**JancoBlog个人博客V2.0**

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\config\AllowOriginIntercepter.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/9/16

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.config;

import org.springframework.web.servlet.HandlerInterceptor;

import org.springframework.web.servlet.ModelAndView;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

public class AllowOriginIntercepter implements HandlerInterceptor {

@Override

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object handler) {

// 过滤器

// response.setHeader("Access-Control-Allow-Origin", "\*");

response.setHeader("Access-Control-Allow-Origin", request.getHeader("Origin"));

response.setHeader("Access-Control-Allow-Credentials", "true");

response.setHeader("Access-Control-Allow-Methods","GET, POST, OPTIONS, PUT");

response.setHeader("Access-Control-Allow-Headers","Authorization, Content-Type, token");

return true;

}

@Override

public void postHandle(HttpServletRequest request, HttpServletResponse response, Object handler,

ModelAndView modelAndView) throws Exception {

}

@Override

public void afterCompletion(HttpServletRequest request, HttpServletResponse response, Object handler, Exception ex)

throws Exception {

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\config\MvcConfig.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/9/16

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.config;

import com.jancoyan.jancoblog.utils.ConstantUtil;

import org.springframework.boot.system.ApplicationHome;

import org.springframework.context.annotation.Configuration;

import org.springframework.web.servlet.config.annotation.InterceptorRegistry;

import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;

import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

import java.io.File;

@Configuration

public class MvcConfig implements WebMvcConfigurer {

/\*\*

\* 将jar文件下的对应静态资源文件路径对应到磁盘的路径(根据个人的情况修改"file:static/"的static的值)

\*

\* @param registry

\*/

@Override

public void addResourceHandlers(ResourceHandlerRegistry registry) {

String path = "file:" + ConstantUtil.STATIC\_RESOURCES;

registry.

addResourceHandler("/static/\*\*").

addResourceLocations("classpath:/static/", path);

}

// 拦截器

@Override

public void addInterceptors(InterceptorRegistry registry) {

registry.addInterceptor(new AllowOriginIntercepter());

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\config\MybatisPlusConfig.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/9/14

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.config;

import com.baomidou.mybatisplus.annotation.DbType;

import com.baomidou.mybatisplus.extension.plugins.MybatisPlusInterceptor;

import com.baomidou.mybatisplus.extension.plugins.inner.PaginationInnerInterceptor;

import org.mybatis.spring.annotation.MapperScan;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.transaction.annotation.EnableTransactionManagement;

@EnableTransactionManagement

@Configuration

@MapperScan("com.jancoyan.jancoblog.mapper")

public class MybatisPlusConfig {

// 分页拦截器对象，目前分页不支持表连接，只能单表查询

@Bean

public MybatisPlusInterceptor mybatisPlusInterceptor() {

MybatisPlusInterceptor interceptor = new MybatisPlusInterceptor();

interceptor.addInnerInterceptor(new PaginationInnerInterceptor(DbType.MYSQL));

return interceptor;

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\config\RedisConfig.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/9/25

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.config;

import com.fasterxml.jackson.annotation.JsonAutoDetect;

import com.fasterxml.jackson.annotation.PropertyAccessor;

import com.fasterxml.jackson.databind.ObjectMapper;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import org.springframework.data.redis.connection.RedisConnectionFactory;

import org.springframework.data.redis.core.RedisTemplate;

import org.springframework.data.redis.serializer.Jackson2JsonRedisSerializer;

import org.springframework.data.redis.serializer.StringRedisSerializer;

@Configuration

public class RedisConfig {

@Bean

@SuppressWarnings("all")

public RedisTemplate<String, Object> redisTemplate(

RedisConnectionFactory factory) {

RedisTemplate<String, Object> template =

new RedisTemplate<String, Object>();

template.setConnectionFactory(factory);

Jackson2JsonRedisSerializer jackson2JsonRedisSerializer =

new Jackson2JsonRedisSerializer(Object.class);

ObjectMapper om = new ObjectMapper();

om.setVisibility(PropertyAccessor.ALL, JsonAutoDetect.Visibility.ANY);

om.enableDefaultTyping(ObjectMapper.DefaultTyping.NON\_FINAL);

jackson2JsonRedisSerializer.setObjectMapper(om);

StringRedisSerializer stringRedisSerializer =

new StringRedisSerializer();

// key采用String的序列化方式

template.setKeySerializer(stringRedisSerializer);

// hash的key也采用String的序列化方式

template.setHashKeySerializer(stringRedisSerializer);

// value序列化方式采用jackson

template.setValueSerializer(jackson2JsonRedisSerializer);

// hash的value序列化方式采用jackson

template.setHashValueSerializer(jackson2JsonRedisSerializer);

template.afterPropertiesSet();

return template;

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\controller\ArticleController.java

package com.jancoyan.jancoblog.controller;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.\*;

import com.jancoyan.jancoblog.service.ArticleService;

import com.jancoyan.jancoblog.service.CommentService;

import com.jancoyan.jancoblog.service.LikeRecordService;

import com.jancoyan.jancoblog.utils.\*;

import org.springframework.beans.factory.annotation.Autowired;

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.RestController;

import org.springframework.web.multipart.MultipartFile;

import javax.servlet.http.HttpServletRequest;

import java.io.File;

import java.io.IOException;

import java.io.UnsupportedEncodingException;

import java.sql.SQLIntegrityConstraintViolationException;

import java.util.Date;

import java.util.List;

import java.util.Objects;

import java.util.UUID;

/\*\*

\* <p>

\* 前端控制器

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@RestController

@RequestMapping("/article")

public class ArticleController {

@Autowired

ArticleService service;

@Autowired

CommentService commentService;

@Autowired

LikeRecordService likeRecordService;

@Autowired

RedisUtil redisUtil;

/\*\*

\* 获取首页的文章，带有搜索功能

\* @param pn 第几页

\* @param limit 每一页的大小

\* @param condition 条件

\* @return 标准的pageInfo

\*/

@RequestMapping(value = "/all")

public Msg listArticleIndex(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition

){

IPage<Article> iPage = service.listArticleIndex(Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 获取管理的文章管理列表的文章

\* @param pn 第几页

\* @param limit 容量

\* @param condition 条件

\* @return 成功

\*/

@RequestMapping(value = "/manage")

public Msg listArticleManageAll(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition,

HttpServletRequest request

){

String token = request.getHeader("token");

if(null == token){

// 用户登录信息过期了

return Msg.expire();

}

IPage<Article> iPage = service.listArticleManage(null,

Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 获取当前登录的用户发表的所有文章

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @param request 获取token

\* @return 成功

\*/

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

public Msg listArticleManageUser(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition,

HttpServletRequest request){

// 从token中拿到用户

String token = request.getHeader("token");

if(null == token){

// 用户信息已经过期了

return Msg.expire();

}

User user = (User) redisUtil.get(token);

if(null == user){

return Msg.fail();

}

IPage<Article> iPage = service.listArticleManage(

user.getUserId(),

Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 获取全站所有删除的文章

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @param request 获取token

\* @return

\*/

@RequestMapping(value = "/deleted/all", method = RequestMethod.GET)

public Msg listDeletedAll(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition,

HttpServletRequest request

){

String token = request.getHeader("token");

if(null == token){

// 用户登录信息过期了

return Msg.expire();

}

IPage<Article> iPage = service.listDeleted(

null,

Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 获取用户删除的文章

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @param request 获取token

\* @return

\*/

@RequestMapping(value = "/deleted/user", method = RequestMethod.GET)

public Msg listDeletedUser(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition,

HttpServletRequest request

){

// 从token中拿到用户

String token = request.getHeader("token");

if(null == token){

// 用户信息已经过期了

return Msg.fail();

}

User user = (User) redisUtil.get(token);

if(null == user){

return Msg.expire();

}

IPage<Article> iPage = service.listDeleted(

user.getUserId(),

Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 彻底删除已经删除了的文章

\* @param ids id

\* @param request

\* @return

\*/

@RequestMapping(value = "/deleted/delete", method = RequestMethod.POST)

public Msg deleteArticleDeleted(

String ids,

HttpServletRequest request

){

String token = request.getHeader("token");

if(null == token){

// 未登录

return Msg.fail();

}

boolean suc = service.deleteCompletely(ids);

return Msg.success().add("suc", suc);

}

/\*\*

\* 批量恢复用户已经删除的文章

\* @param ids id

\* @param request

\* @return

\*/

@RequestMapping(value = "/deleted/recover", method = RequestMethod.POST)

public Msg recoverArticle(

String ids,

HttpServletRequest request

){

// 验证用户登录

String token = request.getHeader("token");

if(null == token){

return Msg.fail();

}

// 恢复删除的文章

boolean suc = service.batchRecoverDeletedArticle(ids);

// 恢复删除的评论

commentService.recoverCommentByArticle(ids);

return Msg.success().add("suc", suc);

}

/\*\*

\* 批量删除文章

\* @param ids 文章id，以 & 连接

\* @return 成功/失败

\*/

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

public Msg deleteArticle(

String ids,

HttpServletRequest request

){

String token = request.getHeader("token");

if(null == token){

// 用户未登录

return Msg.fail();

}

// 批量删除文章

boolean suc = service.batchDeleteArticle(ids);

commentService.deleteCommentByArticle(ids);

return Msg.success().add("suc", suc);

}

/\*\*

\* 获取文章用于编辑

\* @param id 文章id

\* @param request

\* @return

\*/

@RequestMapping(value = "/edit", method = RequestMethod.GET)

public Msg getArticleEdit(

@RequestParam(value = "id") String id,

HttpServletRequest request){

String token = request.getHeader("token");

if(null == token){

// 用户未登录

return Msg.fail();

}

Article article = service.getArticleEdit(id);

return Msg.success().add("article", article);

}

/\*\*

\* 查看文章的时候获取单个文章

\* @param articleId 文章ID

\* @return 成功

\*/

@RequestMapping(value = "/single", method = RequestMethod.GET)

public Msg getArticleSingle(

@RequestParam(value = "id") String articleId

){

Article article = service.getArticleSingle(articleId);

return Msg.success().add("article", article);

}

/\*\*

\* 查看被删除的文章的时候获取单个文章

\* @param articleId 文章ID

\* @return 成功

\*/

@RequestMapping(value = "/single/deleted", method = RequestMethod.GET)

public Msg getArticleSingleDeleted(

@RequestParam(value = "id") String articleId

){

Article article = service.getArticleSingleDeleted(articleId);

return Msg.success().add("article", article);

}

/\*\*

\* 获取用户最近发布的文章 10 个

\* @param id 用户id

\* @param pn 页码

\* @param limit 容量

\* @return

\*/

@RequestMapping(value = "/recent", method = RequestMethod.GET)

public Msg listArticleUserRecently(

@RequestParam(value = "id") String id,

@RequestParam(value = "pn", defaultValue = "1")Integer pn,

@RequestParam(value = "limit" ,defaultValue = "10")Integer limit

){

IPage<PageArticle> iPage = service.listArticleUserRecently(id, pn, limit);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 点赞, 游客不能点赞

\* @param id 点赞的文章

\* @return 成功

\*/

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

public Msg addLikeCount(

@RequestParam(value = "id")String id,

HttpServletRequest request

){

// 登录认证

String token = request.getHeader("token");

if(null == token){

// 未登录，说明是游客

return Msg.loginNeeded();

}

User user = (User) redisUtil.get(token);

if(null == user){

return Msg.expire();

}

service.addLikeCount(id);

likeRecordService.insertRecord(user.getUserId(), id);

return Msg.success();

}

/\*\*

\* 取消点赞

\* @param id 点赞的文章

\* @return 成功

\*/

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

public Msg subLikeCount(

@RequestParam(value = "id")String id,

HttpServletRequest request

){

// 登录认证

String token = request.getHeader("token");

if(null == token){

// 未登录，说明是游客

return Msg.loginNeeded();

}

User user = (User) redisUtil.get(token);

service.subLikeCount(id);

likeRecordService.deleteRecord(user.getUserId(), id);

return Msg.success();

}

/\*\*

\* 浏览

\* @param id 点开的文章的id

\* @return 成功

\*/

@RequestMapping(value = "/view", method = RequestMethod.GET)

public Msg addViewCount(@RequestParam(value = "id")String id){

service.addViewCount(id);

return Msg.success();

}

/\*\*

\* 改变当前评论的状态

\* @param id 评论的id

\* @return

\*/

@RequestMapping(value = "/toggle/comment", method = RequestMethod.POST)

public Msg updateIsComment(

@RequestParam(value = "id")String id,

HttpServletRequest request

){

String token = request.getHeader("token");

if(null == token){

// 用户未登录

return Msg.fail();

}

boolean suc = service.updateIsComment(id);

if(suc) {

return Msg.success();

} else {

return Msg.fail();

}

}

/\*\*

\* 置顶指定的博文

\* @param id 博文的id

\* @return

\*/

@RequestMapping(value = "/toggle/top", method = RequestMethod.POST)

public Msg updateArticleTop(

@RequestParam(value = "id")String id,

HttpServletRequest request

){

String token = request.getHeader("token");

if(null == token){

// 用户未登录

return Msg.fail();

}

boolean suc = service.updateIsTop(id);

if(suc) {

return Msg.success();

} else {

return Msg.fail();

}

}

/\*\*

\* 文章中图片上传

\* @param file 图片

\* @param request

\* @return

\* @throws IOException

\*/

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

public Msg uploadPicture(

@RequestParam(value = "file") MultipartFile file,

HttpServletRequest request

) throws IOException {

if (file == null) {

return Msg.fail().add("msg", "请选择要上传的图片");

}

if (file.getSize() > 1024 \* 1024 \* 10) {

return Msg.fail().add("msg", "文件大小不能大于10M");

}

//获取文件后缀

String suffix = Objects.requireNonNull(file.getOriginalFilename()).substring(file.getOriginalFilename().lastIndexOf(".") + 1);

if (!"jpg,jpeg,gif,png".toUpperCase().contains(suffix.toUpperCase())) {

return Msg.fail().add("msg", "请选择jpg,jpeg,gif,png格式的图片");

}

// 获取当前年-月， 如 2021-01

String nowMonth = TimeUtils.getCurrentTimeString().substring(0, 7);

String savePath = ConstantUtil.STATIC\_RESOURCES + "/p/" + nowMonth + "/";

File savePathFile = new File(savePath);

if (!savePathFile.exists()) {

//若不存在该目录，则创建目录

savePathFile.mkdir();

}

//通过UUID生成唯一文件名

String filename = UUID.randomUUID().toString().replaceAll("-","") + "." + suffix;

try {

//将文件保存指定目录

file.transferTo(new File(savePath + filename));

} catch (Exception e) {

e.printStackTrace();

return Msg.fail().add("msg", "保存文件异常");

}

String url = ConstantUtil.STATIC\_URL + "/p/" + nowMonth + "/" + filename;

//返回文件名称

return Msg.success().add("url", url);

}

/\*\*

\* 发表文章

\* @param title 标题

\* @param type 类型

\* @param summary 摘要

\* @param comment 是否允许评论

\* @param md md格式的内容

\* @param html 不加修饰的html格式的内容

\* @param request request

\* @return 消息

\* @throws UnsupportedEncodingException 设置编码格式

\*/

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

public Msg insertArticle(

@RequestParam(value = "title") String title,

@RequestParam(value = "type") String type,

@RequestParam(value = "summary") String summary,

@RequestParam(value = "comment") String comment,

@RequestParam(value = "md") String md,

@RequestParam(value = "html") String html,

HttpServletRequest request

) throws UnsupportedEncodingException {

request.setCharacterEncoding("utf-8");

// 判断用户登录状态

String token = request.getHeader("token");

User user;

if(null == token){

// 用户未登录

return Msg.fail();

}else{

user = (User) redisUtil.get(token);

if(null == user) {

return Msg.expire();

}

}

// 此时User已经拿到了。组装文章

Article article = new Article();

article.setArticleTitle(title)

.setArticleAuthor(user.getUserId())

.setArticleType(Integer.parseInt(type))

.setArticleHtml(ArticleUtils.simplifyImages(html))

.setArticleMd(ArticleUtils.replaceSingleSlash(md))

.setArticleIsComment("true".equals(comment) ? 1 : 0)

.setArticleRank(0);

// 统一发布时间

long now = System.currentTimeMillis();

article.setArticleId(ArticleUtils.getArticleId(user.getUserId(), now))

.setArticlePostTime(new Date(now))

.setArticleEditTime(new Date(now));

// 填充文章摘要

if(!"".equals(summary)){

article.setArticleSummary(summary);

}else{

article.setArticleSummary(ArticleUtils.getArticleDefaultSummary(html));

}

// 向文章——图片表中插入记录

List<String> images = ArticleUtils.getPicturesInArticle(html);

// 向 file-image 表中插入文章图片记录

ArticleImage articleImage = new ArticleImage();

for (String image : images) {

articleImage.setInsertDate(new Date(now));

articleImage.setArticleId(article.getArticleId());

articleImage.setFilename(image);

articleImage.insert();

}

boolean suc = article.insert();

return Msg.success().add("suc", suc).add("id", article.getArticleId());

}

/\*\*

\* 修改文章

\* @param id 修改的文章的id

\* @param title 标题

\* @param type 类型

\* @param summary 摘要

\* @param comment 是否允许评论

\* @param md md格式的内容

\* @param html 不加修饰的html格式的内容

\* @param request request

\* @return 消息

\* @throws UnsupportedEncodingException 设置编码格式

\*/

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

public Msg updateArticle(

@RequestParam(value = "id") String id,

@RequestParam(value = "title") String title,

@RequestParam(value = "type") String type,

@RequestParam(value = "summary") String summary,

@RequestParam(value = "comment") String comment,

@RequestParam(value = "md") String md,

@RequestParam(value = "html") String html,

HttpServletRequest request

) throws UnsupportedEncodingException {

request.setCharacterEncoding("utf-8");

// 判断用户登录状态

String token = request.getHeader("token");

User user;

if(null == token){

// 用户未登录

return Msg.fail();

}else{

user = (User) redisUtil.get(token);

if(null == user) {

return Msg.expire();

}

}

// 此时User已经拿到了。组装文章

Article article = new Article();

article.setArticleId(id)

.setArticleTitle(title)

.setArticleAuthor(user.getUserId())

.setArticleType(Integer.parseInt(type))

.setArticleHtml(ArticleUtils.simplifyImages(html))

.setArticleMd(ArticleUtils.replaceSingleSlash(md))

.setArticleIsComment("true".equals(comment) ? 1 : 0)

.setArticleRank(0);

// 统一发布时间

long now = System.currentTimeMillis();

article.setArticleEditTime(new Date(now));

// 填充文章摘要

if(!"".equals(summary)){

article.setArticleSummary(summary);

}else{

article.setArticleSummary(ArticleUtils.getArticleDefaultSummary(html));

}

// 向文章——图片表中插入记录

List<String> images = ArticleUtils.getPicturesInArticle(html);

// 向 file-image 表中插入文章图片记录

ArticleImage articleImage = new ArticleImage();

for (String image : images) {

articleImage.setInsertDate(new Date(now));

articleImage.setArticleId(article.getArticleId());

articleImage.setFilename(image);

try {

articleImage.insert();

}catch (Exception e){

// 略过

}

}

boolean suc = article.updateById();

return Msg.success().add("suc", suc).add("id", article.getArticleId());

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\controller\ArticleImageController.java

package com.jancoyan.jancoblog.controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.stereotype.Controller;

/\*\*

\* <p>

\* 前端控制器

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-16

\*/

@Controller

@RequestMapping("/article\_image")

public class ArticleImageController {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\controller\CommentController.java

package com.jancoyan.jancoblog.controller;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.Article;

import com.jancoyan.jancoblog.pojo.Comment;

import com.jancoyan.jancoblog.pojo.PageComment;

import com.jancoyan.jancoblog.pojo.User;

import com.jancoyan.jancoblog.service.CommentService;

import com.jancoyan.jancoblog.utils.Msg;

import com.jancoyan.jancoblog.utils.RedisUtil;

import org.springframework.beans.factory.annotation.Autowired;

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.RestController;

import javax.servlet.http.HttpServletRequest;

import java.util.Date;

/\*\*

\* <p>

\* 前端控制器

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@RestController

@RequestMapping("/comment")

public class CommentController {

@Autowired

CommentService service;

@Autowired

RedisUtil redisUtil;

/\*\*

\* 管理员进行评论管理的时候，获取所有评论

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @return 成功

\*/

@RequestMapping(value = "/all")

public Msg listAll(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition

){

IPage<Comment> iPage = service.listAll(null, Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 获取用户收到的所有评论

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @param request request

\* @return 消息

\*/

@RequestMapping(value = "/receive")

public Msg listCommentByUserReceive(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition,

HttpServletRequest request

){

// 从token中拿到用户

String token = request.getHeader("token");

if(null == token){

return Msg.expire();

}

User user = (User) redisUtil.get(token);

IPage<Comment> iPage = service.listAll(

String.valueOf(user.getUserId()),

Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 获取当前登录的用户所发表的所有评论

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @param request

\* @return

\*/

@RequestMapping(value = "/posted")

public Msg listCommentByUserPosted(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition,

HttpServletRequest request

){

// 从token中拿到用户

String token = request.getHeader("token");

if(null == token){

return Msg.expire();

}

User user = (User) redisUtil.get(token);

if(null == user){

return Msg.fail().add("msg", "获取失败");

}

IPage<Comment> iPage = service.listCommentByUserPosted(

String.valueOf(user.getUserId()),

Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 批量删除评论

\* @param ids 评论id

\* @return 消息

\*/

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

public Msg deleteComment(

@RequestParam(value = "ids") String ids

){

boolean suc = service.batchDeleteComment(ids);

return Msg.success().add("suc", suc ? "success" : "fail");

}

/\*\*

\* 获取某个文章的分页评论

\* @param id 文章id

\* @param pn 页码

\* @param limit 容量，默认为 7

\* @return 消息

\*/

@RequestMapping(value = "/article")

public Msg listCommentByArticle(

@RequestParam(value = "id") String id,

@RequestParam(value = "pn")Integer pn,

@RequestParam(value = "limit", defaultValue = "7") Integer limit

){

IPage<Comment> iPage = service.listCommentByArticle(id, pn, limit);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 发表评论

\* @param articleId 目标文章

\* @param userName 用户名，游客有效

\* @param email 邮箱，游客有效

\* @param content 内容

\* @param request request，用来获取ip

\* @return 消息

\*/

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

public Msg insertComment(

@RequestParam(value = "articleId")String articleId,

@RequestParam(value = "name", defaultValue = "") String userName,

@RequestParam(value = "email", defaultValue = "")String email,

@RequestParam(value = "content")String content,

HttpServletRequest request

){

// 先判断是不是已经登陆了

Comment comment = new Comment();

String token = request.getHeader("token");

User user = (User) redisUtil.get(token);

// 用户信息已经过期了

if(null == user && null != token) {

return Msg.expire();

}

// 如果 token 是 null

if(null == token || "".equals(token)){

// 游客发表的评论

comment.setCommentAuthorEmail(email);

comment.setCommentAuthorName(userName);

} else {

// 已经登录的用户发表的评论

comment.setCommentAuthorName(user.getUserName());

comment.setCommentAuthorId(user.getUserId());

}

comment.setCommentArticleId(articleId);

comment.setCommentDate(new Date());

comment.setCommentContent(content);

comment.setCommentAuthorIp(request.getScheme());

Article article = new Article();

article.setArticleId(articleId);

article = article.selectById();

article.setArticleCommentCount(article.getArticleCommentCount() + 1);

article.updateById();

comment.insert();

return Msg.success();

}

/\*\*

\* 给评论点赞

\* @param id 评论id

\* @return 消息

\*/

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

public Msg likeComment(

@RequestParam(value = "id") Integer id

){

service.likeComment(id);

return Msg.success();

}

/\*\*

\* 获取指定用户最近收到的评论

\* @param authorId 用户id

\* @return

\*/

@RequestMapping(value = "/recent", method = RequestMethod.GET)

public Msg listCommentByUserRecentlyReceive(

@RequestParam(value = "id") String authorId,

@RequestParam(value = "pn", defaultValue = "1") String pn,

@RequestParam(value = "limit", defaultValue = "10") String limit

){

IPage<PageComment> iPage = service.listCommentByUserRecently(authorId);

return Msg.success().add("pageInfo", iPage);

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\controller\DeletedCommentController.java

package com.jancoyan.jancoblog.controller;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.Comment;

import com.jancoyan.jancoblog.pojo.DeletedComment;

import com.jancoyan.jancoblog.service.DeletedCommentService;

import com.jancoyan.jancoblog.utils.Msg;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

/\*\*

\* <p>

\* 前端控制器

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-17

\*/

@RestController

@RequestMapping("/deleted/comment")

public class DeletedCommentController {

@Autowired

DeletedCommentService service;

/\*\*

\* 获取某个文章的分页评论

\* @param id 文章id

\* @param pn 页码

\* @param limit 容量，默认为 7

\* @return 消息

\*/

@RequestMapping(value = "/article", method = RequestMethod.GET)

public Msg getCommentByArticle(

@RequestParam(value = "id") String id,

@RequestParam(value = "pn", defaultValue = "1")Integer pn,

@RequestParam(value = "limit", defaultValue = "7") Integer limit

){

IPage<DeletedComment> iPage = service.getCommentByArticle(id, pn, limit);

return Msg.success().add("pageInfo", iPage);

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\controller\LikeRecordController.java

package com.jancoyan.jancoblog.controller;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.baomidou.mybatisplus.extension.plugins.pagination.Page;

import com.jancoyan.jancoblog.pojo.Article;

import com.jancoyan.jancoblog.pojo.LikeRecord;

import com.jancoyan.jancoblog.pojo.User;

import com.jancoyan.jancoblog.service.LikeRecordService;

import com.jancoyan.jancoblog.utils.Msg;

import com.jancoyan.jancoblog.utils.RedisUtil;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import javax.servlet.http.HttpServletRequest;

/\*\*

\* <p>

\* 前端控制器

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-06

\*/

@RestController

@RequestMapping("/like")

public class LikeRecordController {

@Autowired

LikeRecordService service;

@Autowired

RedisUtil redisUtil;

/\*\*

\* 判断用户是否点赞了这个文章，只有用户登录了之后才会到这个请求中来

\* @param id 文章id

\* @param request

\* @return

\*/

@RequestMapping(value = "/like", method = RequestMethod.GET)

public Msg isLiked(

@RequestParam(value = "id") String id,

HttpServletRequest request

){

// 登录认证

String token = request.getHeader("token");

if(null == token){

// 未登录，说明是游客

return Msg.success().add("suc", true);

}

User user;

user = (User) redisUtil.get(token);

if(null == user){

return Msg.fail();

}

QueryWrapper<LikeRecord> wrapper = new QueryWrapper<>();

wrapper.eq("article\_id", id);

wrapper.eq("author\_id", user.getUserId());

LikeRecord record = new LikeRecord();

record = record.selectOne(wrapper);

if(null == record){

// 没有点赞

return Msg.success().add("suc", false);

}else {

// 点赞了

return Msg.success().add("suc", true);

}

}

/\*\*

\* 获取一个用户最近收到的分页点赞记录

\* @param id 用户id

\* @return

\*/

@RequestMapping(value = "/recent", method = RequestMethod.GET)

public Msg getUserLike(

@RequestParam(value = "id") String id,

@RequestParam(value = "pn")Integer pn,

@RequestParam(value = "limit")Integer limit

) {

// 登录认证

IPage<LikeRecord> pageInfo = service.getUserReceive(id, pn, limit);

return Msg.success().add("pageInfo", pageInfo);

}

}R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\controller\TypeController.java

package com.jancoyan.jancoblog.controller;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.baomidou.mybatisplus.extension.plugins.pagination.Page;

import com.jancoyan.jancoblog.pojo.Type;

import com.jancoyan.jancoblog.service.ArticleService;

import com.jancoyan.jancoblog.service.TypeService;

import com.jancoyan.jancoblog.utils.Msg;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import java.awt.geom.IllegalPathStateException;

/\*\*

\* <p>

\* 前端控制器

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@RestController

@RequestMapping("/type")

public class TypeController {

@Autowired

TypeService service;

/\*\*

\* 获取所有类别，不分页

\* @return pageInfo

\*/

@RequestMapping(value = "/all", method = RequestMethod.GET)

public Msg getAll(){

IPage<Type> iPage = new Page<>(1, 99);

Type type = new Type();

iPage = type.selectPage(iPage, null);

return Msg.success().add("pageInfo", iPage);

}

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

public Msg addType(

@RequestParam(value = "name") String typeName,

@RequestParam(value = "description", defaultValue = "") String description

){

return Msg.success();

}

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

public Msg deleteType(

@RequestParam(value = "ids")String ids

){

return Msg.success();

}

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

public Msg updateType(

@RequestParam(value = "name") String typeName,

@RequestParam(value = "description", defaultValue = "") String description

){

return Msg.success();

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\controller\UserController.java

package com.jancoyan.jancoblog.controller;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.User;

import com.jancoyan.jancoblog.pojo.UserInfo;

import com.jancoyan.jancoblog.pojo.UserLogin;

import com.jancoyan.jancoblog.pojo.VUserTotalData;

import com.jancoyan.jancoblog.service.UserService;

import com.jancoyan.jancoblog.utils.\*;

import io.lettuce.core.output.ScanOutput;

import org.springframework.beans.factory.annotation.Autowired;

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.RestController;

import org.springframework.web.multipart.MultipartFile;

import javax.imageio.ImageIO;

import javax.servlet.http.Cookie;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.awt.\*;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import java.io.OutputStream;

import java.util.Date;

import java.util.Locale;

import java.util.Objects;

import java.util.UUID;

/\*\*

\* <p>

\* 前端控制器

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@RestController

@RequestMapping("/user")

public class UserController {

@Autowired

UserService service;

@Autowired

RedisUtil redisUtil;

/\*\*

\* 登录

\* @param username 用户名

\* @param password 密码

\* @return

\*/

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

public Msg login(

@RequestParam(value = "username") String username,

@RequestParam(value = "password") String password,

HttpServletRequest request

){

String token = null;

// 从视图选择登录

User user = service.login(username, password);

if(null != user){

// 登录成功, 生成token

token = JsonWebTokenUtils.createToken(user.getUserId().longValue());

// 存到redis数据库, 设置过期时间为 60 分钟

redisUtil.set(token, user, 120 \* 60);

// 更新上次登录时间

user.setUserLastLoginDate(new Date());

user.updateById();

} else {

return Msg.fail().add("msg", "登录失败");

}

//登录记录

UserLogin log = new UserLogin();

String userAgent = request.getHeader("User-Agent");

log.setLoginUser(user.getUserId());

log.setLoginDate(new Date());

log.setLoginAddress(NetworkUtils.queryAddressByIp(request.getRemoteAddr()));

log.setLoginIp(request.getRemoteAddr());

log.setUserAgent(userAgent);

log.setBrowserName(UserAgentUtils.getBorderName(userAgent));

log.setBrowserVersion(UserAgentUtils.getBrowserVersion(userAgent));

log.setOsName(UserAgentUtils.getOsName(userAgent));

log.setOsVersion(UserAgentUtils.getOsVersion(userAgent));

if(null != token){

//插入用户登录信息

log.insert();

return Msg.success().add("token", token);

}

return Msg.fail();

}

/\*\*

\* 用户主页的内容

\* @param userId 用户id

\* @param request request

\* @return

\*/

@RequestMapping(value = "/data/total", method = RequestMethod.GET)

public Msg getUserTotalData(

@RequestParam(value = "id", defaultValue = "-1")String userId,

HttpServletRequest request

){

if("-1".equals(userId)){

// 没有提供id, 直接获取当前登录的用户

String token = request.getHeader("token");

if(null == token) {

return Msg.fail();

}

User user = (User)redisUtil.get(token);

userId = String.valueOf(user.getUserId());

}

// 获取数据

VUserTotalData data = service.getUserTotalData(userId);

if (null != data){

return Msg.success().add("data", data);

} else {

return Msg.fail().add("msg", "找不到用户");

}

}

/\*\*

\* 每一次要获取信息的时候都会发送一次这个

\* @return 带有用户消息的信息

\*/

@RequestMapping(value = "/userinfo", method = RequestMethod.GET)

public Msg getUserInfo(HttpServletRequest request){

// 把token放在请求头中

String token = request.getHeader("token");

User user = (User)redisUtil.get(token);

if(null != user){

// 获取用户成功

return Msg.success().add("user", user);

}

return Msg.fail().add("msg", "找不到用户");

}

/\*\*

\* 获取当前已经登录的用户的详细信息

\* @param request req

\* @return

\*/

@RequestMapping(value = "/userdetail", method = RequestMethod.GET)

public Msg getUserDetailInfo(

HttpServletRequest request

){

// 把token放在请求头中

String token = request.getHeader("token");

User user = (User)redisUtil.get(token);

if(null == user){

// 获取用户失败

return Msg.expire().add("msg", "用户已过期");

}

// 获取用户的信息

UserInfo info = service.getUserInfo(user.getUserId());

if(null != info){

return Msg.success().add("info", info);

} else {

return Msg.fail();

}

}

/\*\*

\* 在进入文章的时候，简要显示用户展示给别人看的信息

\* @param userId 用户id

\* @return 消息

\*/

@RequestMapping(value = "/authorinfo", method = RequestMethod.GET)

public Msg getAuthorInfo(

@RequestParam(value = "id")String userId

){

VUserTotalData data = service.getUserTotalData(userId);

if(null != data) {

return Msg.success().add("data", data);

} else {

return Msg.fail().add("msg", "获取数据失败");

}

}

/\*\*

\* 注册

\* @param userName 姓名

\* @param password 密码

\* @param request 用来获取ip

\* @return 消息

\*/

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

public Msg register(

@RequestParam(value = "username") String userName,

@RequestParam(value = "password") String password,

@RequestParam(value = "code") String code,

HttpServletRequest request

) throws IOException {

String verify = (String) request.getSession().getAttribute(VerifyCodeUtil.RANDOMCODEKEY);

if(!verify.toLowerCase(Locale.ROOT).equals(code.toLowerCase(Locale.ROOT))){

return Msg.fail().add("msg", "验证码输入错误");

}

User user = new User();

// 设置信息

user.setUserName(userName)

.setUserLastLoginDate(null)

.setUserId(null)

.setUserSignature("Hello World")

.setUserPassword(MD5Util.getMD5(password))

.setUserRole(2)

.setUserCreateDate(new Date())

.setUserIp(request.getRemoteAddr());

// 生成头像

AvatarUtil.generateImg(userName,

ConstantUtil.STATIC\_RESOURCES + "/avatar/",

userName);

// 插入

boolean insert = user.insert();

if(!insert){

return Msg.fail().add("msg", "用户注册失败");

}

return Msg.success().add("success", insert);

}

/\*\*

\* 用户注册的时候用来检查用户名的唯一性

\* @param userName 用户名

\* @return 消息

\*/

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

public Msg checkUserNameUnique(

@RequestParam(value = "username") String userName

){

User user = new User();

QueryWrapper<User> queryWrapper = new QueryWrapper<>();

queryWrapper.eq("user\_name", userName);

int count = user.selectCount(queryWrapper);

return Msg.success().add("unique", count == 0);

}

/\*\*

\* 用户退出登录

\* @return 成功

\*/

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

public Msg logout(HttpServletRequest request){

String token = request.getHeader("token");

if(null == token || "".equals(token)){

// 登录过期

return Msg.fail();

}

// 登录没过期，移除token

redisUtil.del(token);

return Msg.success().add("suc", true);

}

/\*\*

\* 管理员进行用户管理的时候分页获取所有用户

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @return

\*/

@RequestMapping(value = "/all", method = RequestMethod.GET)

public Msg getAllUser(

@RequestParam(value = "pn")String pn,

@RequestParam(value = "limit", defaultValue = "10")String limit,

@RequestParam(value = "condition", defaultValue = "")String condition

){

IPage<User> iPage = service.getAll(Integer.parseInt(pn),

Integer.parseInt(limit),

condition);

return Msg.success().add("pageInfo", iPage);

}

/\*\*

\* 批量删除用户

\* @param ids 用户id 用 & 拼接

\* @return

\*/

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

public Msg batchDelte(

@RequestParam(value = "ids") String ids

){

User user = new User();

boolean suc = false;

if(!ids.contains("&")){

user.setUserId(Integer.parseInt(ids));

suc = user.deleteById();

} else {

String[] id = ids.split("&");

for (String item : id) {

user.setUserId(Integer.parseInt(item));

suc = user.deleteById();

}

}

return Msg.success().add("suc", suc ? "success" : "fail");

}

/\*\*

\* 更换头像

\* @return

\*/

@RequestMapping(value = "/upload/avatar", method = RequestMethod.POST)

public Msg uploadAvatar(

@RequestParam(value = "file") MultipartFile file,

HttpServletRequest request

){

// 登录状态

String token = request.getHeader("token");

if (null == token){

return Msg.fail();

}

// 获取用户

User user = (User) redisUtil.get(token);

if (null == user){

return Msg.expire();

}

// 文件判定

if (null == file) {

return Msg.fail().add("msg", "请选择要上传的图片");

}

if (file.getSize() > 1024 \* 1024 \* 10) {

return Msg.fail().add("msg", "文件大小不能大于10M");

}

//获取文件后缀

String suffix = Objects.requireNonNull(file.getOriginalFilename()).substring(file.getOriginalFilename().lastIndexOf(".") + 1);

if (!"jpg,jpeg,gif,png".toUpperCase().contains(suffix.toUpperCase())) {

return Msg.fail().add("msg", "请选择jpg,jpeg,gif,png格式的图片");

}

String savePath = null;

// savePath = new File(".").getCanonicalPath() + "\\target\\classes\\static\\avatar\\";

savePath = ConstantUtil.STATIC\_RESOURCES + "/avatar/";

File savePathFile = new File(savePath);

if (!savePathFile.exists()) {

//若不存在该目录，则创建目录

savePathFile.mkdir();

}

//用户头像名称就是用户的id

String filename = user.getUserName() + "." + suffix;

// 如果头衔已存在就先删除

File oldFile = new File(savePath + filename);

if(oldFile.exists()){

oldFile.delete();

}

try {

//将文件保存指定目录

file.transferTo(new File(savePath + filename));

} catch (Exception e) {

e.printStackTrace();

return Msg.fail().add("msg", "保存文件异常");

}

//返回文件名称

return Msg.success().add("suc", true);

}

/\*\*

\* 修改密码

\* @param userId 用户id，如果没有就是修改自己的密码

\* @param oldPassword 旧密码

\* @param newPassword 新密码

\* @param request

\* @return

\*/

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

public Msg changePassword(

@RequestParam(value = "id", defaultValue = "-1") String userId,

@RequestParam(value = "old") String oldPassword,

@RequestParam(value = "new") String newPassword,

HttpServletRequest request

){

boolean suc = false;

// 先判断有没有id

if("-1".equals(userId)){

// 没有id，修改目前登录的用户的id

String token = request.getHeader("token");

if(null == token) {

return Msg.fail();

}

User user = (User) redisUtil.get(token);

if(user.getUserPassword().equals(MD5Util.getMD5(oldPassword))){

// 密码正确

user.setUserPassword(MD5Util.getMD5(newPassword));

suc = user.updateById();

return Msg.success().add("suc", suc);

} else {

return Msg.success().add("suc", suc);

}

} //修改当前登录的用户的密码

// 管理员修改用户的密码

User user = new User();

user.setUserId(Integer.parseInt(userId));

user.setUserPassword(MD5Util.getMD5(newPassword));

// 不需要验证旧密码

suc = user.updateById();

// 直接修改

return Msg.success().add("suc", suc);

}

/\*\*

\* 修改用户信息

\* @param userInfo 用户信息

\* @param request

\* @return

\*/

@RequestMapping(value = "/info/update", method = RequestMethod.POST)

public Msg updateUserInfo(

@RequestParam(value = "user")String userInfo,

HttpServletRequest request

){

String token = request.getHeader("token");

if(null == token){

return Msg.fail().add("msg", "用户未登录");

}

User user = new User();

user = (User) redisUtil.get(token);

if(null == user){

return Msg.expire();

}

UserInfo info = new UserInfo();

// 收到一个userInfo - Json字符串，直接从这里拆开，然后update

userInfo = userInfo.substring(1, userInfo.length() - 1);

// 取出 “userId”：00000 这种

String[] kvs = userInfo.split(",");

for (String kv: kvs) {

String[] nameAndValue = kv.split(":");

// 下标0 为 “userID” 下标1为 “asdasd” “1” 10929 null

// userId

String name = nameAndValue[0].substring(1, nameAndValue[0].length() - 1);

String value = nameAndValue[1];

// 查找与选择逐个排查

if((!"null".equals(value)) && (!"".equals(value))){

// 此字段不是null

// 逐个判断 userInfo 中的属性

if("userId".equals(name)){

info.setUserId(Integer.parseInt(value));

continue;

}

if("userEmail".equals(name)){

info.setUserEmail(value.substring(1, value.length() - 1));

continue;

}

if("userSex".equals(name)){

if("\"1\"".equals(value)){

// 男

info.setUserSex(1);

} else if("\"0\"".equals(value)){

// 女

info.setUserSex(0);

}

continue;

}

if("userRegion".equals(name)){

info.setUserRegion(value.substring(1, value.length() - 1));

continue;

}

if("userBirthdate".equals(name)){

info.setUserBirthdate(TimeUtils.castDateStringToDateTypeYMD(

value.substring(1, 11)));

continue;

}

if("userTelephone".equals(name)){

info.setUserTelephone(value.substring(1, value.length() - 1));

continue;

}

if("userRealName".equals(name)){

info.setUserRealName(value.substring(1, value.length() - 1));

continue;

}

if("userSchool".equals(name)){

info.setUserSchool(value.substring(1, value.length() - 1));

continue;

}

if("userMajor".equals(name)){

info.setUserMajor(value.substring(1, value.length() - 1));

continue;

}

if("userEnterSchoolDate".equals(name)) {

info.setUserEnterSchoolDate(TimeUtils.castDateStringToDateTypeYMD(

value.substring(1, 11)));

continue;

}

if("userAcademicDegree".equals(name)){

info.setUserAcademicDegree(value.substring(1, value.length() - 1));

continue;

}

if("userCompany".equals(name)){

info.setUserCompany(value.substring(1, value.length() - 1));

continue;

}

if("userPosition".equals(name)){

info.setUserPosition(value.substring(1, value.length() - 1));

continue;

}

if("userField".equals(name)){

info.setUserField(value.substring(1, value.length() - 1));

}

}

}

boolean suc = info.updateById();

return Msg.success().add("suc", suc);

}

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

public Msg updateUser(

@RequestParam(value = "username") String username,

@RequestParam(value = "signature") String signature,

HttpServletRequest request

){

// 判断用户登录状态

String token = request.getHeader("token");

if(null == token){

return Msg.fail().add("msg", "用户未登录");

}

User user = new User();

user = (User) redisUtil.get(token);

if(null == user){

return Msg.expire();

}

// 修改用户

user.setUserSignature(signature);

user.setUserName(username);

boolean suc = user.updateById();

if(suc){

redisUtil.set(token, user, 20 \* 60);

}

return Msg.success().add("suc", suc);

}

/\*\*

\* 获取验证码

\* @param request

\* @param response

\*/

@RequestMapping(value = "/getverifycode", method = RequestMethod.GET)

public void generateValidationCode(

HttpServletRequest request,

HttpServletResponse response,

HttpSession session

){

try {

//设置相应类型,告诉浏览器输出的内容为图片

response.setContentType("image/jpeg");

//设置响应头信息，告诉浏览器不要缓存此内容

response.setHeader("Pragma", "No-cache");

response.setHeader("Cache-Control", "no-cache");

response.setDateHeader("Expire", 0);

VerifyCodeUtil verifyCodeUtil = new VerifyCodeUtil();

//输出验证码图片方法

verifyCodeUtil.getRandcode(request, response);

} catch (Exception e) {

e.printStackTrace();

}

}

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

public Msg addUser(

@RequestParam("username") String userName,

@RequestParam("password") String password,

HttpServletRequest request

) throws IOException {

User user = new User();

// 设置信息

user.setUserName(userName)

.setUserLastLoginDate(null)

.setUserId(null)

.setUserSignature("Hello World")

.setUserPassword(MD5Util.getMD5(password))

.setUserRole(2)

.setUserCreateDate(new Date())

.setUserIp(request.getRemoteAddr());

// 生成头像

AvatarUtil.generateImg(userName,

ConstantUtil.STATIC\_RESOURCES + "/avatar",

userName);

// 插入

boolean insert = user.insert();

if(!insert){

return Msg.fail().add("msg", "用户注册失败");

}

return Msg.success().add("success", insert);

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\controller\UserLoginController.java

package com.jancoyan.jancoblog.controller;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.baomidou.mybatisplus.extension.plugins.pagination.Page;

import com.jancoyan.jancoblog.pojo.User;

import com.jancoyan.jancoblog.pojo.UserLogin;

import com.jancoyan.jancoblog.service.UserLoginService;

import com.jancoyan.jancoblog.utils.Msg;

import com.jancoyan.jancoblog.utils.RedisUtil;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RequestParam;

import org.springframework.web.bind.annotation.RestController;

import javax.servlet.http.HttpServletRequest;

import java.util.List;

/\*\*

\* <p>

\* 前端控制器

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-15

\*/

@RestController

@RequestMapping("/login")

public class UserLoginController {

@Autowired

UserLoginService service;

@Autowired

RedisUtil redisUtil;

/\*\*

\* 查询特定用户的登录记录，未指定id就是查找当前用户的登录记录

\* 只能查找最近10次的

\* @param userId 用户id

\* @param request

\* @return

\*/

@RequestMapping(value = "/all", method = RequestMethod.GET)

public Msg getAll(

@RequestParam(value = "id", defaultValue = "") String userId,

HttpServletRequest request

){

String token = request.getHeader("token");

if(null == token){

return Msg.fail().add("msg", "用户验证失败");

}

User user = (User) redisUtil.get(token);

if(null == user){

return Msg.expire();

}

// 获取10条用户登录信息

UserLogin userLogin = new UserLogin();

// 构造Wrapper

QueryWrapper<UserLogin> wrapper = new QueryWrapper<>();

wrapper.eq("login\_user", user.getUserId());

wrapper.orderByDesc("login\_date");

// 构造Page

IPage<UserLogin> iPage = new Page<>(1, 10);

// 查询

iPage = userLogin.selectPage(iPage, wrapper);

return Msg.success().add("pageInfo", iPage);

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\JancoBlogApplication.java

package com.jancoyan.jancoblog;

import org.apache.catalina.connector.Connector;

import org.mybatis.spring.annotation.MapperScan;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.boot.web.embedded.tomcat.TomcatConnectorCustomizer;

import org.springframework.boot.web.embedded.tomcat.TomcatServletWebServerFactory;

import org.springframework.boot.web.servlet.server.ConfigurableServletWebServerFactory;

import org.springframework.context.annotation.Bean;

@SpringBootApplication

@MapperScan("com.jancoyan.jancoblog.mapper")

public class JancoBlogApplication {

public static void main(String[] args) {

SpringApplication.run(JancoBlogApplication.class, args);

}

@Bean

public ConfigurableServletWebServerFactory webServerFactory() {

TomcatServletWebServerFactory factory = new TomcatServletWebServerFactory();

factory.addConnectorCustomizers(new TomcatConnectorCustomizer() {

@Override

public void customize(Connector connector) {

connector.setProperty("relaxedQueryChars", "|{}[]");

}

});

return factory;

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\ArticleImageMapper.java

package com.jancoyan.jancoblog.mapper;

import com.jancoyan.jancoblog.pojo.ArticleImage;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-16

\*/

public interface ArticleImageMapper extends BaseMapper<ArticleImage> {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\ArticleMapper.java

package com.jancoyan.jancoblog.mapper;

import com.baomidou.mybatisplus.core.conditions.Wrapper;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.Article;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

import com.jancoyan.jancoblog.pojo.PageArticle;

import javax.swing.event.InternalFrameAdapter;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

public interface ArticleMapper extends BaseMapper<Article> {

IPage<Article> getIndexList(IPage<Article> iPage, Wrapper ew);

IPage<Article> getManageList(IPage<Article> iPage, Wrapper ew);

Article getSingleArticle(String id);

IPage<Article> getDeletedList(IPage<Article> iPage, Wrapper ew);

void deleteCompletely(String articleId);

void batchRecover(String articleId);

IPage<PageArticle> getArticleByUserRecently(IPage<PageArticle> iPage, Wrapper ew);

Article getSingleArticleDeleted(String id);

Article getArticleEdit(String id);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\CommentMapper.java

package com.jancoyan.jancoblog.mapper;

import com.baomidou.mybatisplus.core.conditions.Wrapper;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.Comment;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

import com.jancoyan.jancoblog.pojo.PageComment;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

public interface CommentMapper extends BaseMapper<Comment> {

IPage<Comment> getAll(IPage<Comment> iPage, Wrapper ew);

IPage<Comment> getCommentByArticle(IPage<Comment> page, QueryWrapper<Comment> ew);

IPage<Comment> getCommentByUserPosted(IPage<Comment> iPage, Wrapper ew);

IPage<PageComment> getCommentByUserRecently(IPage<PageComment> iPage, Wrapper ew);

void deleteComment(Integer commentId);

void recoverComment(Integer commentId);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\DeletedCommentMapper.java

package com.jancoyan.jancoblog.mapper;

import com.baomidou.mybatisplus.core.conditions.Wrapper;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.DeletedComment;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-17

\*/

public interface DeletedCommentMapper extends BaseMapper<DeletedComment> {

IPage<DeletedComment> getCommentByArticle(IPage<DeletedComment> iPage, Wrapper ew);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\LikeRecordMapper.java

package com.jancoyan.jancoblog.mapper;

import com.baomidou.mybatisplus.core.conditions.Wrapper;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.LikeRecord;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-06

\*/

public interface LikeRecordMapper extends BaseMapper<LikeRecord> {

IPage<LikeRecord> getUserReceive(IPage<LikeRecord> iPage,

Wrapper<LikeRecord> ew, String userId);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\TypeMapper.java

package com.jancoyan.jancoblog.mapper;

import com.jancoyan.jancoblog.pojo.Type;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

public interface TypeMapper extends BaseMapper<Type> {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\UserInfoMapper.java

package com.jancoyan.jancoblog.mapper;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

import com.jancoyan.jancoblog.pojo.UserInfo;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-14

