.selectWithArticleTitle();
    }
    public void deleteByPrimaryKey(ArticleCommentKey articleCommentKey) {
        articleCommentMapper.deleteByPrimaryKey(articleCommentKey);
    }
    public ArticleComment getCommentByPrimaryKey(ArticleCommentKey articleCommentKey) {
        ArticleCommentExample example = new ArticleCommentExample();
        ArticleCommentExample.Criteria criteria = example.createCriteria();
        criteria.andArticleIdEqualTo(articleCommentKey.getArticleId());
        criteria.andCommentDateEqualTo(articleCommentKey.getCommentDate());
        criteria.andAuthorEmailEqualTo(articleCommentKey.getAuthorEmail());
        List<ArticleComment> comments = articleCommentMapper.selectByExampleWithBLOBs(exa
mple);
        return comments.get(0);
    }
}

package com.Jancoyan.service;
import com.Jancoyan.dao.TypeMapper;
import com.Jancoyan.domain.Type;
import com.Jancoyan.domain.TypeExample;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

```

```

import java.util.List;
@Service
public class TypeService {
    @Autowired
    TypeMapper typeMapper;
    public Type getTypeByPrimarykey(Integer id){
        return typeMapper.selectByPrimaryKey(id);
    }
    public List<Type> getAll(){
        TypeExample typeExample = new TypeExample();
        TypeExample.Criteria criteria = typeExample.createCriteria();
        criteria.andPreTypeIdIsNull();
        List<Type> typeList = typeMapper.selectByExample(typeExample);
        for (Type type : typeList){
            TypeExample typeExample1 = new TypeExample();
            TypeExample.Criteria criteria1 = typeExample1.createCriteria();
            criteria1.andPreTypeIdEqualTo(type.getTypeId());
            List<Type> subTypes = typeMapper.selectByExample(typeExample1);
            type.setSubTypes(subTypes);
        }
        return typeList;
    }
}

package com.Jancoyan.service;
import com.Jancoyan.dao.UserMapper;
import com.Jancoyan.domain.User;
import com.Jancoyan.domain.UserExample;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import java.util.List;
@Service
public class UserService {
    @Autowired
    UserMapper userMapper;
    public User updateUserSelective(User user) {
        userMapper.updateByPrimaryKeySelective(user);
        return user;
    }
    public User exitLogin(User user) {
        UserExample example = new UserExample();
        UserExample.Criteria criteria = example.createCriteria();
        criteria.andUserNameEqualTo(user.getUserName());
        criteria.andUserPwdEqualTo(user.getUserPwd());
        List<User> users = userMapper.selectByExample(example);
        if(users.size() == 1){
            return users.get(0);
            System.out.println("已退出");
        } else return null;
    }
}

```