\*/

public interface UserInfoMapper extends BaseMapper<UserInfo> {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\UserLoginMapper.java

package com.jancoyan.jancoblog.mapper;

import com.jancoyan.jancoblog.pojo.UserLogin;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-15

\*/

public interface UserLoginMapper extends BaseMapper<UserLogin> {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\mapper\UserMapper.java

package com.jancoyan.jancoblog.mapper;

import com.baomidou.mybatisplus.core.conditions.Wrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.User;

import com.baomidou.mybatisplus.core.mapper.BaseMapper;

import com.jancoyan.jancoblog.pojo.UserInfo;

import com.jancoyan.jancoblog.pojo.VUserTotalData;

import java.util.List;

/\*\*

\* <p>

\* Mapper 接口

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

public interface UserMapper extends BaseMapper<User> {

IPage<User> getAll(IPage<User> iPage, Wrapper ew);

VUserTotalData getUserTotalData(String userId);

UserInfo getUserInfo(Integer userId);

User login(String username, String password);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\Article.java

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.TableField;

import com.baomidou.mybatisplus.annotation.TableId;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.time.LocalDateTime;

import java.io.Serializable;

import java.util.Date;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@Data

@EqualsAndHashCode(callSuper = true)

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

@TableName("tbl\_article")

public class Article extends Model<Article> {

private static final long serialVersionUID = 1L;

@TableId(value = "article\_id")

private String articleId;

private String articleTitle;

private Integer articleAuthor;

private Integer articleType;

private String articleSummary;

private String articleHtml;

private String articleMd;

private Integer articleIsComment;

private Integer articleRank;

private Date articlePostTime;

private Date articleEditTime;

private Integer articleViewCount;

private Integer articleCommentCount;

private Integer articleLikeCount;

@TableField(exist = false)

private String userName;

@TableField(exist = false)

private String typeName;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\ArticleImage.java

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.TableId;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import lombok.Data;

import lombok.EqualsAndHashCode;

import java.util.Date;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-16

\*/

@EqualsAndHashCode(callSuper = true)

@TableName("tbl\_article\_image")

@Data

public class ArticleImage extends Model<ArticleImage> {

private static final long serialVersionUID = 1L;

@TableId(value = "filename")

private String filename;

private String articleId;

private Date insertDate;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\Comment.java

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.TableField;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.annotation.IdType;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import com.baomidou.mybatisplus.annotation.TableId;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.time.LocalDateTime;

import java.io.Serializable;

import java.util.Date;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@Data

@EqualsAndHashCode(callSuper = true)

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

@TableName("tbl\_comment")

public class Comment extends Model<Comment> {

private static final long serialVersionUID = 1L;

@TableId(value = "comment\_id", type = IdType.AUTO)

private Integer commentId;

private Integer preCommentId;

private String commentArticleId;

private Integer commentAuthorId;

private String commentAuthorName;

private String commentAuthorEmail;

private String commentContent;

private Date commentDate;

private String commentAuthorIp;

private Integer commentLikeCount;

@TableField(exist = false)

private String articleTitle;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\DeletedComment.java

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.TableField;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.time.LocalDateTime;

import java.util.Date;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-17

\*/

@Data

@EqualsAndHashCode(callSuper = true)

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

@TableName("tbl\_deleted\_comment")

public class DeletedComment extends Model<DeletedComment> {

private static final long serialVersionUID = 1L;

private Integer commentId;

private Integer preCommentId;

private String commentArticleId;

private Integer commentAuthorId;

private String commentAuthorName;

private String commentAuthorEmail;

private String commentContent;

private Date commentDate;

private String commentAuthorIp;

private Integer commentLikeCount;

@TableField(exist = false)

private String articleTitle;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\LikeRecord.java

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.TableField;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.annotation.IdType;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import com.baomidou.mybatisplus.annotation.TableId;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.time.LocalDateTime;

import java.io.Serializable;

import java.util.Date;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-06

\*/

@Data

@EqualsAndHashCode(callSuper = true)

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

@TableName("tbl\_like\_record")

public class LikeRecord extends Model<LikeRecord> {

private static final long serialVersionUID = 1L;

@TableId(value = "like\_id", type = IdType.AUTO)

private Integer likeId;

/\*\*

\* 被点赞文章id

\*/

private String articleId;

/\*\*

\* 点赞的人

\*/

private Integer authorId;

/\*\*

\* 点赞时间

\*/

private Date likeDate;

/\*\*

\* 点赞人用户名

\*/

@TableField(exist = false)

private String userName;

/\*\*

\* 被点赞的文章标题

\*/

@TableField(exist = false)

private String articleTitle;

/\*\*

\* 被点赞的文章的作者

\*/

@TableField(exist = false)

private String articleAuthor;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\PageArticle.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/10/8

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.util.Date;

@Data

@AllArgsConstructor

@NoArgsConstructor

@EqualsAndHashCode(callSuper = true)

@Accessors(chain = true)

public class PageArticle extends Model<PageArticle>{

private String articleId;

private Integer articleAuthor;

private String articleTitle;

private Date articlePostTime;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\PageComment.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/10/8

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.util.Date;

@EqualsAndHashCode(callSuper = true)

@Data

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

public class PageComment extends Model<LikeRecord> {

private String commentAuthorName;

private String articleId;

private Integer articleAuthor;

private String articleTitle;

private Date commentDate;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\Type.java

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.annotation.IdType;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import com.baomidou.mybatisplus.annotation.TableId;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.io.Serializable;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@Data

@EqualsAndHashCode(callSuper = true)

@Accessors(chain = true)

@AllArgsConstructor

@NoArgsConstructor

@TableName("tbl\_type")

public class Type extends Model<Type> {

private static final long serialVersionUID = 1L;

@TableId(value = "type\_id", type = IdType.AUTO)

private Integer typeId;

private String typeName;

private String typeDescription;

@Override

protected Serializable pkVal() {

return this.typeId;

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\User.java

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.TableField;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.annotation.IdType;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import com.baomidou.mybatisplus.annotation.TableId;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.util.Date;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@Data

@EqualsAndHashCode(callSuper = true)

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

@TableName("tbl\_user")

public class User extends Model<User> {

@TableId(value = "user\_id", type = IdType.AUTO)

private Integer userId;

private String userName;

private String userSignature;

private String userPassword;

private Integer userRole;

private Date userCreateDate;

private Date userLastLoginDate;

private String userIp;

@TableField(exist = false)

private String roleName;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\UserInfo.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/10/3

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.IdType;

import com.baomidou.mybatisplus.annotation.TableId;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.util.Date;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-03

\*/

@Data

@EqualsAndHashCode(callSuper = true)

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

@TableName("tbl\_user\_info")

public class UserInfo extends Model<UserInfo> {

private static final long serialVersionUID = 1L;

@TableId(value = "user\_id", type = IdType.AUTO)

private long userId;

private String userEmail;

private Integer userSex;

private String userRegion;

private Date userBirthdate;

private String userTelephone;

private String userRealName;

private String userSchool;

private String userMajor;

private Date userEnterSchoolDate;

private String userAcademicDegree;

private String userCompany;

private String userPosition;

private String userField;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\UserLogin.java

package com.jancoyan.jancoblog.pojo;

import com.baomidou.mybatisplus.annotation.TableName;

import com.baomidou.mybatisplus.annotation.IdType;

import com.baomidou.mybatisplus.extension.activerecord.Model;

import com.baomidou.mybatisplus.annotation.TableId;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.EqualsAndHashCode;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

import java.time.LocalDateTime;

import java.io.Serializable;

import java.util.Date;

/\*\*

\* <p>

\*

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-15

\*/

@EqualsAndHashCode(callSuper = true)

@Data

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

@TableName("tbl\_user\_login")

public class UserLogin extends Model<UserLogin> {

private static final long serialVersionUID = 1L;

@TableId(value = "id", type = IdType.AUTO)

private Integer id;

private Integer loginUser;

private Date loginDate;

private String loginIp;

private String loginAddress;

private String userAgent;

private String browserName;

private String browserVersion;

private String osName;

private String osVersion;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\pojo\VUserTotalData.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/10/3

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.pojo;

import lombok.AllArgsConstructor;

import lombok.Data;

import lombok.NoArgsConstructor;

import lombok.experimental.Accessors;

/\*\*

\* @author Jancoyan

\* @since 2021-10-03

\*/

@Data

@AllArgsConstructor

@NoArgsConstructor

@Accessors(chain = true)

public class VUserTotalData {

private Integer articleAuthor;

private String userName;

private String userSignature;

private Integer totalArticle;

private Integer totalViewCount;

private Integer totalLikeCount;

private Integer totalCommentCount;

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\ArticleImageService.java

package com.jancoyan.jancoblog.service;

import com.jancoyan.jancoblog.pojo.ArticleImage;

import com.baomidou.mybatisplus.extension.service.IService;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-16

\*/

public interface ArticleImageService extends IService<ArticleImage> {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\ArticleService.java

package com.jancoyan.jancoblog.service;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.Article;

import com.baomidou.mybatisplus.extension.service.IService;

import com.jancoyan.jancoblog.pojo.PageArticle;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

public interface ArticleService extends IService<Article> {

/\*\*

\* 获取首页的文章列表

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @return

\*/

IPage<Article> listArticleIndex(Integer pn, Integer limit, String condition);

/\*\*

\* 文章管理的时候获得用于管理的文章列表

\* @param userId 用户id，可选

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @return

\*/

IPage<Article> listArticleManage(Integer userId, Integer pn, Integer limit, String condition);

/\*\*

\* 获取一篇文章

\* @param articleId 文章id

\* @return

\*/

Article getArticleSingle(String articleId);

/\*\*

\* 获取单个已经删除的文章

\* @param articleId

\* @return

\*/

Article getArticleSingleDeleted(String articleId);

/\*\*

\* 获取所有删除的文章

\* @param userId 用户名，可选

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @return

\*/

IPage<Article> listDeleted(Integer userId, Integer pn, Integer limit, String condition);

/\*\*

\* 彻底删除已经删除了的文章

\* @param ids id

\* @return

\*/

boolean deleteCompletely(String ids);

/\*\*

\* 批量恢复指定的文章

\* @param ids ids

\* @return

\*/

boolean batchRecoverDeletedArticle(String ids);

/\*\*

\* 批量删除指定文章

\* @param ids

\* @return

\*/

boolean batchDeleteArticle(String ids);

/\*\*

\* 获区用户最近发表的文章

\* @param id 用户id

\* @param pn 页码

\* @param limit 容量

\* @return

\*/

IPage<PageArticle> listArticleUserRecently(String id, Integer pn, Integer limit);

/\*\*

\* 给文章点赞

\* @param id 文章id

\*/

void addLikeCount(String id);

/\*\*

\* 取消文章的点赞

\* @param id 文章id

\*/

void subLikeCount(String id);

/\*\*

\* 增加文章的浏览量

\* @param id

\*/

void addViewCount(String id);

/\*\*

\* 改变文章是否评论的状态

\* @param id 文章id

\* @return

\*/

boolean updateIsComment(String id);

/\*\*

\* 改变文章的置顶状态

\* @return

\*/

boolean updateIsTop(String id);

/\*\*

\* 获取文章用于修改

\* @return

\*/

Article getArticleEdit(String id);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\CommentService.java

package com.jancoyan.jancoblog.service;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.Comment;

import com.baomidou.mybatisplus.extension.service.IService;

import com.jancoyan.jancoblog.pojo.PageComment;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

public interface CommentService extends IService<Comment> {

/\*\*

\* 评论管理的时候获取所有评论，用户是获得用户的所有评论

\* @param userId 用户id

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @return

\*/

IPage<Comment> listAll(String userId, Integer pn, Integer limit, String condition);

/\*\*

\* 获取某个文章的所有评论

\* @param id 文章id

\* @param pn 页码

\* @param limit 容量

\* @return

\*/

IPage<Comment> listCommentByArticle(String id, Integer pn, Integer limit);

/\*\*

\* 评论管理，获取用户发表的所有评论

\* @param id 用户id

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @return

\*/

IPage<Comment> listCommentByUserPosted(String id, Integer pn, Integer limit,

String condition);

/\*\*

\* 获取用户最近获取的评论

\* @param authorId 作者id

\* @return

\*/

IPage<PageComment> listCommentByUserRecently(String authorId);

/\*\*

\* 按照文章“删除”评论

\* @param articleId

\*/

void deleteCommentByArticle(String articleId);

/\*\*

\* 按照文章“恢复”评论

\* @param ids 多个id

\*/

void recoverCommentByArticle(String ids);

/\*\*

\* 批量删除评论

\* @param ids

\* @return

\*/

boolean batchDeleteComment(String ids);

/\*\*

\* 给评论点赞

\* @param id

\*/

void likeComment(Integer id);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\DeletedCommentService.java

package com.jancoyan.jancoblog.service;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.Comment;

import com.jancoyan.jancoblog.pojo.DeletedComment;

import com.baomidou.mybatisplus.extension.service.IService;

import org.apache.ibatis.annotations.Delete;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-17

\*/

public interface DeletedCommentService extends IService<DeletedComment> {

IPage<DeletedComment> getCommentByArticle(String id, Integer pn, Integer limit);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\ArticleImageServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.jancoyan.jancoblog.pojo.ArticleImage;

import com.jancoyan.jancoblog.mapper.ArticleImageMapper;

import com.jancoyan.jancoblog.service.ArticleImageService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import org.springframework.stereotype.Service;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-16

\*/

@Service

public class ArticleImageServiceImpl extends ServiceImpl<ArticleImageMapper, ArticleImage> implements ArticleImageService {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\ArticleServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.baomidou.mybatisplus.extension.plugins.pagination.Page;

import com.jancoyan.jancoblog.pojo.Article;

import com.jancoyan.jancoblog.mapper.ArticleMapper;

import com.jancoyan.jancoblog.pojo.ArticleImage;

import com.jancoyan.jancoblog.pojo.DeletedComment;

import com.jancoyan.jancoblog.pojo.PageArticle;

import com.jancoyan.jancoblog.service.ArticleService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import com.jancoyan.jancoblog.utils.ArticleUtils;

import com.jancoyan.jancoblog.utils.FileUtils;

import org.springframework.stereotype.Service;

import java.util.List;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@Service

public class ArticleServiceImpl extends ServiceImpl<ArticleMapper, Article> implements ArticleService {

@Override

public IPage<Article> listArticleIndex(Integer pn, Integer limit, String condition) {

// 分页查询

IPage<Article> iPage = new Page<>(pn, limit);

QueryWrapper<Article> wrapper = new QueryWrapper<>();

wrapper.orderByDesc("article\_rank");

wrapper = ArticleUtils.generateManageArticleWrapperByCondition(wrapper, condition);

wrapper.orderByDesc("article\_post\_time");

return baseMapper.getIndexList(iPage, wrapper);

}

@Override

public IPage<Article> listArticleManage(Integer userId,

Integer pn,

Integer limit,

String condition) {

// 分页查询

IPage<Article> iPage = new Page<>(pn, limit);

QueryWrapper<Article> wrapper = new QueryWrapper<>();

// 单一用户的文章获取

if(null != userId) {

wrapper.eq("user\_id", userId);

}

wrapper = ArticleUtils.generateManageArticleWrapperByCondition(wrapper, condition);

wrapper.orderByDesc("article\_post\_time");

return baseMapper.getManageList(iPage, wrapper);

}

@Override

public Article getArticleSingle(String articleId) {

return baseMapper.getSingleArticle(articleId);

}

@Override

public Article getArticleSingleDeleted(String articleId) {

return baseMapper.getSingleArticleDeleted(articleId);

}

@Override

public IPage<Article> listDeleted(Integer userId,

Integer pn,

Integer limit,

String condition) {

// 分页查询

IPage<Article> iPage = new Page<>(pn, limit);

QueryWrapper<Article> wrapper = new QueryWrapper<>();

// 单一用户的文章获取

if(null != userId) {

wrapper.eq("user\_id", userId);

}

wrapper = ArticleUtils.generateManageArticleWrapperByCondition(wrapper, condition);

wrapper.orderByDesc("article\_post\_time");

return baseMapper.getDeletedList(iPage, wrapper);

}

@Override

public boolean deleteCompletely(String ids) {

QueryWrapper<ArticleImage> wrapper = new QueryWrapper<>();

ArticleImage articleImage = new ArticleImage();

QueryWrapper<DeletedComment> deletedCommentQueryWrapper = new QueryWrapper<>();

DeletedComment deletedComment = new DeletedComment();

if(!ids.contains("&")){

wrapper.eq("article\_id", ids);

deletedCommentQueryWrapper.eq("comment\_article\_id", ids);

List<ArticleImage> articleImages = articleImage.selectList(wrapper);

for (ArticleImage image : articleImages) {

// 依次删除图片文件

FileUtils.deleteImageIfExists(image.getInsertDate(), image.getFilename());

}

// 从表中删除

articleImage.delete(wrapper);

deletedComment.delete(deletedCommentQueryWrapper);

// 删除所有已删除的评论

baseMapper.deleteCompletely(ids);

} else {

String[] id = ids.split("&");

for (String item : id) {

// 删除图片文件

wrapper.eq("article\_id", item);

deletedCommentQueryWrapper.eq("comment\_article\_id", item);

List<ArticleImage> articleImages = articleImage.selectList(wrapper);

for (ArticleImage image : articleImages) {

// 依次删除图片文件

FileUtils.deleteImageIfExists(image.getInsertDate(), image.getFilename());

}

// 从表中删除

articleImage.delete(wrapper);

deletedComment.delete(deletedCommentQueryWrapper);

// 从文章表中删除文章

baseMapper.deleteCompletely(item);

}

}

return true;

}

@Override

public boolean batchRecoverDeletedArticle(String ids) {

if(!ids.contains("&")){

baseMapper.batchRecover(ids);

} else {

String[] id = ids.split("&");

for (String item : id) {

baseMapper.batchRecover(item);

}

}

return true;

}

@Override

public boolean batchDeleteArticle(String ids) {

Article article = new Article();

if(!ids.contains("&")){

article.setArticleId(ids);

article.deleteById();

} else {

String[] id = ids.split("&");

for (String item : id) {

article.setArticleId(item);

article.deleteById();

}

}

return true;

}

@Override

public IPage<PageArticle> listArticleUserRecently(String id, Integer pn, Integer limit) {

IPage<PageArticle> iPage = new Page<>(pn, limit);

QueryWrapper<PageArticle> wrapper = new QueryWrapper<>();

wrapper.eq("article\_author", id);

wrapper.orderByDesc("article\_post\_time");

return baseMapper.getArticleByUserRecently(iPage, wrapper);

}

@Override

public void addLikeCount(String id) {

// 增加点赞量

Article article = new Article();

article.setArticleId(id);

article = article.selectById();

article.setArticleLikeCount(article.getArticleLikeCount() + 1);

article.updateById();

}

@Override

public void subLikeCount(String id) {

Article article = new Article();

article.setArticleId(id);

article = article.selectById();

article.setArticleLikeCount(article.getArticleLikeCount() - 1);

article.updateById();

}

@Override

public void addViewCount(String id) {

Article article = new Article();

article.setArticleId(id);

article = article.selectById();

article.setArticleViewCount(article.getArticleViewCount() + 1);

article.updateById();

}

@Override

public boolean updateIsComment(String id) {

// 改变评论的状态

Article article = new Article();

article.setArticleId(id);

article = article.selectById();

article.setArticleIsComment(1 - article.getArticleIsComment());

return article.updateById();

}

@Override

public boolean updateIsTop(String id) {

// 改变置顶的状态

Article article = new Article();

article.setArticleId(id);

article = article.selectById();

article.setArticleRank(1 - article.getArticleRank());

return article.updateById();

}

@Override

public Article getArticleEdit(String id) {

return baseMapper.getArticleEdit(id);

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\CommentServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.baomidou.mybatisplus.extension.plugins.pagination.Page;

import com.jancoyan.jancoblog.pojo.Article;

import com.jancoyan.jancoblog.pojo.Comment;

import com.jancoyan.jancoblog.mapper.CommentMapper;

import com.jancoyan.jancoblog.pojo.DeletedComment;

import com.jancoyan.jancoblog.pojo.PageComment;

import com.jancoyan.jancoblog.service.CommentService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import io.lettuce.core.dynamic.annotation.CommandNaming;

import org.springframework.stereotype.Service;

import java.util.List;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@Service

public class CommentServiceImpl extends ServiceImpl<CommentMapper, Comment> implements CommentService {

@Override

public IPage<Comment> listAll(String userId, Integer pn, Integer limit, String condition) {

// 分页查询

IPage<Comment> iPage = new Page<>(pn, limit);

QueryWrapper<Comment> wrapper = new QueryWrapper<>();

if(userId != null) wrapper.eq("user\_id", userId);

String[] split = condition.split("--");

for (String item : split) {

String[] split2 = item.split("=");

if(split2.length < 2){

continue;

}

if("comment\_author\_name".equals(split2[0])){

wrapper.like("comment\_author\_name", split2[1]);

}else if("article\_title".equals(split2[0])){

wrapper.like("article\_title", split2[1]);

}else if("start".equals(split2[0])){

wrapper.gt("comment\_date", split2[1]);

}else if("end".equals(split2[0])){

wrapper.lt("comment\_date", split2[1]);

}else if("rank\_like".equals(split2[0])){

if ("1".equals(split2[1])) {

wrapper.orderByAsc("comment\_like\_count");

} else {

wrapper.orderByDesc("comment\_like\_count");

}

}

}

return baseMapper.getAll(iPage, wrapper);

}

@Override

public IPage<Comment> listCommentByArticle(String id, Integer pn, Integer limit) {

QueryWrapper<Comment> wrapper = new QueryWrapper<>();

IPage<Comment> page = new Page<>(pn, limit);

wrapper.eq("comment\_article\_id", id);

wrapper.orderByDesc("comment\_date");

return baseMapper.getCommentByArticle(page, wrapper);

}

@Override

public IPage<Comment> listCommentByUserPosted(String id, Integer pn, Integer limit, String condition) {

// 分页查询

IPage<Comment> iPage = new Page<>(pn, limit);

QueryWrapper<Comment> wrapper = new QueryWrapper<>();

if(id != null) {

wrapper.eq("comment\_author\_id", id);

}

String[] split = condition.split("--");

for (String item : split) {

String[] split2 = item.split("=");

if(split2.length < 2){

continue;

}

if("comment\_author\_name".equals(split2[0])){

wrapper.like("user\_id", split2[1]);

}else if("article\_title".equals(split2[0])){

wrapper.like("article\_title", split2[1]);

}else if("start".equals(split2[0])){

wrapper.gt("comment\_date", split2[1]);

}else if("end".equals(split2[0])){

wrapper.lt("comment\_date", split2[1]);

}else if("rank\_like".equals(split2[0])){

if ("1".equals(split2[1])) {

wrapper.orderByAsc("comment\_like\_count");

} else {

wrapper.orderByDesc("comment\_like\_count");

}

}

}

return baseMapper.getCommentByUserPosted(iPage, wrapper);

}

@Override

public IPage<PageComment> listCommentByUserRecently(String authorId) {

IPage<PageComment> iPage = new Page<>(1, 10);

QueryWrapper<PageComment> wrapper = new QueryWrapper<>();

wrapper.eq("article\_author", authorId);

wrapper.orderByDesc("comment\_date");

return baseMapper.getCommentByUserRecently(iPage, wrapper);

}

@Override

public void deleteCommentByArticle(String ids) {

// 先查出来评论，再依次删除

QueryWrapper<Comment> wrapper = new QueryWrapper<>();

Comment comment = new Comment();

Article article = new Article();

if(!ids.contains("&")){

// 将删除文章的评论挪到另一张表中

wrapper.eq("comment\_article\_id", ids);

List<Comment> list = comment.selectList(wrapper);

if(!list.isEmpty()){

for (Comment item : list){

baseMapper.deleteComment(item.getCommentId());

}

}

} else {

String[] id = ids.split("&");

for (String articleId : id) {

wrapper.eq("comment\_article\_id", articleId);

List<Comment> list = comment.selectList(wrapper);

if(!list.isEmpty()){

for (Comment item : list){

baseMapper.deleteComment(item.getCommentId());

}

}

}

}

}

@Override

public void recoverCommentByArticle(String ids) {

QueryWrapper<DeletedComment> wrapper = new QueryWrapper<>();

DeletedComment comment = new DeletedComment();

// 先查出来评论，再依次恢复

if(!ids.contains("&")){

wrapper.eq("comment\_article\_id", ids);

List<DeletedComment> list = comment.selectList(wrapper);

if(!list.isEmpty()){

for (DeletedComment item : list){

baseMapper.recoverComment(item.getCommentId());

}

}

} else {

String[] id = ids.split("&");

for (String article : id) {

wrapper.eq("comment\_article\_id", article);

List<DeletedComment> list = comment.selectList(wrapper);

if(!list.isEmpty()){

for (DeletedComment item : list){

baseMapper.recoverComment(item.getCommentId());

}

}

}

}

}

@Override

public boolean batchDeleteComment(String ids) {

Comment comment = new Comment();

boolean suc = false;

if(!ids.contains("&")){

comment.setCommentId(Integer.parseInt(ids));

suc = comment.deleteById();

} else {

String[] id = ids.split("&");

for (String item : id) {

comment.setCommentId(Integer.parseInt(item));

suc = comment.deleteById();

}

}

return suc;

}

@Override

public void likeComment(Integer id) {

Comment comment = new Comment();

comment.setCommentId(id);

comment = comment.selectById();

comment.setCommentLikeCount(comment.getCommentLikeCount() + 1);

comment.updateById();

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\DeletedCommentServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.baomidou.mybatisplus.extension.plugins.pagination.Page;

import com.jancoyan.jancoblog.pojo.Comment;

import com.jancoyan.jancoblog.pojo.DeletedComment;

import com.jancoyan.jancoblog.mapper.DeletedCommentMapper;

import com.jancoyan.jancoblog.service.DeletedCommentService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import org.springframework.stereotype.Service;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-17

\*/

@Service

public class DeletedCommentServiceImpl extends ServiceImpl<DeletedCommentMapper, DeletedComment> implements DeletedCommentService {

@Override

public IPage<DeletedComment> getCommentByArticle(String id, Integer pn, Integer limit) {

QueryWrapper<DeletedComment> wrapper = new QueryWrapper<>();

IPage<DeletedComment> page = new Page<>(pn, limit);

wrapper.eq("comment\_article\_id", id);

wrapper.orderByDesc("comment\_date");

return baseMapper.getCommentByArticle(page, wrapper);

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\LikeRecordServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.baomidou.mybatisplus.extension.plugins.pagination.Page;

import com.jancoyan.jancoblog.pojo.LikeRecord;

import com.jancoyan.jancoblog.mapper.LikeRecordMapper;

import com.jancoyan.jancoblog.service.LikeRecordService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import org.springframework.stereotype.Service;

import java.util.Date;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-06

\*/

@Service

public class LikeRecordServiceImpl extends ServiceImpl<LikeRecordMapper, LikeRecord> implements LikeRecordService {

@Override

public IPage<LikeRecord> getUserReceive(String userId, Integer pn, Integer limit) {

IPage<LikeRecord> iPage = new Page<>(pn, limit);

QueryWrapper<LikeRecord> wrapper = new QueryWrapper<>();

wrapper.eq("article\_author", userId);

wrapper.orderByDesc("like\_date");

return baseMapper.getUserReceive(iPage, wrapper, userId);

}

@Override

public void insertRecord(Integer userId, String articleId) {

LikeRecord record = new LikeRecord();

record.setAuthorId(userId);

record.setLikeDate(new Date());

record.setArticleId(articleId);

record.insert();

}

@Override

public void deleteRecord(Integer userId, String articleId) {

LikeRecord record = new LikeRecord();

QueryWrapper<LikeRecord> wrapper = new QueryWrapper<>();

wrapper.eq("article\_id", articleId);

wrapper.eq("author\_id", userId);

record.delete(wrapper);

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\TypeServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.jancoyan.jancoblog.pojo.Type;

import com.jancoyan.jancoblog.mapper.TypeMapper;

import com.jancoyan.jancoblog.service.TypeService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import org.springframework.stereotype.Service;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@Service

public class TypeServiceImpl extends ServiceImpl<TypeMapper, Type> implements TypeService {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\UserInfoServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.jancoyan.jancoblog.mapper.UserInfoMapper;

import com.jancoyan.jancoblog.pojo.UserInfo;

import com.jancoyan.jancoblog.service.UserInfoService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import org.springframework.stereotype.Service;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-14

\*/

@Service

public class UserInfoServiceImpl extends ServiceImpl<UserInfoMapper, UserInfo> implements UserInfoService {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\UserLoginServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.jancoyan.jancoblog.pojo.UserLogin;

import com.jancoyan.jancoblog.mapper.UserLoginMapper;

import com.jancoyan.jancoblog.service.UserLoginService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import org.springframework.stereotype.Service;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-15

\*/

@Service

public class UserLoginServiceImpl extends ServiceImpl<UserLoginMapper, UserLogin> implements UserLoginService {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\impl\UserServiceImpl.java

package com.jancoyan.jancoblog.service.impl;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.baomidou.mybatisplus.extension.plugins.pagination.Page;

import com.jancoyan.jancoblog.pojo.User;

import com.jancoyan.jancoblog.mapper.UserMapper;

import com.jancoyan.jancoblog.pojo.UserInfo;

import com.jancoyan.jancoblog.pojo.VUserTotalData;

import com.jancoyan.jancoblog.service.UserService;

import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;

import com.jancoyan.jancoblog.utils.MD5Util;

import org.springframework.stereotype.Service;

import java.util.List;

/\*\*

\* <p>

\* 服务实现类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

@Service

public class UserServiceImpl extends ServiceImpl<UserMapper, User> implements UserService {

@Override

public IPage<User> getAll(Integer pn, Integer limit, String condition) {

// 分页查询

IPage<User> iPage = new Page<>(pn, limit);

QueryWrapper<User> wrapper = new QueryWrapper<>();

String[] split = condition.split("--");

for (String item : split) {

String[] split2 = item.split("=");

if(split2.length < 2){

continue;

}

if("user\_name".equals(split2[0])){

wrapper.like("user\_name", split2[1]);

}else if("start".equals(split2[0])){

wrapper.gt("user\_create\_date", split2[1]);

}else if("end".equals(split2[0])){

wrapper.lt("user\_create\_date", split2[1]);

}

}

return baseMapper.getAll(iPage, wrapper);

}

@Override

public VUserTotalData getUserTotalData(String userId) {

return baseMapper.getUserTotalData(userId);

}

@Override

public UserInfo getUserInfo(Integer userId) {

return baseMapper.getUserInfo(userId);

}

@Override

public User login(String username, String password) {

// 登录数据校验

password = MD5Util.getMD5(password);

return baseMapper.login(username, password);

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\LikeRecordService.java

package com.jancoyan.jancoblog.service;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.LikeRecord;

import com.baomidou.mybatisplus.extension.service.IService;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-06

\*/

public interface LikeRecordService extends IService<LikeRecord> {

/\*\*

\* 获取用户收到的点赞列表（时间顺序）

\* @param userId 用户id

\* @param pn 页码

\* @param limit 容量

\* @return

\*/

IPage<LikeRecord> getUserReceive(String userId, Integer pn, Integer limit);

/\*\*

\* 增加点赞记录

\* @param userId 用户id

\* @param articleId 文章id

\*/

void insertRecord(Integer userId, String articleId);

/\*\*

\* 删除点赞记录

\* @param userId 用户id

\* @param articleId 文章id

\*/

void deleteRecord(Integer userId, String articleId);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\TypeService.java

package com.jancoyan.jancoblog.service;

import com.jancoyan.jancoblog.pojo.Type;

import com.baomidou.mybatisplus.extension.service.IService;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

public interface TypeService extends IService<Type> {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\UserInfoService.java

package com.jancoyan.jancoblog.service;

import com.baomidou.mybatisplus.extension.service.IService;

import com.jancoyan.jancoblog.pojo.UserInfo;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-14

\*/

public interface UserInfoService extends IService<UserInfo> {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\UserLoginService.java

package com.jancoyan.jancoblog.service;

import com.jancoyan.jancoblog.pojo.UserLogin;

import com.baomidou.mybatisplus.extension.service.IService;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-10-15

\*/

public interface UserLoginService extends IService<UserLogin> {

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\service\UserService.java

package com.jancoyan.jancoblog.service;

import com.baomidou.mybatisplus.core.metadata.IPage;

import com.jancoyan.jancoblog.pojo.User;

import com.baomidou.mybatisplus.extension.service.IService;

import com.jancoyan.jancoblog.pojo.UserInfo;

import com.jancoyan.jancoblog.pojo.VUserTotalData;

import java.util.List;

/\*\*

\* <p>

\* 服务类

\* </p>

\*

\* @author Jancoyan

\* @since 2021-09-14

\*/

public interface UserService extends IService<User> {

/\*\*

\* 管理员管理的时候获取所有用户

\* @param pn 页码

\* @param limit 容量

\* @param condition 条件

\* @return

\*/

IPage<User> getAll(Integer pn, Integer limit, String condition);

/\*\*

\* 获取用户汇总信息，用于数据展板

\* @param userId 用户id

\* @return

\*/

VUserTotalData getUserTotalData(String userId);

/\*\*

\* 获取用户登录信息

\* @param userId 用户id

\* @return

\*/

UserInfo getUserInfo(Integer userId);

User login(String username, String password);

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\ArticleUtils.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/7/1

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;

import com.jancoyan.jancoblog.pojo.Article;

import java.util.ArrayList;

import java.util.List;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class ArticleUtils {

/\*\*

\* 获取管理文章时候的wrapper，只有从 v\_article\_manage\_show 中查询的时候才能用这个util

\* @param condition 条件

\* @return wrapper

\*/

public static QueryWrapper<Article> generateManageArticleWrapperByCondition(QueryWrapper<Article> wrapper,

String condition){

String[] split = condition.split("--");

for (String item : split) {

String[] split2 = item.split("=");

if(split2.length < 2){

continue;

}

if("article\_author\_name".equals(split2[0])){

wrapper.like("user\_name", split2[1]);

}else if("article\_title".equals(split2[0])){

wrapper.like("article\_title", split2[1]);

}else if("type".equals(split2[0])){

wrapper.eq("article\_type", split2[1]);

}else if("start".equals(split2[0])){

wrapper.gt("article\_post\_time", split2[1]);

}else if("end".equals(split2[0])){

wrapper.lt("article\_post\_time", split2[1]);

}else if("rank\_view".equals(split2[0])){

if ("1".equals(split2[1])) {

wrapper.orderByAsc("article\_view\_count");

} else {

wrapper.orderByDesc("article\_view\_count");

}

}else if("rank\_like".equals(split2[0])){

if ("1".equals(split2[1])) {

wrapper.orderByAsc("article\_like\_count");

} else {

wrapper.orderByDesc("article\_like\_count");

}

}else if("rank\_comment".equals(split2[0])){

if ("1".equals(split2[1])) {

wrapper.orderByAsc("article\_comment\_count");

} else {

wrapper.orderByDesc("article\_comment\_count");

}

}

}

return wrapper;

}

/\*\*

\* 把换行的\n转换为文本的\n

\* @param str 要替换的文本

\* @return 替换完毕的文本

\*/

public static String nextLineToText(String str){

return str.replaceAll("\n", "\\\\n");

}

/\*\*

\* 把单引号替换为 \' 进行转义

\* @param str 文章的内容

\* @return 转换好的文章

\*/

public static String replaceSingleSlash(String str){

return str.replaceAll("'", "\\\\'");

}

/\*\*

\* 获取文章的默认摘要

\* @param innerHTML 作者写入的md转成的html（带有格式）

\* @return 文章的摘要

\*/

public static String getArticleDefaultSummary(String innerHTML){

// 去掉所有的HTML标签

String rst = innerHTML.replaceAll("<.\*?>", "");

// 转换所有的空格

rst = rst.replaceAll("\\s", " ");

// 过长截取

if(rst.length() >= 100){

rst = rst.substring(0, 100) + "......";

}

return rst;

}

/\*\*

\* 获取文章id

\* @param userId 用户id

\* @return 文章id

\*/

public static String getArticleId(Integer userId, long now){

String str = String.valueOf(userId);

str += String.valueOf(now);

return str;

}

/\*\*

\* 标准化博文中的图片，宽度调整为容器大小的70%

\* @param html

\* @return

\*/

public static String simplifyImages(String html){

return html.replaceAll("<img src", "<img style=\"max-width:70%\" src");

}

/\*\*

\* 获取文章中的图片名称

\* @param html html标签

\* @return

\*/

public static List<String> getPicturesInArticle(String html){

String regex = "src=\".\*?\"";

Pattern p = Pattern.compile(regex);

Matcher matcher = p.matcher(html);

List<String> fileNames = new ArrayList<>(16);

while (matcher.find()){

// 当前匹配到的字符串

String fileName = matcher.group();

fileName = fileName.substring(fileName.lastIndexOf('/') + 1, fileName.length() - 1);

fileNames.add(fileName);

System.out.println(fileName);

}

return fileNames;

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\AvatarUtil.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/11/17

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import java.awt.\*;

import java.awt.geom.RoundRectangle2D;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import java.util.Random;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.imageio.ImageIO;

/\*\*

\* 根据姓名创建图片

\*/

public class AvatarUtil {

/\*\*

\* 绘制字体头像

\* 如果是英文名，只显示首字母大写

\* 如果是中文名，只显示最后两个字

\* @param name

\* @param outputPath

\* @param outputName

\* @throws IOException

\*/

public static void generateImg(String name, String outputPath, String outputName)

throws IOException {

int width = 100;

int height = 100;

int nameLen = name.length();

String nameWritten;

//如果用户输入的姓名少于等于2个字符，不用截取

if (nameLen <= 2) {

nameWritten = name;

} else {

//如果用户输入的姓名大于等于3个字符，截取后面两位

String first = name.substring(0, 1);

if (isChinese(first)) {

//截取倒数两位汉字

nameWritten = name.substring(nameLen - 2);

} else {

//截取前面的两个英文字母

nameWritten = name.substring(0, 2).toUpperCase();

}

}

String filename = outputPath + File.separator + outputName + ".png";

File file = new File(filename);

//Font font = new Font("微软雅黑", Font.PLAIN, 30);

BufferedImage bi = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_RGB);

Graphics2D g2 = (Graphics2D) bi.getGraphics();

g2.setRenderingHint(RenderingHints.KEY\_TEXT\_ANTIALIASING,

RenderingHints.VALUE\_TEXT\_ANTIALIAS\_ON);

g2.setBackground(getRandomColor());

g2.clearRect(0, 0, width, height);

g2.setPaint(Color.WHITE);

Font font = null;

//两个字及以上

if(nameWritten.length() >= 2) {

font = new Font("微软雅黑", Font.PLAIN, 30);

g2.setFont(font);

String firstWritten = nameWritten.substring(0, 1);

String secondWritten = nameWritten.substring(1, 2);

//两个中文 如 言曌

if (isChinese(firstWritten) && isChinese(secondWritten)) {

g2.drawString(nameWritten, 20, 60);

}

//首中次英 如 罗Q

else if (isChinese(firstWritten) && !isChinese(secondWritten)) {

g2.drawString(nameWritten, 27, 60);

//首英,如 AB

} else {

nameWritten = nameWritten.substring(0,1);

}

}

//一个字

if(nameWritten.length() ==1) {

//中文

if(isChinese(nameWritten)) {

font = new Font("微软雅黑", Font.PLAIN, 50);

g2.setFont(font);

g2.drawString(nameWritten, 25, 70);

}

//英文

else {

font = new Font("微软雅黑", Font.PLAIN, 55);

g2.setFont(font);

g2.drawString(nameWritten.toUpperCase(), 33, 67);

}

}

BufferedImage rounded = makeRoundedCorner(bi, 99);

ImageIO.write(rounded, "png", file);

}

/\*\*

\* 判断字符串是否为中文

\* @param str

\* @return

\*/

public static boolean isChinese(String str) {

String regEx = "[\\u4e00-\\u9fa5]+";

Pattern p = Pattern.compile(regEx);

Matcher m = p.matcher(str);

if (m.find())

return true;

else

return false;

}

/\*\*

\* 获得随机颜色

\* @return

\*/

private static Color getRandomColor() {

String[] beautifulColors =

new String[]{"232,221,203", "205,179,128", "3,101,100", "3,54,73", "3,22,52",

"237,222,139", "251,178,23", "96,143,159", "1,77,103", "254,67,101", "252,157,154",

"249,205,173", "200,200,169", "131,175,155", "229,187,129", "161,23,21", "34,8,7",

"118,77,57", "17,63,61", "60,79,57", "95,92,51", "179,214,110", "248,147,29",

"227,160,93", "178,190,126", "114,111,238", "56,13,49", "89,61,67", "250,218,141",

"3,38,58", "179,168,150", "222,125,44", "20,68,106", "130,57,53", "137,190,178",

"201,186,131", "222,211,140", "222,156,83", "23,44,60", "39,72,98", "153,80,84",

"217,104,49", "230,179,61", "174,221,129", "107,194,53", "6,128,67", "38,157,128",

"178,200,187", "69,137,148", "117,121,71", "114,83,52", "87,105,60", "82,75,46",

"171,92,37", "100,107,48", "98,65,24", "54,37,17", "137,157,192", "250,227,113",

"29,131,8", "220,87,18", "29,191,151", "35,235,185", "213,26,33", "160,191,124",

"101,147,74", "64,116,52", "255,150,128", "255,94,72", "38,188,213", "167,220,224",

"1,165,175", "179,214,110", "248,147,29", "230,155,3", "209,73,78", "62,188,202",

"224,160,158", "161,47,47", "0,90,171", "107,194,53", "174,221,129", "6,128,67",

"38,157,128", "201,138,131", "220,162,151", "137,157,192", "175,215,237", "92,167,186",

"255,66,93", "147,224,255", "247,68,97", "185,227,217"};

int len = beautifulColors.length;

Random random = new Random();

String[] color = beautifulColors[random.nextInt(len)].split(",");

return new Color(Integer.parseInt(color[0]), Integer.parseInt(color[1]),

Integer.parseInt(color[2]));

}

/\*\*

\* 图片做圆角处理

\* @param image

\* @param cornerRadius

\* @return

\*/

public static BufferedImage makeRoundedCorner(BufferedImage image, int cornerRadius) {

int w = image.getWidth();

int h = image.getHeight();

BufferedImage output = new BufferedImage(w, h, BufferedImage.TYPE\_INT\_ARGB);

Graphics2D g2 = output.createGraphics();

g2.setComposite(AlphaComposite.Src);

g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);

g2.setColor(Color.WHITE);

g2.fill(new RoundRectangle2D.Float(0, 0, w, h, cornerRadius, cornerRadius));

g2.setComposite(AlphaComposite.SrcAtop);

g2.drawImage(image, 0, 0, null);

g2.dispose();

return output;

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\ConstantUtil.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/10/23

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

public class ConstantUtil {

// public static final String STATIC\_RESOURCES = "C:/Users/Administrator/Pictures/webstatic";

// public static final String STATIC\_URL = "http://localhost:8000";

// deploy

public static final String STATIC\_RESOURCES = "/home/jancoyan/springbootApp/blog/static";

public static final String STATIC\_URL = "http://101.201.64.102:8000";

}R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\FileUtils.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/10/16

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import java.io.File;

import java.io.IOException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class FileUtils {

/\*\*

\* 删除文章中的指定图片

\* @param date 日期

\* @param imageName 图片的名称

\*/

public static void deleteImageIfExists(Date date, String imageName) {

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM");

String formatMonth = simpleDateFormat.format(date);

String path = null;

path = ConstantUtil.STATIC\_RESOURCES + "/p/" + formatMonth + "/" + imageName;

File file = new File(path);

if(file.exists()){

file.delete();

}

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\JsonWebTokenUtils.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/9/25

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import com.auth0.jwt.JWT;

import com.auth0.jwt.JWTVerifier;

import com.auth0.jwt.algorithms.Algorithm;

import com.auth0.jwt.interfaces.Claim;

import com.auth0.jwt.interfaces.DecodedJWT;

import java.util.Calendar;

import java.util.Date;

import java.util.HashMap;

import java.util.Map;

public class JsonWebTokenUtils {

/\*\* token秘钥，请勿泄露，请勿随便修改 backups:JKKLJOoasdlfj \*/

public static final String SECRET = "JKKLJOoasdlfj";

/\*\* token 过期时间: 1天 \*/

public static final int calendarField = Calendar.DATE;

public static final int calendarInterval = 1;

/\*\*

\* JWT生成Token.<br/>

\* JWT构成: header, payload, signature

\* @param user\_id 登录成功后用户user\_id, 参数user\_id不可传空

\*/

public static String createToken(Long user\_id){

Date iatDate = new Date();

// expire time

Calendar nowTime = Calendar.getInstance();

nowTime.add(calendarField, calendarInterval);

Date expiresDate = nowTime.getTime();

// header Map

Map<String, Object> map = new HashMap<>();

map.put("alg", "HS256");

map.put("typ", "JWT");

// build token

// param backups {iss:Service, aud:APP}

String token = JWT.create().withHeader(map) // header

.withClaim("iss", "Service") // payload

.withClaim("aud", "APP")

.withClaim("user\_id", null == user\_id ? null : user\_id.toString())

.withIssuedAt(iatDate) // sign time

.withExpiresAt(expiresDate) // expire time

.sign(Algorithm.HMAC256(SECRET)); // signature

return token;

}

/\*\*

\* 解密Token

\* @param token token字符串

\* @return 对token进行解密

\* @throws Exception token非法异常

\*/

public static Map<String, Claim> verifyToken(String token) {

// String[] header = token.split("Bearer");

// token = header[1];

DecodedJWT jwt = null;

try {

JWTVerifier verifier = JWT.require(Algorithm.HMAC256(SECRET)).build();

jwt = verifier.verify(token);

} catch (Exception e) {

e.printStackTrace();

System.out.println("Token 非法");

}

return jwt.getClaims();

}

/\*\*

\* 根据Token获取user\_id

\* @param token token

\* @return user\_id

\*/

public static Long getAppUID(String token) {

Map<String, Claim> claims = verifyToken(token);

Claim user\_id\_claim = claims.get("user\_id");

if (null == user\_id\_claim || "".equals(user\_id\_claim.asString())) {

// token 校验失败, 抛出Token验证非法异常

System.out.println("token 验证非法");

}

return Long.valueOf(user\_id\_claim.asString());

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\MD5Util.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/9/18

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

public class MD5Util {

public static String getMD5(String password) {

try {

// 得到一个信息摘要器

MessageDigest digest = MessageDigest.getInstance("md5");

byte[] result = digest.digest(password.getBytes());

StringBuffer buffer = new StringBuffer();

// 把每一个byte 做一个与运算 0xff;

for (byte b : result) {

// 与运算

int number = b & 0xff;// 加盐

String str = Integer.toHexString(number);

if (str.length() == 1) {

buffer.append("0");

}

buffer.append(str);

}

// 标准的md5加密后的结果

return buffer.toString();

} catch (NoSuchAlgorithmException e) {

e.printStackTrace();

return "";

}

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\Msg.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/9/14

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import lombok.Data;

import java.util.HashMap;

import java.util.Map;

/\*\*

\* 通用的返回的类

\* @author Jancoyan

\*/

@Data

public class Msg {

/\*\*

\* 状态码

\* 100-成功

\* 200-失败

\*

\* \*/

private int code;

//提示信息

private String msg;

//用户要返回给浏览器的数据

private Map<String, Object> extend = new HashMap<String, Object>();

/\*\*

\* 用户信息过期

\* @return 结果

\*/

public static Msg expire(){

Msg result = new Msg();

result.setCode(601);

result.setMsg("用户信息已过期");

return result;

}

/\*\*

\* 非法token

\* @return 结果

\*/

public static Msg illegalToken(){

Msg result = new Msg();

result.setCode(602);

result.setMsg("登录信息无效");

return result;

}

/\*\*

\* 数据传输成功

\* @return 结果

\*/

public static Msg success(){

Msg result = new Msg();

result.setCode(100);

result.setMsg("成功");

return result;

}

/\*\*

\* 用户未登录

\* @return 结果

\*/

public static Msg loginNeeded(){

Msg result = new Msg();

result.setCode(603);

result.setMsg("用户未登录");

return result;

}

/\*\*

\* 数据传输失败

\* @return 结果

\*/

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;

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\NetworkUtils.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/10/15

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import org.json.JSONObject;

import java.io.BufferedReader;

import java.io.DataOutputStream;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URL;

import java.nio.charset.StandardCharsets;

public class NetworkUtils {

/\*\*

\* 查找ip归属地

\* @param ip ip地址

\* @return

\*/

public static String queryAddressByIp(String ip) {

String address = null;

try {

address = getAddress(ip);

} catch (Exception e) {

e.printStackTrace();

}

return address;

}

/\*\*

\* 获取地址

\* @param ip

\* @return

\* @throws Exception

\*/

private static String getAddress(String ip) throws Exception{

String path = "https://ip.taobao.com//outGetIpInfo?ip=" + ip + "&accessKey=alibaba-inc";

String returnStr = getRs(path);

JSONObject json;

if(returnStr != null){

json = new JSONObject(returnStr);

if("0".equals(json.get("code").toString())){

StringBuffer buffer = new StringBuffer();

//国家

buffer.append(decodeUnicode(json.optJSONObject("data").getString("country")));

//省份

buffer.append(decodeUnicode(json.optJSONObject("data").getString("region")));

//市区

buffer.append(decodeUnicode(json.optJSONObject("data").getString("city")));

return buffer.toString();

}else{

return "获取地址失败";

}

}

return null;

}

/\*\*

\* 从url获取结果

\* @param path

\* @param params

\* @param encoding

\* @return

\*/

private static String getRs(String path){

URL url = null;

HttpURLConnection connection = null;

try {

url = new URL(path);

// 新建连接实例

connection = (HttpURLConnection)url.openConnection();

// 设置连接超时时间，单位毫秒

connection.setConnectTimeout(2000);

// 设置读取数据超时时间，单位毫秒

connection.setReadTimeout(2000);

// 是否打开输出流 true|false

connection.setDoInput(true);

// 是否打开输入流true|false

connection.setDoOutput(true);

// 提交方法POST|GET

connection.setRequestMethod("POST");

// 是否缓存true|false

connection.setUseCaches(false);

// 打开连接端口

connection.connect();

DataOutputStream out = new DataOutputStream(connection.getOutputStream());

out.writeBytes(path);

out.flush();

out.close();

BufferedReader reader =

new BufferedReader(new InputStreamReader(connection.getInputStream(), StandardCharsets.UTF\_8));

StringBuilder buffer = new StringBuilder();

String line = "";

while ((line = reader.readLine())!= null) {

buffer.append(line);

}

reader.close();

return buffer.toString();

} catch (Exception e) {

e.printStackTrace();

}finally{

connection.disconnect();// 关闭连接

}

return null;

}

/\*\*

\* 对字符串进行解码

\* @param theString

\* @return

\*/

private static String decodeUnicode(String theString){

char aChar;

int len = theString.length();

StringBuffer buffer = new StringBuffer(len);

for (int i = 0; i < len;) {

aChar = theString.charAt(i++);

if(aChar == '\\'){

aChar = theString.charAt(i++);

if(aChar == 'u'){

int val = 0;

for(int j = 0; j < 4; j++){

aChar = theString.charAt(i++);

switch (aChar) {

case '0':case '1':case '2':case '3':

case '4':case '5':case '6':case '7':

case '8':case '9':

val = (val << 4) + aChar - '0';

break;

case 'a':case 'b':case 'c':case 'd':

case 'e':case 'f':

val = (val << 4) + 10 + aChar - 'a';

break;

case 'A':case 'B':case 'C':case 'D':

case 'E':case 'F':

val = (val << 4) + 10 + aChar - 'A';

break;

default:

throw new IllegalArgumentException("Malformed encoding.");

}

}

buffer.append((char) val);

}else{

if(aChar == 't'){ aChar = '\t';}

if(aChar == 'r'){ aChar = '\r';}

if(aChar == 'n'){ aChar = '\n';}

if(aChar == 'f'){ aChar = '\f';}

buffer.append(aChar);

}

}else{

buffer.append(aChar);

}

}

return buffer.toString();

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\RedisUtil.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/9/25

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.redis.core.RedisTemplate;

import org.springframework.stereotype.Component;

import org.springframework.util.CollectionUtils;

import java.util.Collection;

import java.util.List;

import java.util.Map;

import java.util.Set;

import java.util.concurrent.TimeUnit;

@Component

public class RedisUtil {

@Autowired

private RedisTemplate<String, Object> redisTemplate;

// =============================common============================

/\*\*

\* 指定缓存失效时间

\*

\* @param key 键

\* @param time 时间(秒)

\*/

public boolean expire(String key, long time) {

try {

if (time > 0) {

redisTemplate.expire(key, time, TimeUnit.SECONDS);

}

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 根据key 获取过期时间

\*

\* @param key 键 不能为null

\* @return 时间(秒) 返回0代表为永久有效

\*/

public long getExpire(String key) {

return redisTemplate.getExpire(key, TimeUnit.SECONDS);

}

/\*\*

\* 判断key是否存在

\*

\* @param key 键

\* @return true 存在 false不存在

\*/

public boolean hasKey(String key) {

try {

return redisTemplate.hasKey(key);

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 删除缓存

\* @param key 可以传一个值 或多个

\*/

@SuppressWarnings("unchecked")

public void del(String... key) {

if (key != null && key.length > 0) {

if (key.length == 1) {

redisTemplate.delete(key[0]);

} else {

redisTemplate.delete((Collection<String>) CollectionUtils.arrayToList(key));

}

}

}

// ============================String=============================

/\*\*

\* 普通缓存获取

\*

\* @param key 键

\* @return 值

\*/

public Object get(String key) {

return key == null ? null : redisTemplate.opsForValue().get(key);

}

/\*\*

\* 普通缓存放入

\*

\* @param key 键

\* @param value 值

\* @return true成功 false失败

\*/

public boolean set(String key, Object value) {

try {

redisTemplate.opsForValue().set(key, value);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 普通缓存放入并设置时间

\*

\* @param key 键

\* @param value 值

\* @param time 时间(秒) time要大于0 如果time小于等于0 将设置无限期

\* @return true成功 false 失败

\*/

public boolean set(String key, Object value, long time) {

try {

if (time > 0) {

redisTemplate.opsForValue().set(key, value, time,

TimeUnit.SECONDS);

} else {

set(key, value);

}

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 递增

\*

\* @param key 键

\* @param delta 要增加几(大于0)

\*/

public long incr(String key, long delta) {

if (delta < 0) {

throw new RuntimeException("递增因子必须大于0");

}

return redisTemplate.opsForValue().increment(key, delta);

}

/\*\*

\* 递减

\*

\* @param key 键

\* @param delta 要减少几(小于0)

\*/

public long decr(String key, long delta) {

if (delta < 0) {

throw new RuntimeException("递减因子必须大于0");

}

return redisTemplate.opsForValue().increment(key, -delta);

}

// ================================Map=================================

/\*\*

\* HashGet

\*

\* @param key 键 不能为null

\* @param item 项 不能为null

\*/

public Object hget(String key, String item) {

return redisTemplate.opsForHash().get(key, item);

}

/\*\*

\* 获取hashKey对应的所有键值

\*

\* @param key 键

\* @return 对应的多个键值

\*/

public Map<Object, Object> hmget(String key) {

return redisTemplate.opsForHash().entries(key);

}

/\*\*

\* HashSet

\*

\* @param key 键

\* @param map 对应多个键值

\*/

public boolean hmset(String key, Map<String, Object> map) {

try {

redisTemplate.opsForHash().putAll(key, map);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* HashSet 并设置时间

\*

\* @param key 键

\* @param map 对应多个键值

\* @param time 时间(秒)

\* @return true成功 false失败

\*/

public boolean hmset(String key, Map<String, Object> map, long time) {

try {

redisTemplate.opsForHash().putAll(key, map);

if (time > 0) {

expire(key, time);

}

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 向一张hash表中放入数据,如果不存在将创建

\*

\* @param key 键

\* @param item 项

\* @param value 值

\* @return true 成功 false失败

\*/

public boolean hset(String key, String item, Object value) {

try {

redisTemplate.opsForHash().put(key, item, value);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 向一张hash表中放入数据,如果不存在将创建

\*

\* @param key 键

\* @param item 项

\* @param value 值

\* @param time 时间(秒) 注意:如果已存在的hash表有时间,这里将会替换原有的时间

\* @return true 成功 false失败

\*/

public boolean hset(String key, String item, Object value, long time) {

try {

redisTemplate.opsForHash().put(key, item, value);

if (time > 0) {

expire(key, time);

}

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 删除hash表中的值

\*

\* @param key 键 不能为null

\* @param item 项 可以使多个 不能为null

\*/

public void hdel(String key, Object... item) {

redisTemplate.opsForHash().delete(key, item);

}

/\*\*

\* 判断hash表中是否有该项的值

\*

\* @param key 键 不能为null

\* @param item 项 不能为null

\* @return true 存在 false不存在

\*/

public boolean hHasKey(String key, String item) {

return redisTemplate.opsForHash().hasKey(key, item);

}

/\*\*

\* hash递增 如果不存在,就会创建一个 并把新增后的值返回

\*

\* @param key 键

\* @param item 项

\* @param by 要增加几(大于0)

\*/

public double hincr(String key, String item, double by) {

return redisTemplate.opsForHash().increment(key, item, by);

}

/\*\*

\* hash递减

\*

\* @param key 键

\* @param item 项

\* @param by 要减少记(小于0)

\*/

public double hdecr(String key, String item, double by) {

return redisTemplate.opsForHash().increment(key, item, -by);

}

// ============================set=============================

/\*\*

\* 根据key获取Set中的所有值

\*

\* @param key 键

\*/

public Set<Object> sGet(String key) {

try {

return redisTemplate.opsForSet().members(key);

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

/\*\*

\* 根据value从一个set中查询,是否存在

\*

\* @param key 键

\* @param value 值

\* @return true 存在 false不存在

\*/

public boolean sHasKey(String key, Object value) {

try {

return redisTemplate.opsForSet().isMember(key, value);

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 将数据放入set缓存

\*

\* @param key 键

\* @param values 值 可以是多个

\* @return 成功个数

\*/

public long sSet(String key, Object... values) {

try {

return redisTemplate.opsForSet().add(key, values);

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

/\*\*

\* 将set数据放入缓存

\*

\* @param key 键

\* @param time 时间(秒)

\* @param values 值 可以是多个

\* @return 成功个数

\*/

public long sSetAndTime(String key, long time, Object... values) {

try {

Long count = redisTemplate.opsForSet().add(key, values);

if (time > 0)

expire(key, time);

return count;

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

/\*\*

\* 获取set缓存的长度

\*

\* @param key 键

\*/

public long sGetSetSize(String key) {

try {

return redisTemplate.opsForSet().size(key);

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

/\*\*

\* 移除值为value的

\*

\* @param key 键

\* @param values 值 可以是多个

\* @return 移除的个数

\*/

public long setRemove(String key, Object... values) {

try {

Long count = redisTemplate.opsForSet().remove(key, values);

return count;

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

// ===============================list=================================

/\*\*

\* 获取list缓存的内容

\*

\* @param key 键

\* @param start 开始

\* @param end 结束 0 到 -1代表所有值

\*/

public List<Object> lGet(String key, long start, long end) {

try {

return redisTemplate.opsForList().range(key, start, end);

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

/\*\*

\* 获取list缓存的长度

\*

\* @param key 键

\*/

public long lGetListSize(String key) {

try {

return redisTemplate.opsForList().size(key);

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

/\*\*

\* 通过索引 获取list中的值

\*

\* @param key 键

\* @param index 索引 index>=0时， 0 表头，1 第二个元素，依次类推；index<0

\* 时，-1，表尾，-2倒数第二个元素，依次类推

\*/

public Object lGetIndex(String key, long index) {

try {

return redisTemplate.opsForList().index(key, index);

} catch (Exception e) {

e.printStackTrace();

return null;

}

}

/\*\*

\* 将list放入缓存

\*

\* @param key 键

\* @param value 值

\*/

public boolean lSet(String key, Object value) {

try {

redisTemplate.opsForList().rightPush(key, value);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 将list放入缓存

\*

\* @param key 键

\* @param value 值

\* @param time 时间(秒)

\*/

public boolean lSet(String key, Object value, long time) {

try {

redisTemplate.opsForList().rightPush(key, value);

if (time > 0)

expire(key, time);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 将list放入缓存

\*

\* @param key 键

\* @param value 值

\* @return

\*/

public boolean lSet(String key, List<Object> value) {

try {

redisTemplate.opsForList().rightPushAll(key, value);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 将list放入缓存

\*

\* @param key 键

\* @param value 值

\* @param time 时间(秒)

\* @return

\*/

public boolean lSet(String key, List<Object> value, long time) {

try {

redisTemplate.opsForList().rightPushAll(key, value);

if (time > 0)

expire(key, time);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 根据索引修改list中的某条数据

\*

\* @param key 键

\* @param index 索引

\* @param value 值

\* @return

\*/

public boolean lUpdateIndex(String key, long index, Object value) {

try {

redisTemplate.opsForList().set(key, index, value);

return true;

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

/\*\*

\* 移除N个值为value

\*

\* @param key 键

\* @param count 移除多少个

\* @param value 值

\* @return 移除的个数

\*/

public long lRemove(String key, long count, Object value) {

try {

Long remove = redisTemplate.opsForList().remove(key, count,

value);

return remove;

} catch (Exception e) {

e.printStackTrace();

return 0;

}

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\TimeUtils.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/5/28

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

public class TimeUtils {

/\*\*

\* 获取当前系统时间，格式为 yyyy-MM-dd HH:mm:ss

\* @return 时间字符串

\*/

public static String getCurrentTimeString(){

SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");//设置日期格式

return df.format(new Date());

}

/\*\*

\* 将java Date 将对象转化为日期字符串形式

\* @param date

\* @return

\*/

public static String castDateTypeToDateString(Date date){

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyy-MM-dd");

return simpleDateFormat.format(date);

}

/\*\*

\* 把日期字符串转化为 yyyy-MM-dd 日期类型

\* @param dateStr 日期字符串

\* @return 日期类型

\*/

public static Date castDateStringToDateTypeYMD(String dateStr){

Date date = null;

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

try {

date = sdf.parse(dateStr);

} catch (ParseException e) {

e.printStackTrace();

}

return date;

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\UserAgentUtils.java

/\*\*

\* @Author: Yan Jingcun

\* @Date: 2021/10/15

\* @Description:

\* @Version: 1.0

\*/

package com.jancoyan.jancoblog.utils;

import eu.bitwalker.useragentutils.Browser;

import eu.bitwalker.useragentutils.OperatingSystem;

import eu.bitwalker.useragentutils.UserAgent;

import org.apache.commons.lang3.StringUtils;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

import javax.servlet.http.HttpServletRequest;

public class UserAgentUtils {

private static Logger logger = LoggerFactory.getLogger(UserAgentUtils.class);

/\*\*

\* 根据http获取userAgent信息

\* @param request

\* @return

\*/

public static String getUserAgent(HttpServletRequest request) {

String userAgent=request.getHeader("User-Agent");

return userAgent;

}

/\*\*

\* 根据request获取userAgent，然后解析出osVersion

\* @param request

\* @return

\*/

public static String getOsVersion(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getOsVersion(userAgent);

}

/\*\*

\* 根据userAgent解析出osVersion

\* @param userAgent

\* @return

\*/

public static String getOsVersion(String userAgent) {

String osVersion = "";

if(StringUtils.isBlank(userAgent))

return osVersion;

String[] strArr = userAgent.substring(userAgent.indexOf("(")+1,

userAgent.indexOf(")")).split(";");

if(null == strArr || strArr.length == 0)

return osVersion;

osVersion = strArr[1];

logger.info("osVersion is:{}", osVersion);

return osVersion;

}

/\*\*

\* 获取操作系统对象

\* @param request

\* @return

\*/

private static OperatingSystem getOperatingSystem(String userAgent) {

UserAgent agent = UserAgent.parseUserAgentString(userAgent);

OperatingSystem operatingSystem = agent.getOperatingSystem();

return operatingSystem;

}

/\*\*

\* 获取os：Windows/ios/Android

\* @param request

\* @return

\*/

public static String getOs(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getOs(userAgent);

}

/\*\*

\* 获取os：Windows/ios/Android

\* @param request

\* @return

\*/

public static String getOs(String userAgent) {

OperatingSystem operatingSystem = getOperatingSystem(userAgent);

String os = operatingSystem.getGroup().getName();

logger.info("os is:{}", os);

return os;

}

/\*\*

\* 获取deviceType

\* @param request

\* @return

\*/

public static String getDevicetype(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getDevicetype(userAgent);

}

/\*\*

\* 获取deviceType

\* @return

\*/

public static String getDevicetype(String userAgent) {

OperatingSystem operatingSystem = getOperatingSystem(userAgent);

String deviceType = operatingSystem.getDeviceType().toString();

logger.info("deviceType is:{}", deviceType);

return deviceType;

}

/\*\*

\* 获取操作系统的名字

\* @param request

\* @return

\*/

public static String getOsName(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getOsName(userAgent);

}

/\*\*

\* 获取操作系统的名字

\* @return

\*/

public static String getOsName(String userAgent) {

OperatingSystem operatingSystem = getOperatingSystem(userAgent);

String osName = operatingSystem.getName();

logger.info("osName is:{}", osName);

return osName;

}

/\*\*

\* 获取device的生产厂家

\* @param request

\*/

public static String getDeviceManufacturer(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getDeviceManufacturer(userAgent);

}

/\*\*

\* 获取device的生产厂家

\*/

public static String getDeviceManufacturer(String userAgent) {

OperatingSystem operatingSystem = getOperatingSystem(userAgent);

String deviceManufacturer = operatingSystem.getManufacturer().toString();

logger.info("deviceManufacturer is:{}", deviceManufacturer);

return deviceManufacturer;

}

/\*\*

\* 获取浏览器对象

\* @return

\*/

public static Browser getBrowser(String agent) {

UserAgent userAgent = UserAgent.parseUserAgentString(agent);

Browser browser = userAgent.getBrowser();

return browser;

}

/\*\*

\* 获取browser name

\* @param request

\* @return

\*/

public static String getBorderName(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getBorderName(userAgent);

}

/\*\*

\* 获取browser name

\* @return

\*/

public static String getBorderName(String userAgent) {

Browser browser = getBrowser(userAgent);

String borderName = browser.getName();

logger.info("borderName is:{}", borderName);

return borderName;

}

/\*\*

\* 获取浏览器的类型

\* @param request

\* @return

\*/

public static String getBorderType(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getBorderType(userAgent);

}

/\*\*

\* 获取浏览器的类型

\* @return

\*/

public static String getBorderType(String userAgent) {

Browser browser = getBrowser(userAgent);

String borderType = browser.getBrowserType().getName();

logger.info("borderType is:{}", borderType);

return borderType;

}

/\*\*

\* 获取浏览器组： CHROME、IE

\* @param request

\* @return

\*/

public static String getBorderGroup(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getBorderGroup(userAgent);

}

/\*\*

\* 获取浏览器组： CHROME、IE

\* @return

\*/

public static String getBorderGroup(String userAgent) {

Browser browser = getBrowser(userAgent);

String browerGroup = browser.getGroup().getName();

logger.info("browerGroup is:{}", browerGroup);

return browerGroup;

}

/\*\*

\* 获取浏览器的生产厂商

\* @param request

\* @return

\*/

public static String getBrowserManufacturer(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getBrowserManufacturer(userAgent);

}

/\*\*

\* 获取浏览器的生产厂商

\* @return

\*/

public static String getBrowserManufacturer(String userAgent) {

Browser browser = getBrowser(userAgent);

String browserManufacturer = browser.getManufacturer().getName();

logger.info("browserManufacturer is:{}", browserManufacturer);

return browserManufacturer;

}

/\*\*

\* 获取浏览器使用的渲染引擎

\* @param request

\* @return

\*/

public static String getBorderRenderingEngine(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getBorderRenderingEngine(userAgent);

}

/\*\*

\* 获取浏览器使用的渲染引擎

\* @return

\*/

public static String getBorderRenderingEngine(String userAgent) {

Browser browser = getBrowser(userAgent);

String renderingEngine = browser.getRenderingEngine().name();

logger.info("renderingEngine is:{}", renderingEngine);

return renderingEngine;

}

/\*\*

\* 获取浏览器版本

\* @param request

\* @return

\*/

public static String getBrowserVersion(HttpServletRequest request) {

String userAgent = getUserAgent(request);

return getBrowserVersion(userAgent);

}

/\*\*

\* 获取浏览器版本

\* @return

\*/

public static String getBrowserVersion(String userAgent) {

Browser browser = getBrowser(userAgent);

String borderVersion = browser.getVersion( userAgent).toString();

return borderVersion;

}

// public static void main(String[] args) {

// String winUserAgent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36";

// System.out.println("浏览器名字："+getBorderName(winUserAgent));

// System.out.println("浏览器版本："+getBrowserVersion(winUserAgent));

// System.out.println("操作系统的名字："+getOsName(winUserAgent));

// System.out.println("操作系统的版本号："+getOsVersion(winUserAgent));

// }

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\java\com\jancoyan\jancoblog\utils\VerifyCodeUtil.java

package com.jancoyan.jancoblog.utils;

import javax.imageio.ImageIO;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import java.awt.\*;

import java.awt.image.BufferedImage;

import java.util.Random;

/\*\*

\* @author 遗憾就遗憾吧

\* @version 1.3

\* @Date 2019/9/16

\* @datatime:?serverTimezone=UTC&characterEncoding=utf-8&userSSL=yes

\* @since 1.2

\*/

public class VerifyCodeUtil {

//TODO 图片验证生成器

public static final String RANDOMCODEKEY= "RANDOMVALIDATECODEKEY";//放到session中的key

private String randString = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";//随机产生数字与字母组合的字符串

private int width = 160;// 图片宽

private int height = 40;// 图片高

private int lineSize = 40;// 干扰线数量

private int stringNum = 4;// 随机产生字符数量

private Random random = new Random();

/\*\*

\* 获得字体

\*/

private Font getFont() {

return new Font("Fixedsys", Font.CENTER\_BASELINE, 28);

}

/\*\*

\* 获得颜色

\*/

private Color getRandColor(int fc, int bc) {

if (fc > 255)

fc = 255;

if (bc > 255)

bc = 255;

int r = fc + random.nextInt(bc - fc - 16);

int g = fc + random.nextInt(bc - fc - 14);

int b = fc + random.nextInt(bc - fc - 18);

return new Color(r, g, b);

}

/\*\*

\* 生成随机图片

\*/

public String getRandcode(HttpServletRequest request, HttpServletResponse response) {

HttpSession session = request.getSession();

// BufferedImage类是具有缓冲区的Image类,Image类是用于描述图像信息的类

BufferedImage image = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_BGR);

Graphics g = image.getGraphics();// 产生Image对象的Graphics对象,改对象可以在图像上进行各种绘制操作

g.fillRect(0, 0, width, height);//图片大小

g.setFont(new Font("Times New Roman", Font.ROMAN\_BASELINE, 18));//字体大小

g.setColor(getRandColor(110, 133));//字体颜色

// 绘制干扰线

for (int i = 0; i <= lineSize; i++) {

drowLine(g);

}

// 绘制随机字符

String randomString = "";

for (int i = 1; i <= stringNum; i++) {

randomString = drowString(g, randomString, i);

}

session.removeAttribute(RANDOMCODEKEY);

session.setAttribute(RANDOMCODEKEY, randomString);

g.dispose();

try {

// 将内存中的图片通过流动形式输出到客户端

ImageIO.write(image, "JPEG", response.getOutputStream());

} catch (Exception e) {

e.printStackTrace();

}

return randomString;

}

/\*\*

\* 绘制字符串

\*/

private String drowString(Graphics g, String randomString, int i) {

g.setFont(getFont());

g.setColor(new Color(random.nextInt(101), random.nextInt(111), random

.nextInt(121)));

String rand = String.valueOf(getRandomString(random.nextInt(randString

.length())));

randomString += rand;

g.translate(random.nextInt(3), random.nextInt(3));

g.drawString(rand, 30 \* i, 21);

return randomString;

}

/\*\*

\* 绘制干扰线

\*/

private void drowLine(Graphics g) {

int x = random.nextInt(width);

int y = random.nextInt(height);

int xl = random.nextInt(13);

int yl = random.nextInt(15);

g.drawLine(x, y, x + xl, y + yl);

}

/\*\*

\* 获取随机的字符

\*/

public String getRandomString(int num) {

return String.valueOf(randString.charAt(num));

}

}

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\application.yml

jar:

fileUrl: /home/jancoyan/springbootApp/blog/static

# fileUrl: C:/Users/Administrator/Pictures/webstatic

spring:

datasource:

type: com.alibaba.druid.pool.DruidDataSource

driver-class-name: com.mysql.cj.jdbc.Driver

# url: jdbc:mysql://101.201.64.102/jancoblog?serverTimezone=UTC&characterEncoding=utf-8

url: jdbc:mysql://localhost/jancoblog?serverTimezone=UTC&characterEncoding=utf-8

username: jancoblog

password: 333

devtools:

restart:

enabled: true

additional-paths: src/main/java

mvc:

view:

prefix: /

suffix: .html

web:

resources:

static-locations: file:${jar.fileUrl},classpath:/static/,

redis:

host: 101.201.64.102

port: 6379

password: 333

servlet:

multipart:

max-file-size: 50MB

max-request-size: 50MB

server:

max-http-header-size: 1MB

port: 8000

logging:

level:

com.jancoyan.jancoblog.mapper: debug

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\ArticleImageMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.ArticleImageMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.ArticleImage">

<id column="filename" property="filename" />

<result column="article\_id" property="articleId" />

<result column="insert\_date" property="insertDate" />

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

filename, article\_id, insert\_date

</sql>

</mapper>

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\ArticleMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.ArticleMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.Article">

<id column="article\_id" property="articleId" />

<result column="article\_title" property="articleTitle" />

<result column="article\_author" property="articleAuthor" />

<result column="article\_type" property="articleType" />

<result column="article\_summary" property="articleSummary" />

<result column="article\_html" property="articleHtml" />

<result column="article\_md" property="articleMd" />

<result column="article\_is\_comment" property="articleIsComment" />

<result column="article\_rank" property="articleRank" />

<result column="article\_post\_time" property="articlePostTime" />

<result column="article\_edit\_time" property="articleEditTime" />

<result column="article\_view\_count" property="articleViewCount" />

<result column="article\_comment\_count" property="articleCommentCount" />

<result column="article\_like\_count" property="articleLikeCount" />

<result column="user\_name" property="userName" />

<result column="type\_name" property="typeName"/>

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

article\_id, article\_title, article\_author, article\_type, article\_summary, article\_html, article\_md, article\_is\_comment, article\_rank, article\_post\_time, article\_edit\_time, article\_view\_count, article\_comment\_count, article\_like\_count

</sql>

<!-- 首页文章列表-->

<select id="getIndexList" resultType="com.jancoyan.jancoblog.pojo.Article">

select \* from v\_article\_list\_show ${ew.customSqlSegment}

</select>

<!-- 管理员能看到的文章列表-->

<select id="getManageList" resultType="com.jancoyan.jancoblog.pojo.Article">

select \* from v\_article\_manage\_list ${ew.customSqlSegment}

</select>

<!-- 查看文章的时候获取文章整体信息 -->

<select id="getSingleArticle" resultType="com.jancoyan.jancoblog.pojo.Article">

select \* from v\_article\_get\_single where article\_id = '${id}'

</select>

<select id="getSingleArticleDeleted" resultType="com.jancoyan.jancoblog.pojo.Article">

select \* from v\_article\_get\_single\_deleted where article\_id = '${id}'

</select>

<!-- 获取删除的文章-->

<select id="getDeletedList" resultType="com.jancoyan.jancoblog.pojo.Article">

select \* from v\_article\_deleted\_list ${ew.customSqlSegment}

</select>

<!-- 彻底删除文章-->

<delete id="deleteCompletely">

delete from tbl\_deleted\_article where article\_id = '${articleId}'

</delete>

<!-- 恢复用户的文章-->

<select id="batchRecover">

CALL proc\_recover\_article(${articleId});

</select>

<!-- 获取用户最近发表的文章 -->

<select id="getArticleByUserRecently"

resultType="com.jancoyan.jancoblog.pojo.PageArticle">

select \* from v\_page\_article ${ew.customSqlSegment}

</select>

<!-- 获取用户文章用于编辑文章-->

<select id="getArticleEdit" resultType="com.jancoyan.jancoblog.pojo.Article">

select article\_id,

article\_title,

article\_summary,

article\_md,

article\_type,

article\_is\_comment

from tbl\_article where article\_id = '${id}'

</select>

</mapper>

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\CommentMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.CommentMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.Comment">

<id column="comment\_id" property="commentId" />

<result column="pre\_comment\_id" property="preCommentId" />

<result column="comment\_article\_id" property="commentArticleId" />

<result column="comment\_author\_id" property="commentAuthorId" />

<result column="comment\_author\_name" property="commentAuthorName" />

<result column="comment\_author\_email" property="commentAuthorEmail" />

<result column="comment\_content" property="commentContent" />

<result column="comment\_date" property="commentDate" />

<result column="comment\_author\_ip" property="commentAuthorIp" />

<result column="comment\_like\_count" property="commentLikeCount" />

<result column="article\_title" property="articleTitle" />

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

comment\_id, pre\_comment\_id, comment\_article\_id, comment\_author\_id, comment\_author\_name, comment\_author\_email, comment\_content, comment\_date, comment\_author\_ip, comment\_like\_count

</sql>

<update id="recoverComment">

CALL proc\_recover\_comment(${commentId});

</update>

<delete id="deleteComment">

CALL proc\_transfer\_comment(${commentId});

</delete>

<select id="getAll" resultType="com.jancoyan.jancoblog.pojo.Comment">

select \* from v\_comment\_manage\_list ${ew.customSqlSegment}

</select>

<select id="getCommentByArticle" resultType="com.jancoyan.jancoblog.pojo.Comment">

select \* from v\_comment\_article ${ew.customSqlSegment}

</select>

<select id="getCommentByUserPosted" resultType="com.jancoyan.jancoblog.pojo.Comment">

select \* from v\_comment\_manage\_list ${ew.customSqlSegment}

</select>

<!-- 通用查询映射结果 -->

<resultMap id="PageComment" type="com.jancoyan.jancoblog.pojo.PageComment">

<result column="comment\_author\_name" property="commentAuthorName" />

<result column="article\_id" property="articleId" />

<result column="article\_author" property="articleAuthor" />

<result column="article\_title" property="articleTitle" />

<result column="comment\_date" property="commentDate" />

</resultMap>

<sql id="Page\_Comment">

comment\_author\_name, article\_id, article\_author, article\_title, comment\_date

</sql>

<select id="getCommentByUserRecently"

resultType="com.jancoyan.jancoblog.pojo.PageComment">

select <include refid="Page\_Comment" />

from v\_page\_comment ${ew.customSqlSegment}

</select>

</mapper>

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\DeletedCommentMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.DeletedCommentMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.DeletedComment">

<id column="comment\_id" property="commentId" />

<result column="pre\_comment\_id" property="preCommentId" />

<result column="comment\_article\_id" property="commentArticleId" />

<result column="comment\_author\_id" property="commentAuthorId" />

<result column="comment\_author\_name" property="commentAuthorName" />

<result column="comment\_author\_email" property="commentAuthorEmail" />

<result column="comment\_content" property="commentContent" />

<result column="comment\_date" property="commentDate" />

<result column="comment\_author\_ip" property="commentAuthorIp" />

<result column="comment\_like\_count" property="commentLikeCount" />

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

comment\_id, pre\_comment\_id, comment\_article\_id, comment\_author\_id, comment\_author\_name, comment\_author\_email, comment\_content, comment\_date, comment\_author\_ip, comment\_like\_count

</sql>

<select id="getCommentByArticle"

resultType="com.jancoyan.jancoblog.pojo.DeletedComment">

select \* from v\_comment\_article\_deleted ${ew.customSqlSegment}

</select>

</mapper>

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\LikeRecordMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.LikeRecordMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.LikeRecord">

<id column="like\_id" property="likeId" />

<result column="article\_id" property="articleId" />

<result column="author\_id" property="authorId" />

<result column="like\_date" property="likeDate" />

<result column="user\_name" property="userName" />

<result column="article\_title" property="articleTitle" />

<result column="article\_author" property="articleAuthor" />

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

like\_id, article\_id, author\_id, like\_date,

</sql>

<sql id="Column\_List">

article\_id, article\_author, article\_title, like\_date, user\_name

</sql>

<select id="getUserReceive" resultType="com.jancoyan.jancoblog.pojo.LikeRecord">

select <include refid="Column\_List" />

from v\_page\_like ${ew.customSqlSegment}

</select>

</mapper>

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\TypeMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.TypeMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.Type">

<id column="type\_id" property="typeId" />

<result column="type\_name" property="typeName" />

<result column="type\_description" property="typeDescription" />

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

type\_id, type\_name, type\_description

</sql>

</mapper>

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\UserInfoMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.UserInfoMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.UserInfo">

<id column="user\_id" property="userId" />

<result column="user\_email" property="userEmail" />

<result column="user\_sex" property="userSex" />

<result column="user\_region" property="userRegion" />

<result column="user\_birthdate" property="userBirthdate" />

<result column="user\_telephone" property="userTelephone" />

<result column="user\_real\_name" property="userRealName" />

<result column="user\_school" property="userSchool" />

<result column="user\_major" property="userMajor" />

<result column="user\_enter\_school\_date" property="userEnterSchoolDate" />

<result column="user\_academic\_degree" property="userAcademicDegree" />

<result column="user\_company" property="userCompany" />

<result column="user\_position" property="userPosition" />

<result column="user\_field" property="userField" />

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

user\_id, user\_email, user\_sex, user\_region, user\_birthdate, user\_telephone, user\_real\_name, user\_school, user\_major, user\_enter\_school\_date, user\_academic\_degree, user\_company, user\_position, user\_field

</sql>

</mapper>

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\UserLoginMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.UserLoginMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.UserLogin">

<id column="id" property="id" />

<result column="login\_user" property="loginUser" />

<result column="login\_date" property="loginDate" />

<result column="login\_ip" property="loginIp" />

<result column="login\_address" property="loginAddress" />

<result column="user\_agent" property="userAgent" />

<result column="browser\_name" property="browserName" />

<result column="browser\_version" property="browserVersion" />

<result column="os\_name" property="osName" />

<result column="os\_version" property="osVersion" />

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

id, login\_user, login\_date, login\_ip, login\_address, user\_agent, browser\_name,

browser\_version, os\_name, os\_version

</sql>

</mapper>

R:\GITHUB\JancoBlog\JancoBlog-v3.0\src\main\resources\mapper\UserMapper.xml

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.jancoyan.jancoblog.mapper.UserMapper">

<!-- 通用查询映射结果 -->

<resultMap id="BaseResultMap" type="com.jancoyan.jancoblog.pojo.User">

<id column="user\_id" property="userId" />

<result column="user\_name" property="userName" />

<result column="user\_password" property="userPassword" />

<result column="user\_signature" property="userSignature" />

<result column="user\_role" property="userRole" />

<result column="user\_create\_date" property="userCreateDate" />

<result column="user\_last\_login\_date" property="userLastLoginDate" />

<result column="user\_ip" property="userIp" />

</resultMap>

<!-- 通用查询结果列 -->

<sql id="Base\_Column\_List">

user\_id, user\_name, user\_signature, user\_password, user\_role, user\_create\_date,

user\_last\_login\_date, user\_ip

</sql>

<select id="getAll" resultType="com.jancoyan.jancoblog.pojo.User">

select \* from v\_user\_manage\_show ${ew.customSqlSegment}

</select>

<!-- 用户的数据统计信息查询结果列 -->

<resultMap id="UserTotalData" type="com.jancoyan.jancoblog.pojo.VUserTotalData">

<result column="article\_author" property="articleAuthor" />

<result column="user\_name" property="userName" />

<result column="user\_signature" property="userSignature" />

<result column="total\_article" property="totalArticle" />

<result column="total\_view\_count" property="totalViewCount" />

<result column="total\_like\_count" property="totalLikeCount" />

<result column="total\_comment\_count" property="totalCommentCount"/>

</resultMap>

<sql id="User\_Total\_Data\_List">

article\_author, user\_name, user\_signature, total\_article, total\_view\_count,

total\_like\_count, total\_comment\_count

</sql>

<select id="getUserTotalData" resultType="com.jancoyan.jancoblog.pojo.VUserTotalData">

select \* from v\_user\_total\_data where article\_author = ${userId};

</select>

<!-- 用户信息的查询结果-->

<resultMap id="UserInfoMap" type="com.jancoyan.jancoblog.pojo.UserInfo">

<id column="user\_id" property="userId" />

<result column="user\_email" property="userEmail" />

<result column="user\_sex" property="userSex" />

<result column="user\_region" property="userRegion"/>

<result column="user\_birthday" property="userBirthdate" />

<result column="user\_telephone" property="userTelephone" />

<result column="user\_real\_name" property="userRealName" />

<result column="user\_school" property="userSchool" />

<result column="user\_major" property="userMajor"/>

<result column="user\_enter\_school\_date" property="userEnterSchoolDate"/>

<result column="user\_academic\_degree" property="userAcademicDegree"/>

<result column="user\_company" property="userCompany"/>

<result column="user\_position" property="userPosition"/>

<result column="user\_field" property="userField"/>

</resultMap>

<!-- 用户详细信息的结果列-->

<sql id="User\_Info\_List">

user\_id, user\_email, user\_sex, user\_region, user\_birthdate,

user\_telephone, user\_real\_name, user\_school, user\_major,

user\_enter\_school\_date, user\_academic\_degree, user\_company, user\_position, user\_field

</sql>

<select id="getUserInfo" resultType="com.jancoyan.jancoblog.pojo.UserInfo">

select \* from tbl\_user\_info where user\_id = ${userId}

</select>

<select id="login" resultType="com.jancoyan.jancoblog.pojo.User">

select user\_id, user\_name, user\_signature, role\_name

from v\_login

where user\_name = '${username}' and user\_password = '${password}';

</select>

</mapper>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\api\article.js

import request from '../utils/request';

/\*\*

\* 获取首页文章列表

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getIndexArticleList(pn, limit, condition) {

return request({

url: '/article/all',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 通过首页查看文章的时候获取文章信息

\* @param {String} id

\* @returns

\*/

export function getSingleArticle(id) {

return request({

url: '/article/single',

method: 'get',

params: {

id: id

}

})

}

/\*\*

\* 通过首页查看文章的时候获取文章信息

\* @param {String} id

\* @returns

\*/

export function getArticleEdit(id) {

return request({

url: '/article/edit',

method: 'get',

params: {

id: id

}

})

}

/\*\*

\* 通过首页查看文章的时候获取文章信息

\* @param {String} id

\* @returns

\*/

export function getSingleArticleDeleted(id) {

return request({

url: '/article/single/deleted',

method: 'get',

params: {

id: id

}

})

}

/\*\*

\* 点赞

\* @param {Stirng} id

\* @returns

\*/

export function likeArticle(id) {

return request({

url: '/article/like',

method: 'post',

params: {

id: id

}

})

}

/\*\*

\* 取消点赞

\* @param {Stirng} id

\* @returns

\*/

export function dislikeArticle(id) {

return request({

url: '/article/dislike',

method: 'post',

params: {

id: id

}

})

}

/\*\*

\* 浏览文章

\* @param {String} id

\* @returns

\*/

export function viewArticle(id) {

return request({

url: '/article/view?id='+id,

method: 'get'

})

}

/\*\*

\* 管理员视角获取所有文章

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getAll(pn, limit, condition) {

return request({

url: '/article/manage',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 管理员视角获取所有已经删除的文章

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getAllDeleted(pn, limit, condition) {

return request({

url: '/article/deleted/all',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 用户视角获取所有已经删除的文章

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getArticleByUserDeleted(pn, limit, condition) {

return request({

url: '/article/deleted/user',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 用户视角获取所有的文章

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getArticleByUser(pn, limit, condition) {

return request({

url: '/article/user',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 批量彻底删除文章

\* @param {String} ids 文章id的&拼接

\* @returns

\*/

export function batchDeleteDeletedArticles(ids) {

return request({

url: '/article/deleted/delete',

method: 'post',

params: {

ids: ids

}

})

}

/\*\*

\* 批量恢复文章

\* @param {String} ids 文章id的&拼接

\* @returns

\*/

export function batchRecoverArticles(ids) {

return request({

url: '/article/deleted/recover',

method: 'post',

params: {

ids: ids

}

})

}

/\*\*

\* 批量删除文章

\* @param {String} ids 文章id的&拼接

\* @returns

\*/

export function batchDeleteArticles(ids) {

return request({

url: '/article/delete',

method: 'post',

params: {

ids: ids

}

})

}

/\*\*

\* 发表文章

\* @param {String} title 标题

\* @param {Integee} type 类型

\* @param {String} summary 总结

\* @param {Integer} comment 是否允许评论

\* @param {String} md md格式的文章

\* @param {String} html HTML格式的文章

\* @returns

\*/

export function postArticle(title, type, summary, comment, md, html){

return request({

url: '/article/post',

method: 'post',

params: {

title: title,

type: type,

summary: summary,

comment: comment,

md: md,

html: html

}

})

}

/\*\*

\* 修改文章

\* @param {String} title 标题

\* @param {Integee} type 类型

\* @param {String} summary 总结

\* @param {Integer} comment 是否允许评论

\* @param {String} md md格式的文章

\* @param {String} html HTML格式的文章

\* @returns

\*/

export function updateArticle(id, title, type, summary, comment, md, html){

return request({

url: '/article/update',

method: 'post',

params: {

id: id,

title: title,

type: type,

summary: summary,

comment: comment,

md: md,

html: html

}

})

}

/\*\*

\* 改变文章允许评论的状态

\* @param {String} id 文章id

\* @returns

\*/

export function toggleArticleIsComment(id) {

return request({

url: '/article/toggle/comment',

method: 'post',

params: {

id: id

}

})

}

/\*\*

\* 文章置顶

\* @param {String} id 文章id

\* @returns

\*/

export function toggleArticleStickTop(id) {

return request({

url: '/article/toggle/top',

method: 'post',

params: {

id: id

}

})

}

/\*\*

\* 文章中的图片的上传

\* @param {blob} blob

\* @returns

\*/

export function uploadPicture(file) {

return request({

url: '/article/picture',

method: 'post',

headers: {

'Content-Type': 'multipart/form-data'

},

params: {

file: file

},

})

}

/\*\*

\* 获取用户最近发表的文章

\* @param {String} id 用户id

\* @returns

\*/

export function getUserArticleRecently(id, pn, limit) {

return request({

url: '/article/recent',

method: 'get',

params: {

id: id,

pn: pn,

limit: limit

}

})

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\api\comment.js

import request from '../utils/request';

/\*\*

\* 获取收到的评论

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getAll(pn, limit, condition) {

return request({

url: '/comment/all',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 发表评论

\* @param {Map} param 参数列表

\* @returns

\*/

export function postComment(param) {

return request({

url: '/comment/post',

method: 'post',

params: param

})

}

/\*\*

\* 赞同评论

\* @param {Integer} id 评论id

\* @returns

\*/

export function likeComment(id) {

return request({

url: '/comment/like',

method: 'post',

params: {

id: id

}

})

}

/\*\*

\* 获取文章下面的所有评论

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getCommentByArticle(id, pn, limit) {

return request({

url: '/comment/article',

method: 'get',

params: {

id: id,

pn : pn,

limit: limit

}

})

}

/\*\*

\* 获取已经删除的文章下面的评论

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getCommentByDeletedArticle(id, pn, limit) {

return request({

url: '/deleted/comment/article',

method: 'get',

params: {

id: id,

pn : pn,

limit: limit

}

})

}

/\*\*

\* 获取用户收到的所有评论

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getCommentByUserReceive(pn, limit, condition) {

return request({

url: '/comment/receive',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 批量删除

\* @param {String} ids id拼接

\* @returns

\*/

export function batchDeleteComments(ids) {

return request({

url: '/comment/delete',

method: 'post',

params: {

ids: ids

}

})

}

/\*\*

\* 获取用户发表的所有评论

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getCommentByUserPosted(pn, limit, condition) {

return request({

url: '/comment/posted',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 获取用户最近收到的评论数量

\* @param {String} id 用户id

\* @returns

\*/

export function getUserCommentRecently(id, pn, limit) {

return request({

url: '/comment/recent',

method: 'get',

params: {

id: id,

pn: pn,

limit: limit

}

})

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\api\like.js

import request from '../utils/request';

/\*\*

\* 在用户已经登录的情况下才会走这个，判断用户有没有对这个文章点赞

\* @param {String} id 文章id

\* @returns

\*/

export function isLiked(id) {

return request({

url: '/like/like',

method: 'get',

params:{

id: id

}

})

}

/\*\*

\* 获取用户最近收到的点赞数量

\* @param {String} id 用户id

\* @returns

\*/

export function getUserLikeRecently(id, pn, limit) {

return request({

url: '/like/recent',

method: 'get',

params: {

id: id,

pn: pn,

limit: limit

}

})

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\api\login.js

import request from '../utils/request';

/\*\*

\* 获取当前登录的用户的登录记录

\* @returns

\*/

export function getLoginLog() {

return request({

url: '/login/all',

method: 'get',

})

}

/\*\*

\* 获取指定用户的登陆记录

\* @returns

\*/

export function getLoginLogByUser(id) {

return request({

url: '/login/all',

method: 'get',

params: {

id: id

}

})

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\api\type.js

import request from '../utils/request';

export function getAllType(pn) {

return request({

url: '/type/all',

method: 'get',

})

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\api\user.js

import request from '@/utils/request'

/\*\*

\* 登录

\* @param {Object} data 表单

\* @returns

\*/

export function login(data) {

return request({

url: '/user/login',

method: 'post',

params: data

})

}

/\*\*

\* 检查用户名的唯一性

\* @param {String} userName 用户名

\* @returns

\*/

export function checkUserNameUnique(userName) {

return request({

url: '/user/checkusername',

method: 'post',

params:{

username: userName

}

})

}

/\*\*

\* 注册

\* @param {\*} registerForm 表单

\* @returns

\*/

export function register(registerForm) {

return request({

url: '/user/register',

method: 'post',

params:{

username: registerForm.username,

password: registerForm.password,

code: registerForm.code

}

})

}

/\*\*

\* 添加用户

\* @param {\*} registerForm 表单

\* @returns

\*/

export function addUser(registerForm) {

return request({

url: '/user/add',

method: 'post',

params:{

username: registerForm.username,

password: registerForm.password

}

})

}

/\*\*

\* 获取作者信息

\* @param {String} id 作者id

\* @returns

\*/

export function getAuthorInfo(id) {

return request({

url: '/user/authorinfo',

method: 'get',

params:{

id: id

}

})

}

/\*\*

\* 获取当前已经登录的用户信息

\* @returns

\*/

export function getUserInfo() {

return request({

url: '/user/userinfo',

method: 'get'

})

}

/\*\*

\* 获取当前已经登录的用户详细信息

\* @returns

\*/

export function getUserDetailInfo() {

return request({

url: '/user/userdetail',

method: 'get'

})

}

/\*\*

\* 获取用户的总的数据统计

\* @param {String} userId 用户id

\* @returns

\*/

export function getUserTotalData(userId) {

return request({

url: '/user/data/total',

method: 'get',

params: {

id: userId

}

})

}

/\*\*

\* 退出登录

\* @returns

\*/

export function logout() {

return request({

url: '/user/logout',

method: 'post'

})

}

/\*\*

\* 管理员获取所有用户信息

\* @param {Integer} pn 页码

\* @param {Integer} limit 容量

\* @param {String} condition 条件

\* @returns

\*/

export function getAll(pn, limit, condition) {

return request({

url: '/user/all',

method: 'get',

params: {

pn: pn,

limit: limit,

condition: condition

}

})

}

/\*\*

\* 批量删除用户

\* @param {String} ids 用户id

\* @returns

\*/

export function batchDeleteUsers(ids) {

return request({

url: '/user/delete',

method: 'post',

params: {

ids: ids

}

})

}

/\*\*

\* 修改用户密码

\* @param {Stirng} id 要修改密码的用户id

\* @param {String} oldpwd 旧密码

\* @param {String} newpwd 新密码

\* @returns

\*/

export function changePassword(id = '-1', oldpwd, newpwd){

return request({

url: '/user/changepwd',

method: 'post',

params: {

id: id,

old: oldpwd,

new: newpwd

}

})

}

/\*\*

\* 修改用户个人信息

\* @param {String} user 用户

\* @returns

\*/

export function updateUserInfo(user){

return request({

url: '/user/info/update',

method: 'post',

params: {

user: user

}

})

}

/\*\*

\* 修改用户信息

\* @param {String} username 用户名

\* @param {String} signature 用户签名

\* @returns

\*/

export function updateUser(username, signature){

return request({

url: '/user/info/user',

method: 'post',

params: {

username: username,

signature: signature

}

})

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\App.vue

<template>

<div id="app">

<router-view />

</div>

</template>

<script>

export default {

name: 'App'

}

</script>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\assets\css\highlight.css

pre code.hljs{display:block;overflow-x:auto;padding:1em}code.hljs{padding:3px 5px}/\*!

Theme: a11y-light

Author: @ericwbailey

Maintainer: @ericwbailey

Based on the Tomorrow Night Eighties theme: https://github.com/isagalaev/highlight.js/blob/master/src/styles/tomorrow-night-eighties.css

\*/.hljs{background:#fefefe;color:#545454}.hljs-comment,.hljs-quote{color:#696969}.hljs-deletion,.hljs-name,.hljs-regexp,.hljs-selector-class,.hljs-selector-id,.hljs-tag,.hljs-template-variable,.hljs-variable{color:#d91e18}.hljs-attribute,.hljs-built\_in,.hljs-link,.hljs-literal,.hljs-meta,.hljs-number,.hljs-params,.hljs-type{color:#aa5d00}.hljs-addition,.hljs-bullet,.hljs-string,.hljs-symbol{color:green}.hljs-section,.hljs-title{color:#007faa}.hljs-keyword,.hljs-selector-tag{color:#7928a1}.hljs-emphasis{font-style:italic}.hljs-strong{font-weight:700}@media screen and (-ms-high-contrast:active){.hljs-addition,.hljs-attribute,.hljs-built\_in,.hljs-bullet,.hljs-comment,.hljs-link,.hljs-literal,.hljs-meta,.hljs-number,.hljs-params,.hljs-quote,.hljs-string,.hljs-symbol,.hljs-type{color:highlight}.hljs-keyword,.hljs-selector-tag{font-weight:700}}R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\assets\js\highhight.js

/\*!

Highlight.js v11.2.0 (git: 2a5c592e5e)

(c) 2006-2021 Ivan Sagalaev and other contributors

License: BSD-3-Clause

\*/

var hljs = function () {

"use strict"; var e = { exports: {} }; function n(e) {

return e instanceof Map ? e.clear = e.delete = e.set = () => {

throw Error("map is read-only")

} : e instanceof Set && (e.add = e.clear = e.delete = () => {

throw Error("set is read-only")

}), Object.freeze(e), Object.getOwnPropertyNames(e).forEach((t => {

var a = e[t]

; "object" != typeof a || Object.isFrozen(a) || n(a)

})), e

}

e.exports = n, e.exports.default = n; var t = e.exports; class a {

constructor(e) {

void 0 === e.data && (e.data = {}), this.data = e.data, this.isMatchIgnored = !1

}

ignoreMatch() { this.isMatchIgnored = !0 }

} function i(e) {

return e.replace(/&/g, "&amp;").replace(/</g, "&lt;").replace(/>/g, "&gt;").replace(/"/g, "&quot;").replace(/'/g, "&#x27;")

} function s(e, ...n) {

const t = Object.create(null); for (const n in e) t[n] = e[n]

; return n.forEach((e => { for (const n in e) t[n] = e[n] })), t

} const r = e => !!e.kind

; class o {

constructor(e, n) {

this.buffer = "", this.classPrefix = n.classPrefix, e.walk(this)

} addText(e) {

this.buffer += i(e)

} openNode(e) {

if (!r(e)) return; let n = e.kind

; n = e.sublanguage ? "language-" + n : ((e, { prefix: n }) => {

if (e.includes(".")) {

const t = e.split(".")

; return [`${n}${t.shift()}`, ...t.map(((e, n) => `${e}${"\_".repeat(n + 1)}`))].join(" ")

} return `${n}${e}`

})(n, { prefix: this.classPrefix }), this.span(n)

} closeNode(e) {

r(e) && (this.buffer += "</span>")

} value() { return this.buffer } span(e) {

this.buffer += `<span class="${e}">`

}

} class l {

constructor() {

this.rootNode = {

children: []

}, this.stack = [this.rootNode]

} get top() {

return this.stack[this.stack.length - 1]

} get root() { return this.rootNode } add(e) {

this.top.children.push(e)

} openNode(e) {

const n = { kind: e, children: [] }

; this.add(n), this.stack.push(n)

} closeNode() {

if (this.stack.length > 1) return this.stack.pop()

} closeAllNodes() {

for (; this.closeNode(););

} toJSON() { return JSON.stringify(this.rootNode, null, 4) }

walk(e) { return this.constructor.\_walk(e, this.rootNode) } static \_walk(e, n) {

return "string" == typeof n ? e.addText(n) : n.children && (e.openNode(n),

n.children.forEach((n => this.\_walk(e, n))), e.closeNode(n)), e

} static \_collapse(e) {

"string" != typeof e && e.children && (e.children.every((e => "string" == typeof e)) ? e.children = [e.children.join("")] : e.children.forEach((e => {

l.\_collapse(e)

})))

}

} class c extends l {

constructor(e) { super(), this.options = e }

addKeyword(e, n) { "" !== e && (this.openNode(n), this.addText(e), this.closeNode()) }

addText(e) { "" !== e && this.add(e) } addSublanguage(e, n) {

const t = e.root

; t.kind = n, t.sublanguage = !0, this.add(t)

} toHTML() {

return new o(this, this.options).value()

} finalize() { return !0 }

} function d(e) {

return e ? "string" == typeof e ? e : e.source : null

} function g(e) { return b("(?=", e, ")") }

function u(e) { return b("(?:", e, ")?") } function b(...e) {

return e.map((e => d(e))).join("")

} function m(...e) {

return "(" + ((e => {

const n = e[e.length - 1]

; return "object" == typeof n && n.constructor === Object ? (e.splice(e.length - 1, 1), n) : {}

})(e).capture ? "" : "?:") + e.map((e => d(e))).join("|") + ")"

} function p(e) {

return RegExp(e.toString() + "|").exec("").length - 1

}

const \_ = /\[(?:[^\\\]]|\\.)\*\]|\(\??|\\([1-9][0-9]\*)|\\./

; function h(e, { joinWith: n }) {

let t = 0; return e.map((e => {

t += 1; const n = t

; let a = d(e), i = ""; for (; a.length > 0;) {

const e = \_.exec(a); if (!e) { i += a; break }

i += a.substring(0, e.index),

a = a.substring(e.index + e[0].length), "\\" === e[0][0] && e[1] ? i += "\\" + (Number(e[1]) + n) : (i += e[0],

"(" === e[0] && t++)

} return i

})).map((e => `(${e})`)).join(n)

}

const f = "[a-zA-Z]\\w\*", E = "[a-zA-Z\_]\\w\*", y = "\\b\\d+(\\.\\d+)?", N = "(-?)(\\b0[xX][a-fA-F0-9]+|(\\b\\d+(\\.\\d\*)?|\\.\\d+)([eE][-+]?\\d+)?)", w = "\\b(0b[01]+)", v = {

begin: "\\\\[\\s\\S]", relevance: 0

}, O = {

scope: "string", begin: "'", end: "'",

illegal: "\\n", contains: [v]

}, M = {

scope: "string", begin: '"', end: '"', illegal: "\\n",

contains: [v]

}, x = (e, n, t = {}) => {

const a = s({

scope: "comment", begin: e, end: n,

contains: []

}, t); a.contains.push({

scope: "doctag",

begin: "[ ]\*(?=(TODO|FIXME|NOTE|BUG|OPTIMIZE|HACK|XXX):)",

end: /(TODO|FIXME|NOTE|BUG|OPTIMIZE|HACK|XXX):/, excludeBegin: !0, relevance: 0

})

; const i = m("I", "a", "is", "so", "us", "to", "at", "if", "in", "it", "on", /[A-Za-z]+['](d|ve|re|ll|t|s|n)/, /[A-Za-z]+[-][a-z]+/, /[A-Za-z][a-z]{2,}/)

; return a.contains.push({ begin: b(/[ ]+/, "(", i, /[.]?[:]?([.][ ]|[ ])/, "){3}") }), a

}, S = x("//", "$"), k = x("/\\\*", "\\\*/"), A = x("#", "$"); var C = Object.freeze({

\_\_proto\_\_: null, MATCH\_NOTHING\_RE: /\b\B/, IDENT\_RE: f, UNDERSCORE\_IDENT\_RE: E,

NUMBER\_RE: y, C\_NUMBER\_RE: N, BINARY\_NUMBER\_RE: w,

RE\_STARTERS\_RE: "!|!=|!==|%|%=|&|&&|&=|\\\*|\\\*=|\\+|\\+=|,|-|-=|/=|/|:|;|<<|<<=|<=|<|===|==|=|>>>=|>>=|>=|>>>|>>|>|\\?|\\[|\\{|\\(|\\^|\\^=|\\||\\|=|\\|\\||~",

SHEBANG: (e = {}) => {

const n = /^#![ ]\*\//

; return e.binary && (e.begin = b(n, /.\*\b/, e.binary, /\b.\*/)), s({

scope: "meta", begin: n,

end: /$/, relevance: 0, "on:begin": (e, n) => { 0 !== e.index && n.ignoreMatch() }

}, e)

},

BACKSLASH\_ESCAPE: v, APOS\_STRING\_MODE: O, QUOTE\_STRING\_MODE: M, PHRASAL\_WORDS\_MODE: {

begin: /\b(a|an|the|are|I'm|isn't|don't|doesn't|won't|but|just|should|pretty|simply|enough|gonna|going|wtf|so|such|will|you|your|they|like|more)\b/

}, COMMENT: x, C\_LINE\_COMMENT\_MODE: S, C\_BLOCK\_COMMENT\_MODE: k, HASH\_COMMENT\_MODE: A,

NUMBER\_MODE: { scope: "number", begin: y, relevance: 0 }, C\_NUMBER\_MODE: {

scope: "number",

begin: N, relevance: 0

}, BINARY\_NUMBER\_MODE: { scope: "number", begin: w, relevance: 0 },

REGEXP\_MODE: {

begin: /(?=\/[^/\n]\*\/)/, contains: [{

scope: "regexp", begin: /\//,

end: /\/[gimuy]\*/, illegal: /\n/, contains: [v, {

begin: /\[/, end: /\]/, relevance: 0,

contains: [v]

}]

}]

}, TITLE\_MODE: { scope: "title", begin: f, relevance: 0 },

UNDERSCORE\_TITLE\_MODE: { scope: "title", begin: E, relevance: 0 }, METHOD\_GUARD: {

begin: "\\.\\s\*[a-zA-Z\_]\\w\*", relevance: 0

}, END\_SAME\_AS\_BEGIN: e => Object.assign(e, {

"on:begin": (e, n) => { n.data.\_beginMatch = e[1] }, "on:end": (e, n) => {

n.data.\_beginMatch !== e[1] && n.ignoreMatch()

}

})

}); function T(e, n) {

"." === e.input[e.index - 1] && n.ignoreMatch()

} function R(e, n) {

void 0 !== e.className && (e.scope = e.className, delete e.className)

} function D(e, n) {

n && e.beginKeywords && (e.begin = "\\b(" + e.beginKeywords.split(" ").join("|") + ")(?!\\.)(?=\\b|\\s)",

e.\_\_beforeBegin = T, e.keywords = e.keywords || e.beginKeywords, delete e.beginKeywords,

void 0 === e.relevance && (e.relevance = 0))

} function I(e, n) {

Array.isArray(e.illegal) && (e.illegal = m(...e.illegal))

} function L(e, n) {

if (e.match) {

if (e.begin || e.end) throw Error("begin & end are not supported with match")

; e.begin = e.match, delete e.match

}

} function B(e, n) {

void 0 === e.relevance && (e.relevance = 1)

} const $ = (e, n) => {

if (!e.beforeMatch) return

; if (e.starts) throw Error("beforeMatch cannot be used with starts")

; const t = Object.assign({}, e); Object.keys(e).forEach((n => {

delete e[n]

})), e.keywords = t.keywords, e.begin = b(t.beforeMatch, g(t.begin)), e.starts = {

relevance: 0, contains: [Object.assign(t, { endsParent: !0 })]

}, e.relevance = 0, delete t.beforeMatch

}, z = ["of", "and", "for", "in", "not", "or", "if", "then", "parent", "list", "value"]

; function F(e, n, t = "keyword") {

const a = Object.create(null)

; return "string" == typeof e ? i(t, e.split(" ")) : Array.isArray(e) ? i(t, e) : Object.keys(e).forEach((t => {

Object.assign(a, F(e[t], n, t))

})), a; function i(e, t) {

n && (t = t.map((e => e.toLowerCase()))), t.forEach((n => {

const t = n.split("|")

; a[t[0]] = [e, j(t[0], t[1])]

}))

}

} function j(e, n) {

return n ? Number(n) : (e => z.includes(e.toLowerCase()))(e) ? 0 : 1

} const U = {}, P = e => {

console.error(e)

}, K = (e, ...n) => { console.log("WARN: " + e, ...n) }, q = (e, n) => {

U[`${e}/${n}`] || (console.log(`Deprecated as of ${e}. ${n}`), U[`${e}/${n}`] = !0)

}, H = Error(); function G(e, n, { key: t }) {

let a = 0; const i = e[t], s = {}, r = {}

; for (let e = 1; e <= n.length; e++)r[e + a] = i[e], s[e + a] = !0, a += p(n[e - 1])

; e[t] = r, e[t].\_emit = s, e[t].\_multi = !0

} function Z(e) {

(e => {

e.scope && "object" == typeof e.scope && null !== e.scope && (e.beginScope = e.scope,

delete e.scope)

})(e), "string" == typeof e.beginScope && (e.beginScope = {

\_wrap: e.beginScope

}), "string" == typeof e.endScope && (e.endScope = {

\_wrap: e.endScope

}), (e => {

if (Array.isArray(e.begin)) {

if (e.skip || e.excludeBegin || e.returnBegin) throw P("skip, excludeBegin, returnBegin not compatible with beginScope: {}"),

H

; if ("object" != typeof e.beginScope || null === e.beginScope) throw P("beginScope must be object"),

H; G(e, e.begin, { key: "beginScope" }), e.begin = h(e.begin, { joinWith: "" })

}

})(e), (e => {

if (Array.isArray(e.end)) {

if (e.skip || e.excludeEnd || e.returnEnd) throw P("skip, excludeEnd, returnEnd not compatible with endScope: {}"),

H

; if ("object" != typeof e.endScope || null === e.endScope) throw P("endScope must be object"),

H; G(e, e.end, { key: "endScope" }), e.end = h(e.end, { joinWith: "" })

}

})(e)

} function W(e) {

function n(n, t) { return RegExp(d(n), "m" + (e.case\_insensitive ? "i" : "") + (t ? "g" : "")) }

class t {

constructor() {

this.matchIndexes = {}, this.regexes = [], this.matchAt = 1, this.position = 0

}

addRule(e, n) {

n.position = this.position++, this.matchIndexes[this.matchAt] = n, this.regexes.push([n, e]),

this.matchAt += p(e) + 1

} compile() {

0 === this.regexes.length && (this.exec = () => null)

; const e = this.regexes.map((e => e[1])); this.matcherRe = n(h(e, {

joinWith: "|"

}), !0), this.lastIndex = 0

} exec(e) {

this.matcherRe.lastIndex = this.lastIndex

; const n = this.matcherRe.exec(e); if (!n) return null

; const t = n.findIndex(((e, n) => n > 0 && void 0 !== e)), a = this.matchIndexes[t]

; return n.splice(0, t), Object.assign(n, a)

}

} class a {

constructor() {

this.rules = [], this.multiRegexes = [],

this.count = 0, this.lastIndex = 0, this.regexIndex = 0

} getMatcher(e) {

if (this.multiRegexes[e]) return this.multiRegexes[e]; const n = new t

; return this.rules.slice(e).forEach((([e, t]) => n.addRule(e, t))),

n.compile(), this.multiRegexes[e] = n, n

} resumingScanAtSamePosition() {

return 0 !== this.regexIndex

} considerAll() { this.regexIndex = 0 } addRule(e, n) {

this.rules.push([e, n]), "begin" === n.type && this.count++

} exec(e) {

const n = this.getMatcher(this.regexIndex); n.lastIndex = this.lastIndex

; let t = n.exec(e)

; if (this.resumingScanAtSamePosition()) if (t && t.index === this.lastIndex); else {

const n = this.getMatcher(0); n.lastIndex = this.lastIndex + 1, t = n.exec(e)

}

return t && (this.regexIndex += t.position + 1,

this.regexIndex === this.count && this.considerAll()), t

}

}

if (e.compilerExtensions || (e.compilerExtensions = []),

e.contains && e.contains.includes("self")) throw Error("ERR: contains `self` is not supported at the top-level of a language. See documentation.")

; return e.classNameAliases = s(e.classNameAliases || {}), function t(i, r) {

const o = i

; if (i.isCompiled) return o

;[R, L, Z, $].forEach((e => e(i, r))), e.compilerExtensions.forEach((e => e(i, r))),

i.\_\_beforeBegin = null, [D, I, B].forEach((e => e(i, r))), i.isCompiled = !0; let l = null

; return "object" == typeof i.keywords && i.keywords.$pattern && (i.keywords = Object.assign({}, i.keywords),

l = i.keywords.$pattern,

delete i.keywords.$pattern), l = l || /\w+/, i.keywords && (i.keywords = F(i.keywords, e.case\_insensitive)),

o.keywordPatternRe = n(l, !0),

r && (i.begin || (i.begin = /\B|\b/), o.beginRe = n(i.begin), i.end || i.endsWithParent || (i.end = /\B|\b/),

i.end && (o.endRe = n(i.end)),

o.terminatorEnd = d(i.end) || "", i.endsWithParent && r.terminatorEnd && (o.terminatorEnd += (i.end ? "|" : "") + r.terminatorEnd)),

i.illegal && (o.illegalRe = n(i.illegal)),

i.contains || (i.contains = []), i.contains = [].concat(...i.contains.map((e => (e => (e.variants && !e.cachedVariants && (e.cachedVariants = e.variants.map((n => s(e, {

variants: null

}, n)))), e.cachedVariants ? e.cachedVariants : Q(e) ? s(e, {

starts: e.starts ? s(e.starts) : null

}) : Object.isFrozen(e) ? s(e) : e))("self" === e ? i : e)))), i.contains.forEach((e => {

t(e, o)

})), i.starts && t(i.starts, r), o.matcher = (e => {

const n = new a

; return e.contains.forEach((e => n.addRule(e.begin, {

rule: e, type: "begin"

}))), e.terminatorEnd && n.addRule(e.terminatorEnd, {

type: "end"

}), e.illegal && n.addRule(e.illegal, { type: "illegal" }), n

})(o), o

}(e)

} function Q(e) {

return !!e && (e.endsWithParent || Q(e.starts))

} const X = i, V = s, J = Symbol("nomatch")

; var Y = (e => {

const n = Object.create(null), i = Object.create(null), s = []; let r = !0

; const o = "Could not find the language '{}', did you forget to load/include a language module?", l = {

disableAutodetect: !0, name: "Plain text", contains: []

}; let d = {

ignoreUnescapedHTML: !1, noHighlightRe: /^(no-?highlight)$/i,

languageDetectRe: /\blang(?:uage)?-([\w-]+)\b/i, classPrefix: "hljs-",

cssSelector: "pre code", languages: null, \_\_emitter: c

}; function g(e) {

return d.noHighlightRe.test(e)

} function u(e, n, t) {

let a = "", i = ""

; "object" == typeof n ? (a = e,

t = n.ignoreIllegals, i = n.language) : (q("10.7.0", "highlight(lang, code, ...args) has been deprecated."),

q("10.7.0", "Please use highlight(code, options) instead.\nhttps://github.com/highlightjs/highlight.js/issues/2277"),

i = e, a = n), void 0 === t && (t = !0); const s = { code: a, language: i }; N("before:highlight", s)

; const r = s.result ? s.result : b(s.language, s.code, t)

; return r.code = s.code, N("after:highlight", r), r

} function b(e, t, i, s) {

const l = Object.create(null); function c() {

if (!M.keywords) return void S.addText(k)

; let e = 0; M.keywordPatternRe.lastIndex = 0; let n = M.keywordPatternRe.exec(k), t = ""

; for (; n;) {

t += k.substring(e, n.index)

; const i = w.case\_insensitive ? n[0].toLowerCase() : n[0], s = (a = i, M.keywords[a]); if (s) {

const [e, a] = s

; if (S.addText(t), t = "", l[i] = (l[i] || 0) + 1, l[i] <= 7 && (A += a), e.startsWith("\_")) t += n[0]; else {

const t = w.classNameAliases[e] || e; S.addKeyword(n[0], t)

}

} else t += n[0]

; e = M.keywordPatternRe.lastIndex, n = M.keywordPatternRe.exec(k)

} var a

; t += k.substr(e), S.addText(t)

} function g() {

null != M.subLanguage ? (() => {

if ("" === k) return; let e = null; if ("string" == typeof M.subLanguage) {

if (!n[M.subLanguage]) return void S.addText(k)

; e = b(M.subLanguage, k, !0, x[M.subLanguage]), x[M.subLanguage] = e.\_top

} else e = m(k, M.subLanguage.length ? M.subLanguage : null)

; M.relevance > 0 && (A += e.relevance), S.addSublanguage(e.\_emitter, e.language)

})() : c(), k = ""

} function u(e, n) {

let t = 1; for (; void 0 !== n[t];) {

if (!e.\_emit[t]) {

t++

; continue

} const a = w.classNameAliases[e[t]] || e[t], i = n[t]

; a ? S.addKeyword(i, a) : (k = i, c(), k = ""), t++

}

} function p(e, n) {

return e.scope && "string" == typeof e.scope && S.openNode(w.classNameAliases[e.scope] || e.scope),

e.beginScope && (e.beginScope.\_wrap ? (S.addKeyword(k, w.classNameAliases[e.beginScope.\_wrap] || e.beginScope.\_wrap),

k = "") : e.beginScope.\_multi && (u(e.beginScope, n), k = "")), M = Object.create(e, {

parent: {

value: M

}

}), M

} function \_(e, n, t) {

let i = ((e, n) => {

const t = e && e.exec(n)

; return t && 0 === t.index

})(e.endRe, t); if (i) {

if (e["on:end"]) {

const t = new a(e)

; e["on:end"](n, t), t.isMatchIgnored && (i = !1)

} if (i) {

for (; e.endsParent && e.parent;)e = e.parent; return e

}

}

if (e.endsWithParent) return \_(e.parent, n, t)

} function h(e) {

return 0 === M.matcher.regexIndex ? (k += e[0], 1) : (R = !0, 0)

} function E(e) {

const n = e[0], a = t.substr(e.index), i = \_(M, e, a); if (!i) return J; const s = M

; M.endScope && M.endScope.\_wrap ? (g(),

S.addKeyword(n, M.endScope.\_wrap)) : M.endScope && M.endScope.\_multi ? (g(),

u(M.endScope, e)) : s.skip ? k += n : (s.returnEnd || s.excludeEnd || (k += n),

g(), s.excludeEnd && (k = n)); do {

M.scope && S.closeNode(), M.skip || M.subLanguage || (A += M.relevance), M = M.parent

} while (M !== i.parent); return i.starts && p(i.starts, e), s.returnEnd ? 0 : n.length

}

let y = {}; function N(n, s) {

const o = s && s[0]; if (k += n, null == o) return g(), 0

; if ("begin" === y.type && "end" === s.type && y.index === s.index && "" === o) {

if (k += t.slice(s.index, s.index + 1), !r) {

const n = Error(`0 width match regex (${e})`)

; throw n.languageName = e, n.badRule = y.rule, n

} return 1

}

if (y = s, "begin" === s.type) return (e => {

const n = e[0], t = e.rule, i = new a(t), s = [t.\_\_beforeBegin, t["on:begin"]]

; for (const t of s) if (t && (t(e, i), i.isMatchIgnored)) return h(n)

; return t.skip ? k += n : (t.excludeBegin && (k += n),

g(), t.returnBegin || t.excludeBegin || (k = n)), p(t, e), t.returnBegin ? 0 : n.length

})(s)

; if ("illegal" === s.type && !i) {

const e = Error('Illegal lexeme "' + o + '" for mode "' + (M.scope || "<unnamed>") + '"')

; throw e.mode = M, e

} if ("end" === s.type) { const e = E(s); if (e !== J) return e }

if ("illegal" === s.type && "" === o) return 1

; if (T > 1e5 && T > 3 \* s.index) throw Error("potential infinite loop, way more iterations than matches")

; return k += o, o.length

} const w = f(e)

; if (!w) throw P(o.replace("{}", e)), Error('Unknown language: "' + e + '"')

; const v = W(w); let O = "", M = s || v; const x = {}, S = new d.\_\_emitter(d); (() => {

const e = []

; for (let n = M; n !== w; n = n.parent)n.scope && e.unshift(n.scope)

; e.forEach((e => S.openNode(e)))

})(); let k = "", A = 0, C = 0, T = 0, R = !1; try {

for (M.matcher.considerAll(); ;) {

T++, R ? R = !1 : M.matcher.considerAll(), M.matcher.lastIndex = C

; const e = M.matcher.exec(t); if (!e) break; const n = N(t.substring(C, e.index), e)

; C = e.index + n

} return N(t.substr(C)), S.closeAllNodes(), S.finalize(), O = S.toHTML(), {

language: e, value: O, relevance: A, illegal: !1, \_emitter: S, \_top: M

}

} catch (n) {

if (n.message && n.message.includes("Illegal")) return {

language: e, value: X(t),

illegal: !0, relevance: 0, \_illegalBy: {

message: n.message, index: C,

context: t.slice(C - 100, C + 100), mode: n.mode, resultSoFar: O

}, \_emitter: S

}; if (r) return {

language: e, value: X(t), illegal: !1, relevance: 0, errorRaised: n, \_emitter: S, \_top: M

}

; throw n

}

} function m(e, t) {

t = t || d.languages || Object.keys(n); const a = (e => {

const n = { value: X(e), illegal: !1, relevance: 0, \_top: l, \_emitter: new d.\_\_emitter(d) }

; return n.\_emitter.addText(e), n

})(e), i = t.filter(f).filter(y).map((n => b(n, e, !1)))

; i.unshift(a); const s = i.sort(((e, n) => {

if (e.relevance !== n.relevance) return n.relevance - e.relevance

; if (e.language && n.language) {

if (f(e.language).supersetOf === n.language) return 1

; if (f(n.language).supersetOf === e.language) return -1

} return 0

})), [r, o] = s, c = r

; return c.secondBest = o, c

} function p(e) {

let n = null; const t = (e => {

let n = e.className + " "; n += e.parentNode ? e.parentNode.className : ""

; const t = d.languageDetectRe.exec(n); if (t) {

const n = f(t[1])

; return n || (K(o.replace("{}", t[1])),

K("Falling back to no-highlight mode for this block.", e)), n ? t[1] : "no-highlight"

}

return n.split(/\s+/).find((e => g(e) || f(e)))

})(e); if (g(t)) return

; N("before:highlightElement", {

el: e, language: t

}), !d.ignoreUnescapedHTML && e.children.length > 0 && (console.warn("One of your code blocks includes unescaped HTML. This is a potentially serious security risk."),

console.warn("https://github.com/highlightjs/highlight.js/issues/2886"),

console.warn(e)), n = e; const a = n.textContent, s = t ? u(a, {

language: t, ignoreIllegals: !0

}) : m(a); e.innerHTML = s.value, ((e, n, t) => {

const a = n && i[n] || t

; e.classList.add("hljs"), e.classList.add("language-" + a)

})(e, t, s.language), e.result = {

language: s.language, re: s.relevance,

relevance: s.relevance

}, s.secondBest && (e.secondBest = {

language: s.secondBest.language, relevance: s.secondBest.relevance

}), N("after:highlightElement", { el: e, result: s, text: a })

} let \_ = !1; function h() {

"loading" !== document.readyState ? document.querySelectorAll(d.cssSelector).forEach(p) : \_ = !0

} function f(e) { return e = (e || "").toLowerCase(), n[e] || n[i[e]] }

function E(e, { languageName: n }) {

"string" == typeof e && (e = [e]), e.forEach((e => {

i[e.toLowerCase()] = n

}))

} function y(e) {

const n = f(e)

; return n && !n.disableAutodetect

} function N(e, n) {

const t = e; s.forEach((e => {

e[t] && e[t](n)

}))

}

"undefined" != typeof window && window.addEventListener && window.addEventListener("DOMContentLoaded", (() => {

\_ && h()

}), !1), Object.assign(e, {

highlight: u, highlightAuto: m, highlightAll: h,

highlightElement: p,

highlightBlock: e => (q("10.7.0", "highlightBlock will be removed entirely in v12.0"),

q("10.7.0", "Please use highlightElement now."), p(e)), configure: e => { d = V(d, e) },

initHighlighting: () => {

h(), q("10.6.0", "initHighlighting() deprecated. Use highlightAll() now.")

},

initHighlightingOnLoad: () => {

h(), q("10.6.0", "initHighlightingOnLoad() deprecated. Use highlightAll() now.")

}, registerLanguage: (t, a) => {

let i = null; try { i = a(e) } catch (e) {

if (P("Language definition for '{}' could not be registered.".replace("{}", t)),

!r) throw e; P(e), i = l

}

i.name || (i.name = t), n[t] = i, i.rawDefinition = a.bind(null, e), i.aliases && E(i.aliases, {

languageName: t

})

}, unregisterLanguage: e => {

delete n[e]

; for (const n of Object.keys(i)) i[n] === e && delete i[n]

},

listLanguages: () => Object.keys(n), getLanguage: f, registerAliases: E,

autoDetection: y, inherit: V, addPlugin: e => {

(e => {

e["before:highlightBlock"] && !e["before:highlightElement"] && (e["before:highlightElement"] = n => {

e["before:highlightBlock"](Object.assign({ block: n.el }, n))

}), e["after:highlightBlock"] && !e["after:highlightElement"] && (e["after:highlightElement"] = n => {

e["after:highlightBlock"](Object.assign({ block: n.el }, n))

})

})(e), s.push(e)

}

}), e.debugMode = () => { r = !1 }, e.safeMode = () => { r = !0 }, e.versionString = "11.2.0"

; for (const e in C) "object" == typeof C[e] && t(C[e]); return Object.assign(e, C), e

})({}); const ee = e => ({

IMPORTANT: { scope: "meta", begin: "!important" }, HEXCOLOR: {

scope: "number", begin: "#([a-fA-F0-9]{6}|[a-fA-F0-9]{3})"

},

ATTRIBUTE\_SELECTOR\_MODE: {

scope: "selector-attr", begin: /\[/, end: /\]/, illegal: "$",

contains: [e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE]

}, CSS\_NUMBER\_MODE: {

scope: "number",

begin: e.NUMBER\_RE + "(%|em|ex|ch|rem|vw|vh|vmin|vmax|cm|mm|in|pt|pc|px|deg|grad|rad|turn|s|ms|Hz|kHz|dpi|dpcm|dppx)?",

relevance: 0

}, CSS\_VARIABLE: { className: "attr", begin: /--[A-Za-z][A-Za-z0-9\_-]\*/ }

}), ne = ["a", "abbr", "address", "article", "aside", "audio", "b", "blockquote", "body", "button", "canvas", "caption", "cite", "code", "dd", "del", "details", "dfn", "div", "dl", "dt", "em", "fieldset", "figcaption", "figure", "footer", "form", "h1", "h2", "h3", "h4", "h5", "h6", "header", "hgroup", "html", "i", "iframe", "img", "input", "ins", "kbd", "label", "legend", "li", "main", "mark", "menu", "nav", "object", "ol", "p", "q", "quote", "samp", "section", "span", "strong", "summary", "sup", "table", "tbody", "td", "textarea", "tfoot", "th", "thead", "time", "tr", "ul", "var", "video"], te = ["any-hover", "any-pointer", "aspect-ratio", "color", "color-gamut", "color-index", "device-aspect-ratio", "device-height", "device-width", "display-mode", "forced-colors", "grid", "height", "hover", "inverted-colors", "monochrome", "orientation", "overflow-block", "overflow-inline", "pointer", "prefers-color-scheme", "prefers-contrast", "prefers-reduced-motion", "prefers-reduced-transparency", "resolution", "scan", "scripting", "update", "width", "min-width", "max-width", "min-height", "max-height"], ae = ["active", "any-link", "blank", "checked", "current", "default", "defined", "dir", "disabled", "drop", "empty", "enabled", "first", "first-child", "first-of-type", "fullscreen", "future", "focus", "focus-visible", "focus-within", "has", "host", "host-context", "hover", "indeterminate", "in-range", "invalid", "is", "lang", "last-child", "last-of-type", "left", "link", "local-link", "not", "nth-child", "nth-col", "nth-last-child", "nth-last-col", "nth-last-of-type", "nth-of-type", "only-child", "only-of-type", "optional", "out-of-range", "past", "placeholder-shown", "read-only", "read-write", "required", "right", "root", "scope", "target", "target-within", "user-invalid", "valid", "visited", "where"], ie = ["after", "backdrop", "before", "cue", "cue-region", "first-letter", "first-line", "grammar-error", "marker", "part", "placeholder", "selection", "slotted", "spelling-error"], se = ["align-content", "align-items", "align-self", "animation", "animation-delay", "animation-direction", "animation-duration", "animation-fill-mode", "animation-iteration-count", "animation-name", "animation-play-state", "animation-timing-function", "auto", "backface-visibility", "background", "background-attachment", "background-clip", "background-color", "background-image", "background-origin", "background-position", "background-repeat", "background-size", "border", "border-bottom", "border-bottom-color", "border-bottom-left-radius", "border-bottom-right-radius", "border-bottom-style", "border-bottom-width", "border-collapse", "border-color", "border-image", "border-image-outset", "border-image-repeat", "border-image-slice", "border-image-source", "border-image-width", "border-left", "border-left-color", "border-left-style", "border-left-width", "border-radius", "border-right", "border-right-color", "border-right-style", "border-right-width", "border-spacing", "border-style", "border-top", "border-top-color", "border-top-left-radius", "border-top-right-radius", "border-top-style", "border-top-width", "border-width", "bottom", "box-decoration-break", "box-shadow", "box-sizing", "break-after", "break-before", "break-inside", "caption-side", "clear", "clip", "clip-path", "color", "column-count", "column-fill", "column-gap", "column-rule", "column-rule-color", "column-rule-style", "column-rule-width", "column-span", "column-width", "columns", "content", "counter-increment", "counter-reset", "cursor", "direction", "display", "empty-cells", "filter", "flex", "flex-basis", "flex-direction", "flex-flow", "flex-grow", "flex-shrink", "flex-wrap", "float", "font", "font-display", "font-family", "font-feature-settings", "font-kerning", "font-language-override", "font-size", "font-size-adjust", "font-smoothing", "font-stretch", "font-style", "font-variant", "font-variant-ligatures", "font-variation-settings", "font-weight", "height", "hyphens", "icon", "image-orientation", "image-rendering", "image-resolution", "ime-mode", "inherit", "initial", "justify-content", "left", "letter-spacing", "line-height", "list-style", "list-style-image", "list-style-position", "list-style-type", "margin", "margin-bottom", "margin-left", "margin-right", "margin-top", "marks", "mask", "max-height", "max-width", "min-height", "min-width", "nav-down", "nav-index", "nav-left", "nav-right", "nav-up", "none", "normal", "object-fit", "object-position", "opacity", "order", "orphans", "outline", "outline-color", "outline-offset", "outline-style", "outline-width", "overflow", "overflow-wrap", "overflow-x", "overflow-y", "padding", "padding-bottom", "padding-left", "padding-right", "padding-top", "page-break-after", "page-break-before", "page-break-inside", "perspective", "perspective-origin", "pointer-events", "position", "quotes", "resize", "right", "src", "tab-size", "table-layout", "text-align", "text-align-last", "text-decoration", "text-decoration-color", "text-decoration-line", "text-decoration-style", "text-indent", "text-overflow", "text-rendering", "text-shadow", "text-transform", "text-underline-position", "top", "transform", "transform-origin", "transform-style", "transition", "transition-delay", "transition-duration", "transition-property", "transition-timing-function", "unicode-bidi", "vertical-align", "visibility", "white-space", "widows", "width", "word-break", "word-spacing", "word-wrap", "z-index"].reverse(), re = ae.concat(ie)

; var oe = "\\.([0-9](\_\*[0-9])\*)", le = "[0-9a-fA-F](\_\*[0-9a-fA-F])\*", ce = {

className: "number", variants: [{

begin: `(\\b([0-9](\_\*[0-9])\*)((${oe})|\\.)?|(${oe}))[eE][+-]?([0-9](\_\*[0-9])\*)[fFdD]?\\b`

}, { begin: `\\b([0-9](\_\*[0-9])\*)((${oe})[fFdD]?\\b|\\.([fFdD]\\b)?)` }, {

begin: `(${oe})[fFdD]?\\b`

}, { begin: "\\b([0-9](\_\*[0-9])\*)[fFdD]\\b" }, {

begin: `\\b0[xX]((${le})\\.?|(${le})?\\.(${le}))[pP][+-]?([0-9](\_\*[0-9])\*)[fFdD]?\\b`

}, { begin: "\\b(0|[1-9](\_\*[0-9])\*)[lL]?\\b" }, { begin: `\\b0[xX](${le})[lL]?\\b` }, {

begin: "\\b0(\_\*[0-7])\*[lL]?\\b"

}, { begin: "\\b0[bB][01](\_\*[01])\*[lL]?\\b" }],

relevance: 0

}; function de(e, n, t) { return -1 === t ? "" : e.replace(n, (a => de(e, n, t - 1))) }

const ge = "[A-Za-z$\_][0-9A-Za-z$\_]\*", ue = ["as", "in", "of", "if", "for", "while", "finally", "var", "new", "function", "do", "return", "void", "else", "break", "catch", "instanceof", "with", "throw", "case", "default", "try", "switch", "continue", "typeof", "delete", "let", "yield", "const", "class", "debugger", "async", "await", "static", "import", "from", "export", "extends"], be = ["true", "false", "null", "undefined", "NaN", "Infinity"], me = ["Intl", "DataView", "Number", "Math", "Date", "String", "RegExp", "Object", "Function", "Boolean", "Error", "Symbol", "Set", "Map", "WeakSet", "WeakMap", "Proxy", "Reflect", "JSON", "Promise", "Float64Array", "Int16Array", "Int32Array", "Int8Array", "Uint16Array", "Uint32Array", "Float32Array", "Array", "Uint8Array", "Uint8ClampedArray", "ArrayBuffer", "BigInt64Array", "BigUint64Array", "BigInt"], pe = ["EvalError", "InternalError", "RangeError", "ReferenceError", "SyntaxError", "TypeError", "URIError"], \_e = ["setInterval", "setTimeout", "clearInterval", "clearTimeout", "require", "exports", "eval", "isFinite", "isNaN", "parseFloat", "parseInt", "decodeURI", "decodeURIComponent", "encodeURI", "encodeURIComponent", "escape", "unescape"], he = ["arguments", "this", "super", "console", "window", "document", "localStorage", "module", "global"], fe = [].concat(\_e, me, pe)

; function Ee(e) {

const n = ge, t = {

begin: /<[A-Za-z0-9\\.\_:-]+/,

end: /\/[A-Za-z0-9\\.\_:-]+>|\/>/, isTrulyOpeningTag: (e, n) => {

const t = e[0].length + e.index, a = e.input[t]; "<" !== a ? ">" === a && (((e, { after: n }) => {

const t = "</" + e[0].slice(1); return -1 !== e.input.indexOf(t, n)

})(e, {

after: t

}) || n.ignoreMatch()) : n.ignoreMatch()

}

}, a = {

$pattern: ge, keyword: ue, literal: be,

built\_in: fe, "variable.language": he

}, i = "\\.([0-9](\_?[0-9])\*)", s = "0|[1-9](\_?[0-9])\*|0[0-7]\*[89][0-9]\*", r = {

className: "number", variants: [{

begin: `(\\b(${s})((${i})|\\.)?|(${i}))[eE][+-]?([0-9](\_?[0-9])\*)\\b`

}, {

begin: `\\b(${s})\\b((${i})\\b|\\.)?|(${i})\\b`

}, {

begin: "\\b(0|[1-9](\_?[0-9])\*)n\\b"

}, {

begin: "\\b0[xX][0-9a-fA-F](\_?[0-9a-fA-F])\*n?\\b"

}, {

begin: "\\b0[bB][0-1](\_?[0-1])\*n?\\b"

}, { begin: "\\b0[oO][0-7](\_?[0-7])\*n?\\b" }, {

begin: "\\b0[0-7]+n?\\b"

}], relevance: 0

}, o = {

className: "subst", begin: "\\$\\{",

end: "\\}", keywords: a, contains: []

}, l = {

begin: "html`", end: "", starts: {

end: "`",

returnEnd: !1, contains: [e.BACKSLASH\_ESCAPE, o], subLanguage: "xml"

}

}, c = {

begin: "css`", end: "", starts: {

end: "`", returnEnd: !1,

contains: [e.BACKSLASH\_ESCAPE, o], subLanguage: "css"

}

}, d = {

className: "string",

begin: "`", end: "`", contains: [e.BACKSLASH\_ESCAPE, o]

}, u = {

className: "comment",

variants: [e.COMMENT(/\/\\*\\*(?!\/)/, "\\\*/", {

relevance: 0, contains: [{

begin: "(?=@[A-Za-z]+)", relevance: 0, contains: [{

className: "doctag",

begin: "@[A-Za-z]+"

}, {

className: "type", begin: "\\{", end: "\\}", excludeEnd: !0,

excludeBegin: !0, relevance: 0

}, {

className: "variable", begin: n + "(?=\\s\*(-)|$)",

endsParent: !0, relevance: 0

}, { begin: /(?=[^\n])\s/, relevance: 0 }]

}]

}), e.C\_BLOCK\_COMMENT\_MODE, e.C\_LINE\_COMMENT\_MODE]

}, m = [e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE, l, c, d, r, e.REGEXP\_MODE]

; o.contains = m.concat({

begin: /\{/, end: /\}/, keywords: a, contains: ["self"].concat(m)

}); const p = [].concat(u, o.contains), \_ = p.concat([{

begin: /\(/, end: /\)/, keywords: a,

contains: ["self"].concat(p)

}]), h = {

className: "params", begin: /\(/, end: /\)/,

excludeBegin: !0, excludeEnd: !0, keywords: a, contains: \_

}, f = {

variants: [{

match: [/class/, /\s+/, n], scope: { 1: "keyword", 3: "title.class" }

}, {

match: [/extends/, /\s+/, b(n, "(", b(/\./, n), ")\*")], scope: {

1: "keyword",

3: "title.class.inherited"

}

}]

}, E = {

relevance: 0,

match: /\b[A-Z][a-z]+([A-Z][a-z]+)\*/, className: "title.class", keywords: {

\_: [...me, ...pe]

}

}, y = {

variants: [{ match: [/function/, /\s+/, n, /(?=\s\*\()/] }, {

match: [/function/, /\s\*(?=\()/]

}], className: { 1: "keyword", 3: "title.function" },

label: "func.def", contains: [h], illegal: /%/

}, N = {

match: b(/\b/, (w = [...\_e, "super"], b("(?!", w.join("|"), ")")), n, g(/\(/)),

className: "title.function", relevance: 0

}; var w; const v = {

begin: b(/\./, g(b(n, /(?![0-9A-Za-z$\_(])/))), end: n, excludeBegin: !0,

keywords: "prototype", className: "property", relevance: 0

}, O = {

match: [/get|set/, /\s+/, n, /(?=\()/], className: { 1: "keyword", 3: "title.function" },

contains: [{ begin: /\(\)/ }, h]

}, M = "(\\([^()]\*(\\([^()]\*(\\([^()]\*\\)[^()]\*)\*\\)[^()]\*)\*\\)|" + e.UNDERSCORE\_IDENT\_RE + ")\\s\*=>", x = {

match: [/const|var|let/, /\s+/, n, /\s\*/, /=\s\*/, g(M)], className: {

1: "keyword",

3: "title.function"

}, contains: [h]

}; return {

name: "Javascript",

aliases: ["js", "jsx", "mjs", "cjs"], keywords: a, exports: { PARAMS\_CONTAINS: \_ },

illegal: /#(?![$\_A-z])/, contains: [e.SHEBANG({

label: "shebang", binary: "node",

relevance: 5

}), {

label: "use\_strict", className: "meta", relevance: 10,

begin: /^\s\*['"]use (strict|asm)['"]/

}, e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE, l, c, d, u, r, E, {

className: "attr",

begin: n + g(":"), relevance: 0

}, x, {

begin: "(" + e.RE\_STARTERS\_RE + "|\\b(case|return|throw)\\b)\\s\*",

keywords: "return throw case", relevance: 0, contains: [u, e.REGEXP\_MODE, {

className: "function", begin: M, returnBegin: !0, end: "\\s\*=>", contains: [{

className: "params", variants: [{ begin: e.UNDERSCORE\_IDENT\_RE, relevance: 0 }, {

className: null, begin: /\(\s\*\)/, skip: !0

}, {

begin: /\(/, end: /\)/, excludeBegin: !0,

excludeEnd: !0, keywords: a, contains: \_

}]

}]

}, { begin: /,/, relevance: 0 }, {

match: /\s+/,

relevance: 0

}, {

variants: [{ begin: "<>", end: "</>" }, {

begin: t.begin,

"on:begin": t.isTrulyOpeningTag, end: t.end

}], subLanguage: "xml", contains: [{

begin: t.begin, end: t.end, skip: !0, contains: ["self"]

}]

}]

}, y, {

beginKeywords: "while if switch catch for"

}, {

begin: "\\b(?!function)" + e.UNDERSCORE\_IDENT\_RE + "\\([^()]\*(\\([^()]\*(\\([^()]\*\\)[^()]\*)\*\\)[^()]\*)\*\\)\\s\*\\{",

returnBegin: !0, label: "func.def", contains: [h, e.inherit(e.TITLE\_MODE, {

begin: n,

className: "title.function"

})]

}, { match: /\.\.\./, relevance: 0 }, v, {

match: "\\$" + n,

relevance: 0

}, {

match: [/\bconstructor(?=\s\*\()/], className: { 1: "title.function" },

contains: [h]

}, N, {

relevance: 0, match: /\b[A-Z][A-Z\_0-9]+\b/,

className: "variable.constant"

}, f, O, { match: /\$[(.]/ }]

}

}

const ye = e => b(/\b/, e, /\w$/.test(e) ? /\b/ : /\B/), Ne = ["Protocol", "Type"].map(ye), we = ["init", "self"].map(ye), ve = ["Any", "Self"], Oe = ["actor", "associatedtype", "async", "await", /as\?/, /as!/, "as", "break", "case", "catch", "class", "continue", "convenience", "default", "defer", "deinit", "didSet", "do", "dynamic", "else", "enum", "extension", "fallthrough", /fileprivate\(set\)/, "fileprivate", "final", "for", "func", "get", "guard", "if", "import", "indirect", "infix", /init\?/, /init!/, "inout", /internal\(set\)/, "internal", "in", "is", "isolated", "nonisolated", "lazy", "let", "mutating", "nonmutating", /open\(set\)/, "open", "operator", "optional", "override", "postfix", "precedencegroup", "prefix", /private\(set\)/, "private", "protocol", /public\(set\)/, "public", "repeat", "required", "rethrows", "return", "set", "some", "static", "struct", "subscript", "super", "switch", "throws", "throw", /try\?/, /try!/, "try", "typealias", /unowned\(safe\)/, /unowned\(unsafe\)/, "unowned", "var", "weak", "where", "while", "willSet"], Me = ["false", "nil", "true"], xe = ["assignment", "associativity", "higherThan", "left", "lowerThan", "none", "right"], Se = ["#colorLiteral", "#column", "#dsohandle", "#else", "#elseif", "#endif", "#error", "#file", "#fileID", "#fileLiteral", "#filePath", "#function", "#if", "#imageLiteral", "#keyPath", "#line", "#selector", "#sourceLocation", "#warn\_unqualified\_access", "#warning"], ke = ["abs", "all", "any", "assert", "assertionFailure", "debugPrint", "dump", "fatalError", "getVaList", "isKnownUniquelyReferenced", "max", "min", "numericCast", "pointwiseMax", "pointwiseMin", "precondition", "preconditionFailure", "print", "readLine", "repeatElement", "sequence", "stride", "swap", "swift\_unboxFromSwiftValueWithType", "transcode", "type", "unsafeBitCast", "unsafeDowncast", "withExtendedLifetime", "withUnsafeMutablePointer", "withUnsafePointer", "withVaList", "withoutActuallyEscaping", "zip"], Ae = m(/[/=\-+!\*%<>&|^~?]/, /[\u00A1-\u00A7]/, /[\u00A9\u00AB]/, /[\u00AC\u00AE]/, /[\u00B0\u00B1]/, /[\u00B6\u00BB\u00BF\u00D7\u00F7]/, /[\u2016-\u2017]/, /[\u2020-\u2027]/, /[\u2030-\u203E]/, /[\u2041-\u2053]/, /[\u2055-\u205E]/, /[\u2190-\u23FF]/, /[\u2500-\u2775]/, /[\u2794-\u2BFF]/, /[\u2E00-\u2E7F]/, /[\u3001-\u3003]/, /[\u3008-\u3020]/, /[\u3030]/), Ce = m(Ae, /[\u0300-\u036F]/, /[\u1DC0-\u1DFF]/, /[\u20D0-\u20FF]/, /[\uFE00-\uFE0F]/, /[\uFE20-\uFE2F]/), Te = b(Ae, Ce, "\*"), Re = m(/[a-zA-Z\_]/, /[\u00A8\u00AA\u00AD\u00AF\u00B2-\u00B5\u00B7-\u00BA]/, /[\u00BC-\u00BE\u00C0-\u00D6\u00D8-\u00F6\u00F8-\u00FF]/, /[\u0100-\u02FF\u0370-\u167F\u1681-\u180D\u180F-\u1DBF]/, /[\u1E00-\u1FFF]/, /[\u200B-\u200D\u202A-\u202E\u203F-\u2040\u2054\u2060-\u206F]/, /[\u2070-\u20CF\u2100-\u218F\u2460-\u24FF\u2776-\u2793]/, /[\u2C00-\u2DFF\u2E80-\u2FFF]/, /[\u3004-\u3007\u3021-\u302F\u3031-\u303F\u3040-\uD7FF]/, /[\uF900-\uFD3D\uFD40-\uFDCF\uFDF0-\uFE1F\uFE30-\uFE44]/, /[\uFE47-\uFEFE\uFF00-\uFFFD]/), De = m(Re, /\d/, /[\u0300-\u036F\u1DC0-\u1DFF\u20D0-\u20FF\uFE20-\uFE2F]/), Ie = b(Re, De, "\*"), Le = b(/[A-Z]/, De, "\*"), Be = ["autoclosure", b(/convention\(/, m("swift", "block", "c"), /\)/), "discardableResult", "dynamicCallable", "dynamicMemberLookup", "escaping", "frozen", "GKInspectable", "IBAction", "IBDesignable", "IBInspectable", "IBOutlet", "IBSegueAction", "inlinable", "main", "nonobjc", "NSApplicationMain", "NSCopying", "NSManaged", b(/objc\(/, Ie, /\)/), "objc", "objcMembers", "propertyWrapper", "requires\_stored\_property\_inits", "resultBuilder", "testable", "UIApplicationMain", "unknown", "usableFromInline"], $e = ["iOS", "iOSApplicationExtension", "macOS", "macOSApplicationExtension", "macCatalyst", "macCatalystApplicationExtension", "watchOS", "watchOSApplicationExtension", "tvOS", "tvOSApplicationExtension", "swift"]

; var ze = Object.freeze({

\_\_proto\_\_: null, grmr\_bash: e => {

const n = {}, t = {

begin: /\$\{/,

end: /\}/, contains: ["self", { begin: /:-/, contains: [n] }]

}; Object.assign(n, {

className: "variable", variants: [{

begin: b(/\$[\w\d#@][\w\d\_]\*/, "(?![\\w\\d])(?![$])")

}, t]

}); const a = {

className: "subst", begin: /\$\(/, end: /\)/, contains: [e.BACKSLASH\_ESCAPE]

}, i = {

begin: /<<-?\s\*(?=\w+)/, starts: {

contains: [e.END\_SAME\_AS\_BEGIN({

begin: /(\w+)/,

end: /(\w+)/, className: "string"

})]

}

}, s = {

className: "string", begin: /"/, end: /"/,

contains: [e.BACKSLASH\_ESCAPE, n, a]

}; a.contains.push(s); const r = {

begin: /\$\(\(/,

end: /\)\)/, contains: [{ begin: /\d+#[0-9a-f]+/, className: "number" }, e.NUMBER\_MODE, n]

}, o = e.SHEBANG({

binary: "(fish|bash|zsh|sh|csh|ksh|tcsh|dash|scsh)", relevance: 10

}), l = {

className: "function", begin: /\w[\w\d\_]\*\s\*\(\s\*\)\s\*\{/, returnBegin: !0,

contains: [e.inherit(e.TITLE\_MODE, { begin: /\w[\w\d\_]\*/ })], relevance: 0

}; return {

name: "Bash", aliases: ["sh"], keywords: {

$pattern: /\b[a-z.\_-]+\b/,

keyword: ["if", "then", "else", "elif", "fi", "for", "while", "in", "do", "done", "case", "esac", "function"],

literal: ["true", "false"],

built\_in: "break cd continue eval exec exit export getopts hash pwd readonly return shift test times trap umask unset alias bind builtin caller command declare echo enable help let local logout mapfile printf read readarray source type typeset ulimit unalias set shopt autoload bg bindkey bye cap chdir clone comparguments compcall compctl compdescribe compfiles compgroups compquote comptags comptry compvalues dirs disable disown echotc echoti emulate fc fg float functions getcap getln history integer jobs kill limit log noglob popd print pushd pushln rehash sched setcap setopt stat suspend ttyctl unfunction unhash unlimit unsetopt vared wait whence where which zcompile zformat zftp zle zmodload zparseopts zprof zpty zregexparse zsocket zstyle ztcp"

}, contains: [o, e.SHEBANG(), l, r, e.HASH\_COMMENT\_MODE, i, s, {

className: "", begin: /\\"/

}, { className: "string", begin: /'/, end: /'/ }, n]

}

}, grmr\_c: e => {

const n = e.COMMENT("//", "$", {

contains: [{ begin: /\\\n/ }]

}), t = "[a-zA-Z\_]\\w\*::", a = "(decltype\\(auto\\)|" + u(t) + "[a-zA-Z\_]\\w\*" + u("<[^<>]+>") + ")", i = {

className: "type", variants: [{ begin: "\\b[a-z\\d\_]\*\_t\\b" }, {

match: /\batomic\_[a-z]{3,6}\b/

}]

}, s = {

className: "string", variants: [{

begin: '(u8?|U|L)?"', end: '"', illegal: "\\n", contains: [e.BACKSLASH\_ESCAPE]

}, {

begin: "(u8?|U|L)?'(\\\\(x[0-9A-Fa-f]{2}|u[0-9A-Fa-f]{4,8}|[0-7]{3}|\\S)|.)",

end: "'", illegal: "."

}, e.END\_SAME\_AS\_BEGIN({

begin: /(?:u8?|U|L)?R"([^()\\ ]{0,16})\(/, end: /\)([^()\\ ]{0,16})"/

})]

}, r = {

className: "number", variants: [{ begin: "\\b(0b[01']+)" }, {

begin: "(-?)\\b([\\d']+(\\.[\\d']\*)?|\\.[\\d']+)((ll|LL|l|L)(u|U)?|(u|U)(ll|LL|l|L)?|f|F|b|B)"

}, {

begin: "(-?)(\\b0[xX][a-fA-F0-9']+|(\\b[\\d']+(\\.[\\d']\*)?|\\.[\\d']+)([eE][-+]?[\\d']+)?)"

}], relevance: 0

}, o = {

className: "meta", begin: /#\s\*[a-z]+\b/, end: /$/, keywords: {

keyword: "if else elif endif define undef warning error line pragma \_Pragma ifdef ifndef include"

}, contains: [{ begin: /\\\n/, relevance: 0 }, e.inherit(s, { className: "string" }), {

className: "string", begin: /<.\*?>/

}, n, e.C\_BLOCK\_COMMENT\_MODE]

}, l = {

className: "title", begin: u(t) + e.IDENT\_RE, relevance: 0

}, c = u(t) + e.IDENT\_RE + "\\s\*\\(", d = {

keyword: ["asm", "auto", "break", "case", "const", "continue", "default", "do", "else", "enum", "extern", "for", "fortran", "goto", "if", "inline", "register", "restrict", "return", "sizeof", "static", "struct", "switch", "typedef", "union", "volatile", "while", "\_Alignas", "\_Alignof", "\_Atomic", "\_Generic", "\_Noreturn", "\_Static\_assert", "\_Thread\_local", "alignas", "alignof", "noreturn", "static\_assert", "thread\_local", "\_Pragma"],

type: ["float", "double", "signed", "unsigned", "int", "short", "long", "char", "void", "\_Bool", "\_Complex", "\_Imaginary", "\_Decimal32", "\_Decimal64", "\_Decimal128", "complex", "bool", "imaginary"],

literal: "true false NULL",

built\_in: "std string wstring cin cout cerr clog stdin stdout stderr stringstream istringstream ostringstream auto\_ptr deque list queue stack vector map set pair bitset multiset multimap unordered\_set unordered\_map unordered\_multiset unordered\_multimap priority\_queue make\_pair array shared\_ptr abort terminate abs acos asin atan2 atan calloc ceil cosh cos exit exp fabs floor fmod fprintf fputs free frexp fscanf future isalnum isalpha iscntrl isdigit isgraph islower isprint ispunct isspace isupper isxdigit tolower toupper labs ldexp log10 log malloc realloc memchr memcmp memcpy memset modf pow printf putchar puts scanf sinh sin snprintf sprintf sqrt sscanf strcat strchr strcmp strcpy strcspn strlen strncat strncmp strncpy strpbrk strrchr strspn strstr tanh tan vfprintf vprintf vsprintf endl initializer\_list unique\_ptr"

}, g = [o, i, n, e.C\_BLOCK\_COMMENT\_MODE, r, s], b = {

variants: [{ begin: /=/, end: /;/ }, {

begin: /\(/, end: /\)/

}, { beginKeywords: "new throw return else", end: /;/ }],

keywords: d, contains: g.concat([{

begin: /\(/, end: /\)/, keywords: d,

contains: g.concat(["self"]), relevance: 0

}]), relevance: 0

}, m = {

begin: "(" + a + "[\\\*&\\s]+)+" + c, returnBegin: !0, end: /[{;=]/, excludeEnd: !0,

keywords: d, illegal: /[^\w\s\\*&:<>.]/, contains: [{

begin: "decltype\\(auto\\)",

keywords: d, relevance: 0

}, {

begin: c, returnBegin: !0, contains: [e.inherit(l, {

className: "title.function"

})], relevance: 0

}, { relevance: 0, match: /,/ }, {

className: "params", begin: /\(/, end: /\)/, keywords: d, relevance: 0,

contains: [n, e.C\_BLOCK\_COMMENT\_MODE, s, r, i, {

begin: /\(/, end: /\)/, keywords: d,

relevance: 0, contains: ["self", n, e.C\_BLOCK\_COMMENT\_MODE, s, r, i]

}]

}, i, n, e.C\_BLOCK\_COMMENT\_MODE, o]

}; return {

name: "C", aliases: ["h"], keywords: d,

disableAutodetect: !0, illegal: "</", contains: [].concat(b, m, g, [o, {

begin: e.IDENT\_RE + "::", keywords: d

}, {

className: "class",

beginKeywords: "enum class struct union", end: /[{;:<>=]/, contains: [{

beginKeywords: "final class struct"

}, e.TITLE\_MODE]

}]), exports: {

preprocessor: o,

strings: s, keywords: d

}

}

}, grmr\_cpp: e => {

const n = e.COMMENT("//", "$", {

contains: [{

begin: /\\\n/

}]

}), t = "[a-zA-Z\_]\\w\*::", a = "(?!struct)(decltype\\(auto\\)|" + u(t) + "[a-zA-Z\_]\\w\*" + u("<[^<>]+>") + ")", i = {

className: "type", begin: "\\b[a-z\\d\_]\*\_t\\b"

}, s = {

className: "string", variants: [{

begin: '(u8?|U|L)?"', end: '"', illegal: "\\n", contains: [e.BACKSLASH\_ESCAPE]

}, {

begin: "(u8?|U|L)?'(\\\\(x[0-9A-Fa-f]{2}|u[0-9A-Fa-f]{4,8}|[0-7]{3}|\\S)|.)",

end: "'", illegal: "."

}, e.END\_SAME\_AS\_BEGIN({

begin: /(?:u8?|U|L)?R"([^()\\ ]{0,16})\(/, end: /\)([^()\\ ]{0,16})"/

})]

}, r = {

className: "number", variants: [{ begin: "\\b(0b[01']+)" }, {

begin: "(-?)\\b([\\d']+(\\.[\\d']\*)?|\\.[\\d']+)((ll|LL|l|L)(u|U)?|(u|U)(ll|LL|l|L)?|f|F|b|B)"

}, {

begin: "(-?)(\\b0[xX][a-fA-F0-9']+|(\\b[\\d']+(\\.[\\d']\*)?|\\.[\\d']+)([eE][-+]?[\\d']+)?)"

}], relevance: 0

}, o = {

className: "meta", begin: /#\s\*[a-z]+\b/, end: /$/, keywords: {

keyword: "if else elif endif define undef warning error line pragma \_Pragma ifdef ifndef include"

}, contains: [{ begin: /\\\n/, relevance: 0 }, e.inherit(s, { className: "string" }), {

className: "string", begin: /<.\*?>/

}, n, e.C\_BLOCK\_COMMENT\_MODE]

}, l = {

className: "title", begin: u(t) + e.IDENT\_RE, relevance: 0

}, c = u(t) + e.IDENT\_RE + "\\s\*\\(", d = {

type: ["bool", "char", "char16\_t", "char32\_t", "char8\_t", "double", "float", "int", "long", "short", "void", "wchar\_t"],

keyword: ["alignas", "alignof", "and", "and\_eq", "asm", "atomic\_cancel", "atomic\_commit", "atomic\_noexcept", "auto", "bitand", "bitor", "break", "case", "catch", "class", "co\_await", "co\_return", "co\_yield", "compl", "concept", "const", "const\_cast|10", "consteval", "constexpr", "constinit", "continue", "decltype", "default", "delete", "do", "dynamic\_cast|10", "else", "enum", "explicit", "export", "extern", "false", "final", "for", "friend", "goto", "if", "import", "inline", "module", "mutable", "namespace", "new", "noexcept", "not", "not\_eq", "nullptr", "operator", "or", "or\_eq", "override", "private", "protected", "public", "reflexpr", "register", "reinterpret\_cast|10", "requires", "return", "signed", "sizeof", "static", "static\_assert", "static\_cast|10", "struct", "switch", "synchronized", "template", "this", "thread\_local", "throw", "transaction\_safe", "transaction\_safe\_dynamic", "true", "try", "typedef", "typeid", "typename", "union", "unsigned", "using", "virtual", "volatile", "while", "xor", "xor\_eq,"],

literal: ["NULL", "false", "nullopt", "nullptr", "true"], built\_in: ["\_Pragma"],

\_type\_hints: ["any", "auto\_ptr", "barrier", "binary\_semaphore", "bitset", "complex", "condition\_variable", "condition\_variable\_any", "counting\_semaphore", "deque", "false\_type", "future", "imaginary", "initializer\_list", "istringstream", "jthread", "latch", "lock\_guard", "multimap", "multiset", "mutex", "optional", "ostringstream", "packaged\_task", "pair", "promise", "priority\_queue", "queue", "recursive\_mutex", "recursive\_timed\_mutex", "scoped\_lock", "set", "shared\_future", "shared\_lock", "shared\_mutex", "shared\_timed\_mutex", "shared\_ptr", "stack", "string\_view", "stringstream", "timed\_mutex", "thread", "true\_type", "tuple", "unique\_lock", "unique\_ptr", "unordered\_map", "unordered\_multimap", "unordered\_multiset", "unordered\_set", "variant", "vector", "weak\_ptr", "wstring", "wstring\_view"]

}, m = {

className: "function.dispatch", relevance: 0, keywords: {

\_hint: ["abort", "abs", "acos", "apply", "as\_const", "asin", "atan", "atan2", "calloc", "ceil", "cerr", "cin", "clog", "cos", "cosh", "cout", "declval", "endl", "exchange", "exit", "exp", "fabs", "floor", "fmod", "forward", "fprintf", "fputs", "free", "frexp", "fscanf", "future", "invoke", "isalnum", "isalpha", "iscntrl", "isdigit", "isgraph", "islower", "isprint", "ispunct", "isspace", "isupper", "isxdigit", "labs", "launder", "ldexp", "log", "log10", "make\_pair", "make\_shared", "make\_shared\_for\_overwrite", "make\_tuple", "make\_unique", "malloc", "memchr", "memcmp", "memcpy", "memset", "modf", "move", "pow", "printf", "putchar", "puts", "realloc", "scanf", "sin", "sinh", "snprintf", "sprintf", "sqrt", "sscanf", "std", "stderr", "stdin", "stdout", "strcat", "strchr", "strcmp", "strcpy", "strcspn", "strlen", "strncat", "strncmp", "strncpy", "strpbrk", "strrchr", "strspn", "strstr", "swap", "tan", "tanh", "terminate", "to\_underlying", "tolower", "toupper", "vfprintf", "visit", "vprintf", "vsprintf"]

},

begin: b(/\b/, /(?!decltype)/, /(?!if)/, /(?!for)/, /(?!while)/, e.IDENT\_RE, g(/(<[^<>]+>|)\s\*\(/))

}, p = [m, o, i, n, e.C\_BLOCK\_COMMENT\_MODE, r, s], \_ = {

variants: [{ begin: /=/, end: /;/ }, {

begin: /\(/, end: /\)/

}, { beginKeywords: "new throw return else", end: /;/ }],

keywords: d, contains: p.concat([{

begin: /\(/, end: /\)/, keywords: d,

contains: p.concat(["self"]), relevance: 0

}]), relevance: 0

}, h = {

className: "function",

begin: "(" + a + "[\\\*&\\s]+)+" + c, returnBegin: !0, end: /[{;=]/, excludeEnd: !0,

keywords: d, illegal: /[^\w\s\\*&:<>.]/, contains: [{

begin: "decltype\\(auto\\)",

keywords: d, relevance: 0

}, { begin: c, returnBegin: !0, contains: [l], relevance: 0 }, {

begin: /::/, relevance: 0

}, { begin: /:/, endsWithParent: !0, contains: [s, r] }, {

relevance: 0, match: /,/

}, {

className: "params", begin: /\(/, end: /\)/, keywords: d,

relevance: 0, contains: [n, e.C\_BLOCK\_COMMENT\_MODE, s, r, i, {

begin: /\(/, end: /\)/,

keywords: d, relevance: 0, contains: ["self", n, e.C\_BLOCK\_COMMENT\_MODE, s, r, i]

}]

}, i, n, e.C\_BLOCK\_COMMENT\_MODE, o]

}; return {

name: "C++",

aliases: ["cc", "c++", "h++", "hpp", "hh", "hxx", "cxx"], keywords: d, illegal: "</",

classNameAliases: { "function.dispatch": "built\_in" },

contains: [].concat(\_, h, m, p, [o, {

begin: "\\b(deque|list|queue|priority\_queue|pair|stack|vector|map|set|bitset|multiset|multimap|unordered\_map|unordered\_set|unordered\_multiset|unordered\_multimap|array|tuple|optional|variant|function)\\s\*<",

end: ">", keywords: d, contains: ["self", i]

}, { begin: e.IDENT\_RE + "::", keywords: d }, {

match: [/\b(?:enum(?:\s+(?:class|struct))?|class|struct|union)/, /\s+/, /\w+/],

className: { 1: "keyword", 3: "title.class" }

}])

}

}, grmr\_csharp: e => {

const n = {

keyword: ["abstract", "as", "base", "break", "case", "catch", "class", "const", "continue", "do", "else", "event", "explicit", "extern", "finally", "fixed", "for", "foreach", "goto", "if", "implicit", "in", "interface", "internal", "is", "lock", "namespace", "new", "operator", "out", "override", "params", "private", "protected", "public", "readonly", "record", "ref", "return", "sealed", "sizeof", "stackalloc", "static", "struct", "switch", "this", "throw", "try", "typeof", "unchecked", "unsafe", "using", "virtual", "void", "volatile", "while"].concat(["add", "alias", "and", "ascending", "async", "await", "by", "descending", "equals", "from", "get", "global", "group", "init", "into", "join", "let", "nameof", "not", "notnull", "on", "or", "orderby", "partial", "remove", "select", "set", "unmanaged", "value|0", "var", "when", "where", "with", "yield"]),

built\_in: ["bool", "byte", "char", "decimal", "delegate", "double", "dynamic", "enum", "float", "int", "long", "nint", "nuint", "object", "sbyte", "short", "string", "ulong", "uint", "ushort"],

literal: ["default", "false", "null", "true"]

}, t = e.inherit(e.TITLE\_MODE, {

begin: "[a-zA-Z](\\.?\\w)\*"

}), a = {

className: "number", variants: [{

begin: "\\b(0b[01']+)"

}, {

begin: "(-?)\\b([\\d']+(\\.[\\d']\*)?|\\.[\\d']+)(u|U|l|L|ul|UL|f|F|b|B)"

}, {

begin: "(-?)(\\b0[xX][a-fA-F0-9']+|(\\b[\\d']+(\\.[\\d']\*)?|\\.[\\d']+)([eE][-+]?[\\d']+)?)"

}], relevance: 0

}, i = {

className: "string", begin: '@"', end: '"', contains: [{ begin: '""' }]

}, s = e.inherit(i, { illegal: /\n/ }), r = {

className: "subst", begin: /\{/, end: /\}/,

keywords: n

}, o = e.inherit(r, { illegal: /\n/ }), l = {

className: "string", begin: /\$"/,

end: '"', illegal: /\n/, contains: [{ begin: /\{\{/ }, {

begin: /\}\}/

}, e.BACKSLASH\_ESCAPE, o]

}, c = {

className: "string", begin: /\$@"/, end: '"', contains: [{

begin: /\{\{/

}, { begin: /\}\}/ }, { begin: '""' }, r]

}, d = e.inherit(c, {

illegal: /\n/,

contains: [{ begin: /\{\{/ }, { begin: /\}\}/ }, { begin: '""' }, o]

})

; r.contains = [c, l, i, e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE, a, e.C\_BLOCK\_COMMENT\_MODE],

o.contains = [d, l, s, e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE, a, e.inherit(e.C\_BLOCK\_COMMENT\_MODE, {

illegal: /\n/

})]; const g = {

variants: [c, l, i, e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE]

}, u = {

begin: "<", end: ">", contains: [{ beginKeywords: "in out" }, t]

}, b = e.IDENT\_RE + "(<" + e.IDENT\_RE + "(\\s\*,\\s\*" + e.IDENT\_RE + ")\*>)?(\\[\\])?", m = {

begin: "@" + e.IDENT\_RE, relevance: 0

}; return {

name: "C#", aliases: ["cs", "c#"],

keywords: n, illegal: /::/, contains: [e.COMMENT("///", "$", {

returnBegin: !0,

contains: [{

className: "doctag", variants: [{ begin: "///", relevance: 0 }, {

begin: "\x3c!--|--\x3e"

}, { begin: "</?", end: ">" }]

}]

}), e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE, {

className: "meta", begin: "#",

end: "$", keywords: {

keyword: "if else elif endif define undef warning error line region endregion pragma checksum"

}

}, g, a, {

beginKeywords: "class interface", relevance: 0, end: /[{;=]/,

illegal: /[^\s:,]/, contains: [{

beginKeywords: "where class"

}, t, u, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE]

}, {

beginKeywords: "namespace",

relevance: 0, end: /[{;=]/, illegal: /[^\s:]/,

contains: [t, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE]

}, {

beginKeywords: "record", relevance: 0, end: /[{;=]/, illegal: /[^\s:]/,

contains: [t, u, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE]

}, {

className: "meta",

begin: "^\\s\*\\[(?=[\\w])", excludeBegin: !0, end: "\\]", excludeEnd: !0, contains: [{

className: "string", begin: /"/, end: /"/

}]

}, {

beginKeywords: "new return throw await else", relevance: 0

}, {

className: "function",

begin: "(" + b + "\\s+)+" + e.IDENT\_RE + "\\s\*(<.+>\\s\*)?\\(", returnBegin: !0,

end: /\s\*[{;=]/, excludeEnd: !0, keywords: n, contains: [{

beginKeywords: "public private protected static internal protected abstract async extern override unsafe virtual new sealed partial",

relevance: 0

}, {

begin: e.IDENT\_RE + "\\s\*(<.+>\\s\*)?\\(", returnBegin: !0,

contains: [e.TITLE\_MODE, u], relevance: 0

}, {

className: "params", begin: /\(/, end: /\)/,

excludeBegin: !0, excludeEnd: !0, keywords: n, relevance: 0,

contains: [g, a, e.C\_BLOCK\_COMMENT\_MODE]

}, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE]

}, m]

}

}, grmr\_css: e => {

const n = ee(e), t = [e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE]; return {

name: "CSS",

case\_insensitive: !0, illegal: /[=|'\$]/, keywords: { keyframePosition: "from to" },

classNameAliases: { keyframePosition: "selector-tag" },

contains: [e.C\_BLOCK\_COMMENT\_MODE, {

begin: /-(webkit|moz|ms|o)-(?=[a-z])/

}, n.CSS\_NUMBER\_MODE, {

className: "selector-id", begin: /#[A-Za-z0-9\_-]+/, relevance: 0

}, {

className: "selector-class", begin: "\\.[a-zA-Z-][a-zA-Z0-9\_-]\*", relevance: 0

}, n.ATTRIBUTE\_SELECTOR\_MODE, {

className: "selector-pseudo", variants: [{

begin: ":(" + ae.join("|") + ")"

}, { begin: "::(" + ie.join("|") + ")" }]

}, n.CSS\_VARIABLE, {

className: "attribute", begin: "\\b(" + se.join("|") + ")\\b"

}, {

begin: ":", end: "[;}]",

contains: [n.HEXCOLOR, n.IMPORTANT, n.CSS\_NUMBER\_MODE, ...t, {

begin: /(url|data-uri)\(/, end: /\)/, relevance: 0, keywords: {

built\_in: "url data-uri"

}, contains: [{ className: "string", begin: /[^)]/, endsWithParent: !0, excludeEnd: !0 }]

}, { className: "built\_in", begin: /[\w-]+(?=\()/ }]

}, {

begin: g(/@/), end: "[{;]",

relevance: 0, illegal: /:/, contains: [{

className: "keyword", begin: /@-?\w[\w]\*(-\w+)\*/

}, {

begin: /\s/, endsWithParent: !0, excludeEnd: !0, relevance: 0, keywords: {

$pattern: /[a-z-]+/, keyword: "and or not only", attribute: te.join(" ")

}, contains: [{

begin: /[a-z-]+(?=:)/, className: "attribute"

}, ...t, n.CSS\_NUMBER\_MODE]

}]

}, {

className: "selector-tag", begin: "\\b(" + ne.join("|") + ")\\b"

}]

}

}, grmr\_diff: e => ({

name: "Diff", aliases: ["patch"], contains: [{

className: "meta", relevance: 10,

match: m(/^@@ +-\d+,\d+ +\+\d+,\d+ +@@/, /^\\*\\*\\* +\d+,\d+ +\\*\\*\\*\\*$/, /^--- +\d+,\d+ +----$/)

}, {

className: "comment", variants: [{

begin: m(/Index: /, /^index/, /={3,}/, /^-{3}/, /^\\*{3} /, /^\+{3}/, /^diff --git/),

end: /$/

}, { match: /^\\*{15}$/ }]

}, { className: "addition", begin: /^\+/, end: /$/ }, {

className: "deletion", begin: /^-/, end: /$/

}, {

className: "addition", begin: /^!/,

end: /$/

}]

}), grmr\_go: e => {

const n = {

keyword: ["break", "default", "func", "interface", "select", "case", "map", "struct", "chan", "else", "goto", "package", "switch", "const", "fallthrough", "if", "range", "type", "continue", "for", "import", "return", "var", "go", "defer", "bool", "byte", "complex64", "complex128", "float32", "float64", "int8", "int16", "int32", "int64", "string", "uint8", "uint16", "uint32", "uint64", "int", "uint", "uintptr", "rune"],

literal: ["true", "false", "iota", "nil"],

built\_in: ["append", "cap", "close", "complex", "copy", "imag", "len", "make", "new", "panic", "print", "println", "real", "recover", "delete"]

}; return {

name: "Go", aliases: ["golang"], keywords: n, illegal: "</",

contains: [e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE, {

className: "string",

variants: [e.QUOTE\_STRING\_MODE, e.APOS\_STRING\_MODE, { begin: "`", end: "`" }]

}, {

className: "number", variants: [{

begin: e.C\_NUMBER\_RE + "[i]", relevance: 1

}, e.C\_NUMBER\_MODE]

}, { begin: /:=/ }, {

className: "function", beginKeywords: "func",

end: "\\s\*(\\{|$)", excludeEnd: !0, contains: [e.TITLE\_MODE, {

className: "params",

begin: /\(/, end: /\)/, keywords: n, illegal: /["']/

}]

}]

}

}, grmr\_ini: e => {

const n = {

className: "number", relevance: 0, variants: [{ begin: /([+-]+)?[\d]+\_[\d\_]+/ }, {

begin: e.NUMBER\_RE

}]

}, t = e.COMMENT(); t.variants = [{ begin: /;/, end: /$/ }, {

begin: /#/,

end: /$/

}]; const a = {

className: "variable", variants: [{ begin: /\$[\w\d"][\w\d\_]\*/ }, {

begin: /\$\{(.\*?)\}/

}]

}, i = {

className: "literal",

begin: /\bon|off|true|false|yes|no\b/

}, s = {

className: "string",

contains: [e.BACKSLASH\_ESCAPE], variants: [{ begin: "'''", end: "'''", relevance: 10 }, {

begin: '"""', end: '"""', relevance: 10

}, { begin: '"', end: '"' }, { begin: "'", end: "'" }]

}, r = {

begin: /\[/, end: /\]/, contains: [t, i, a, s, n, "self"], relevance: 0

}, o = m(/[A-Za-z0-9\_-]+/, /"(\\"|[^"])\*"/, /'[^']\*'/); return {

name: "TOML, also INI",

aliases: ["toml"], case\_insensitive: !0, illegal: /\S/, contains: [t, {

className: "section", begin: /\[+/, end: /\]+/

}, {

begin: b(o, "(\\s\*\\.\\s\*", o, ")\*", g(/\s\*=\s\*[^#\s]/)), className: "attr", starts: {

end: /$/, contains: [t, r, i, a, s, n]

}

}]

}

}, grmr\_java: e => {

const n = "[\xc0-\u02b8a-zA-Z\_$][\xc0-\u02b8a-zA-Z\_$0-9]\*", t = n + de("(?:<" + n + "~~~(?:\\s\*,\\s\*" + n + "~~~)\*>)?", /~~~/g, 2), a = {

keyword: ["synchronized", "abstract", "private", "var", "static", "if", "const ", "for", "while", "strictfp", "finally", "protected", "import", "native", "final", "void", "enum", "else", "break", "transient", "catch", "instanceof", "volatile", "case", "assert", "package", "default", "public", "try", "switch", "continue", "throws", "protected", "public", "private", "module", "requires", "exports", "do"],

literal: ["false", "true", "null"],

type: ["char", "boolean", "long", "float", "int", "byte", "short", "double"],

built\_in: ["super", "this"]

}, i = {

className: "meta", begin: "@" + n, contains: [{

begin: /\(/, end: /\)/, contains: ["self"]

}]

}, s = {

className: "params", begin: /\(/,

end: /\)/, keywords: a, relevance: 0, contains: [e.C\_BLOCK\_COMMENT\_MODE], endsParent: !0

}

; return {

name: "Java", aliases: ["jsp"], keywords: a, illegal: /<\/|#/,

contains: [e.COMMENT("/\\\*\\\*", "\\\*/", {

relevance: 0, contains: [{

begin: /\w+@/,

relevance: 0

}, { className: "doctag", begin: "@[A-Za-z]+" }]

}), {

begin: /import java\.[a-z]+\./, keywords: "import", relevance: 2

}, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE, e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE, {

match: [/\b(?:class|interface|enum|extends|implements|new)/, /\s+/, n], className: {

1: "keyword", 3: "title.class"

}

}, {

begin: [n, /\s+/, n, /\s+/, /=/], className: {

1: "type",

3: "variable", 5: "operator"

}

}, {

begin: [/record/, /\s+/, n], className: {

1: "keyword",

3: "title.class"

}, contains: [s, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE]

}, {

beginKeywords: "new throw return else", relevance: 0

}, {

begin: ["(?:" + t + "\\s+)", e.UNDERSCORE\_IDENT\_RE, /\s\*(?=\()/], className: {

2: "title.function"

}, keywords: a, contains: [{

className: "params", begin: /\(/,

end: /\)/, keywords: a, relevance: 0,

contains: [i, e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE, ce, e.C\_BLOCK\_COMMENT\_MODE]

}, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE]

}, ce, i]

}

}, grmr\_javascript: Ee,

grmr\_json: e => ({

name: "JSON", contains: [{

className: "attr",

begin: /"(\\.|[^\\"\r\n])\*"(?=\s\*:)/, relevance: 1.01

}, {

match: /[{}[\],:]/,

className: "punctuation", relevance: 0

}, e.QUOTE\_STRING\_MODE, {

beginKeywords: "true false null"

}, e.C\_NUMBER\_MODE, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE], illegal: "\\S"

}),

grmr\_kotlin: e => {

const n = {

keyword: "abstract as val var vararg get set class object open private protected public noinline crossinline dynamic final enum if else do while for when throw try catch finally import package is in fun override companion reified inline lateinit init interface annotation data sealed internal infix operator out by constructor super tailrec where const inner suspend typealias external expect actual",

built\_in: "Byte Short Char Int Long Boolean Float Double Void Unit Nothing",

literal: "true false null"

}, t = {

className: "symbol", begin: e.UNDERSCORE\_IDENT\_RE + "@"

}, a = { className: "subst", begin: /\$\{/, end: /\}/, contains: [e.C\_NUMBER\_MODE] }, i = {

className: "variable", begin: "\\$" + e.UNDERSCORE\_IDENT\_RE

}, s = {

className: "string",

variants: [{ begin: '"""', end: '"""(?=[^"])', contains: [i, a] }, {

begin: "'", end: "'",

illegal: /\n/, contains: [e.BACKSLASH\_ESCAPE]

}, {

begin: '"', end: '"', illegal: /\n/,

contains: [e.BACKSLASH\_ESCAPE, i, a]

}]

}; a.contains.push(s); const r = {

className: "meta",

begin: "@(?:file|property|field|get|set|receiver|param|setparam|delegate)\\s\*:(?:\\s\*" + e.UNDERSCORE\_IDENT\_RE + ")?"

}, o = {

className: "meta", begin: "@" + e.UNDERSCORE\_IDENT\_RE, contains: [{

begin: /\(/,

end: /\)/, contains: [e.inherit(s, { className: "string" })]

}]

}, l = ce, c = e.COMMENT("/\\\*", "\\\*/", { contains: [e.C\_BLOCK\_COMMENT\_MODE] }), d = {

variants: [{ className: "type", begin: e.UNDERSCORE\_IDENT\_RE }, {

begin: /\(/, end: /\)/,

contains: []

}]

}, g = d; return g.variants[1].contains = [d], d.variants[1].contains = [g],

{

name: "Kotlin", aliases: ["kt", "kts"], keywords: n,

contains: [e.COMMENT("/\\\*\\\*", "\\\*/", {

relevance: 0, contains: [{

className: "doctag",

begin: "@[A-Za-z]+"

}]

}), e.C\_LINE\_COMMENT\_MODE, c, {

className: "keyword",

begin: /\b(break|continue|return|this)\b/, starts: {

contains: [{

className: "symbol",

begin: /@\w+/

}]

}

}, t, r, o, {

className: "function", beginKeywords: "fun", end: "[(]|$",

returnBegin: !0, excludeEnd: !0, keywords: n, relevance: 5, contains: [{

begin: e.UNDERSCORE\_IDENT\_RE + "\\s\*\\(", returnBegin: !0, relevance: 0,

contains: [e.UNDERSCORE\_TITLE\_MODE]

}, {

className: "type", begin: /</, end: />/,

keywords: "reified", relevance: 0

}, {

className: "params", begin: /\(/, end: /\)/,

endsParent: !0, keywords: n, relevance: 0, contains: [{

begin: /:/, end: /[=,\/]/,

endsWithParent: !0, contains: [d, e.C\_LINE\_COMMENT\_MODE, c], relevance: 0

}, e.C\_LINE\_COMMENT\_MODE, c, r, o, s, e.C\_NUMBER\_MODE]

}, c]

}, {

className: "class",

beginKeywords: "class interface trait", end: /[:\{(]|$/, excludeEnd: !0,

illegal: "extends implements", contains: [{

beginKeywords: "public protected internal private constructor"

}, e.UNDERSCORE\_TITLE\_MODE, {

className: "type", begin: /</, end: />/, excludeBegin: !0,

excludeEnd: !0, relevance: 0

}, {

className: "type", begin: /[,:]\s\*/, end: /[<\(,]|$/,

excludeBegin: !0, returnEnd: !0

}, r, o]

}, s, {

className: "meta", begin: "^#!/usr/bin/env",

end: "$", illegal: "\n"

}, l]

}

}, grmr\_less: e => {

const n = ee(e), t = re, a = "([\\w-]+|@\\{[\\w-]+\\})", i = [], s = [], r = e => ({

className: "string", begin: "~?" + e + ".\*?" + e

}), o = (e, n, t) => ({

className: e, begin: n,

relevance: t

}), l = {

$pattern: /[a-z-]+/, keyword: "and or not only",

attribute: te.join(" ")

}, c = {

begin: "\\(", end: "\\)", contains: s, keywords: l,

relevance: 0

}

; s.push(e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE, r("'"), r('"'), n.CSS\_NUMBER\_MODE, {

begin: "(url|data-uri)\\(", starts: {

className: "string", end: "[\\)\\n]",

excludeEnd: !0

}

}, n.HEXCOLOR, c, o("variable", "@@?[\\w-]+", 10), o("variable", "@\\{[\\w-]+\\}"), o("built\_in", "~?`[^`]\*?`"), {

className: "attribute", begin: "[\\w-]+\\s\*:", end: ":", returnBegin: !0, excludeEnd: !0

}, n.IMPORTANT); const d = s.concat({ begin: /\{/, end: /\}/, contains: i }), g = {

beginKeywords: "when", endsWithParent: !0, contains: [{

beginKeywords: "and not"

}].concat(s)

}, u = {

begin: a + "\\s\*:", returnBegin: !0, end: /[;}]/, relevance: 0,

contains: [{ begin: /-(webkit|moz|ms|o)-/ }, n.CSS\_VARIABLE, {

className: "attribute",

begin: "\\b(" + se.join("|") + ")\\b", end: /(?=:)/, starts: {

endsWithParent: !0,

illegal: "[<=$]", relevance: 0, contains: s

}

}]

}, b = {

className: "keyword",

begin: "@(import|media|charset|font-face|(-[a-z]+-)?keyframes|supports|document|namespace|page|viewport|host)\\b",

starts: { end: "[;{}]", keywords: l, returnEnd: !0, contains: s, relevance: 0 }

}, m = {

className: "variable", variants: [{ begin: "@[\\w-]+\\s\*:", relevance: 15 }, {

begin: "@[\\w-]+"

}], starts: { end: "[;}]", returnEnd: !0, contains: d }

}, p = {

variants: [{

begin: "[\\.#:&\\[>]", end: "[;{}]"

}, { begin: a, end: /\{/ }], returnBegin: !0,

returnEnd: !0, illegal: "[<='$\"]", relevance: 0,

contains: [e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE, g, o("keyword", "all\\b"), o("variable", "@\\{[\\w-]+\\}"), {

begin: "\\b(" + ne.join("|") + ")\\b", className: "selector-tag"

}, o("selector-tag", a + "%?", 0), o("selector-id", "#" + a), o("selector-class", "\\." + a, 0), o("selector-tag", "&", 0), n.ATTRIBUTE\_SELECTOR\_MODE, {

className: "selector-pseudo", begin: ":(" + ae.join("|") + ")"

}, {

className: "selector-pseudo", begin: "::(" + ie.join("|") + ")"

}, {

begin: /\(/, end: /\)/,

relevance: 0, contains: d

}, { begin: "!important" }]

}, \_ = {

begin: `[\\w-]+:(:)?(${t.join("|")})`, returnBegin: !0, contains: [p]

}

; return i.push(e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE, b, m, \_, u, p), {

name: "Less", case\_insensitive: !0, illegal: "[=>'/<($\"]", contains: i

}

}, grmr\_lua: e => {

const n = "\\[=\*\\[", t = "\\]=\*\\]", a = {

begin: n, end: t, contains: ["self"]

}, i = [e.COMMENT("--(?!\\[=\*\\[)", "$"), e.COMMENT("--\\[=\*\\[", t, {

contains: [a],

relevance: 10

})]; return {

name: "Lua", keywords: {

$pattern: e.UNDERSCORE\_IDENT\_RE,

literal: "true false nil",

keyword: "and break do else elseif end for goto if in local not or repeat return then until while",

built\_in: "\_G \_ENV \_VERSION \_\_index \_\_newindex \_\_mode \_\_call \_\_metatable \_\_tostring \_\_len \_\_gc \_\_add \_\_sub \_\_mul \_\_div \_\_mod \_\_pow \_\_concat \_\_unm \_\_eq \_\_lt \_\_le assert collectgarbage dofile error getfenv getmetatable ipairs load loadfile loadstring module next pairs pcall print rawequal rawget rawset require select setfenv setmetatable tonumber tostring type unpack xpcall arg self coroutine resume yield status wrap create running debug getupvalue debug sethook getmetatable gethook setmetatable setlocal traceback setfenv getinfo setupvalue getlocal getregistry getfenv io lines write close flush open output type read stderr stdin input stdout popen tmpfile math log max acos huge ldexp pi cos tanh pow deg tan cosh sinh random randomseed frexp ceil floor rad abs sqrt modf asin min mod fmod log10 atan2 exp sin atan os exit setlocale date getenv difftime remove time clock tmpname rename execute package preload loadlib loaded loaders cpath config path seeall string sub upper len gfind rep find match char dump gmatch reverse byte format gsub lower table setn insert getn foreachi maxn foreach concat sort remove"

}, contains: i.concat([{

className: "function", beginKeywords: "function", end: "\\)",

contains: [e.inherit(e.TITLE\_MODE, {

begin: "([\_a-zA-Z]\\w\*\\.)\*([\_a-zA-Z]\\w\*:)?[\_a-zA-Z]\\w\*"

}), {

className: "params",

begin: "\\(", endsWithParent: !0, contains: i

}].concat(i)

}, e.C\_NUMBER\_MODE, e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE, {

className: "string",

begin: n, end: t, contains: [a], relevance: 5

}])

}

}, grmr\_makefile: e => {

const n = {

className: "variable", variants: [{

begin: "\\$\\(" + e.UNDERSCORE\_IDENT\_RE + "\\)",

contains: [e.BACKSLASH\_ESCAPE]

}, { begin: /\$[@%<?\^\+\\*]/ }]

}, t = {

className: "string",

begin: /"/, end: /"/, contains: [e.BACKSLASH\_ESCAPE, n]

}, a = {

className: "variable",

begin: /\$\([\w-]+\s/, end: /\)/, keywords: {

built\_in: "subst patsubst strip findstring filter filter-out sort word wordlist firstword lastword dir notdir suffix basename addsuffix addprefix join wildcard realpath abspath error warning shell origin flavor foreach if or and call eval file value"

}, contains: [n]

}, i = { begin: "^" + e.UNDERSCORE\_IDENT\_RE + "\\s\*(?=[:+?]?=)" }, s = {

className: "section", begin: /^[^\s]+:/, end: /$/, contains: [n]

}; return {

name: "Makefile", aliases: ["mk", "mak", "make"], keywords: {

$pattern: /[\w-]+/,

keyword: "define endef undefine ifdef ifndef ifeq ifneq else endif include -include sinclude override export unexport private vpath"

}, contains: [e.HASH\_COMMENT\_MODE, n, t, a, i, {

className: "meta", begin: /^\.PHONY:/,

end: /$/, keywords: { $pattern: /[\.\w]+/, keyword: ".PHONY" }

}, s]

}

}, grmr\_xml: e => {

const n = b(/[A-Z\_]/, u(/[A-Z0-9\_.-]\*:/), /[A-Z0-9\_.-]\*/), t = {

className: "symbol",

begin: /&[a-z]+;|&#[0-9]+;|&#x[a-f0-9]+;/

}, a = {

begin: /\s/, contains: [{

className: "keyword", begin: /#?[a-z\_][a-z1-9\_-]+/, illegal: /\n/

}]

}, i = e.inherit(a, {

begin: /\(/, end: /\)/

}), s = e.inherit(e.APOS\_STRING\_MODE, {

className: "string"

}), r = e.inherit(e.QUOTE\_STRING\_MODE, { className: "string" }), o = {

endsWithParent: !0,

illegal: /</, relevance: 0, contains: [{

className: "attr", begin: /[A-Za-z0-9.\_:-]+/,

relevance: 0

}, {

begin: /=\s\*/, relevance: 0, contains: [{

className: "string",

endsParent: !0, variants: [{ begin: /"/, end: /"/, contains: [t] }, {

begin: /'/, end: /'/,

contains: [t]

}, { begin: /[^\s"'=<>`]+/ }]

}]

}]

}; return {

name: "HTML, XML",

aliases: ["html", "xhtml", "rss", "atom", "xjb", "xsd", "xsl", "plist", "wsf", "svg"],

case\_insensitive: !0, contains: [{

className: "meta", begin: /<![a-z]/, end: />/,

relevance: 10, contains: [a, r, s, i, {

begin: /\[/, end: /\]/, contains: [{

className: "meta",

begin: /<![a-z]/, end: />/, contains: [a, i, r, s]

}]

}]

}, e.COMMENT(/<!--/, /-->/, {

relevance: 10

}), { begin: /<!\[CDATA\[/, end: /\]\]>/, relevance: 10 }, t, {

className: "meta", begin: /<\?xml/, end: /\?>/, relevance: 10

}, {

className: "tag",

begin: /<style(?=\s|>)/, end: />/, keywords: { name: "style" }, contains: [o], starts: {

end: /<\/style>/, returnEnd: !0, subLanguage: ["css", "xml"]

}

}, {

className: "tag",

begin: /<script(?=\s|>)/, end: />/, keywords: { name: "script" }, contains: [o], starts: {

end: /<\/script>/, returnEnd: !0, subLanguage: ["javascript", "handlebars", "xml"]

}

}, {

className: "tag", begin: /<>|<\/>/

}, {

className: "tag",

begin: b(/</, g(b(n, m(/\/>/, />/, /\s/)))), end: /\/?>/, contains: [{

className: "name",

begin: n, relevance: 0, starts: o

}]

}, {

className: "tag", begin: b(/<\//, g(b(n, />/))),

contains: [{ className: "name", begin: n, relevance: 0 }, {

begin: />/, relevance: 0,

endsParent: !0

}]

}]

}

}, grmr\_markdown: e => {

const n = {

begin: /<\/?[A-Za-z\_]/, end: ">",

subLanguage: "xml", relevance: 0

}, t = {

variants: [{

begin: /\[.+?\]\[.\*?\]/, relevance: 0

}, {

begin: /\[.+?\]\(((data|javascript|mailto):|(?:http|ftp)s?:\/\/).\*?\)/,

relevance: 2

}, {

begin: b(/\[.+?\]\(/, /[A-Za-z][A-Za-z0-9+.-]\*/, /:\/\/.\*?\)/),

relevance: 2

}, { begin: /\[.+?\]\([./?&#].\*?\)/, relevance: 1 }, {

begin: /\[.\*?\]\(.\*?\)/, relevance: 0

}], returnBegin: !0, contains: [{

match: /\[(?=\])/

}, {

className: "string", relevance: 0, begin: "\\[", end: "\\]", excludeBegin: !0,

returnEnd: !0

}, {

className: "link", relevance: 0, begin: "\\]\\(", end: "\\)",

excludeBegin: !0, excludeEnd: !0

}, {

className: "symbol", relevance: 0, begin: "\\]\\[",

end: "\\]", excludeBegin: !0, excludeEnd: !0

}]

}, a = {

className: "strong", contains: [],

variants: [{ begin: /\_{2}/, end: /\_{2}/ }, { begin: /\\*{2}/, end: /\\*{2}/ }]

}, i = {

className: "emphasis", contains: [], variants: [{ begin: /\\*(?!\\*)/, end: /\\*/ }, {

begin: /\_(?!\_)/, end: /\_/, relevance: 0

}]

}; a.contains.push(i), i.contains.push(a)

; let s = [n, t]

; return a.contains = a.contains.concat(s), i.contains = i.contains.concat(s),

s = s.concat(a, i), {

name: "Markdown", aliases: ["md", "mkdown", "mkd"], contains: [{

className: "section", variants: [{ begin: "^#{1,6}", end: "$", contains: s }, {

begin: "(?=^.+?\\n[=-]{2,}$)", contains: [{ begin: "^[=-]\*$" }, {

begin: "^", end: "\\n",

contains: s

}]

}]

}, n, {

className: "bullet", begin: "^[ \t]\*([\*+-]|(\\d+\\.))(?=\\s+)",

end: "\\s+", excludeEnd: !0

}, a, i, {

className: "quote", begin: "^>\\s+", contains: s,

end: "$"

}, {

className: "code", variants: [{ begin: "(`{3,})[^`](.|\\n)\*?\\1`\*[ ]\*" }, {

begin: "(~{3,})[^~](.|\\n)\*?\\1~\*[ ]\*"

}, { begin: "```", end: "```+[ ]\*$" }, {

begin: "~~~", end: "~~~+[ ]\*$"

}, { begin: "`.+?`" }, {

begin: "(?=^( {4}|\\t))",

contains: [{ begin: "^( {4}|\\t)", end: "(\\n)$" }], relevance: 0

}]

}, {

begin: "^[-\\\*]{3,}", end: "$"

}, t, {

begin: /^\[[^\n]+\]:/, returnBegin: !0, contains: [{

className: "symbol", begin: /\[/, end: /\]/, excludeBegin: !0, excludeEnd: !0

}, {

className: "link", begin: /:\s\*/, end: /$/, excludeBegin: !0

}]

}]

}

}, grmr\_objectivec: e => {

const n = /[a-zA-Z@][a-zA-Z0-9\_]\*/, t = {

$pattern: n,

keyword: ["@interface", "@class", "@protocol", "@implementation"]

}; return {

name: "Objective-C", aliases: ["mm", "objc", "obj-c", "obj-c++", "objective-c++"],

keywords: {

$pattern: n,

keyword: ["int", "float", "while", "char", "export", "sizeof", "typedef", "const", "struct", "for", "union", "unsigned", "long", "volatile", "static", "bool", "mutable", "if", "do", "return", "goto", "void", "enum", "else", "break", "extern", "asm", "case", "short", "default", "double", "register", "explicit", "signed", "typename", "this", "switch", "continue", "wchar\_t", "inline", "readonly", "assign", "readwrite", "self", "@synchronized", "id", "typeof", "nonatomic", "super", "unichar", "IBOutlet", "IBAction", "strong", "weak", "copy", "in", "out", "inout", "bycopy", "byref", "oneway", "\_\_strong", "\_\_weak", "\_\_block", "\_\_autoreleasing", "@private", "@protected", "@public", "@try", "@property", "@end", "@throw", "@catch", "@finally", "@autoreleasepool", "@synthesize", "@dynamic", "@selector", "@optional", "@required", "@encode", "@package", "@import", "@defs", "@compatibility\_alias", "\_\_bridge", "\_\_bridge\_transfer", "\_\_bridge\_retained", "\_\_bridge\_retain", "\_\_covariant", "\_\_contravariant", "\_\_kindof", "\_Nonnull", "\_Nullable", "\_Null\_unspecified", "\_\_FUNCTION\_\_", "\_\_PRETTY\_FUNCTION\_\_", "\_\_attribute\_\_", "getter", "setter", "retain", "unsafe\_unretained", "nonnull", "nullable", "null\_unspecified", "null\_resettable", "class", "instancetype", "NS\_DESIGNATED\_INITIALIZER", "NS\_UNAVAILABLE", "NS\_REQUIRES\_SUPER", "NS\_RETURNS\_INNER\_POINTER", "NS\_INLINE", "NS\_AVAILABLE", "NS\_DEPRECATED", "NS\_ENUM", "NS\_OPTIONS", "NS\_SWIFT\_UNAVAILABLE", "NS\_ASSUME\_NONNULL\_BEGIN", "NS\_ASSUME\_NONNULL\_END", "NS\_REFINED\_FOR\_SWIFT", "NS\_SWIFT\_NAME", "NS\_SWIFT\_NOTHROW", "NS\_DURING", "NS\_HANDLER", "NS\_ENDHANDLER", "NS\_VALUERETURN", "NS\_VOIDRETURN"],

literal: ["false", "true", "FALSE", "TRUE", "nil", "YES", "NO", "NULL"],

built\_in: ["BOOL", "dispatch\_once\_t", "dispatch\_queue\_t", "dispatch\_sync", "dispatch\_async", "dispatch\_once"]

}, illegal: "</", contains: [{

className: "built\_in",

begin: "\\b(AV|CA|CF|CG|CI|CL|CM|CN|CT|MK|MP|MTK|MTL|NS|SCN|SK|UI|WK|XC)\\w+"

}, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE, e.C\_NUMBER\_MODE, e.QUOTE\_STRING\_MODE, e.APOS\_STRING\_MODE, {

className: "string", variants: [{

begin: '@"', end: '"', illegal: "\\n",

contains: [e.BACKSLASH\_ESCAPE]

}]

}, {

className: "meta", begin: /#\s\*[a-z]+\b/, end: /$/,

keywords: {

keyword: "if else elif endif define undef warning error line pragma ifdef ifndef include"

}, contains: [{ begin: /\\\n/, relevance: 0 }, e.inherit(e.QUOTE\_STRING\_MODE, {

className: "string"

}), {

className: "string", begin: /<.\*?>/, end: /$/, illegal: "\\n"

}, e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE]

}, {

className: "class",

begin: "(" + t.keyword.join("|") + ")\\b", end: /(\{|$)/, excludeEnd: !0, keywords: t,

contains: [e.UNDERSCORE\_TITLE\_MODE]

}, {

begin: "\\." + e.UNDERSCORE\_IDENT\_RE,

relevance: 0

}]

}

}, grmr\_perl: e => {

const n = /[dualxmsipngr]{0,12}/, t = {

$pattern: /[\w.]+/,

keyword: "abs accept alarm and atan2 bind binmode bless break caller chdir chmod chomp chop chown chr chroot close closedir connect continue cos crypt dbmclose dbmopen defined delete die do dump each else elsif endgrent endhostent endnetent endprotoent endpwent endservent eof eval exec exists exit exp fcntl fileno flock for foreach fork format formline getc getgrent getgrgid getgrnam gethostbyaddr gethostbyname gethostent getlogin getnetbyaddr getnetbyname getnetent getpeername getpgrp getpriority getprotobyname getprotobynumber getprotoent getpwent getpwnam getpwuid getservbyname getservbyport getservent getsockname getsockopt given glob gmtime goto grep gt hex if index int ioctl join keys kill last lc lcfirst length link listen local localtime log lstat lt ma map mkdir msgctl msgget msgrcv msgsnd my ne next no not oct open opendir or ord our pack package pipe pop pos print printf prototype push q|0 qq quotemeta qw qx rand read readdir readline readlink readpipe recv redo ref rename require reset return reverse rewinddir rindex rmdir say scalar seek seekdir select semctl semget semop send setgrent sethostent setnetent setpgrp setpriority setprotoent setpwent setservent setsockopt shift shmctl shmget shmread shmwrite shutdown sin sleep socket socketpair sort splice split sprintf sqrt srand stat state study sub substr symlink syscall sysopen sysread sysseek system syswrite tell telldir tie tied time times tr truncate uc ucfirst umask undef unless unlink unpack unshift untie until use utime values vec wait waitpid wantarray warn when while write x|0 xor y|0"

}, a = { className: "subst", begin: "[$@]\\{", end: "\\}", keywords: t }, i = {

begin: /->\{/,

end: /\}/

}, s = {

variants: [{ begin: /\$\d/ }, {

begin: b(/[$%@](\^\w\b|#\w+(::\w+)\*|\{\w+\}|\w+(::\w\*)\*)/, "(?![A-Za-z])(?![@$%])")

}, { begin: /[$%@][^\s\w{]/, relevance: 0 }]

}, r = [e.BACKSLASH\_ESCAPE, a, s], o = [/!/, /\//, /\|/, /\?/, /'/, /"/, /#/], l = (e, t, a = "\\1") => {

const i = "\\1" === a ? a : b(a, t)

; return b(b("(?:", e, ")"), t, /(?:\\.|[^\\\/])\*?/, i, /(?:\\.|[^\\\/])\*?/, a, n)

}, c = (e, t, a) => b(b("(?:", e, ")"), t, /(?:\\.|[^\\\/])\*?/, a, n), d = [s, e.HASH\_COMMENT\_MODE, e.COMMENT(/^=\w/, /=cut/, {

endsWithParent: !0

}), i, {

className: "string", contains: r, variants: [{

begin: "q[qwxr]?\\s\*\\(", end: "\\)", relevance: 5

}, {

begin: "q[qwxr]?\\s\*\\[",

end: "\\]", relevance: 5

}, { begin: "q[qwxr]?\\s\*\\{", end: "\\}", relevance: 5 }, {

begin: "q[qwxr]?\\s\*\\|", end: "\\|", relevance: 5

}, {

begin: "q[qwxr]?\\s\*<", end: ">",

relevance: 5

}, { begin: "qw\\s+q", end: "q", relevance: 5 }, {

begin: "'", end: "'",

contains: [e.BACKSLASH\_ESCAPE]

}, { begin: '"', end: '"' }, {

begin: "`", end: "`",

contains: [e.BACKSLASH\_ESCAPE]

}, { begin: /\{\w+\}/, relevance: 0 }, {

begin: "-?\\w+\\s\*=>", relevance: 0

}]

}, {

className: "number",

begin: "(\\b0[0-7\_]+)|(\\b0x[0-9a-fA-F\_]+)|(\\b[1-9][0-9\_]\*(\\.[0-9\_]+)?)|[0\_]\\b",

relevance: 0

}, {

begin: "(\\/\\/|" + e.RE\_STARTERS\_RE + "|\\b(split|return|print|reverse|grep)\\b)\\s\*",

keywords: "split return print reverse grep", relevance: 0,

contains: [e.HASH\_COMMENT\_MODE, {

className: "regexp", variants: [{

begin: l("s|tr|y", m(...o, { capture: !0 }))

}, { begin: l("s|tr|y", "\\(", "\\)") }, {

begin: l("s|tr|y", "\\[", "\\]")

}, { begin: l("s|tr|y", "\\{", "\\}") }], relevance: 2

}, {

className: "regexp", variants: [{ begin: /(m|qr)\/\//, relevance: 0 }, {

begin: c("(?:m|qr)?", /\//, /\//)

}, { begin: c("m|qr", m(...o, { capture: !0 }), /\1/) }, {

begin: c("m|qr", /\(/, /\)/)

}, { begin: c("m|qr", /\[/, /\]/) }, {

begin: c("m|qr", /\{/, /\}/)

}]

}]

}, {

className: "function", beginKeywords: "sub",

end: "(\\s\*\\(.\*?\\))?[;{]", excludeEnd: !0, relevance: 5, contains: [e.TITLE\_MODE]

}, {

begin: "-\\w\\b", relevance: 0

}, {

begin: "^\_\_DATA\_\_$", end: "^\_\_END\_\_$",

subLanguage: "mojolicious", contains: [{ begin: "^@@.\*", end: "$", className: "comment" }]

}]; return a.contains = d, i.contains = d, {

name: "Perl", aliases: ["pl", "pm"], keywords: t,

contains: d

}

}, grmr\_php: e => {

const n = {

className: "variable",

begin: "\\$+[a-zA-Z\_\x7f-\xff][a-zA-Z0-9\_\x7f-\xff]\*(?![A-Za-z0-9])(?![$])"

}, t = {

className: "meta", variants: [{ begin: /<\?php/, relevance: 10 }, { begin: /<\?[=]?/ }, {

begin: /\?>/

}]

}, a = {

className: "subst", variants: [{ begin: /\$\w+/ }, {

begin: /\{\$/,

end: /\}/

}]

}, i = e.inherit(e.APOS\_STRING\_MODE, {

illegal: null

}), s = e.inherit(e.QUOTE\_STRING\_MODE, {

illegal: null,

contains: e.QUOTE\_STRING\_MODE.contains.concat(a)

}), r = e.END\_SAME\_AS\_BEGIN({

begin: /<<<[ \t]\*(\w+)\n/, end: /[ \t]\*(\w+)\b/,

contains: e.QUOTE\_STRING\_MODE.contains.concat(a)

}), o = {

className: "string",

contains: [e.BACKSLASH\_ESCAPE, t], variants: [e.inherit(i, {

begin: "b'", end: "'"

}), e.inherit(s, { begin: 'b"', end: '"' }), s, i, r]

}, l = {

className: "number", variants: [{

begin: "\\b0b[01]+(?:\_[01]+)\*\\b"

}, { begin: "\\b0o[0-7]+(?:\_[0-7]+)\*\\b" }, {

begin: "\\b0x[\\da-f]+(?:\_[\\da-f]+)\*\\b"

}, {

begin: "(?:\\b\\d+(?:\_\\d+)\*(\\.(?:\\d+(?:\_\\d+)\*))?|\\B\\.\\d+)(?:e[+-]?\\d+)?"

}], relevance: 0

}, c = {

keyword: "\_\_CLASS\_\_ \_\_DIR\_\_ \_\_FILE\_\_ \_\_FUNCTION\_\_ \_\_LINE\_\_ \_\_METHOD\_\_ \_\_NAMESPACE\_\_ \_\_TRAIT\_\_ die echo exit include include\_once print require require\_once array abstract and as binary bool boolean break callable case catch class clone const continue declare default do double else elseif empty enddeclare endfor endforeach endif endswitch endwhile enum eval extends final finally float for foreach from global goto if implements instanceof insteadof int integer interface isset iterable list match|0 mixed new object or private protected public real return string switch throw trait try unset use var void while xor yield",

literal: "false null true",

built\_in: "Error|0 AppendIterator ArgumentCountError ArithmeticError ArrayIterator ArrayObject AssertionError BadFunctionCallException BadMethodCallException CachingIterator CallbackFilterIterator CompileError Countable DirectoryIterator DivisionByZeroError DomainException EmptyIterator ErrorException Exception FilesystemIterator FilterIterator GlobIterator InfiniteIterator InvalidArgumentException IteratorIterator LengthException LimitIterator LogicException MultipleIterator NoRewindIterator OutOfBoundsException OutOfRangeException OuterIterator OverflowException ParentIterator ParseError RangeException RecursiveArrayIterator RecursiveCachingIterator RecursiveCallbackFilterIterator RecursiveDirectoryIterator RecursiveFilterIterator RecursiveIterator RecursiveIteratorIterator RecursiveRegexIterator RecursiveTreeIterator RegexIterator RuntimeException SeekableIterator SplDoublyLinkedList SplFileInfo SplFileObject SplFixedArray SplHeap SplMaxHeap SplMinHeap SplObjectStorage SplObserver SplObserver SplPriorityQueue SplQueue SplStack SplSubject SplSubject SplTempFileObject TypeError UnderflowException UnexpectedValueException UnhandledMatchError ArrayAccess Closure Generator Iterator IteratorAggregate Serializable Stringable Throwable Traversable WeakReference WeakMap Directory \_\_PHP\_Incomplete\_Class parent php\_user\_filter self static stdClass"

}; return {

case\_insensitive: !0, keywords: c,

contains: [e.HASH\_COMMENT\_MODE, e.COMMENT("//", "$", {

contains: [t]

}), e.COMMENT("/\\\*", "\\\*/", {

contains: [{ className: "doctag", begin: "@[A-Za-z]+" }]

}), e.COMMENT("\_\_halt\_compiler.+?;", !1, {

endsWithParent: !0,

keywords: "\_\_halt\_compiler"

}), t, { className: "keyword", begin: /\$this\b/ }, n, {

begin: /(::|->)+[a-zA-Z\_\x7f-\xff][a-zA-Z0-9\_\x7f-\xff]\*/

}, {

className: "function",

relevance: 0, beginKeywords: "fn function", end: /[;{]/, excludeEnd: !0,

illegal: "[$%\\[]", contains: [{ beginKeywords: "use" }, e.UNDERSCORE\_TITLE\_MODE, {

begin: "=>", endsParent: !0

}, {

className: "params", begin: "\\(", end: "\\)",

excludeBegin: !0, excludeEnd: !0, keywords: c,

contains: ["self", n, e.C\_BLOCK\_COMMENT\_MODE, o, l]

}]

}, {

className: "class", variants: [{

beginKeywords: "enum", illegal: /[($"]/

}, {

beginKeywords: "class interface trait",

illegal: /[:($"]/

}], relevance: 0, end: /\{/, excludeEnd: !0, contains: [{

beginKeywords: "extends implements"

}, e.UNDERSCORE\_TITLE\_MODE]

}, {

beginKeywords: "namespace", relevance: 0, end: ";", illegal: /[.']/,

contains: [e.UNDERSCORE\_TITLE\_MODE]

}, {

beginKeywords: "use", relevance: 0, end: ";",

contains: [e.UNDERSCORE\_TITLE\_MODE]

}, o, l]

}

}, grmr\_php\_template: e => ({

name: "PHP template", subLanguage: "xml", contains: [{

begin: /<\?(php|=)?/, end: /\?>/,

subLanguage: "php", contains: [{ begin: "/\\\*", end: "\\\*/", skip: !0 }, {

begin: 'b"',

end: '"', skip: !0

}, { begin: "b'", end: "'", skip: !0 }, e.inherit(e.APOS\_STRING\_MODE, {

illegal: null, className: null, contains: null, skip: !0

}), e.inherit(e.QUOTE\_STRING\_MODE, {

illegal: null, className: null, contains: null,

skip: !0

})]

}]

}), grmr\_plaintext: e => ({

name: "Plain text", aliases: ["text", "txt"],

disableAutodetect: !0

}), grmr\_python: e => {

const n = {

$pattern: /[A-Za-z]\w+|\_\_\w+\_\_/,

keyword: ["and", "as", "assert", "async", "await", "break", "class", "continue", "def", "del", "elif", "else", "except", "finally", "for", "from", "global", "if", "import", "in", "is", "lambda", "nonlocal|10", "not", "or", "pass", "raise", "return", "try", "while", "with", "yield"],

built\_in: ["\_\_import\_\_", "abs", "all", "any", "ascii", "bin", "bool", "breakpoint", "bytearray", "bytes", "callable", "chr", "classmethod", "compile", "complex", "delattr", "dict", "dir", "divmod", "enumerate", "eval", "exec", "filter", "float", "format", "frozenset", "getattr", "globals", "hasattr", "hash", "help", "hex", "id", "input", "int", "isinstance", "issubclass", "iter", "len", "list", "locals", "map", "max", "memoryview", "min", "next", "object", "oct", "open", "ord", "pow", "print", "property", "range", "repr", "reversed", "round", "set", "setattr", "slice", "sorted", "staticmethod", "str", "sum", "super", "tuple", "type", "vars", "zip"],

literal: ["\_\_debug\_\_", "Ellipsis", "False", "None", "NotImplemented", "True"],

type: ["Any", "Callable", "Coroutine", "Dict", "List", "Literal", "Generic", "Optional", "Sequence", "Set", "Tuple", "Type", "Union"]

}, t = { className: "meta", begin: /^(>>>|\.\.\.) / }, a = {

className: "subst", begin: /\{/,

end: /\}/, keywords: n, illegal: /#/

}, i = { begin: /\{\{/, relevance: 0 }, s = {

className: "string", contains: [e.BACKSLASH\_ESCAPE], variants: [{

begin: /([uU]|[bB]|[rR]|[bB][rR]|[rR][bB])?'''/, end: /'''/,

contains: [e.BACKSLASH\_ESCAPE, t], relevance: 10

}, {

begin: /([uU]|[bB]|[rR]|[bB][rR]|[rR][bB])?"""/, end: /"""/,

contains: [e.BACKSLASH\_ESCAPE, t], relevance: 10

}, {

begin: /([fF][rR]|[rR][fF]|[fF])'''/, end: /'''/,

contains: [e.BACKSLASH\_ESCAPE, t, i, a]

}, {

begin: /([fF][rR]|[rR][fF]|[fF])"""/,

end: /"""/, contains: [e.BACKSLASH\_ESCAPE, t, i, a]

}, {

begin: /([uU]|[rR])'/, end: /'/,

relevance: 10

}, { begin: /([uU]|[rR])"/, end: /"/, relevance: 10 }, {

begin: /([bB]|[bB][rR]|[rR][bB])'/, end: /'/

}, {

begin: /([bB]|[bB][rR]|[rR][bB])"/,

end: /"/

}, {

begin: /([fF][rR]|[rR][fF]|[fF])'/, end: /'/,

contains: [e.BACKSLASH\_ESCAPE, i, a]

}, {

begin: /([fF][rR]|[rR][fF]|[fF])"/, end: /"/,

contains: [e.BACKSLASH\_ESCAPE, i, a]

}, e.APOS\_STRING\_MODE, e.QUOTE\_STRING\_MODE]

}, r = "[0-9](\_?[0-9])\*", o = `(\\b(${r}))?\\.(${r})|\\b(${r})\\.`, l = {

className: "number", relevance: 0, variants: [{

begin: `(\\b(${r})|(${o}))[eE][+-]?(${r})[jJ]?\\b`

}, { begin: `(${o})[jJ]?` }, {

begin: "\\b([1-9](\_?[0-9])\*|0+(\_?0)\*)[lLjJ]?\\b"

}, {

begin: "\\b0[bB](\_?[01])+[lL]?\\b"

}, { begin: "\\b0[oO](\_?[0-7])+[lL]?\\b" }, {

begin: "\\b0[xX](\_?[0-9a-fA-F])+[lL]?\\b"

}, { begin: `\\b(${r})[jJ]\\b` }]

}, c = {

className: "comment", begin: g(/# type:/), end: /$/, keywords: n, contains: [{

begin: /# type:/

}, { begin: /#/, end: /\b\B/, endsWithParent: !0 }]

}, d = {

className: "params", variants: [{ className: "", begin: /\(\s\*\)/, skip: !0 }, {

begin: /\(/,

end: /\)/, excludeBegin: !0, excludeEnd: !0, keywords: n,

contains: ["self", t, l, s, e.HASH\_COMMENT\_MODE]

}]

}; return a.contains = [s, l, t], {

name: "Python", aliases: ["py", "gyp", "ipython"], keywords: n,

illegal: /(<\/|->|\?)|=>/, contains: [t, l, { begin: /\bself\b/ }, {

beginKeywords: "if",

relevance: 0

}, s, c, e.HASH\_COMMENT\_MODE, {

match: [/def/, /\s+/, E], scope: {

1: "keyword",

3: "title.function"

}, contains: [d]

}, {

variants: [{

match: [/class/, /\s+/, E, /\s\*/, /\(\s\*/, E, /\s\*\)/]

}, { match: [/class/, /\s+/, E] }],

scope: { 1: "keyword", 3: "title.class", 6: "title.class.inherited" }

}, {

className: "meta", begin: /^[\t ]\*@/, end: /(?=#)|$/, contains: [l, d, s]

}]

}

},

grmr\_python\_repl: e => ({

aliases: ["pycon"], contains: [{

className: "meta", starts: {

end: / |$/, starts: { end: "$", subLanguage: "python" }

}, variants: [{

begin: /^>>>(?=[ ]|$)/

}, { begin: /^\.\.\.(?=[ ]|$)/ }]

}]

}), grmr\_r: e => {

const n = /(?:(?:[a-zA-Z]|\.[.\_a-zA-Z])[.\_a-zA-Z0-9]\*)|\.(?!\d)/, t = m(/0[xX][0-9a-fA-F]+\.[0-9a-fA-F]\*[pP][+-]?\d+i?/, /0[xX][0-9a-fA-F]+(?:[pP][+-]?\d+)?[Li]?/, /(?:\d+(?:\.\d\*)?|\.\d+)(?:[eE][+-]?\d+)?[Li]?/), a = /[=!<>:]=|\|\||&&|:::?|<-|<<-|->>|->|\|>|[-+\*\/?!$&|:<=>@^~]|\\*\\*/, i = m(/[()]/, /[{}]/, /\[\[/, /[[\]]/, /\\/, /,/)

; return {

name: "R", keywords: {

$pattern: n,

keyword: "function if in break next repeat else for while",

literal: "NULL NA TRUE FALSE Inf NaN NA\_integer\_|10 NA\_real\_|10 NA\_character\_|10 NA\_complex\_|10",

built\_in: "LETTERS letters month.abb month.name pi T F abs acos acosh all any anyNA Arg as.call as.character as.complex as.double as.environment as.integer as.logical as.null.default as.numeric as.raw asin asinh atan atanh attr attributes baseenv browser c call ceiling class Conj cos cosh cospi cummax cummin cumprod cumsum digamma dim dimnames emptyenv exp expression floor forceAndCall gamma gc.time globalenv Im interactive invisible is.array is.atomic is.call is.character is.complex is.double is.environment is.expression is.finite is.function is.infinite is.integer is.language is.list is.logical is.matrix is.na is.name is.nan is.null is.numeric is.object is.pairlist is.raw is.recursive is.single is.symbol lazyLoadDBfetch length lgamma list log max min missing Mod names nargs nzchar oldClass on.exit pos.to.env proc.time prod quote range Re rep retracemem return round seq\_along seq\_len seq.int sign signif sin sinh sinpi sqrt standardGeneric substitute sum switch tan tanh tanpi tracemem trigamma trunc unclass untracemem UseMethod xtfrm"

}, contains: [e.COMMENT(/#'/, /$/, {

contains: [{

scope: "doctag", match: /@examples/,

starts: { end: g(m(/\n^#'\s\*(?=@[a-zA-Z]+)/, /\n^(?!#')/)), endsParent: !0 }

}, {

scope: "doctag", begin: "@param", end: /$/, contains: [{

scope: "variable", variants: [{

match: n

}, { match: /`(?:\\.|[^`\\])+`/ }], endsParent: !0

}]

}, {

scope: "doctag",

match: /@[a-zA-Z]+/

}, { scope: "keyword", match: /\\[a-zA-Z]+/ }]

}), e.HASH\_COMMENT\_MODE, {

scope: "string", contains: [e.BACKSLASH\_ESCAPE],

variants: [e.END\_SAME\_AS\_BEGIN({

begin: /[rR]"(-\*)\(/, end: /\)(-\*)"/

}), e.END\_SAME\_AS\_BEGIN({

begin: /[rR]"(-\*)\{/, end: /\}(-\*)"/

}), e.END\_SAME\_AS\_BEGIN({

begin: /[rR]"(-\*)\[/, end: /\](-\*)"/

}), e.END\_SAME\_AS\_BEGIN({

begin: /[rR]'(-\*)\(/, end: /\)(-\*)'/

}), e.END\_SAME\_AS\_BEGIN({

begin: /[rR]'(-\*)\{/, end: /\}(-\*)'/

}), e.END\_SAME\_AS\_BEGIN({ begin: /[rR]'(-\*)\[/, end: /\](-\*)'/ }), {

begin: '"', end: '"',

relevance: 0

}, { begin: "'", end: "'", relevance: 0 }]

}, {

relevance: 0, variants: [{

scope: {

1: "operator", 2: "number"

}, match: [a, t]

}, {

scope: { 1: "operator", 2: "number" },

match: [/%[^%]\*%/, t]

}, { scope: { 1: "punctuation", 2: "number" }, match: [i, t] }, {

scope: {

2: "number"

}, match: [/[^a-zA-Z0-9.\_]|^/, t]

}]

}, {

scope: { 3: "operator" },

match: [n, /\s+/, /<-/, /\s+/]

}, {

scope: "operator", relevance: 0, variants: [{ match: a }, {

match: /%[^%]\*%/

}]

}, { scope: "punctuation", relevance: 0, match: i }, {

begin: "`", end: "`",

contains: [{ begin: /\\./ }]

}]

}

}, grmr\_ruby: e => {

const n = "([a-zA-Z\_]\\w\*[!?=]?|[-+~]@|<<|>>|=~|===?|<=>|[<>]=?|\\\*\\\*|[-/+%^&\*~`|]|\\[\\]=?)", t = {

keyword: "and then defined module in return redo if BEGIN retry end for self when next until do begin unless END rescue else break undef not super class case require yield alias while ensure elsif or include attr\_reader attr\_writer attr\_accessor \_\_FILE\_\_",

built\_in: "proc lambda", literal: "true false nil"

}, a = {

className: "doctag",

begin: "@[A-Za-z]+"

}, i = { begin: "#<", end: ">" }, s = [e.COMMENT("#", "$", {

contains: [a]

}), e.COMMENT("^=begin", "^=end", {

contains: [a], relevance: 10

}), e.COMMENT("^\_\_END\_\_", "\\n$")], r = {

className: "subst", begin: /#\{/, end: /\}/,

keywords: t

}, o = {

className: "string", contains: [e.BACKSLASH\_ESCAPE, r], variants: [{

begin: /'/, end: /'/

}, { begin: /"/, end: /"/ }, { begin: /`/, end: /`/ }, {

begin: /%[qQwWx]?\(/,

end: /\)/

}, { begin: /%[qQwWx]?\[/, end: /\]/ }, { begin: /%[qQwWx]?\{/, end: /\}/ }, {

begin: /%[qQwWx]?</, end: />/

}, { begin: /%[qQwWx]?\//, end: /\// }, {

begin: /%[qQwWx]?%/,

end: /%/

}, { begin: /%[qQwWx]?-/, end: /-/ }, { begin: /%[qQwWx]?\|/, end: /\|/ }, {

begin: /\B\?(\\\d{1,3})/

}, { begin: /\B\?(\\x[A-Fa-f0-9]{1,2})/ }, {

begin: /\B\?(\\u\{?[A-Fa-f0-9]{1,6}\}?)/

}, {

begin: /\B\?(\\M-\\C-|\\M-\\c|\\c\\M-|\\M-|\\C-\\M-)[\x20-\x7e]/

}, {

begin: /\B\?\\(c|C-)[\x20-\x7e]/

}, { begin: /\B\?\\?\S/ }, {

begin: b(/<<[-~]?'?/, g(/(\w+)(?=\W)[^\n]\*\n(?:[^\n]\*\n)\*?\s\*\1\b/)),

contains: [e.END\_SAME\_AS\_BEGIN({

begin: /(\w+)/, end: /(\w+)/,

contains: [e.BACKSLASH\_ESCAPE, r]

})]

}]

}, l = "[0-9](\_?[0-9])\*", c = {

className: "number",

relevance: 0, variants: [{

begin: `\\b([1-9](\_?[0-9])\*|0)(\\.(${l}))?([eE][+-]?(${l})|r)?i?\\b`

}, {

begin: "\\b0[dD][0-9](\_?[0-9])\*r?i?\\b"

}, {

begin: "\\b0[bB][0-1](\_?[0-1])\*r?i?\\b"

}, { begin: "\\b0[oO][0-7](\_?[0-7])\*r?i?\\b" }, {

begin: "\\b0[xX][0-9a-fA-F](\_?[0-9a-fA-F])\*r?i?\\b"

}, {

begin: "\\b0(\_?[0-7])+r?i?\\b"

}]

}, d = {

className: "params", begin: "\\(", end: "\\)",

endsParent: !0, keywords: t

}, u = [o, {

className: "class", beginKeywords: "class module",

end: "$|;", illegal: /=/, contains: [e.inherit(e.TITLE\_MODE, {

begin: "[A-Za-z\_]\\w\*(::\\w+)\*(\\?|!)?"

}), {

begin: "<\\s\*", contains: [{

begin: "(" + e.IDENT\_RE + "::)?" + e.IDENT\_RE, relevance: 0

}]

}].concat(s)

}, {

className: "function", begin: b(/def\s+/, g(n + "\\s\*(\\(|;|$)")), relevance: 0,

keywords: "def", end: "$|;", contains: [e.inherit(e.TITLE\_MODE, {

begin: n

}), d].concat(s)

}, { begin: e.IDENT\_RE + "::" }, {

className: "symbol",

begin: e.UNDERSCORE\_IDENT\_RE + "(!|\\?)?:", relevance: 0

}, {

className: "symbol",

begin: ":(?!\\s)", contains: [o, { begin: n }], relevance: 0

}, c, {

className: "variable",

begin: "(\\$\\W)|((\\$|@@?)(\\w+))(?=[^@$?])(?![A-Za-z])(?![@$?'])"

}, {

className: "params", begin: /\|/, end: /\|/, relevance: 0, keywords: t

}, {

begin: "(" + e.RE\_STARTERS\_RE + "|unless)\\s\*", keywords: "unless", contains: [{

className: "regexp", contains: [e.BACKSLASH\_ESCAPE, r], illegal: /\n/, variants: [{

begin: "/", end: "/[a-z]\*"

}, { begin: /%r\{/, end: /\}[a-z]\*/ }, {

begin: "%r\\(",

end: "\\)[a-z]\*"

}, { begin: "%r!", end: "![a-z]\*" }, { begin: "%r\\[", end: "\\][a-z]\*" }]

}].concat(i, s), relevance: 0

}].concat(i, s); r.contains = u, d.contains = u; const m = [{

begin: /^\s\*=>/, starts: { end: "$", contains: u }

}, {

className: "meta",

begin: "^([>?]>|[\\w#]+\\(\\w+\\):\\d+:\\d+>|(\\w+-)?\\d+\\.\\d+\\.\\d+(p\\d+)?[^\\d][^>]+>)(?=[ ])",

starts: { end: "$", contains: u }

}]; return s.unshift(i), {

name: "Ruby",

aliases: ["rb", "gemspec", "podspec", "thor", "irb"], keywords: t, illegal: /\/\\*/,

contains: [e.SHEBANG({ binary: "ruby" })].concat(m).concat(s).concat(u)

}

},

grmr\_rust: e => {

const n = {

className: "title.function.invoke", relevance: 0,

begin: b(/\b/, /(?!let\b)/, e.IDENT\_RE, g(/\s\*\(/))

}, t = "([ui](8|16|32|64|128|size)|f(32|64))?", a = ["drop ", "Copy", "Send", "Sized", "Sync", "Drop", "Fn", "FnMut", "FnOnce", "ToOwned", "Clone", "Debug", "PartialEq", "PartialOrd", "Eq", "Ord", "AsRef", "AsMut", "Into", "From", "Default", "Iterator", "Extend", "IntoIterator", "DoubleEndedIterator", "ExactSizeIterator", "SliceConcatExt", "ToString", "assert!", "assert\_eq!", "bitflags!", "bytes!", "cfg!", "col!", "concat!", "concat\_idents!", "debug\_assert!", "debug\_assert\_eq!", "env!", "panic!", "file!", "format!", "format\_args!", "include\_bin!", "include\_str!", "line!", "local\_data\_key!", "module\_path!", "option\_env!", "print!", "println!", "select!", "stringify!", "try!", "unimplemented!", "unreachable!", "vec!", "write!", "writeln!", "macro\_rules!", "assert\_ne!", "debug\_assert\_ne!"]

; return {

name: "Rust", aliases: ["rs"], keywords: {

$pattern: e.IDENT\_RE + "!?",

type: ["i8", "i16", "i32", "i64", "i128", "isize", "u8", "u16", "u32", "u64", "u128", "usize", "f32", "f64", "str", "char", "bool", "Box", "Option", "Result", "String", "Vec"],

keyword: ["abstract", "as", "async", "await", "become", "box", "break", "const", "continue", "crate", "do", "dyn", "else", "enum", "extern", "false", "final", "fn", "for", "if", "impl", "in", "let", "loop", "macro", "match", "mod", "move", "mut", "override", "priv", "pub", "ref", "return", "self", "Self", "static", "struct", "super", "trait", "true", "try", "type", "typeof", "unsafe", "unsized", "use", "virtual", "where", "while", "yield"],

literal: ["true", "false", "Some", "None", "Ok", "Err"], built\_in: a

}, illegal: "</",

contains: [e.C\_LINE\_COMMENT\_MODE, e.COMMENT("/\\\*", "\\\*/", {

contains: ["self"]

}), e.inherit(e.QUOTE\_STRING\_MODE, { begin: /b?"/, illegal: null }), {

className: "string", variants: [{ begin: /b?r(#\*)"(.|\n)\*?"\1(?!#)/ }, {

begin: /b?'\\?(x\w{2}|u\w{4}|U\w{8}|.)'/

}]

}, {

className: "symbol",

begin: /'[a-zA-Z\_][a-zA-Z0-9\_]\*/

}, {

className: "number", variants: [{

begin: "\\b0b([01\_]+)" + t

}, { begin: "\\b0o([0-7\_]+)" + t }, {

begin: "\\b0x([A-Fa-f0-9\_]+)" + t

}, {

begin: "\\b(\\d[\\d\_]\*(\\.[0-9\_]+)?([eE][+-]?[0-9\_]+)?)" + t

}], relevance: 0

}, {

begin: [/fn/, /\s+/, e.UNDERSCORE\_IDENT\_RE], className: {

1: "keyword",

3: "title.function"

}

}, {

className: "meta", begin: "#!?\\[", end: "\\]", contains: [{

className: "string", begin: /"/, end: /"/

}]

}, {

begin: [/let/, /\s+/, /(?:mut\s+)?/, e.UNDERSCORE\_IDENT\_RE], className: {

1: "keyword",

3: "keyword", 4: "variable"

}

}, {

begin: [/for/, /\s+/, e.UNDERSCORE\_IDENT\_RE, /\s+/, /in/], className: {

1: "keyword",

3: "variable", 5: "keyword"

}

}, {

begin: [/type/, /\s+/, e.UNDERSCORE\_IDENT\_RE],

className: { 1: "keyword", 3: "title.class" }

}, {

begin: [/(?:trait|enum|struct|union|impl|for)/, /\s+/, e.UNDERSCORE\_IDENT\_RE],

className: { 1: "keyword", 3: "title.class" }

}, {

begin: e.IDENT\_RE + "::", keywords: {

keyword: "Self", built\_in: a

}

}, { className: "punctuation", begin: "->" }, n]

}

},

grmr\_scss: e => {

const n = ee(e), t = ie, a = ae, i = "@[a-z-]+", s = {

className: "variable",

begin: "(\\$[a-zA-Z-][a-zA-Z0-9\_-]\*)\\b"

}; return {

name: "SCSS", case\_insensitive: !0,

illegal: "[=/|']", contains: [e.C\_LINE\_COMMENT\_MODE, e.C\_BLOCK\_COMMENT\_MODE, {

className: "selector-id", begin: "#[A-Za-z0-9\_-]+", relevance: 0

}, {

className: "selector-class", begin: "\\.[A-Za-z0-9\_-]+", relevance: 0

}, n.ATTRIBUTE\_SELECTOR\_MODE, {

className: "selector-tag",

begin: "\\b(" + ne.join("|") + ")\\b", relevance: 0

}, {

className: "selector-pseudo",

begin: ":(" + a.join("|") + ")"

}, {

className: "selector-pseudo",

begin: "::(" + t.join("|") + ")"

}, s, {

begin: /\(/, end: /\)/, contains: [n.CSS\_NUMBER\_MODE]

}, n.CSS\_VARIABLE, { className: "attribute", begin: "\\b(" + se.join("|") + ")\\b" }, {

begin: "\\b(whitespace|wait|w-resize|visible|vertical-text|vertical-ideographic|uppercase|upper-roman|upper-alpha|underline|transparent|top|thin|thick|text|text-top|text-bottom|tb-rl|table-header-group|table-footer-group|sw-resize|super|strict|static|square|solid|small-caps|separate|se-resize|scroll|s-resize|rtl|row-resize|ridge|right|repeat|repeat-y|repeat-x|relative|progress|pointer|overline|outside|outset|oblique|nowrap|not-allowed|normal|none|nw-resize|no-repeat|no-drop|newspaper|ne-resize|n-resize|move|middle|medium|ltr|lr-tb|lowercase|lower-roman|lower-alpha|loose|list-item|line|line-through|line-edge|lighter|left|keep-all|justify|italic|inter-word|inter-ideograph|inside|inset|inline|inline-block|inherit|inactive|ideograph-space|ideograph-parenthesis|ideograph-numeric|ideograph-alpha|horizontal|hidden|help|hand|groove|fixed|ellipsis|e-resize|double|dotted|distribute|distribute-space|distribute-letter|distribute-all-lines|disc|disabled|default|decimal|dashed|crosshair|collapse|col-resize|circle|char|center|capitalize|break-word|break-all|bottom|both|bolder|bold|block|bidi-override|below|baseline|auto|always|all-scroll|absolute|table|table-cell)\\b"

}, {

begin: ":", end: ";",

contains: [s, n.HEXCOLOR, n.CSS\_NUMBER\_MODE, e.QUOTE\_STRING\_MODE, e.APOS\_STRING\_MODE, n.IMPORTANT]

}, { begin: "@(page|font-face)", keywords: { $pattern: i, keyword: "@page @font-face" } }, {

begin: "@", end: "[{;]", returnBegin: !0, keywords: {

$pattern: /[a-z-]+/,

keyword: "and or not only", attribute: te.join(" ")

}, contains: [{

begin: i,

className: "keyword"

}, {

begin: /[a-z-]+(?=:)/, className: "attribute"

}, s, e.QUOTE\_STRING\_MODE, e.APOS\_STRING\_MODE, n.HEXCOLOR, n.CSS\_NUMBER\_MODE]

}]

}

},

grmr\_shell: e => ({

name: "Shell Session", aliases: ["console", "shellsession"],

contains: [{

className: "meta", begin: /^\s{0,3}[/~\w\d[\]()@-]\*[>%$#][ ]?/, starts: {

end: /[^\\](?=\s\*$)/, subLanguage: "bash"

}

}]

}), grmr\_sql: e => {

const n = e.COMMENT("--", "$"), t = ["true", "false", "unknown"], a = ["bigint", "binary", "blob", "boolean", "char", "character", "clob", "date", "dec", "decfloat", "decimal", "float", "int", "integer", "interval", "nchar", "nclob", "national", "numeric", "real", "row", "smallint", "time", "timestamp", "varchar", "varying", "varbinary"], i = ["abs", "acos", "array\_agg", "asin", "atan", "avg", "cast", "ceil", "ceiling", "coalesce", "corr", "cos", "cosh", "count", "covar\_pop", "covar\_samp", "cume\_dist", "dense\_rank", "deref", "element", "exp", "extract", "first\_value", "floor", "json\_array", "json\_arrayagg", "json\_exists", "json\_object", "json\_objectagg", "json\_query", "json\_table", "json\_table\_primitive", "json\_value", "lag", "last\_value", "lead", "listagg", "ln", "log", "log10", "lower", "max", "min", "mod", "nth\_value", "ntile", "nullif", "percent\_rank", "percentile\_cont", "percentile\_disc", "position", "position\_regex", "power", "rank", "regr\_avgx", "regr\_avgy", "regr\_count", "regr\_intercept", "regr\_r2", "regr\_slope", "regr\_sxx", "regr\_sxy", "regr\_syy", "row\_number", "sin", "sinh", "sqrt", "stddev\_pop", "stddev\_samp", "substring", "substring\_regex", "sum", "tan", "tanh", "translate", "translate\_regex", "treat", "trim", "trim\_array", "unnest", "upper", "value\_of", "var\_pop", "var\_samp", "width\_bucket"], s = ["create table", "insert into", "primary key", "foreign key", "not null", "alter table", "add constraint", "grouping sets", "on overflow", "character set", "respect nulls", "ignore nulls", "nulls first", "nulls last", "depth first", "breadth first"], r = i, o = ["abs", "acos", "all", "allocate", "alter", "and", "any", "are", "array", "array\_agg", "array\_max\_cardinality", "as", "asensitive", "asin", "asymmetric", "at", "atan", "atomic", "authorization", "avg", "begin", "begin\_frame", "begin\_partition", "between", "bigint", "binary", "blob", "boolean", "both", "by", "call", "called", "cardinality", "cascaded", "case", "cast", "ceil", "ceiling", "char", "char\_length", "character", "character\_length", "check", "classifier", "clob", "close", "coalesce", "collate", "collect", "column", "commit", "condition", "connect", "constraint", "contains", "convert", "copy", "corr", "corresponding", "cos", "cosh", "count", "covar\_pop", "covar\_samp", "create", "cross", "cube", "cume\_dist", "current", "current\_catalog", "current\_date", "current\_default\_transform\_group", "current\_path", "current\_role", "current\_row", "current\_schema", "current\_time", "current\_timestamp", "current\_path", "current\_role", "current\_transform\_group\_for\_type", "current\_user", "cursor", "cycle", "date", "day", "deallocate", "dec", "decimal", "decfloat", "declare", "default", "define", "delete", "dense\_rank", "deref", "describe", "deterministic", "disconnect", "distinct", "double", "drop", "dynamic", "each", "element", "else", "empty", "end", "end\_frame", "end\_partition", "end-exec", "equals", "escape", "every", "except", "exec", "execute", "exists", "exp", "external", "extract", "false", "fetch", "filter", "first\_value", "float", "floor", "for", "foreign", "frame\_row", "free", "from", "full", "function", "fusion", "get", "global", "grant", "group", "grouping", "groups", "having", "hold", "hour", "identity", "in", "indicator", "initial", "inner", "inout", "insensitive", "insert", "int", "integer", "intersect", "intersection", "interval", "into", "is", "join", "json\_array", "json\_arrayagg", "json\_exists", "json\_object", "json\_objectagg", "json\_query", "json\_table", "json\_table\_primitive", "json\_value", "lag", "language", "large", "last\_value", "lateral", "lead", "leading", "left", "like", "like\_regex", "listagg", "ln", "local", "localtime", "localtimestamp", "log", "log10", "lower", "match", "match\_number", "match\_recognize", "matches", "max", "member", "merge", "method", "min", "minute", "mod", "modifies", "module", "month", "multiset", "national", "natural", "nchar", "nclob", "new", "no", "none", "normalize", "not", "nth\_value", "ntile", "null", "nullif", "numeric", "octet\_length", "occurrences\_regex", "of", "offset", "old", "omit", "on", "one", "only", "open", "or", "order", "out", "outer", "over", "overlaps", "overlay", "parameter", "partition", "pattern", "per", "percent", "percent\_rank", "percentile\_cont", "percentile\_disc", "period", "portion", "position", "position\_regex", "power", "precedes", "precision", "prepare", "primary", "procedure", "ptf", "range", "rank", "reads", "real", "recursive", "ref", "references", "referencing", "regr\_avgx", "regr\_avgy", "regr\_count", "regr\_intercept", "regr\_r2", "regr\_slope", "regr\_sxx", "regr\_sxy", "regr\_syy", "release", "result", "return", "returns", "revoke", "right", "rollback", "rollup", "row", "row\_number", "rows", "running", "savepoint", "scope", "scroll", "search", "second", "seek", "select", "sensitive", "session\_user", "set", "show", "similar", "sin", "sinh", "skip", "smallint", "some", "specific", "specifictype", "sql", "sqlexception", "sqlstate", "sqlwarning", "sqrt", "start", "static", "stddev\_pop", "stddev\_samp", "submultiset", "subset", "substring", "substring\_regex", "succeeds", "sum", "symmetric", "system", "system\_time", "system\_user", "table", "tablesample", "tan", "tanh", "then", "time", "timestamp", "timezone\_hour", "timezone\_minute", "to", "trailing", "translate", "translate\_regex", "translation", "treat", "trigger", "trim", "trim\_array", "true", "truncate", "uescape", "union", "unique", "unknown", "unnest", "update", "upper", "user", "using", "value", "values", "value\_of", "var\_pop", "var\_samp", "varbinary", "varchar", "varying", "versioning", "when", "whenever", "where", "width\_bucket", "window", "with", "within", "without", "year", "add", "asc", "collation", "desc", "final", "first", "last", "view"].filter((e => !i.includes(e))), l = {

begin: b(/\b/, m(...r), /\s\*\(/), relevance: 0, keywords: { built\_in: r }

}; return {

name: "SQL", case\_insensitive: !0, illegal: /[{}]|<\//, keywords: {

$pattern: /\b[\w\.]+/, keyword: ((e, { exceptions: n, when: t } = {}) => {

const a = t

; return n = n || [], e.map((e => e.match(/\|\d+$/) || n.includes(e) ? e : a(e) ? e + "|0" : e))

})(o, { when: e => e.length < 3 }), literal: t, type: a,

built\_in: ["current\_catalog", "current\_date", "current\_default\_transform\_group", "current\_path", "current\_role", "current\_schema", "current\_transform\_group\_for\_type", "current\_user", "session\_user", "system\_time", "system\_user", "current\_time", "localtime", "current\_timestamp", "localtimestamp"]

}, contains: [{

begin: m(...s), relevance: 0, keywords: {

$pattern: /[\w\.]+/,

keyword: o.concat(s), literal: t, type: a

}

}, {

className: "type",

begin: m("double precision", "large object", "with timezone", "without timezone")

}, l, { className: "variable", begin: /@[a-z0-9]+/ }, {

className: "string", variants: [{

begin: /'/, end: /'/, contains: [{ begin: /''/ }]

}]

}, {

begin: /"/, end: /"/, contains: [{

begin: /""/

}]

}, e.C\_NUMBER\_MODE, e.C\_BLOCK\_COMMENT\_MODE, n, {

className: "operator",

begin: /[-+\*/=%^~]|&&?|\|\|?|!=?|<(?:=>?|<|>)?|>[>=]?/, relevance: 0

}]

}

},

grmr\_swift: e => {

const n = { match: /\s+/, relevance: 0 }, t = e.COMMENT("/\\\*", "\\\*/", {

contains: ["self"]

}), a = [e.C\_LINE\_COMMENT\_MODE, t], i = {

match: [/\./, m(...Ne, ...we)],

className: { 2: "keyword" }

}, s = {

match: b(/\./, m(...Oe)), relevance: 0

}, r = Oe.filter((e => "string" == typeof e)).concat(["\_|0"]), o = {

variants: [{

className: "keyword",

match: m(...Oe.filter((e => "string" != typeof e)).concat(ve).map(ye), ...we)

}]

}, l = {

$pattern: m(/\b\w+/, /#\w+/), keyword: r.concat(Se), literal: Me

}, c = [i, s, o], d = [{

match: b(/\./, m(...ke)), relevance: 0

}, {

className: "built\_in",

match: b(/\b/, m(...ke), /(?=\()/)

}], u = { match: /->/, relevance: 0 }, p = [u, {

className: "operator", relevance: 0, variants: [{ match: Te }, { match: `\\.(\\.|${Ce})+` }]

}], \_ = "([0-9a-fA-F]\_\*)+", h = {

className: "number", relevance: 0, variants: [{

match: "\\b(([0-9]\_\*)+)(\\.(([0-9]\_\*)+))?([eE][+-]?(([0-9]\_\*)+))?\\b"

}, {

match: `\\b0x(${\_})(\\.(${\_}))?([pP][+-]?(([0-9]\_\*)+))?\\b`

}, {

match: /\b0o([0-7]\_\*)+\b/

}, { match: /\b0b([01]\_\*)+\b/ }]

}, f = (e = "") => ({

className: "subst", variants: [{ match: b(/\\/, e, /[0\\tnr"']/) }, {

match: b(/\\/, e, /u\{[0-9a-fA-F]{1,8}\}/)

}]

}), E = (e = "") => ({

className: "subst",

match: b(/\\/, e, /[\t ]\*(?:[\r\n]|\r\n)/)

}), y = (e = "") => ({

className: "subst",

label: "interpol", begin: b(/\\/, e, /\(/), end: /\)/

}), N = (e = "") => ({

begin: b(e, /"""/),

end: b(/"""/, e), contains: [f(e), E(e), y(e)]

}), w = (e = "") => ({

begin: b(e, /"/),

end: b(/"/, e), contains: [f(e), y(e)]

}), v = {

className: "string",

variants: [N(), N("#"), N("##"), N("###"), w(), w("#"), w("##"), w("###")]

}, O = {

match: b(/`/, Ie, /`/)

}, M = [O, { className: "variable", match: /\$\d+/ }, {

className: "variable", match: `\\$${De}+`

}], x = [{

match: /(@|#)available/,

className: "keyword", starts: {

contains: [{

begin: /\(/, end: /\)/, keywords: $e,

contains: [...p, h, v]

}]

}

}, { className: "keyword", match: b(/@/, m(...Be)) }, {

className: "meta", match: b(/@/, Ie)

}], S = {

match: g(/\b[A-Z]/), relevance: 0, contains: [{

className: "type",

match: b(/(AV|CA|CF|CG|CI|CL|CM|CN|CT|MK|MP|MTK|MTL|NS|SCN|SK|UI|WK|XC)/, De, "+")

}, { className: "type", match: Le, relevance: 0 }, { match: /[?!]+/, relevance: 0 }, {

match: /\.\.\./, relevance: 0

}, { match: b(/\s+&\s+/, g(Le)), relevance: 0 }]

}, k = {

begin: /</, end: />/, keywords: l, contains: [...a, ...c, ...x, u, S]

}; S.contains.push(k)

; const A = {

begin: /\(/, end: /\)/, relevance: 0, keywords: l, contains: ["self", {

match: b(Ie, /\s\*:/), keywords: "\_|0", relevance: 0

}, ...a, ...c, ...d, ...p, h, v, ...M, ...x, S]

}, C = {

begin: /</, end: />/, contains: [...a, S]

}, T = {

begin: /\(/, end: /\)/, keywords: l, contains: [{

begin: m(g(b(Ie, /\s\*:/)), g(b(Ie, /\s+/, Ie, /\s\*:/))), end: /:/, relevance: 0,

contains: [{ className: "keyword", match: /\b\_\b/ }, { className: "params", match: Ie }]

}, ...a, ...c, ...p, h, v, ...x, S, A], endsParent: !0, illegal: /["']/

}, R = {

match: [/func/, /\s+/, m(O.match, Ie, Te)], className: {

1: "keyword", 3: "title.function"

}, contains: [C, T, n], illegal: [/\[/, /%/]

}, D = {

match: [/\b(?:subscript|init[?!]?)/, /\s\*(?=[<(])/], className: { 1: "keyword" },

contains: [C, T, n], illegal: /\[|%/

}, I = {

match: [/operator/, /\s+/, Te], className: {

1: "keyword", 3: "title"

}

}, L = {

begin: [/precedencegroup/, /\s+/, Le], className: {

1: "keyword", 3: "title"

}, contains: [S], keywords: [...xe, ...Me], end: /}/

}

; for (const e of v.variants) {

const n = e.contains.find((e => "interpol" === e.label))

; n.keywords = l; const t = [...c, ...d, ...p, h, v, ...M]; n.contains = [...t, {

begin: /\(/,

end: /\)/, contains: ["self", ...t]

}]

} return {

name: "Swift", keywords: l,

contains: [...a, R, D, {

beginKeywords: "struct protocol class extension enum actor",

end: "\\{", excludeEnd: !0, keywords: l, contains: [e.inherit(e.TITLE\_MODE, {

className: "title.class", begin: /[A-Za-z$\_][\u00C0-\u02B80-9A-Za-z$\_]\*/

}), ...c]

}, I, L, {

beginKeywords: "import", end: /$/, contains: [...a], relevance: 0

}, ...c, ...d, ...p, h, v, ...M, ...x, S, A]

}

}, grmr\_typescript: e => {

const n = {

$pattern: ge,

keyword: ue.concat(["type", "namespace", "typedef", "interface", "public", "private", "protected", "implements", "declare", "abstract", "readonly"]),

literal: be,

built\_in: fe.concat(["any", "void", "number", "boolean", "string", "object", "never", "enum"]),

"variable.language": he

}, t = {

className: "meta", begin: "@[A-Za-z$\_][0-9A-Za-z$\_]\*"

}, a = (e, n, t) => {

const a = e.contains.findIndex((e => e.label === n))

; if (-1 === a) throw Error("can not find mode to replace"); e.contains.splice(a, 1, t)

}, i = Ee(e)

; return Object.assign(i.keywords, n), i.exports.PARAMS\_CONTAINS.push(t), i.contains = i.contains.concat([t, {

beginKeywords: "namespace", end: /\{/, excludeEnd: !0

}, {

beginKeywords: "interface",

end: /\{/, excludeEnd: !0, keywords: "interface extends"

}]), a(i, "shebang", e.SHEBANG()), a(i, "use\_strict", {

className: "meta", relevance: 10,

begin: /^\s\*['"]use strict['"]/

}), i.contains.find((e => "func.def" === e.label)).relevance = 0, Object.assign(i, {

name: "TypeScript", aliases: ["ts", "tsx"]

}), i

}, grmr\_vbnet: e => {

const n = /\d{1,2}\/\d{1,2}\/\d{4}/, t = /\d{4}-\d{1,2}-\d{1,2}/, a = /(\d|1[012])(:\d+){0,2} \*(AM|PM)/, i = /\d{1,2}(:\d{1,2}){1,2}/, s = {

className: "literal", variants: [{ begin: b(/# \*/, m(t, n), / \*#/) }, {

begin: b(/# \*/, i, / \*#/)

}, { begin: b(/# \*/, a, / \*#/) }, {

begin: b(/# \*/, m(t, n), / +/, m(a, i), / \*#/)

}]

}, r = e.COMMENT(/'''/, /$/, {

contains: [{

className: "doctag", begin: /<\/?/, end: />/

}]

}), o = e.COMMENT(null, /$/, {

variants: [{

begin: /'/

}, { begin: /([\t ]|^)REM(?=\s)/ }]

}); return {

name: "Visual Basic .NET",

aliases: ["vb"], case\_insensitive: !0, classNameAliases: { label: "symbol" }, keywords: {

keyword: "addhandler alias aggregate ansi as async assembly auto binary by byref byval call case catch class compare const continue custom declare default delegate dim distinct do each equals else elseif end enum erase error event exit explicit finally for friend from function get global goto group handles if implements imports in inherits interface into iterator join key let lib loop me mid module mustinherit mustoverride mybase myclass namespace narrowing new next notinheritable notoverridable of off on operator option optional order overloads overridable overrides paramarray partial preserve private property protected public raiseevent readonly redim removehandler resume return select set shadows shared skip static step stop structure strict sub synclock take text then throw to try unicode until using when where while widening with withevents writeonly yield",

built\_in: "addressof and andalso await directcast gettype getxmlnamespace is isfalse isnot istrue like mod nameof new not or orelse trycast typeof xor cbool cbyte cchar cdate cdbl cdec cint clng cobj csbyte cshort csng cstr cuint culng cushort",

type: "boolean byte char date decimal double integer long object sbyte short single string uinteger ulong ushort",

literal: "true false nothing"

},

illegal: "//|\\{|\\}|endif|gosub|variant|wend|^\\$ ", contains: [{

className: "string", begin: /"(""|[^/n])"C\b/

}, {

className: "string", begin: /"/,

end: /"/, illegal: /\n/, contains: [{ begin: /""/ }]

}, s, {

className: "number", relevance: 0,

variants: [{

begin: /\b\d[\d\_]\*((\.[\d\_]+(E[+-]?[\d\_]+)?)|(E[+-]?[\d\_]+))[RFD@!#]?/

}, { begin: /\b\d[\d\_]\*((U?[SIL])|[%&])?/ }, { begin: /&H[\dA-F\_]+((U?[SIL])|[%&])?/ }, {

begin: /&O[0-7\_]+((U?[SIL])|[%&])?/

}, { begin: /&B[01\_]+((U?[SIL])|[%&])?/ }]

}, {

className: "label", begin: /^\w+:/

}, r, o, {

className: "meta",

begin: /[\t ]\*#(const|disable|else|elseif|enable|end|externalsource|if|region)\b/,

end: /$/, keywords: {

keyword: "const disable else elseif enable end externalsource if region then"

},

contains: [o]

}]

}

}, grmr\_yaml: e => {

const n = "true false yes no null", t = "[\\w#;/?:@&=+$,.~\*'()[\\]]+", a = {

className: "string", relevance: 0, variants: [{ begin: /'/, end: /'/ }, {

begin: /"/, end: /"/

}, { begin: /\S+/ }], contains: [e.BACKSLASH\_ESCAPE, {

className: "template-variable",

variants: [{ begin: /\{\{/, end: /\}\}/ }, { begin: /%\{/, end: /\}/ }]

}]

}, i = e.inherit(a, {

variants: [{ begin: /'/, end: /'/ }, { begin: /"/, end: /"/ }, { begin: /[^\s,{}[\]]+/ }]

}), s = {

end: ",", endsWithParent: !0, excludeEnd: !0, keywords: n, relevance: 0

}, r = {

begin: /\{/,

end: /\}/, contains: [s], illegal: "\\n", relevance: 0

}, o = {

begin: "\\[", end: "\\]",

contains: [s], illegal: "\\n", relevance: 0

}, l = [{

className: "attr", variants: [{

begin: "\\w[\\w :\\/.-]\*:(?=[ \t]|$)"

}, { begin: '"\\w[\\w :\\/.-]\*":(?=[ \t]|$)' }, {

begin: "'\\w[\\w :\\/.-]\*':(?=[ \t]|$)"

}]

}, {

className: "meta", begin: "^---\\s\*$",

relevance: 10

}, {

className: "string",

begin: "[\\|>]([1-9]?[+-])?[ ]\*\\n( +)[^ ][^\\n]\*\\n(\\2[^\\n]+\\n?)\*"

}, {

begin: "<%[%=-]?", end: "[%-]?%>", subLanguage: "ruby", excludeBegin: !0, excludeEnd: !0,

relevance: 0

}, { className: "type", begin: "!\\w+!" + t }, {

className: "type",

begin: "!<" + t + ">"

}, { className: "type", begin: "!" + t }, {

className: "type", begin: "!!" + t

}, { className: "meta", begin: "&" + e.UNDERSCORE\_IDENT\_RE + "$" }, {

className: "meta",

begin: "\\\*" + e.UNDERSCORE\_IDENT\_RE + "$"

}, {

className: "bullet", begin: "-(?=[ ]|$)",

relevance: 0

}, e.HASH\_COMMENT\_MODE, { beginKeywords: n, keywords: { literal: n } }, {

className: "number",

begin: "\\b[0-9]{4}(-[0-9][0-9]){0,2}([Tt \\t][0-9][0-9]?(:[0-9][0-9]){2})?(\\.[0-9]\*)?([ \\t])\*(Z|[-+][0-9][0-9]?(:[0-9][0-9])?)?\\b"

}, { className: "number", begin: e.C\_NUMBER\_RE + "\\b", relevance: 0 }, r, o, a], c = [...l]

; return c.pop(), c.push(i), s.contains = c, {

name: "YAML", case\_insensitive: !0,

aliases: ["yml"], contains: l

}

}

}); const Fe = Y; for (const e of Object.keys(ze)) {

const n = e.replace("grmr\_", "").replace("\_", "-"); Fe.registerLanguage(n, ze[e])

}

return Fe

}()

; "object" == typeof exports && "undefined" != typeof module && (module.exports = hljs);

export {

hljs

}R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\assets\js\timeago.js

/\*\*

\* Copyright (c) 2016 hustcc

\* License: MIT

\* Version: v3.0.2

\* https://github.com/hustcc/timeago.js

\*\*/

/\* jshint expr: true \*/

!function (root, factory) {

if (typeof module === 'object' && module.exports) {

module.exports = factory(root); // nodejs support

module.exports['default'] = module.exports; // es6 support

}

else

root.timeago = factory(root);

}(typeof window !== 'undefined' ? window : this,

function () {

var indexMapEn = 'second\_minute\_hour\_day\_week\_month\_year'.split('\_'),

indexMapZh = '秒\_分钟\_小时\_天\_周\_月\_年'.split('\_'),

// build-in locales: en & zh\_CN

locales = {

'en': function(number, index) {

if (index === 0) return ['just now', 'right now'];

var unit = indexMapEn[parseInt(index / 2)];

if (number > 1) unit += 's';

return [number + ' ' + unit + ' ago', 'in ' + number + ' ' + unit];

},

'zh\_CN': function(number, index) {

if (index === 0) return ['刚刚', '片刻后'];

var unit = indexMapZh[parseInt(index / 2)];

return [number + unit + '前', number + unit + '后'];

}

},

// second, minute, hour, day, week, month, year(365 days)

SEC\_ARRAY = [60, 60, 24, 7, 365/7/12, 12],

SEC\_ARRAY\_LEN = 6,

// ATTR\_DATETIME = 'datetime',

ATTR\_DATA\_TID = 'data-tid',

timers = {}; // real-time render timers

// format Date / string / timestamp to Date instance.

function toDate(input) {

if (input instanceof Date) return input;

if (!isNaN(input)) return new Date(toInt(input));

if (/^\d+$/.test(input)) return new Date(toInt(input));

input = (input || '').trim().replace(/\.\d+/, '') // remove milliseconds

.replace(/-/, '/').replace(/-/, '/')

.replace(/(\d)T(\d)/, '$1 $2').replace(/Z/, ' UTC') // 2017-2-5T3:57:52Z -> 2017-2-5 3:57:52UTC

.replace(/([\+\-]\d\d)\:?(\d\d)/, ' $1$2'); // -04:00 -> -0400

return new Date(input);

}

// change f into int, remove decimal. Just for code compression

function toInt(f) {

return parseInt(f);

}

// format the diff second to \*\*\* time ago, with setting locale

function formatDiff(diff, locale, defaultLocale) {

// if locale is not exist, use defaultLocale.

// if defaultLocale is not exist, use build-in `en`.

// be sure of no error when locale is not exist.

locale = locales[locale] ? locale : (locales[defaultLocale] ? defaultLocale : 'en');

// if (! locales[locale]) locale = defaultLocale;

var i = 0,

agoin = diff < 0 ? 1 : 0, // timein or timeago

total\_sec = diff = Math.abs(diff);

for (; diff >= SEC\_ARRAY[i] && i < SEC\_ARRAY\_LEN; i++) {

diff /= SEC\_ARRAY[i];

}

diff = toInt(diff);

i \*= 2;

if (diff > (i === 0 ? 9 : 1)) i += 1;

return locales[locale](diff, i, total\_sec)[agoin].replace('%s', diff);

}

// calculate the diff second between date to be formated an now date.

function diffSec(date, nowDate) {

nowDate = nowDate ? toDate(nowDate) : new Date();

return (nowDate - toDate(date)) / 1000;

}

/\*\*

\* nextInterval: calculate the next interval time.

\* - diff: the diff sec between now and date to be formated.

\*

\* What's the meaning?

\* diff = 61 then return 59

\* diff = 3601 (an hour + 1 second), then return 3599

\* make the interval with high performace.

\*\*/

function nextInterval(diff) {

var rst = 1, i = 0, d = Math.abs(diff);

for (; diff >= SEC\_ARRAY[i] && i < SEC\_ARRAY\_LEN; i++) {

diff /= SEC\_ARRAY[i];

rst \*= SEC\_ARRAY[i];

}

// return leftSec(d, rst);

d = d % rst;

d = d ? rst - d : rst;

return Math.ceil(d);

}

// get the datetime attribute, `data-timeagp` / `datetime` are supported.

function getDateAttr(node) {

return getAttr(node, 'data-timeago') || getAttr(node, 'datetime');

}

// get the node attribute, native DOM and jquery supported.

function getAttr(node, name) {

if(node.getAttribute) return node.getAttribute(name); // native

if(node.attr) return node.attr(name); // jquery

}

// set the node attribute, native DOM and jquery supported.

function setTidAttr(node, val) {

if(node.setAttribute) return node.setAttribute(ATTR\_DATA\_TID, val); // native

if(node.attr) return node.attr(ATTR\_DATA\_TID, val); // jquery

}

// get the timer id of node.

// remove the function, can save some bytes.

// function getTidFromNode(node) {

// return getAttr(node, ATTR\_DATA\_TID);

// }

/\*\*

\* timeago: the function to get `timeago` instance.

\* - nowDate: the relative date, default is new Date().

\* - defaultLocale: the default locale, default is en. if your set it, then the `locale` parameter of format is not needed of you.

\*

\* How to use it?

\* var timeagoLib = require('timeago.js');

\* var timeago = timeagoLib(); // all use default.

\* var timeago = timeagoLib('2016-09-10'); // the relative date is 2016-09-10, so the 2016-09-11 will be 1 day ago.

\* var timeago = timeagoLib(null, 'zh\_CN'); // set default locale is `zh\_CN`.

\* var timeago = timeagoLib('2016-09-10', 'zh\_CN'); // the relative date is 2016-09-10, and locale is zh\_CN, so the 2016-09-11 will be 1天前.

\*\*/

function Timeago(nowDate, defaultLocale) {

this.nowDate = nowDate;

// if do not set the defaultLocale, set it with `en`

this.defaultLocale = defaultLocale || 'en'; // use default build-in locale

// for dev test

// this.nextInterval = nextInterval;

}

// what the timer will do

Timeago.prototype.doRender = function(node, date, locale) {

var diff = diffSec(date, this.nowDate),

self = this,

tid;

// delete previously assigned timeout's id to node

node.innerHTML = formatDiff(diff, locale, this.defaultLocale);

// waiting %s seconds, do the next render

timers[tid = setTimeout(function() {

self.doRender(node, date, locale);

delete timers[tid];

}, Math.min(nextInterval(diff) \* 1000, 0x7FFFFFFF))] = 0; // there is no need to save node in object.

// set attribute date-tid

setTidAttr(node, tid);

};

/\*\*

\* format: format the date to \*\*\* time ago, with setting or default locale

\* - date: the date / string / timestamp to be formated

\* - locale: the formated string's locale name, e.g. en / zh\_CN

\*

\* How to use it?

\* var timeago = require('timeago.js')();

\* timeago.format(new Date(), 'pl'); // Date instance

\* timeago.format('2016-09-10', 'fr'); // formated date string

\* timeago.format(1473473400269); // timestamp with ms

\*\*/

Timeago.prototype.format = function(date, locale) {

return formatDiff(diffSec(date, this.nowDate), locale, this.defaultLocale);

};

/\*\*

\* render: render the DOM real-time.

\* - nodes: which nodes will be rendered.

\* - locale: the locale name used to format date.

\*

\* How to use it?

\* var timeago = require('timeago.js')();

\* // 1. javascript selector

\* timeago.render(document.querySelectorAll('.need\_to\_be\_rendered'));

\* // 2. use jQuery selector

\* timeago.render($('.need\_to\_be\_rendered'), 'pl');

\*

\* Notice: please be sure the dom has attribute `datetime`.

\*\*/

Timeago.prototype.render = function(nodes, locale) {

if (nodes.length === undefined) nodes = [nodes];

for (var i = 0, len = nodes.length; i < len; i++) {

this.doRender(nodes[i], getDateAttr(nodes[i]), locale); // render item

}

};

/\*\*

\* setLocale: set the default locale name.

\*

\* How to use it?

\* var timeago = require('timeago.js')();

\* timeago.setLocale('fr');

\*\*/

Timeago.prototype.setLocale = function(locale) {

this.defaultLocale = locale;

};

/\*\*

\* timeago: the function to get `timeago` instance.

\* - nowDate: the relative date, default is new Date().

\* - defaultLocale: the default locale, default is en. if your set it, then the `locale` parameter of format is not needed of you.

\*

\* How to use it?

\* var timeagoFactory = require('timeago.js');

\* var timeago = timeagoFactory(); // all use default.

\* var timeago = timeagoFactory('2016-09-10'); // the relative date is 2016-09-10, so the 2016-09-11 will be 1 day ago.

\* var timeago = timeagoFactory(null, 'zh\_CN'); // set default locale is `zh\_CN`.

\* var timeago = timeagoFactory('2016-09-10', 'zh\_CN'); // the relative date is 2016-09-10, and locale is zh\_CN, so the 2016-09-11 will be 1天前.

\*\*/

function timeagoFactory(nowDate, defaultLocale) {

return new Timeago(nowDate, defaultLocale);

}

/\*\*

\* register: register a new language locale

\* - locale: locale name, e.g. en / zh\_CN, notice the standard.

\* - localeFunc: the locale process function

\*

\* How to use it?

\* var timeagoFactory = require('timeago.js');

\*

\* timeagoFactory.register('the locale name', the\_locale\_func);

\* // or

\* timeagoFactory.register('pl', require('timeago.js/locales/pl'));

\*\*/

timeagoFactory.register = function(locale, localeFunc) {

locales[locale] = localeFunc;

};

/\*\*

\* cancel: cancels one or all the timers which are doing real-time render.

\*

\* How to use it?

\* For canceling all the timers:

\* var timeagoFactory = require('timeago.js');

\* var timeago = timeagoFactory();

\* timeago.render(document.querySelectorAll('.need\_to\_be\_rendered'));

\* timeagoFactory.cancel(); // will stop all the timers, stop render in real time.

\*

\* For canceling single timer on specific node:

\* var timeagoFactory = require('timeago.js');

\* var timeago = timeagoFactory();

\* var nodes = document.querySelectorAll('.need\_to\_be\_rendered');

\* timeago.render(nodes);

\* timeagoFactory.cancel(nodes[0]); // will clear a timer attached to the first node, stop render in real time.

\*\*/

timeagoFactory.cancel = function(node) {

var tid;

// assigning in if statement to save space

if (node) {

tid = getAttr(node, ATTR\_DATA\_TID); // get the timer of DOM node(native / jq).

if (tid) {

clearTimeout(tid);

delete timers[tid];

}

} else {

for (tid in timers) clearTimeout(tid);

timers = {};

}

};

return timeagoFactory;

});R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\components\Breadcrumb\index.vue

<template>

<el-breadcrumb class="app-breadcrumb" separator="/">

<transition-group name="breadcrumb">

<el-breadcrumb-item v-for="(item,index) in levelList" :key="item.path">

<span v-if="item.redirect==='noRedirect'||index==levelList.length-1" class="no-redirect">{{ item.meta.title }}</span>

<a v-else @click.prevent="handleLink(item)">{{ item.meta.title }}</a>

</el-breadcrumb-item>

</transition-group>

</el-breadcrumb>

</template>

<script>

import pathToRegexp from 'path-to-regexp'

export default {

data() {

return {

levelList: null

}

},

watch: {

$route() {

this.getBreadcrumb()

}

},

created() {

this.getBreadcrumb()

},

methods: {

getBreadcrumb() {

// only show routes with meta.title

let matched = this.$route.matched.filter(item => item.meta && item.meta.title)

const first = matched[0]

if (!this.isDashboard(first)) {

matched = [{ path: '/dashboard', meta: { title: 'Dashboard' }}].concat(matched)

}

this.levelList = matched.filter(item => item.meta && item.meta.title && item.meta.breadcrumb !== false)

},

isDashboard(route) {

const name = route && route.name

if (!name) {

return false

}

return name.trim().toLocaleLowerCase() === 'Dashboard'.toLocaleLowerCase()

},

pathCompile(path) {

// To solve this problem https://github.com/PanJiaChen/vue-element-admin/issues/561

const { params } = this.$route

var toPath = pathToRegexp.compile(path)

return toPath(params)

},

handleLink(item) {

const { redirect, path } = item

if (redirect) {

this.$router.push(redirect)

return

}

this.$router.push(this.pathCompile(path))

}

}

}

</script>

<style lang="scss" scoped>

.app-breadcrumb.el-breadcrumb {

display: inline-block;

font-size: 14px;

line-height: 50px;

margin-left: 8px;

.no-redirect {

color: #97a8be;

cursor: text;

}

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\components\Hamburger\index.vue

<template>

<div style="padding: 0 15px;" @click="toggleClick">

<svg

:class="{'is-active':isActive}"

class="hamburger"

viewBox="0 0 1024 1024"

xmlns="http://www.w3.org/2000/svg"

width="64"

height="64"

>

<path d="M408 442h480c4.4 0 8-3.6 8-8v-56c0-4.4-3.6-8-8-8H408c-4.4 0-8 3.6-8 8v56c0 4.4 3.6 8 8 8zm-8 204c0 4.4 3.6 8 8 8h480c4.4 0 8-3.6 8-8v-56c0-4.4-3.6-8-8-8H408c-4.4 0-8 3.6-8 8v56zm504-486H120c-4.4 0-8 3.6-8 8v56c0 4.4 3.6 8 8 8h784c4.4 0 8-3.6 8-8v-56c0-4.4-3.6-8-8-8zm0 632H120c-4.4 0-8 3.6-8 8v56c0 4.4 3.6 8 8 8h784c4.4 0 8-3.6 8-8v-56c0-4.4-3.6-8-8-8zM142.4 642.1L298.7 519a8.84 8.84 0 0 0 0-13.9L142.4 381.9c-5.8-4.6-14.4-.5-14.4 6.9v246.3a8.9 8.9 0 0 0 14.4 7z" />

</svg>

</div>

</template>

<script>

export default {

name: 'Hamburger',

props: {

isActive: {

type: Boolean,

default: false

}

},

methods: {

toggleClick() {

this.$emit('toggleClick')

}

}

}

</script>

<style scoped>

.hamburger {

display: inline-block;

vertical-align: middle;

width: 20px;

height: 20px;

}

.hamburger.is-active {

transform: rotate(180deg);

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\components\MarkdownEditor\default-options.js

export default {

minHeight: '200px',

previewStyle: 'vertical',

useCommandShortcut: true,

useDefaultHTMLSanitizer: true,

usageStatistics: false,

hideModeSwitch: false,

toolbarItems: [

'heading',

'bold',

'italic',

'strike',

'divider',

'hr',

'quote',

'divider',

'ul',

'ol',

'task',

'indent',

'outdent',

'divider',

'table',

'image',

'link',

'divider',

'code',

'codeblock'

]

}R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\components\MarkdownEditor\index.vue

<template>

<div :id="id" />

</template>

<script>

// deps for editor

import 'codemirror/lib/codemirror.css' // codemirror

import 'tui-editor/dist/tui-editor.css' // editor ui

import 'tui-editor/dist/tui-editor-contents.css' // editor content

import Editor from 'tui-editor'

import defaultOptions from './default-options'

// import { uploadPicture } from '../../api/article'

import axios from 'axios'

export default {

name: 'MarkdownEditor',

props: {

value: {

type: String,

default: '',

},

id: {

type: String,

required: false,

default() {

return (

'markdown-editor-' +

+new Date() +

((Math.random() \* 1000).toFixed(0) + '')

)

},

},

options: {

type: Object,

default() {

return defaultOptions

},

},

mode: {

type: String,

default: 'markdown',

},

height: {

type: String,

required: false,

default: '300px',

},

language: {

type: String,

required: false,

default: 'en\_US',

},

},

data() {

return {

editor: null,

}

},

computed: {

editorOptions() {

const options = Object.assign({}, defaultOptions, this.options)

options.initialEditType = this.mode

options.height = this.height

options.language = this.language

return options

},

},

watch: {

value(newValue, preValue) {

if (newValue !== preValue && newValue !== this.editor.getValue()) {

this.editor.setValue(newValue)

}

},

language(val) {

this.destroyEditor()

this.initEditor()

},

height(newValue) {

this.editor.height(newValue)

},

mode(newValue) {

this.editor.changeMode(newValue)

},

},

mounted() {

this.initEditor()

},

destroyed() {

this.destroyEditor()

},

methods: {

initEditor() {

this.editor = new Editor({

el: document.getElementById(this.id),

...this.editorOptions,

})

// 删除默认监听事件后，添加自定义监听事件

this.editor.eventManager.removeEventHandler('addImageBlobHook')

this.editor.eventManager.listen('addImageBlobHook', (blob, callback) => {

// 此处填写自己的上传逻辑，url为上传后的图片地址

const formData = new FormData()

formData.append('file', blob)

const ajax = new XMLHttpRequest()

ajax.open('POST', 'http://101.201.64.102:8000/article/picture', true)

// ajax.open('POST', 'http://localhost:8000/article/picture', true)

ajax.send(formData)

ajax.onreadystatechange = function () {

if (ajax.readyState === 4) {

if (

(ajax.status >= 200 && ajax.status < 300) ||

ajax.status === 304

) {

const obj = JSON.parse(ajax.responseText)

if (obj.code && obj.code === 100) {

// 回调，回显上传后图片的地址

callback(obj.extend.url)

}

}

}

}

})

if (this.value) {

this.editor.setValue(this.value)

}

this.editor.on('change', () => {

this.$emit('input', this.editor.getValue())

})

},

destroyEditor() {

if (!this.editor) return

this.editor.off('change')

this.editor.remove()

},

setValue(value) {

this.editor.setValue(value)

},

getValue() {

return this.editor.getValue()

},

setHtml(value) {

this.editor.setHtml(value)

},

getHtml() {

return this.editor.getHtml()

},

},

}

</script>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\components\SearchArticle\index.vue

<template>

<!-- 搜索框 -->

<el-collapse

accordion

class="search-bar"

>

<el-collapse-item>

<template slot="title">

<i

class="header-icon el-icon-search"

style="font-size:20px;margin-right:

5px;"

></i>搜索文章

</template>

<el-form

ref="form"

label-width="80px"

>

<el-form-item label="标题包含">

<el-input v-model="query.article\_title"></el-input>

</el-form-item>

<el-form-item label="作者" v-if="isAdmin">

<el-input v-model="query.article\_author\_name"></el-input>

</el-form-item>

<el-form-item label="发表时间">

<el-col :span="11">

<el-date-picker

type="date"

placeholder="开始日期"

v-model="query.start"

style="width: 100%;"

value-format="yyyy-MM-dd"

></el-date-picker>

</el-col>

<el-col :span="11">

<el-date-picker

type="date"

placeholder="结束日期"

v-model="query.end"

value-format="yyyy-MM-dd"

style="width: 100%;"

></el-date-picker>

</el-col>

</el-form-item>

<el-form-item label="类型">

<el-select

v-model="query.article\_type"

filterable

placeholder="选择或搜索"

>

<el-option

v-for="item in typeList"

:key="item.typeId"

:label="item.typeName"

:value="item.typeId"

>

</el-option>

</el-select>

</el-form-item>

<el-form-item label="浏览量">

<el-radio-group v-model="query.rank\_view">

<el-radio :label="-1">无</el-radio>

<el-radio :label="1">升序</el-radio>

<el-radio :label="0">降序</el-radio>

</el-radio-group>

</el-form-item>

<el-form-item label="点赞">

<el-radio-group v-model="query.rank\_like">

<el-radio :label="-1">无</el-radio>

<el-radio :label="1">升序</el-radio>

<el-radio :label="0">降序</el-radio>

</el-radio-group>

</el-form-item>

<el-form-item label="评论">

<el-radio-group v-model="query.rank\_comment">

<el-radio :label="-1">无</el-radio>

<el-radio :label="1">升序</el-radio>

<el-radio :label="0">降序</el-radio>

</el-radio-group>

</el-form-item>

<el-form-item>

<el-button

type="primary"

@click="submitSearch"

>搜索</el-button>

<el-button @click="resetForm">重置</el-button>

</el-form-item>

</el-form>

</el-collapse-item>

</el-collapse>

</template>

<script>

export default {

name: 'SearchArticle',

props: {

typeList: {

type: Array

},

isAdmin:{

type: Boolean,

default: false

}

},

data() {

return {

condition: '',

query: {

article\_author\_name: '',

article\_title: '',

article\_type: '',

start: '',

end: '',

rank\_view: -1,

rank\_like: -1,

rank\_comment: -1,

},

}

},

mounted() {

this.resetForm()

},

methods: {

submitSearch() {

this.$emit('submit')

},

resetForm() {

this.query.article\_author\_name = ''

this.query.article\_title = ''

this.query.article\_type = ''

this.query.start = ''

this.query.end = ''

this.query.rank\_view = -1

this.query.rank\_like = -1

this.query.rank\_comment = -1

this.condition = ''

this.$emit('reset')

},

generateQueryString() {

let condition = ''

let query = this.query

if (query.article\_author\_name !== '') {

condition += 'article\_author\_name=' + query.article\_author\_name + '--'

}

if (query.article\_title !== '') {

condition += 'article\_title=' + query.article\_title + '--'

}

if (query.article\_type !== '') {

condition += 'type=' + String(query.article\_type) + '--'

}

if (query.start !== '') {

condition += 'start=' + query.start + '--'

}

if (query.end !== '') {

condition += 'end=' + query.end + '--'

}

if (query.rank\_view !== -1) {

condition += 'rank\_view=' + String(query.rank\_view) + '--'

}

if (query.rank\_like !== -1) {

condition += 'rank\_like=' + String(query.rank\_like) + '--'

}

if (query.rank\_comment !== -1) {

condition += 'rank\_comment=' + String(query.rank\_comment) + '--'

}

condition =

condition.lastIndexOf('#') === condition.length - 1

? condition.substr(0, condition.length - 1)

: condition

return condition

},

},

}

</script>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\components\SvgIcon\index.vue

<template>

<div v-if="isExternal" :style="styleExternalIcon" class="svg-external-icon svg-icon" v-on="$listeners" />

<svg v-else :class="svgClass" aria-hidden="true" v-on="$listeners">

<use :xlink:href="iconName" />

</svg>

</template>

<script>

// doc: https://panjiachen.github.io/vue-element-admin-site/feature/component/svg-icon.html#usage

import { isExternal } from '@/utils/validate'

export default {

name: 'SvgIcon',

props: {

iconClass: {

type: String,

required: true

},

className: {

type: String,

default: ''

}

},

computed: {

isExternal() {

return isExternal(this.iconClass)

},

iconName() {

return `#icon-${this.iconClass}`

},

svgClass() {

if (this.className) {

return 'svg-icon ' + this.className

} else {

return 'svg-icon'

}

},

styleExternalIcon() {

return {

mask: `url(${this.iconClass}) no-repeat 50% 50%`,

'-webkit-mask': `url(${this.iconClass}) no-repeat 50% 50%`

}

}

}

}

</script>

<style scoped>

.svg-icon {

width: 1em;

height: 1em;

vertical-align: -0.15em;

fill: currentColor;

overflow: hidden;

}

.svg-external-icon {

background-color: currentColor;

mask-size: cover!important;

display: inline-block;

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\index.js

import Vue from 'vue'

import SvgIcon from '@/components/SvgIcon'// svg component

// register globally

Vue.component('svg-icon', SvgIcon)

const req = require.context('./svg', false, /\.svg$/)

const requireAll = requireContext => requireContext.keys().map(requireContext)

requireAll(req)

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\dashboard.svg

<svg width="128" height="100" xmlns="http://www.w3.org/2000/svg"><path d="M27.429 63.638c0-2.508-.893-4.65-2.679-6.424-1.786-1.775-3.94-2.662-6.464-2.662-2.524 0-4.679.887-6.465 2.662-1.785 1.774-2.678 3.916-2.678 6.424 0 2.508.893 4.65 2.678 6.424 1.786 1.775 3.94 2.662 6.465 2.662 2.524 0 4.678-.887 6.464-2.662 1.786-1.775 2.679-3.916 2.679-6.424zm13.714-31.801c0-2.508-.893-4.65-2.679-6.424-1.785-1.775-3.94-2.662-6.464-2.662-2.524 0-4.679.887-6.464 2.662-1.786 1.774-2.679 3.916-2.679 6.424 0 2.508.893 4.65 2.679 6.424 1.785 1.774 3.94 2.662 6.464 2.662 2.524 0 4.679-.888 6.464-2.662 1.786-1.775 2.679-3.916 2.679-6.424zM71.714 65.98l7.215-27.116c.285-1.23.107-2.378-.536-3.443-.643-1.064-1.56-1.762-2.75-2.094-1.19-.33-2.333-.177-3.429.462-1.095.639-1.81 1.573-2.143 2.804l-7.214 27.116c-2.857.237-5.405 1.266-7.643 3.088-2.238 1.822-3.738 4.152-4.5 6.992-.952 3.644-.476 7.098 1.429 10.364 1.905 3.265 4.69 5.37 8.357 6.317 3.667.947 7.143.474 10.429-1.42 3.285-1.892 5.404-4.66 6.357-8.305.762-2.84.619-5.607-.429-8.305-1.047-2.697-2.762-4.85-5.143-6.46zm47.143-2.342c0-2.508-.893-4.65-2.678-6.424-1.786-1.775-3.94-2.662-6.465-2.662-2.524 0-4.678.887-6.464 2.662-1.786 1.774-2.679 3.916-2.679 6.424 0 2.508.893 4.65 2.679 6.424 1.786 1.775 3.94 2.662 6.464 2.662 2.524 0 4.679-.887 6.465-2.662 1.785-1.775 2.678-3.916 2.678-6.424zm-45.714-45.43c0-2.509-.893-4.65-2.679-6.425C68.68 10.01 66.524 9.122 64 9.122c-2.524 0-4.679.887-6.464 2.661-1.786 1.775-2.679 3.916-2.679 6.425 0 2.508.893 4.65 2.679 6.424 1.785 1.774 3.94 2.662 6.464 2.662 2.524 0 4.679-.888 6.464-2.662 1.786-1.775 2.679-3.916 2.679-6.424zm32 13.629c0-2.508-.893-4.65-2.679-6.424-1.785-1.775-3.94-2.662-6.464-2.662-2.524 0-4.679.887-6.464 2.662-1.786 1.774-2.679 3.916-2.679 6.424 0 2.508.893 4.65 2.679 6.424 1.785 1.774 3.94 2.662 6.464 2.662 2.524 0 4.679-.888 6.464-2.662 1.786-1.775 2.679-3.916 2.679-6.424zM128 63.638c0 12.351-3.357 23.78-10.071 34.286-.905 1.372-2.19 2.058-3.858 2.058H13.93c-1.667 0-2.953-.686-3.858-2.058C3.357 87.465 0 76.037 0 63.638c0-8.613 1.69-16.847 5.071-24.703C8.452 31.08 13 24.312 18.714 18.634c5.715-5.68 12.524-10.199 20.429-13.559C47.048 1.715 55.333.035 64 .035c8.667 0 16.952 1.68 24.857 5.04 7.905 3.36 14.714 7.88 20.429 13.559 5.714 5.678 10.262 12.446 13.643 20.301 3.38 7.856 5.071 16.09 5.071 24.703z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\example.svg

<svg width="128" height="128" xmlns="http://www.w3.org/2000/svg"><path d="M96.258 57.462h31.421C124.794 27.323 100.426 2.956 70.287.07v31.422a32.856 32.856 0 0 1 25.971 25.97zm-38.796-25.97V.07C27.323 2.956 2.956 27.323.07 57.462h31.422a32.856 32.856 0 0 1 25.97-25.97zm12.825 64.766v31.421c30.46-2.885 54.507-27.253 57.713-57.712H96.579c-2.886 13.466-13.146 23.726-26.292 26.291zM31.492 70.287H.07c2.886 30.46 27.253 54.507 57.713 57.713V96.579c-13.466-2.886-23.726-13.146-26.291-26.292z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\eye-open.svg

<svg class="icon" viewBox="0 0 1024 1024" xmlns="http://www.w3.org/2000/svg" width="128" height="128"><defs><style/></defs><path d="M512 128q69.675 0 135.51 21.163t115.498 54.997 93.483 74.837 73.685 82.006 51.67 74.837 32.17 54.827L1024 512q-2.347 4.992-6.315 13.483T998.87 560.17t-31.658 51.669-44.331 59.99-56.832 64.34-69.504 60.16-82.347 51.5-94.848 34.687T512 896q-69.675 0-135.51-21.163t-115.498-54.826-93.483-74.326-73.685-81.493-51.67-74.496-32.17-54.997L0 513.707q2.347-4.992 6.315-13.483t18.816-34.816 31.658-51.84 44.331-60.33 56.832-64.683 69.504-60.331 82.347-51.84 94.848-34.816T512 128.085zm0 85.333q-46.677 0-91.648 12.331t-81.152 31.83-70.656 47.146-59.648 54.485-48.853 57.686-37.675 52.821-26.325 43.99q12.33 21.674 26.325 43.52t37.675 52.351 48.853 57.003 59.648 53.845T339.2 767.02t81.152 31.488T512 810.667t91.648-12.331 81.152-31.659 70.656-46.848 59.648-54.186 48.853-57.344 37.675-52.651T927.957 512q-12.33-21.675-26.325-43.648t-37.675-52.65-48.853-57.345-59.648-54.186-70.656-46.848-81.152-31.659T512 213.334zm0 128q70.656 0 120.661 50.006T682.667 512 632.66 632.661 512 682.667 391.339 632.66 341.333 512t50.006-120.661T512 341.333zm0 85.334q-35.328 0-60.33 25.002T426.666 512t25.002 60.33T512 597.334t60.33-25.002T597.334 512t-25.002-60.33T512 426.666z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\eye.svg

<svg width="128" height="64" xmlns="http://www.w3.org/2000/svg"><path d="M127.072 7.994c1.37-2.208.914-5.152-.914-6.87-2.056-1.717-4.797-1.226-6.396.982-.229.245-25.586 32.382-55.74 32.382-29.24 0-55.74-32.382-55.968-32.627-1.6-1.963-4.57-2.208-6.397-.49C-.17 3.086-.399 6.275 1.2 8.238c.457.736 5.94 7.36 14.62 14.72L4.17 35.96c-1.828 1.963-1.6 5.152.228 6.87.457.98 1.6 1.471 2.742 1.471s2.284-.49 3.198-1.472l12.564-13.983c5.94 4.416 13.021 8.587 20.788 11.53l-4.797 17.418c-.685 2.699.686 5.397 3.198 6.133h1.37c2.057 0 3.884-1.472 4.341-3.68L52.6 42.83c3.655.736 7.538 1.227 11.422 1.227 3.883 0 7.767-.49 11.422-1.227l4.797 17.173c.457 2.208 2.513 3.68 4.34 3.68.457 0 .914 0 1.143-.246 2.513-.736 3.883-3.434 3.198-6.133l-4.797-17.172c7.767-2.944 14.848-7.114 20.788-11.53l12.336 13.738c.913.981 2.056 1.472 3.198 1.472s2.284-.49 3.198-1.472c1.828-1.963 1.828-4.906.228-6.87l-11.65-13.001c9.366-7.36 14.849-14.474 14.849-14.474z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\form.svg

<svg width="128" height="128" xmlns="http://www.w3.org/2000/svg"><path d="M84.068 23.784c-1.02 0-1.877-.32-2.572-.96a8.588 8.588 0 0 1-1.738-2.237 11.524 11.524 0 0 1-1.042-2.621c-.232-.895-.348-1.641-.348-2.238V0h.278c.834 0 1.622.085 2.363.256.742.17 1.645.575 2.711 1.214 1.066.64 2.363 1.535 3.892 2.686 1.53 1.15 3.453 2.664 5.77 4.54 2.502 2.045 4.494 3.771 5.977 5.178 1.483 1.406 2.618 2.6 3.406 3.58.787.98 1.274 1.812 1.46 2.494.185.682.277 1.278.277 1.79v2.046H84.068zM127.3 84.01c.278.682.464 1.535.556 2.558.093 1.023-.37 2.003-1.39 2.94-.463.427-.88.832-1.25 1.215-.372.384-.696.704-.974.96a6.69 6.69 0 0 1-.973.767l-11.816-10.741a44.331 44.331 0 0 0 1.877-1.535 31.028 31.028 0 0 1 1.737-1.406c1.112-.938 2.317-1.343 3.615-1.215 1.297.128 2.363.405 3.197.83.927.427 1.923 1.173 2.989 2.239 1.065 1.065 1.876 2.195 2.432 3.388zM78.23 95.902c2.038 0 3.752-.511 5.143-1.534l-26.969 25.83H18.037c-1.761 0-3.684-.47-5.77-1.407a24.549 24.549 0 0 1-5.838-3.709 21.373 21.373 0 0 1-4.518-5.306c-1.204-2.003-1.807-4.07-1.807-6.202V16.495c0-1.79.44-3.665 1.32-5.626A18.41 18.41 0 0 1 5.04 5.562a21.798 21.798 0 0 1 5.213-3.964C12.198.533 14.237 0 16.37 0h53.24v15.984c0 1.62.278 3.367.834 5.242a16.704 16.704 0 0 0 2.572 5.179c1.159 1.577 2.665 2.898 4.518 3.964 1.853 1.066 4.078 1.598 6.673 1.598h20.295v42.325L85.458 92.45c1.02-1.364 1.529-2.856 1.529-4.476 0-2.216-.857-4.113-2.572-5.69-1.714-1.577-3.776-2.366-6.186-2.366H26.1c-2.409 0-4.448.789-6.116 2.366-1.668 1.577-2.502 3.474-2.502 5.69 0 2.217.834 4.092 2.502 5.626 1.668 1.535 3.707 2.302 6.117 2.302h52.13zM26.1 47.951c-2.41 0-4.449.789-6.117 2.366-1.668 1.577-2.502 3.473-2.502 5.69 0 2.216.834 4.092 2.502 5.626 1.668 1.534 3.707 2.302 6.117 2.302h52.13c2.409 0 4.47-.768 6.185-2.302 1.715-1.534 2.572-3.41 2.572-5.626 0-2.217-.857-4.113-2.572-5.69-1.714-1.577-3.776-2.366-6.186-2.366H26.1zm52.407 64.063l1.807-1.663 3.476-3.196a479.75 479.75 0 0 0 4.587-4.284 500.757 500.757 0 0 1 5.004-4.667c3.985-3.666 8.48-7.758 13.485-12.276l11.677 10.741-13.485 12.404-5.004 4.603-4.587 4.22a179.46 179.46 0 0 0-3.267 3.068c-.88.853-1.367 1.322-1.46 1.407-.463.341-.973.703-1.529 1.087-.556.383-1.112.703-1.668.959-.556.256-1.413.575-2.572.959a83.5 83.5 0 0 1-3.545 1.087 72.2 72.2 0 0 1-3.475.895c-1.112.256-1.946.426-2.502.511-1.112.17-1.854.043-2.224-.383-.371-.426-.464-1.151-.278-2.174.092-.511.278-1.279.556-2.302.278-1.023.602-2.067.973-3.132l1.042-3.005c.325-.938.58-1.577.765-1.918a10.157 10.157 0 0 1 2.224-2.941z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\link.svg

<svg width="128" height="128" xmlns="http://www.w3.org/2000/svg"><path d="M115.625 127.937H.063V12.375h57.781v12.374H12.438v90.813h90.813V70.156h12.374z"/><path d="M116.426 2.821l8.753 8.753-56.734 56.734-8.753-8.745z"/><path d="M127.893 37.982h-12.375V12.375H88.706V0h39.187z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\nested.svg

<svg width="128" height="128" xmlns="http://www.w3.org/2000/svg"><path d="M.002 9.2c0 5.044 3.58 9.133 7.998 9.133 4.417 0 7.997-4.089 7.997-9.133 0-5.043-3.58-9.132-7.997-9.132S.002 4.157.002 9.2zM31.997.066h95.981V18.33H31.997V.066zm0 45.669c0 5.044 3.58 9.132 7.998 9.132 4.417 0 7.997-4.088 7.997-9.132 0-3.263-1.524-6.278-3.998-7.91-2.475-1.63-5.524-1.63-7.998 0-2.475 1.632-4 4.647-4 7.91zM63.992 36.6h63.986v18.265H63.992V36.6zm-31.995 82.2c0 5.043 3.58 9.132 7.998 9.132 4.417 0 7.997-4.089 7.997-9.132 0-5.044-3.58-9.133-7.997-9.133s-7.998 4.089-7.998 9.133zm31.995-9.131h63.986v18.265H63.992V109.67zm0-27.404c0 5.044 3.58 9.133 7.998 9.133 4.417 0 7.997-4.089 7.997-9.133 0-3.263-1.524-6.277-3.998-7.909-2.475-1.631-5.524-1.631-7.998 0-2.475 1.632-4 4.646-4 7.91zm31.995-9.13h31.991V91.4H95.987V73.135z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\password.svg

<svg width="128" height="128" xmlns="http://www.w3.org/2000/svg"><path d="M108.8 44.322H89.6v-5.36c0-9.04-3.308-24.163-25.6-24.163-23.145 0-25.6 16.881-25.6 24.162v5.361H19.2v-5.36C19.2 15.281 36.798 0 64 0c27.202 0 44.8 15.281 44.8 38.961v5.361zm-32 39.356c0-5.44-5.763-9.832-12.8-9.832-7.037 0-12.8 4.392-12.8 9.832 0 3.682 2.567 6.808 6.407 8.477v11.205c0 2.718 2.875 4.962 6.4 4.962 3.524 0 6.4-2.244 6.4-4.962V92.155c3.833-1.669 6.393-4.795 6.393-8.477zM128 64v49.201c0 8.158-8.645 14.799-19.2 14.799H19.2C8.651 128 0 121.359 0 113.201V64c0-8.153 8.645-14.799 19.2-14.799h89.6c10.555 0 19.2 6.646 19.2 14.799z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\table.svg

<svg width="128" height="128" xmlns="http://www.w3.org/2000/svg"><path d="M.006.064h127.988v31.104H.006V.064zm0 38.016h38.396v41.472H.006V38.08zm0 48.384h38.396v41.472H.006V86.464zM44.802 38.08h38.396v41.472H44.802V38.08zm0 48.384h38.396v41.472H44.802V86.464zM89.598 38.08h38.396v41.472H89.598zm0 48.384h38.396v41.472H89.598z"/><path d="M.006.064h127.988v31.104H.006V.064zm0 38.016h38.396v41.472H.006V38.08zm0 48.384h38.396v41.472H.006V86.464zM44.802 38.08h38.396v41.472H44.802V38.08zm0 48.384h38.396v41.472H44.802V86.464zM89.598 38.08h38.396v41.472H89.598zm0 48.384h38.396v41.472H89.598z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\tree.svg

<svg width="128" height="128" xmlns="http://www.w3.org/2000/svg"><path d="M126.713 90.023c.858.985 1.287 2.134 1.287 3.447v29.553c0 1.423-.429 2.6-1.287 3.53-.858.93-1.907 1.395-3.146 1.395H97.824c-1.145 0-2.146-.465-3.004-1.395-.858-.93-1.287-2.107-1.287-3.53V93.47c0-.875.19-1.696.572-2.462.382-.766.906-1.368 1.573-1.806a3.84 3.84 0 0 1 2.146-.657h9.725V69.007a3.84 3.84 0 0 0-.43-1.806 3.569 3.569 0 0 0-1.143-1.313 2.714 2.714 0 0 0-1.573-.492h-36.47v23.149h9.725c1.144 0 2.145.492 3.004 1.478.858.985 1.287 2.134 1.287 3.447v29.553c0 .876-.191 1.696-.573 2.463-.38.766-.905 1.368-1.573 1.806a3.84 3.84 0 0 1-2.145.656H51.915a3.84 3.84 0 0 1-2.145-.656c-.668-.438-1.216-1.04-1.645-1.806a4.96 4.96 0 0 1-.644-2.463V93.47c0-1.313.43-2.462 1.288-3.447.858-.986 1.907-1.478 3.146-1.478h9.582v-23.15h-37.9c-.953 0-1.74.356-2.359 1.068-.62.711-.93 1.56-.93 2.544v19.538h9.726c1.239 0 2.264.492 3.074 1.478.81.985 1.216 2.134 1.216 3.447v29.553c0 1.423-.405 2.6-1.216 3.53-.81.93-1.835 1.395-3.074 1.395H4.29c-.476 0-.93-.082-1.358-.246a4.1 4.1 0 0 1-1.144-.657 4.658 4.658 0 0 1-.93-1.067 5.186 5.186 0 0 1-.643-1.395 5.566 5.566 0 0 1-.215-1.56V93.47c0-.437.048-.875.143-1.313a3.95 3.95 0 0 1 .429-1.15c.19-.328.429-.656.715-.984.286-.329.572-.602.858-.821.286-.22.62-.383 1.001-.493.382-.11.763-.164 1.144-.164h9.726V61.619c0-.985.31-1.833.93-2.544.619-.712 1.358-1.068 2.216-1.068h44.335V39.62h-9.582c-1.24 0-2.288-.492-3.146-1.477a5.09 5.09 0 0 1-1.287-3.448V5.14c0-1.423.429-2.627 1.287-3.612.858-.985 1.907-1.477 3.146-1.477h25.743c.763 0 1.478.246 2.145.739a5.17 5.17 0 0 1 1.573 1.888c.382.766.573 1.587.573 2.462v29.553c0 1.313-.43 2.463-1.287 3.448-.859.985-1.86 1.477-3.004 1.477h-9.725v18.389h42.762c.954 0 1.74.355 2.36 1.067.62.711.93 1.56.93 2.545v26.925h9.582c1.239 0 2.288.492 3.146 1.478z"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svg\user.svg

<svg width="130" height="130" xmlns="http://www.w3.org/2000/svg"><path d="M63.444 64.996c20.633 0 37.359-14.308 37.359-31.953 0-17.649-16.726-31.952-37.359-31.952-20.631 0-37.36 14.303-37.358 31.952 0 17.645 16.727 31.953 37.359 31.953zM80.57 75.65H49.434c-26.652 0-48.26 18.477-48.26 41.27v2.664c0 9.316 21.608 9.325 48.26 9.325H80.57c26.649 0 48.256-.344 48.256-9.325v-2.663c0-22.794-21.605-41.271-48.256-41.271z" stroke="#979797"/></svg>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\icons\svgo.yml

# replace default config

# multipass: true

# full: true

plugins:

# - name

#

# or:

# - name: false

# - name: true

#

# or:

# - name:

# param1: 1

# param2: 2

- removeAttrs:

attrs:

- 'fill'

- 'fill-rule'

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\AppMain.vue

<template>

<section class="app-main">

<transition name="fade-transform" mode="out-in">

<router-view :key="key" />

</transition>

</section>

</template>

<script>

export default {

name: 'AppMain',

computed: {

key() {

return this.$route.path

}

}

}

</script>

<style scoped>

.app-main {

/\*50 = navbar \*/

min-height: calc(100vh - 50px);

width: 100%;

position: relative;

overflow: hidden;

}

.fixed-header+.app-main {

padding-top: 50px;

}

</style>

<style lang="scss">

// fix css style bug in open el-dialog

.el-popup-parent--hidden {

.fixed-header {

padding-right: 15px;

}

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\index.js

export { default as Navbar } from './Navbar'

export { default as Sidebar } from './Sidebar'

export { default as AppMain } from './AppMain'

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\Navbar.vue

<template>

<div class="navbar">

<hamburger :is-active="sidebar.opened" class="hamburger-container" @toggleClick="toggleSideBar" />

<breadcrumb class="breadcrumb-container" />

<div class="right-menu">

<el-dropdown class="avatar-container" trigger="click">

<div class="avatar-wrapper">

<img :src="avatar+'?imageView2/1/w/80/h/80'" class="user-avatar">

<i class="el-icon-caret-bottom" />

</div>

<el-dropdown-menu slot="dropdown" class="user-dropdown">

<router-link to="/">

<el-dropdown-item>

返回主页

</el-dropdown-item>

</router-link>

<el-dropdown-item divided @click.native="logout">

<span style="display:block;">退出登录</span>

</el-dropdown-item>

</el-dropdown-menu>

</el-dropdown>

</div>

</div>

</template>

<script>

import { mapGetters } from 'vuex'

import Breadcrumb from '@/components/Breadcrumb'

import Hamburger from '@/components/Hamburger'

export default {

components: {

Breadcrumb,

Hamburger

},

computed: {

...mapGetters([

'sidebar',

'avatar'

])

},

methods: {

toggleSideBar() {

this.$store.dispatch('app/toggleSideBar')

},

async logout() {

await this.$store.dispatch('user/logout')

this.$router.push(`/index`)

}

}

}

</script>

<style lang="scss" scoped>

.navbar {

height: 50px;

overflow: hidden;

position: relative;

background: #fff;

box-shadow: 0 1px 4px rgba(0,21,41,.08);

.hamburger-container {

line-height: 46px;

height: 100%;

float: left;

cursor: pointer;

transition: background .3s;

-webkit-tap-highlight-color:transparent;

&:hover {

background: rgba(0, 0, 0, .025)

}

}

.breadcrumb-container {

float: left;

}

.right-menu {

float: right;

height: 100%;

line-height: 50px;

&:focus {

outline: none;

}

.right-menu-item {

display: inline-block;

padding: 0 8px;

height: 100%;

font-size: 18px;

color: #5a5e66;

vertical-align: text-bottom;

&.hover-effect {

cursor: pointer;

transition: background .3s;

&:hover {

background: rgba(0, 0, 0, .025)

}

}

}

.avatar-container {

margin-right: 30px;

.avatar-wrapper {

margin-top: 5px;

position: relative;

.user-avatar {

cursor: pointer;

width: 40px;

height: 40px;

border-radius: 10px;

}

.el-icon-caret-bottom {

cursor: pointer;

position: absolute;

right: -20px;

top: 25px;

font-size: 12px;

}

}

}

}

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\Sidebar\FixiOSBug.js

export default {

computed: {

device() {

return this.$store.state.app.device

}

},

mounted() {

// In order to fix the click on menu on the ios device will trigger the mouseleave bug

// https://github.com/PanJiaChen/vue-element-admin/issues/1135

this.fixBugIniOS()

},

methods: {

fixBugIniOS() {

const $subMenu = this.$refs.subMenu

if ($subMenu) {

const handleMouseleave = $subMenu.handleMouseleave

$subMenu.handleMouseleave = (e) => {

if (this.device === 'mobile') {

return

}

handleMouseleave(e)

}

}

}

}

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\Sidebar\index.vue

<template>

<div :class="{'has-logo':showLogo}">

<logo v-if="showLogo" :collapse="isCollapse" />

<el-scrollbar wrap-class="scrollbar-wrapper">

<el-menu

:default-active="activeMenu"

:collapse="isCollapse"

:background-color="variables.menuBg"

:text-color="variables.menuText"

:unique-opened="true"

:active-text-color="variables.menuActiveText"

:collapse-transition="false"

mode="vertical"

>

<sidebar-item v-for="route in routes" :key="route.path" :item="route" :base-path="route.path" />

</el-menu>

</el-scrollbar>

</div>

</template>

<script>

import { mapGetters } from 'vuex'

import Logo from './Logo'

import SidebarItem from './SidebarItem'

import variables from '@/styles/variables.scss'

export default {

components: { SidebarItem, Logo },

computed: {

...mapGetters([

'sidebar'

]),

routes() {

return this.$router.options.routes

},

activeMenu() {

const route = this.$route

const { meta, path } = route

// if set path, the sidebar will highlight the path you set

if (meta.activeMenu) {

return meta.activeMenu

}

return path

},

showLogo() {

return this.$store.state.settings.sidebarLogo

},

variables() {

return variables

},

isCollapse() {

return !this.sidebar.opened

}

}

}

</script>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\Sidebar\Item.vue

<script>

export default {

name: 'MenuItem',

functional: true,

props: {

icon: {

type: String,

default: ''

},

title: {

type: String,

default: ''

}

},

render(h, context) {

const { icon, title } = context.props

const vnodes = []

if (icon) {

if (icon.includes('el-icon')) {

vnodes.push(<i class={[icon, 'sub-el-icon']} />)

} else {

vnodes.push(<svg-icon icon-class={icon}/>)

}

}

if (title) {

vnodes.push(<span slot='title'>{(title)}</span>)

}

return vnodes

}

}

</script>

<style scoped>

.sub-el-icon {

color: currentColor;

width: 1em;

height: 1em;

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\Sidebar\Link.vue

<template>

<component :is="type" v-bind="linkProps(to)">

<slot />

</component>

</template>

<script>

import { isExternal } from '@/utils/validate'

export default {

props: {

to: {

type: String,

required: true

}

},

computed: {

isExternal() {

return isExternal(this.to)

},

type() {

if (this.isExternal) {

return 'a'

}

return 'router-link'

}

},

methods: {

linkProps(to) {

if (this.isExternal) {

return {

href: to,

target: '\_blank',

rel: 'noopener'

}

}

return {

to: to

}

}

}

}

</script>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\Sidebar\Logo.vue

<template>

<div class="sidebar-logo-container" :class="{'collapse':collapse}">

<transition name="sidebarLogoFade">

<router-link v-if="collapse" key="collapse" class="sidebar-logo-link" to="/">

<img v-if="logo" :src="logo" class="sidebar-logo">

<h1 v-else class="sidebar-title">{{ title }} </h1>

</router-link>

<router-link v-else key="expand" class="sidebar-logo-link" to="/">

<img v-if="logo" :src="logo" class="sidebar-logo">

<h1 class="sidebar-title">{{ title }} </h1>

</router-link>

</transition>

</div>

</template>

<script>

export default {

name: 'SidebarLogo',

props: {

collapse: {

type: Boolean,

required: true

}

},

data() {

return {

title: 'Vue Admin Template',

logo: 'https://wpimg.wallstcn.com/69a1c46c-eb1c-4b46-8bd4-e9e686ef5251.png'

}

}

}

</script>

<style lang="scss" scoped>

.sidebarLogoFade-enter-active {

transition: opacity 1.5s;

}

.sidebarLogoFade-enter,

.sidebarLogoFade-leave-to {

opacity: 0;

}

.sidebar-logo-container {

position: relative;

width: 100%;

height: 50px;

line-height: 50px;

background: #2b2f3a;

text-align: center;

overflow: hidden;

& .sidebar-logo-link {

height: 100%;

width: 100%;

& .sidebar-logo {

width: 32px;

height: 32px;

vertical-align: middle;

margin-right: 12px;

}

& .sidebar-title {

display: inline-block;

margin: 0;

color: #fff;

font-weight: 600;

line-height: 50px;

font-size: 14px;

font-family: Avenir, Helvetica Neue, Arial, Helvetica, sans-serif;

vertical-align: middle;

}

}

&.collapse {

.sidebar-logo {

margin-right: 0px;

}

}

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\components\Sidebar\SidebarItem.vue

<template>

<div v-if="!item.hidden">

<template v-if="hasOneShowingChild(item.children,item) && (!onlyOneChild.children||onlyOneChild.noShowingChildren)&&!item.alwaysShow">

<app-link v-if="onlyOneChild.meta" :to="resolvePath(onlyOneChild.path)">

<el-menu-item :index="resolvePath(onlyOneChild.path)" :class="{'submenu-title-noDropdown':!isNest}">

<item :icon="onlyOneChild.meta.icon||(item.meta&&item.meta.icon)" :title="onlyOneChild.meta.title" />

</el-menu-item>

</app-link>

</template>

<el-submenu v-else ref="subMenu" :index="resolvePath(item.path)" popper-append-to-body>

<template slot="title">

<item v-if="item.meta" :icon="item.meta && item.meta.icon" :title="item.meta.title" />

</template>

<sidebar-item

v-for="child in item.children"

:key="child.path"

:is-nest="true"

:item="child"

:base-path="resolvePath(child.path)"

class="nest-menu"

/>

</el-submenu>

</div>

</template>

<script>

import path from 'path'

import { isExternal } from '@/utils/validate'

import Item from './Item'

import AppLink from './Link'

import FixiOSBug from './FixiOSBug'

export default {

name: 'SidebarItem',

components: { Item, AppLink },

mixins: [FixiOSBug],

props: {

// route object

item: {

type: Object,

required: true

},

isNest: {

type: Boolean,

default: false

},

basePath: {

type: String,

default: ''

}

},

data() {

// To fix https://github.com/PanJiaChen/vue-admin-template/issues/237

// TODO: refactor with render function

this.onlyOneChild = null

return {}

},

methods: {

hasOneShowingChild(children = [], parent) {

const showingChildren = children.filter(item => {

if (item.hidden) {

return false

} else {

// Temp set(will be used if only has one showing child)

this.onlyOneChild = item

return true

}

})

// When there is only one child router, the child router is displayed by default

if (showingChildren.length === 1) {

return true

}

// Show parent if there are no child router to display

if (showingChildren.length === 0) {

this.onlyOneChild = { ... parent, path: '', noShowingChildren: true }

return true

}

return false

},

resolvePath(routePath) {

if (isExternal(routePath)) {

return routePath

}

if (isExternal(this.basePath)) {

return this.basePath

}

return path.resolve(this.basePath, routePath)

}

}

}

</script>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\index.vue

<template>

<div :class="classObj" class="app-wrapper">

<div v-if="device==='mobile'&&sidebar.opened" class="drawer-bg" @click="handleClickOutside" />

<sidebar class="sidebar-container" />

<div class="main-container">

<div :class="{'fixed-header':fixedHeader}">

<navbar />

</div>

<app-main />

</div>

</div>

</template>

<script>

import { Navbar, Sidebar, AppMain } from './components'

import ResizeMixin from './mixin/ResizeHandler'

export default {

name: 'Layout',

components: {

Navbar,

Sidebar,

AppMain

},

mixins: [ResizeMixin],

computed: {

sidebar() {

return this.$store.state.app.sidebar

},

device() {

return this.$store.state.app.device

},

fixedHeader() {

return this.$store.state.settings.fixedHeader

},

classObj() {

return {

hideSidebar: !this.sidebar.opened,

openSidebar: this.sidebar.opened,

withoutAnimation: this.sidebar.withoutAnimation,

mobile: this.device === 'mobile'

}

}

},

methods: {

handleClickOutside() {

this.$store.dispatch('app/closeSideBar', { withoutAnimation: false })

}

}

}

</script>

<style lang="scss" scoped>

@import "~@/styles/mixin.scss";

@import "~@/styles/variables.scss";

.app-wrapper {

@include clearfix;

position: relative;

height: 100%;

width: 100%;

&.mobile.openSidebar{

position: fixed;

top: 0;

}

}

.drawer-bg {

background: #000;

opacity: 0.3;

width: 100%;

top: 0;

height: 100%;

position: absolute;

z-index: 999;

}

.fixed-header {

position: fixed;

top: 0;

right: 0;

z-index: 9;

width: calc(100% - #{$sideBarWidth});

transition: width 0.28s;

}

.hideSidebar .fixed-header {

width: calc(100% - 54px)

}

.mobile .fixed-header {

width: 100%;

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\layout\mixin\ResizeHandler.js

import store from '@/store'

const { body } = document

const WIDTH = 992 // refer to Bootstrap's responsive design

export default {

watch: {

$route(route) {

if (this.device === 'mobile' && this.sidebar.opened) {

store.dispatch('app/closeSideBar', { withoutAnimation: false })

}

}

},

beforeMount() {

window.addEventListener('resize', this.$\_resizeHandler)

},

beforeDestroy() {

window.removeEventListener('resize', this.$\_resizeHandler)

},

mounted() {

const isMobile = this.$\_isMobile()

if (isMobile) {

store.dispatch('app/toggleDevice', 'mobile')

store.dispatch('app/closeSideBar', { withoutAnimation: true })

}

},

methods: {

// use $\_ for mixins properties

// https://vuejs.org/v2/style-guide/index.html#Private-property-names-essential

$\_isMobile() {

const rect = body.getBoundingClientRect()

return rect.width - 1 < WIDTH

},

$\_resizeHandler() {

if (!document.hidden) {

const isMobile = this.$\_isMobile()

store.dispatch('app/toggleDevice', isMobile ? 'mobile' : 'desktop')

if (isMobile) {

store.dispatch('app/closeSideBar', { withoutAnimation: true })

}

}

}

}

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\main.js

import Vue from 'vue'

import 'normalize.css/normalize.css' // A modern alternative to CSS resets

import ElementUI from 'element-ui'

import 'element-ui/lib/theme-chalk/index.css'

// import locale from 'element-ui/lib/locale/lang/en' // lang i18n

import '@/styles/index.scss' // global css

import App from './App'

import store from './store'

import router from './router'

import '@/icons' // icon

import '@/permission' // permission control

import Axios from 'axios'

import VueParticles from 'vue-particles'

Vue.use(VueParticles)

Vue.use(ElementUI)

Vue.config.productionTip = false

Axios.defaults.withCredentials = true

new Vue({

el: '#app',

router,

store,

render: h => h(App)

})

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\permission.js

import router from './router'

import store from './store'

import { Message } from 'element-ui'

import NProgress from 'nprogress' // progress bar

import 'nprogress/nprogress.css' // progress bar style

import { getToken } from '@/utils/auth' // get token from cookie

import getPageTitle from '@/utils/get-page-title'

NProgress.configure({ showSpinner: false }) // NProgress Configuration

const whiteList = ['/', '/article', '/login', '/404'] // no redirect whitelist

router.beforeEach(async (to, from, next) => {

// console.log('进入beforeeach, 目标是' + to.path);

// 开始登录的进度条

NProgress.start()

document.title = getPageTitle(to.meta.title)

// 判断token

const hasToken = getToken()

// 已登录

if (hasToken) {

if (to.path === '/login') {

// 即将跳转到 登录 页面, 如果用户已经登录,跳转到 dashboard

next({ path: '/' })

NProgress.done()

} else {

// 跳转到其他页面

const hasGetUserInfo = store.getters.name

if (hasGetUserInfo !== "") {

// 已经拉取用户 -部分- 信息

next()

} else {

// 未获取用户信息,开始获取用户信息并生成可访问的路由表

try {

await store.dispatch('user/getInfo')

const roles = store.getters.roles

store.dispatch('GenerateRoutes', { roles }).then(() => { // 生成可访问的路由表

// 动态添加访问路由表

const routersToAppend = store.getters.addRouters

routersToAppend.forEach(item => {

router.options.routes.push(item)

})

router.addRoutes(routersToAppend)

// hack方法 确保addRoutes已完成

next({ ...to, replace: true })

})

next()

} catch (error) {

// remove token and go to login page to re-login

await store.dispatch('user/resetToken')

Message.error(error || '出现错误')

next(`/login?redirect=${to.path}`)

NProgress.done()

}

}

}

} else {

// 未登录，请求需要登录的页面

if (whiteList.indexOf(to.path) === -1) {

// 跳转到登录页面

next(`/login?redirect=${to.path}`)

}

// 未登录, 判断未登录的情况下能不能到达

if (whiteList.indexOf(to.path) !== -1) {

// 跳到目标页面

next()

} else {

// 没有到达这个页面的权限,

next(`/404`)

NProgress.done()

}

}

})

router.afterEach(() => {

// finish progress bar

NProgress.done()

})

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\router\index.js

import Vue from 'vue'

import Router from 'vue-router'

Vue.use(Router)

/\* Layout \*/

import Layout from '@/layout'

//所有权限通用路由表

//如首页和登录页和一些不用权限的公用页面

export const constantRouterMap = [

{

path: '/',

component: () => import('@/views/index/index'),

hidden: true

},

{

path: '/404',

component: () => import('@/views/404'),

hidden: true

},

{

path: '/deleted-article',

component: () => import('@/views/index/deleted-article'),

hidden: true

},

{

path: '/article',

component: () => import('@/views/index/article'),

hidden: true

},

{

path: '/login',

component: () => import('@/views/login/index'),

hidden: true

},

{

path: '/dashboard',

component: Layout,

children: [{

path: '/dashboard',

name: 'Dashboard', // 路由的名字

component: () => import('@/views/dashboard/index'),

meta: { title: '数据面板', icon: 'dashboard' }

}]

}

]

//实例化vue的时候只挂载constantRouter

export default new Router({

routes: constantRouterMap

});

//异步挂载的路由

//动态加载需要根据权限加载的路由表

export const asyncRouterMap = [

{

path: '/write',

component: Layout,

redirect: '/write',

name: 'Write',

meta: { title: '写博客', icon: 'el-icon-edit', role: ['admin', 'user'] },

children: [

{

path: 'write',

name: 'Write',

component: () => import('@/views/write/write'),

meta: { title: '写博客', icon: 'el-icon-edit', role: ['admin', 'user'] }

}

]

},

{

path: '/personal-info',

component: Layout,

redirect: '/personal-info',

name: 'Write',

meta: { title: '个人信息', icon: 'el-icon-user', role: ['admin', 'user'] },

children: [

{

path: 'personal-info',

name: 'personal-info',

component: () => import('@/views/personal/personal-setting'),

meta: { title: '个人信息', icon: 'el-icon-user', role: ['admin', 'user'] }

},

{

path: 'change-password',

name: 'change-password',

component: () => import('@/views/personal/change-password'),

meta: { title: '修改密码', icon: 'el-icon-key', role: ['admin', 'user'] }

},

{

path: 'login-log',

name: 'login-log',

component: () => import('@/views/personal/login-log'),

meta: { title: '登录记录', icon: 'el-icon-position', role: ['admin', 'user'] }

},

]

},

{

path: '/article-manage',

component: Layout,

redirect: '/article-manage',

name: 'Article',

meta: { title: '文章管理', icon: 'el-icon-document', role: ['admin', 'user'] },

children: [

{

path: 'personal-article',

name: 'personal-article',

component: () => import('@/views/article/personal-article'),

meta: { title: '个人文章', icon: 'el-icon-document', role: ['admin', 'user'] }

},

{

path: 'personal-deleted',

name: 'personal-deleted',

component: () => import('@/views/article/personal-deleted'),

meta: { title: '我删除的文章', icon: 'el-icon-delete', role: ['admin', 'user'] }

},

{

path: 'all-article',

name: 'all-article',

component: () => import('@/views/article/all-article'),

meta: { title: '全站文章', icon: 'el-icon-document', role: ['admin'] }

},

{

path: 'all-deleted',

name: 'all-deleted',

component: () => import('@/views/article/all-deleted'),

meta: { title: '所有已删除文章', icon: 'el-icon-delete', role: ['admin'] }

},

]

},

{

path: '/comment',

component: Layout,

redirect: '/comment/personal-comment',

name: 'Comment',

meta: {

title: '评论管理',

icon: 'el-icon-chat-line-square',

role: ['admin', 'user']

},

children: [

{

path: 'comment-received',

component: () => import('@/views/comment/comment-received'), // Parent router-view

name: 'receive',

meta: { title: '我收到的评论', icon: 'el-icon-chat-dot-round', role: ['admin', 'user'] },

},

{

path: 'comment-posted',

component: () => import('@/views/comment/comment-posted'), // Parent router-view

name: 'personal',

meta: { title: '我发表的评论', icon: 'el-icon-chat-dot-round', role: ['admin', 'user'] },

},

{

path: 'all-comment',

component: () => import('@/views/comment/all-comment'),

name: 'all',

meta: { title: '所有评论', icon: 'el-icon-chat-dot-round', role: ['admin'] }

}

]

},

{

path: '/user',

component: Layout,

redirect: '/user',

name: 'User',

meta: { title: '用户管理', icon: 'el-icon-user', role: ['admin'] },

children: [

{

path: 'user',

name: 'User',

component: () => import('@/views/user/user'),

meta: { title: '用户管理', icon: 'el-icon-user', role: ['admin'] }

}

]

},

{

path: '/settings',

component: Layout,

redirect: '/settings',

name: 'settings',

meta: { title: '设置', icon: 'el-icon-setting', role: ['admin', 'user'] },

children: [

{

path: 'process',

name: 'Process',

component: () => import('@/views/settings/process'),

meta: { title: '开发历程', icon: 'el-icon-goblet-square-full', role: ['admin'] }

}

]

},

// 404 page must be placed at the end !!!

{ path: '\*', redirect: '/404', hidden: true }

];

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\settings.js

module.exports = {

title: 'JancoBlog',

/\*\*

\* @type {boolean} true | false

\* @description Whether fix the header

\*/

fixedHeader: false,

/\*\*

\* @type {boolean} true | false

\* @description Whether show the logo in sidebar

\*/

sidebarLogo: false

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\store\getters.js

const getters = {

sidebar: state => state.app.sidebar,

device: state => state.app.device,

token: state => state.user.token,

id: state => state.user.id,

name: state => state.user.name,

roles: state => state.user.roles,

avatar: state => state.user.avatar,

addRouters: state => state.permission.addRouters

}

export default getters

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\store\index.js

import Vue from 'vue'

import Vuex from 'vuex'

import getters from './getters'

import app from './modules/app'

import settings from './modules/settings'

import user from './modules/user'

import permission from './modules/permission'

Vue.use(Vuex)

const store = new Vuex.Store({

modules: {

app,

settings,

user,

permission

},

getters

})

export default store

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\store\modules\app.js

import Cookies from 'js-cookie'

const state = {

sidebar: {

opened: Cookies.get('sidebarStatus') ? !!+Cookies.get('sidebarStatus') : true,

withoutAnimation: false

},

device: 'desktop'

}

const mutations = {

TOGGLE\_SIDEBAR: state => {

state.sidebar.opened = !state.sidebar.opened

state.sidebar.withoutAnimation = false

if (state.sidebar.opened) {

Cookies.set('sidebarStatus', 1)

} else {

Cookies.set('sidebarStatus', 0)

}

},

CLOSE\_SIDEBAR: (state, withoutAnimation) => {

Cookies.set('sidebarStatus', 0)

state.sidebar.opened = false

state.sidebar.withoutAnimation = withoutAnimation

},

TOGGLE\_DEVICE: (state, device) => {

state.device = device

}

}

const actions = {

toggleSideBar({ commit }) {

commit('TOGGLE\_SIDEBAR')

},

closeSideBar({ commit }, { withoutAnimation }) {

commit('CLOSE\_SIDEBAR', withoutAnimation)

},

toggleDevice({ commit }, device) {

commit('TOGGLE\_DEVICE', device)

}

}

export default {

namespaced: true,

state,

mutations,

actions

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\store\modules\permission.js

import { asyncRouterMap, constantRouterMap } from '../../router/index';

/\*\*

\* 判断权限role能不能拿到route路由

\* @param {roles} roles

\* @param {route} route

\* @returns

\*/

function hasPermission(roles, route) {

if (route.meta && route.meta.role) {

// 路由表的此一级节点有该权限

return route.meta.role.indexOf(roles) >= 0

} else {

return true

}

}

const permission = {

state: {

routers: constantRouterMap,

addRouters: []

},

mutations: {

SET\_ROUTERS: (state, routers) => {

state.addRouters = routers;

state.routers = constantRouterMap.concat(routers);

}

},

actions: {

// 生成路由表

GenerateRoutes({ commit }, {roles}) {

return new Promise(resolve => {

// 异步路由表, 从异步路由表中过滤

const accessedRouters = asyncRouterMap.filter(v => {

// admin 有根节点的所有权限

if (roles.indexOf('admin') >= 0) return true;

// 判断子节点能不能拿到本路由

if (hasPermission(roles, v)) {

// 有一级权限, 判断二级权限

if (v.children && v.children.length > 0) {

// 有子节点

v.children = v.children.filter(child => {

// 过滤子节点路由

if (hasPermission(roles, child)) {

return child

}

return false;

});

return v

} else {

return v

}

}

return false;

});

commit('SET\_ROUTERS', accessedRouters);

resolve();

})

}

}

};

export default permission;

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\store\modules\settings.js

import defaultSettings from '@/settings'

const { showSettings, fixedHeader, sidebarLogo } = defaultSettings

const state = {

showSettings: showSettings,

fixedHeader: fixedHeader,

sidebarLogo: sidebarLogo

}

const mutations = {

CHANGE\_SETTING: (state, { key, value }) => {

// eslint-disable-next-line no-prototype-builtins

if (state.hasOwnProperty(key)) {

state[key] = value

}

}

}

const actions = {

changeSetting({ commit }, data) {

commit('CHANGE\_SETTING', data)

}

}

export default {

namespaced: true,

state,

mutations,

actions

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\store\modules\user.js

import { login, logout, getUserInfo } from '@/api/user'

import { getToken, setToken, removeToken } from '@/utils/auth'

import { resetRouter } from '@/router'

const getDefaultState = () => {

return {

token: getToken(),

id: '',

name: '',

avatar: '',

roles: ''

}

}

const state = getDefaultState()

const mutations = {

RESET\_STATE: (state) => {

Object.assign(state, getDefaultState())

},

SET\_TOKEN: (state, token) => {

state.token = token

},

SET\_NAME: (state, name) => {

state.name = name

},

SET\_ID: (state, id) => {

state.id = id

},

SET\_ROLE: (state, roles) => {

state.roles = roles

},

SET\_AVATAR: (state, url) => {

state.avatar = url

}

}

const actions = {

// user login

login({ commit }, userInfo) {

const { username, password } = userInfo

return new Promise((resolve, reject) => {

login({ username: username.trim(), password: password }).then(response => {

// if(response.code === 200){

// this.$message({

// message: '登录失败',

// type: 'success',

// })

// }

commit('SET\_TOKEN', response.extend.token)

setToken(response.extend.token)

resolve()

}).catch(error => {

reject(error)

})

})

},

// get user info

getInfo({ commit }) {

return new Promise((resolve, reject) => {

getUserInfo().then(response => {

const { data } = response

if (null === data) {

return reject('验证失败，请重新登陆。')

}

// 获取安全的用户信息

const user = response.extend.user

const avatarUrl = 'http://localhost:8080/avatar/' + user.userName + '.png'

// const avatarUrl = 'http://101.201.64.102:8000/avatar/' + user.userName + '.png'

commit('SET\_NAME', user.userName)

commit('SET\_ID', user.userId)

commit('SET\_ROLE', user.roleName)

commit('SET\_AVATAR', avatarUrl)

resolve(data)

}).catch(error => {

reject(error)

})

})

},

// user logout

logout({ commit, state }) {

return new Promise((resolve, reject) => {

logout(state.token).then(() => {

removeToken() // must remove token first

resetRouter()

commit('RESET\_STATE')

resolve()

}).catch(error => {

reject(error)

})

})

},

// remove token

resetToken({ commit }) {

return new Promise(resolve => {

removeToken() // must remove token first

commit('RESET\_STATE')

resolve()

})

}

}

export default {

namespaced: true,

state,

mutations,

actions

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\styles\element-ui.scss

// cover some element-ui styles

.el-breadcrumb\_\_inner,

.el-breadcrumb\_\_inner a {

font-weight: 400 !important;

}

.el-upload {

input[type="file"] {

display: none !important;

}

}

.el-upload\_\_input {

display: none;

}

// to fixed https://github.com/ElemeFE/element/issues/2461

.el-dialog {

transform: none;

left: 0;

position: relative;

margin: 0 auto;

}

// refine element ui upload

.upload-container {

.el-upload {

width: 100%;

.el-upload-dragger {

width: 100%;

height: 200px;

}

}

}

// dropdown

.el-dropdown-menu {

a {

display: block

}

}

// to fix el-date-picker css style

.el-range-separator {

box-sizing: content-box;

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\styles\index.scss

@import './variables.scss';

@import './mixin.scss';

@import './transition.scss';

@import './element-ui.scss';

@import './sidebar.scss';

body {

height: 100%;

-moz-osx-font-smoothing: grayscale;

-webkit-font-smoothing: antialiased;

text-rendering: optimizeLegibility;

font-family: Helvetica Neue, Helvetica, PingFang SC, Hiragino Sans GB, Microsoft YaHei, Arial, sans-serif;

}

label {

font-weight: 700;

}

html {

height: 100%;

box-sizing: border-box;

}

#app {

height: 100%;

}

\*,

\*:before,

\*:after {

box-sizing: inherit;

}

a:focus,

a:active {

outline: none;

}

a,

a:focus,

a:hover {

cursor: pointer;

color: inherit;

text-decoration: none;

}

div:focus {

outline: none;

}

.clearfix {

&:after {

visibility: hidden;

display: block;

font-size: 0;

content: " ";

clear: both;

height: 0;

}

}

// main-container global css

.app-container {

padding: 20px;

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\styles\mixin.scss

@mixin clearfix {

&:after {

content: "";

display: table;

clear: both;

}

}

@mixin scrollBar {

&::-webkit-scrollbar-track-piece {

background: #d3dce6;

}

&::-webkit-scrollbar {

width: 6px;

}

&::-webkit-scrollbar-thumb {

background: #99a9bf;

border-radius: 20px;

}

}

@mixin relative {

position: relative;

width: 100%;

height: 100%;

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\styles\sidebar.scss

#app {

.main-container {

min-height: 100%;

transition: margin-left .28s;

margin-left: $sideBarWidth;

position: relative;

}

.sidebar-container {

transition: width 0.28s;

width: $sideBarWidth !important;

background-color: $menuBg;

height: 100%;

position: fixed;

font-size: 0px;

top: 0;

bottom: 0;

left: 0;

z-index: 1001;

overflow: hidden;

// reset element-ui css

.horizontal-collapse-transition {

transition: 0s width ease-in-out, 0s padding-left ease-in-out, 0s padding-right ease-in-out;

}

.scrollbar-wrapper {

overflow-x: hidden !important;

}

.el-scrollbar\_\_bar.is-vertical {

right: 0px;

}

.el-scrollbar {

height: 100%;

}

&.has-logo {

.el-scrollbar {

height: calc(100% - 50px);

}

}

.is-horizontal {

display: none;

}

a {

display: inline-block;

width: 100%;

overflow: hidden;

}

.svg-icon {

margin-right: 16px;

}

.sub-el-icon {

margin-right: 12px;

margin-left: -2px;

}

.el-menu {

border: none;

height: 100%;

width: 100% !important;

}

// menu hover

.submenu-title-noDropdown,

.el-submenu\_\_title {

&:hover {

background-color: $menuHover !important;

}

}

.is-active>.el-submenu\_\_title {

color: $subMenuActiveText !important;

}

& .nest-menu .el-submenu>.el-submenu\_\_title,

& .el-submenu .el-menu-item {

min-width: $sideBarWidth !important;

background-color: $subMenuBg !important;

&:hover {

background-color: $subMenuHover !important;

}

}

}

.hideSidebar {

.sidebar-container {

width: 54px !important;

}

.main-container {

margin-left: 54px;

}

.submenu-title-noDropdown {

padding: 0 !important;

position: relative;

.el-tooltip {

padding: 0 !important;

.svg-icon {

margin-left: 20px;

}

.sub-el-icon {

margin-left: 19px;

}

}

}

.el-submenu {

overflow: hidden;

&>.el-submenu\_\_title {

padding: 0 !important;

.svg-icon {

margin-left: 20px;

}

.sub-el-icon {

margin-left: 19px;

}

.el-submenu\_\_icon-arrow {

display: none;

}

}

}

.el-menu--collapse {

.el-submenu {

&>.el-submenu\_\_title {

&>span {

height: 0;

width: 0;

overflow: hidden;

visibility: hidden;

display: inline-block;

}

}

}

}

}

.el-menu--collapse .el-menu .el-submenu {

min-width: $sideBarWidth !important;

}

// mobile responsive

.mobile {

.main-container {

margin-left: 0px;

}

.sidebar-container {

transition: transform .28s;

width: $sideBarWidth !important;

}

&.hideSidebar {

.sidebar-container {

pointer-events: none;

transition-duration: 0.3s;

transform: translate3d(-$sideBarWidth, 0, 0);

}

}

}

.withoutAnimation {

.main-container,

.sidebar-container {

transition: none;

}

}

}

// when menu collapsed

.el-menu--vertical {

&>.el-menu {

.svg-icon {

margin-right: 16px;

}

.sub-el-icon {

margin-right: 12px;

margin-left: -2px;

}

}

.nest-menu .el-submenu>.el-submenu\_\_title,

.el-menu-item {

&:hover {

// you can use $subMenuHover

background-color: $menuHover !important;

}

}

// the scroll bar appears when the subMenu is too long

>.el-menu--popup {

max-height: 100vh;

overflow-y: auto;

&::-webkit-scrollbar-track-piece {

background: #d3dce6;

}

&::-webkit-scrollbar {

width: 6px;

}

&::-webkit-scrollbar-thumb {

background: #99a9bf;

border-radius: 20px;

}

}

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\styles\transition.scss

// global transition css

/\* fade \*/

.fade-enter-active,

.fade-leave-active {

transition: opacity 0.28s;

}

.fade-enter,

.fade-leave-active {

opacity: 0;

}

/\* fade-transform \*/

.fade-transform-leave-active,

.fade-transform-enter-active {

transition: all .5s;

}

.fade-transform-enter {

opacity: 0;

transform: translateX(-30px);

}

.fade-transform-leave-to {

opacity: 0;

transform: translateX(30px);

}

/\* breadcrumb transition \*/

.breadcrumb-enter-active,

.breadcrumb-leave-active {

transition: all .5s;

}

.breadcrumb-enter,

.breadcrumb-leave-active {

opacity: 0;

transform: translateX(20px);

}

.breadcrumb-move {

transition: all .5s;

}

.breadcrumb-leave-active {

position: absolute;

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\styles\variables.scss

// sidebar

$menuText:#bfcbd9;

$menuActiveText:#409EFF;

$subMenuActiveText:#f4f4f5; //https://github.com/ElemeFE/element/issues/12951

$menuBg:#304156;

$menuHover:#263445;

$subMenuBg:#1f2d3d;

$subMenuHover:#001528;

$sideBarWidth: 210px;

// the :export directive is the magic sauce for webpack

// https://www.bluematador.com/blog/how-to-share-variables-between-js-and-sass

:export {

menuText: $menuText;

menuActiveText: $menuActiveText;

subMenuActiveText: $subMenuActiveText;

menuBg: $menuBg;

menuHover: $menuHover;

subMenuBg: $subMenuBg;

subMenuHover: $subMenuHover;

sideBarWidth: $sideBarWidth;

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\utils\auth.js

import Cookies from 'js-cookie'

const TokenKey = 'token'

export function getToken() {

return Cookies.get(TokenKey)

}

export function setToken(token) {

return Cookies.set(TokenKey, token)

}

export function removeToken() {

return Cookies.remove(TokenKey)

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\utils\get-page-title.js

import defaultSettings from '@/settings'

const title = defaultSettings.title || '博客后台'

export default function getPageTitle(pageTitle) {

if (pageTitle) {

return `${pageTitle} - ${title}`

}

return `${title}`

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\utils\index.js

/\*\*

\* Created by PanJiaChen on 16/11/18.

\*/

/\*\*

\* Parse the time to string

\* @param {(Object|string|number)} time

\* @param {string} cFormat

\* @returns {string | null}

\*/

export function parseTime(time, cFormat) {

if (arguments.length === 0 || !time) {

return null

}

const format = cFormat || '{y}-{m}-{d} {h}:{i}:{s}'

let date

if (typeof time === 'object') {

date = time

} else {

if ((typeof time === 'string')) {

if ((/^[0-9]+$/.test(time))) {

// support "1548221490638"

time = parseInt(time)

} else {

// support safari

// https://stackoverflow.com/questions/4310953/invalid-date-in-safari

time = time.replace(new RegExp(/-/gm), '/')

}

}

if ((typeof time === 'number') && (time.toString().length === 10)) {

time = time \* 1000

}

date = new Date(time)

}

const formatObj = {

y: date.getFullYear(),

m: date.getMonth() + 1,

d: date.getDate(),

h: date.getHours(),

i: date.getMinutes(),

s: date.getSeconds(),

a: date.getDay()

}

const time\_str = format.replace(/{([ymdhisa])+}/g, (result, key) => {

const value = formatObj[key]

// Note: getDay() returns 0 on Sunday

if (key === 'a') { return ['日', '一', '二', '三', '四', '五', '六'][value ] }

return value.toString().padStart(2, '0')

})

return time\_str

}

/\*\*

\* @param {number} time

\* @param {string} option

\* @returns {string}

\*/

export function formatTime(time, option) {

if (('' + time).length === 10) {

time = parseInt(time) \* 1000

} else {

time = +time

}

const d = new Date(time)

const now = Date.now()

const diff = (now - d) / 1000

if (diff < 30) {

return '刚刚'

} else if (diff < 3600) {

// less 1 hour

return Math.ceil(diff / 60) + '分钟前'

} else if (diff < 3600 \* 24) {

return Math.ceil(diff / 3600) + '小时前'

} else if (diff < 3600 \* 24 \* 2) {

return '1天前'

}

if (option) {

return parseTime(time, option)

} else {

return (

d.getMonth() +

1 +

'月' +

d.getDate() +

'日' +

d.getHours() +

'时' +

d.getMinutes() +

'分'

)

}

}

/\*\*

\* @param {string} url

\* @returns {Object}

\*/

export function param2Obj(url) {

const search = decodeURIComponent(url.split('?')[1]).replace(/\+/g, ' ')

if (!search) {

return {}

}

const obj = {}

const searchArr = search.split('&')

searchArr.forEach(v => {

const index = v.indexOf('=')

if (index !== -1) {

const name = v.substring(0, index)

const val = v.substring(index + 1, v.length)

obj[name] = val

}

})

return obj

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\utils\request.js

import axios from 'axios'

import { MessageBox, Message } from 'element-ui'

import store from '@/store'

import { getToken } from '@/utils/auth'

// create an axios instance

const service = axios.create({

// baseURL: 'http://localhost:8000', // url = base url + request url

baseURL: 'http://101.201.64.102:8000', // url = base url + request url

// withCredentials: true, // send cookies when cross-domain requests

timeout: 5000 // request timeout

})

// request interceptor

service.interceptors.request.use(

config => {

// do something before request is sent

if (store.getters.token) {

config.headers['token'] = getToken()

}

return config

},

error => {

// do something with request error

console.log(error) // for debug

return Promise.reject(error)

}

)

// response interceptor

service.interceptors.response.use(

response => {

const res = response.data

// if the custom code is not 100, it is judged as an error.

if (res.code !== 100) {

// 数据CRUD问题，直接显示错误信息

if (res.code === 200) {

Message({

message: res.extend.msg || '数据错误',

type: 'error',

duration: 5 \* 1000

})

}

// 用户未登录

if (res.code === 603) {

Message({

message: res.msg || '用户未登录',

type: 'error',

duration: 5 \* 1000

})

}

// 用户信息相关错误，重新登录

if (res.code === 601 || res.code === 602) {

// to re-login

MessageBox.confirm('信息已过期，请重新登录', '重新登陆', {

confirmButtonText: '确认',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

store.dispatch('user/resetToken').then(() => {

// location.reload()

this.$router.push('/login')

})

})

}

return Promise.reject(new Error( res.msg || 'Error'))

} else {

return response.data

}

},

error => {

console.log('发生错误 ' + error) // for debug

Message({

message: error.msg || '网络错误',

type: 'error',

duration: 5 \* 1000

})

return Promise.reject(error)

}

)

export default service

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\utils\validate.js

/\*\*

\* Created by PanJiaChen on 16/11/18.

\*/

/\*\*

\* @param {string} path

\* @returns {Boolean}

\*/

export function isExternal(path) {

return /^(https?:|mailto:|tel:)/.test(path)

}

/\*\*

\* @param {string} str

\* @returns {Boolean}

\*/

export function validUsername(str) {

const valid\_map = ['admin', 'editor']

return valid\_map.indexOf(str.trim()) >= 0

}

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\404.vue

<template>

<div class="wscn-http404-container">

<div class="wscn-http404">

<div class="pic-404">

<img class="pic-404\_\_parent" src="@/assets/404\_images/404.png" alt="404">

<img class="pic-404\_\_child left" src="@/assets/404\_images/404\_cloud.png" alt="404">

<img class="pic-404\_\_child mid" src="@/assets/404\_images/404\_cloud.png" alt="404">

<img class="pic-404\_\_child right" src="@/assets/404\_images/404\_cloud.png" alt="404">

</div>

<div class="bullshit">

<div class="bullshit\_\_oops">找不到地址了!</div>

<div class="bullshit\_\_headline">{{ message }}</div>

<div class="bullshit\_\_info">请检查您的URL是否正确，或者点击下方按钮回到首页</div>

<a href="" class="bullshit\_\_return-home">回到首页</a>

</div>

</div>

</div>

</template>

<script>

export default {

name: 'Page404',

computed: {

message() {

return '页面不存在了...'

}

}

}

</script>

<style lang="scss" scoped>

.wscn-http404-container{

transform: translate(-50%,-50%);

position: absolute;

top: 40%;

left: 50%;

}

.wscn-http404 {

position: relative;

width: 1200px;

padding: 0 50px;

overflow: hidden;

.pic-404 {

position: relative;

float: left;

width: 600px;

overflow: hidden;

&\_\_parent {

width: 100%;

}

&\_\_child {

position: absolute;

&.left {

width: 80px;

top: 17px;

left: 220px;

opacity: 0;

animation-name: cloudLeft;

animation-duration: 2s;

animation-timing-function: linear;

animation-fill-mode: forwards;

animation-delay: 1s;

}

&.mid {

width: 46px;

top: 10px;

left: 420px;

opacity: 0;

animation-name: cloudMid;

animation-duration: 2s;

animation-timing-function: linear;

animation-fill-mode: forwards;

animation-delay: 1.2s;

}

&.right {

width: 62px;

top: 100px;

left: 500px;

opacity: 0;

animation-name: cloudRight;

animation-duration: 2s;

animation-timing-function: linear;

animation-fill-mode: forwards;

animation-delay: 1s;

}

@keyframes cloudLeft {

0% {

top: 17px;

left: 220px;

opacity: 0;

}

20% {

top: 33px;

left: 188px;

opacity: 1;

}

80% {

top: 81px;

left: 92px;

opacity: 1;

}

100% {

top: 97px;

left: 60px;

opacity: 0;

}

}

@keyframes cloudMid {

0% {

top: 10px;

left: 420px;

opacity: 0;

}

20% {

top: 40px;

left: 360px;

opacity: 1;

}

70% {

top: 130px;

left: 180px;

opacity: 1;

}

100% {

top: 160px;

left: 120px;

opacity: 0;

}

}

@keyframes cloudRight {

0% {

top: 100px;

left: 500px;

opacity: 0;

}

20% {

top: 120px;

left: 460px;

opacity: 1;

}

80% {

top: 180px;

left: 340px;

opacity: 1;

}

100% {

top: 200px;

left: 300px;

opacity: 0;

}

}

}

}

.bullshit {

position: relative;

float: left;

width: 300px;

padding: 30px 0;

overflow: hidden;

&\_\_oops {

font-size: 32px;

font-weight: bold;

line-height: 40px;

color: #1482f0;

opacity: 0;

margin-bottom: 20px;

animation-name: slideUp;

animation-duration: 0.5s;

animation-fill-mode: forwards;

}

&\_\_headline {

font-size: 20px;

line-height: 24px;

color: #222;

font-weight: bold;

opacity: 0;

margin-bottom: 10px;

animation-name: slideUp;

animation-duration: 0.5s;

animation-delay: 0.1s;

animation-fill-mode: forwards;

}

&\_\_info {

font-size: 13px;

line-height: 21px;

color: grey;

opacity: 0;

margin-bottom: 30px;

animation-name: slideUp;

animation-duration: 0.5s;

animation-delay: 0.2s;

animation-fill-mode: forwards;

}

&\_\_return-home {

display: block;

float: left;

width: 110px;

height: 36px;

background: #1482f0;

border-radius: 100px;

text-align: center;

color: #ffffff;

opacity: 0;

font-size: 14px;

line-height: 36px;

cursor: pointer;

animation-name: slideUp;

animation-duration: 0.5s;

animation-delay: 0.3s;

animation-fill-mode: forwards;

}

@keyframes slideUp {

0% {

transform: translateY(60px);

opacity: 0;

}

100% {

transform: translateY(0);

opacity: 1;

}

}

}

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\article\all-article.vue

<template>

<div class="main">

<!-- 搜索框 -->

<el-col :span="12">

<!-- 组件 - 搜索文章的搜索框 -->

<search-article

ref="searchArticle"

:typeList="typeList"

:isAdmin="true"

@submit="submit"

@reset="resetForm"

></search-article>

</el-col>

<!-- 文章列表 -->

<el-table

:data="tableData"

border

style="width: 100%"

v-loading="loading"

@selection-change="handleSelectionChange"

>

<el-table-column

type="selection"

width="55"

>

</el-table-column>

<el-table-column

prop="articleTitle"

label="标题"

width="280"

>

</el-table-column>

<el-table-column

prop="userName"

label="作者"

width="120"

>

</el-table-column>

<el-table-column

label="类型"

width="120"

>

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.typeName }}</span>

</template>

</el-table-column>

<el-table-column

label="发表时间"

sortable

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.articlePostTime | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="articleEditTime"

label="修改时间"

sortable

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.articleEditTime | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="articleRank"

label="置顶"

width="100"

>

<template slot-scope="scope">

<el-switch

:value="scope.row.articleRank === 1"

@change="handleStickTopChange(scope.row)"

></el-switch>

</template>

</el-table-column>

<el-table-column

prop="articleViewCount"

label="浏览量"

sortable

width="100"

>

</el-table-column>

<el-table-column

prop="articleLikeCount"

label="点赞量"

sortable

width="100"

>

</el-table-column>

<el-table-column

prop="articleCommentCount"

label="评论量"

sortable

width="100"

>

</el-table-column>

<el-table-column

label="操作"

width="120"

>

<template slot-scope="scope">

<el-button

type="text"

size="small"

><a

href="http://localhost:8080"

target="\_blank"

>查看</a></el-button>

<el-button

@click="deleteArticle(scope.row)"

type="text"

size="small"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<div class="btn-group">

<el-button

type="danger"

@click="batchDelete"

>删除选中</el-button>

</div>

<!-- 分页 -->

<div class="pagiation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="pn"

:page-sizes="[10, 20, 30, 40]"

:page-size="limit"

layout="total, sizes, prev, pager, next, jumper"

:total="total"

>

</el-pagination>

</div>

</div>

</template>

<script>

import {

getAll,

batchDeleteArticles,

toggleArticleStickTop,

} from '@/api/article'

import SearchArticle from '@/components/SearchArticle'

import { getAllType } from '@/api/type'

export default {

components: { SearchArticle },

data() {

return {

loading: false,

typeList: [],

tableData: [],

pn: 1,

limit: 10,

total: 0,

multipleSelection: [],

condition: '',

}

},

created() {

this.get\_article\_list(1)

this.get\_type\_list()

},

filters: {

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

},

},

methods: {

// 置顶

handleStickTopChange(col) {

toggleArticleStickTop(col.articleId).then((res) => {

col.articleRank = 1 - col.articleRank

this.$message({

message: '成功',

type: 'success',

})

})

},

// 分页大小

handleSizeChange(val) {

this.limit = val

this.get\_article\_list(1)

},

// 页面跳转

handleCurrentChange(val) {

this.get\_article\_list(val)

},

// 删除文章

deleteArticle(row) {

var msg = '将要删除文章 <' + row.articleTitle + '> , 是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteArticles(row.articleId).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

})

})

},

// 获取文章列表

get\_article\_list(pn) {

this.loading = true

getAll(pn, this.limit, this.condition).then((response) => {

var pageInfo = response.extend.pageInfo

this.tableData = pageInfo.records

this.total = pageInfo.total

})

this.loading = false

},

// 获取类型列表

get\_type\_list() {

getAllType().then((response) => {

this.typeList = response.extend.pageInfo.records

})

},

//多选

handleSelectionChange(val) {

this.multipleSelection = val

},

// 批量删除

batchDelete() {

var ids = ''

if (this.multipleSelection.length === 0) {

this.$message.error('请先选择')

return

}

// 字符串拼接

this.multipleSelection.forEach((item) => {

ids += item.articleId + '&'

})

this.$confirm('确定删除选中文章？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteArticles(ids).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

duration: 1000,

})

})

},

// 提交搜索

submit() {

this.condition = this.$refs.searchArticle.generateQueryString()

this.get\_article\_list(1)

},

// 文章列表重置

resetForm() {

this.condition = ''

this.get\_article\_list(1)

},

},

}

</script>

<style scoped>

.pagiation {

margin-top: 30px;

margin-left: 30px;

}

.btn-group {

margin: 20px;

}

.main {

margin: 20px;

}

.search-bar {

margin-bottom: 20px;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\article\all-deleted.vue

<template>

<div class="main">

<!-- 搜索框 -->

<el-col :span="12">

<!-- 组件 - 搜索文章的搜索框 -->

<search-article

ref="searchArticle"

:typeList="typeList"

:isAdmin="true"

@submit="submit"

@reset="resetForm"

></search-article>

</el-col>

<!-- 文章表格 -->

<el-table

:data="tableData"

border

style="width: 100%"

v-loading="loading"

@selection-change="handleSelectionChange"

>

<el-table-column

type="selection"

width="55"

>

</el-table-column>

<el-table-column

prop="articleTitle"

label="标题"

width="280"

>

</el-table-column>

<el-table-column

prop="userName"

label="作者"

width="120"

>

</el-table-column>

<el-table-column

label="类型"

width="120"

>

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.typeName }}</span>

</template>

</el-table-column>

<el-table-column

label="发表时间"

sortable

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.articlePostTime | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="articleEditTime"

label="删除时间"

sortable

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.articleEditTime | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="articleViewCount"

label="浏览量"

sortable

width="100"

>

</el-table-column>

<el-table-column

prop="articleLikeCount"

label="点赞量"

sortable

width="100"

>

</el-table-column>

<el-table-column

prop="articleCommentCount"

label="评论量"

sortable

width="100"

>

</el-table-column>

<el-table-column

label="操作"

width="220"

>

<template slot-scope="scope">

<el-button

type="text"

size="small"

><a

href="http://localhost:8080"

target="\_blank"

>查看</a></el-button>

<el-button

@click="deleteArticle(scope.row)"

type="text"

size="small"

style="color:#f00"

>彻底删除</el-button>

<el-button

@click="recoverArticle(scope.row)"

type="text"

size="small"

style="color:#5f3"

>恢复</el-button>

</template>

</el-table-column>

</el-table>

<div class="btn-group">

<el-button

type="danger"

@click="batchDelete"

>删除选中</el-button>

<el-button

type="success"

@click="batchRecover"

>恢复选中</el-button>

</div>

<!-- 分页 -->

<div class="pagiation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="pn"

:page-sizes="[10, 20, 30, 40]"

:page-size="limit"

layout="total, sizes, prev, pager, next, jumper"

:total="total"

>

</el-pagination>

</div>

</div>

</template>

<script>

import {

getAllDeleted,

batchDeleteDeletedArticles,

batchRecoverArticles,

} from '@/api/article'

import { getAllType } from '@/api/type'

import SearchArticle from '@/components/SearchArticle'

export default {

components: { SearchArticle },

data() {

return {

loading: false,

typeList: [],

tableData: [],

pn: 1,

limit: 10,

total: 0,

multipleSelection: [],

condition: '',

}

},

created() {

this.get\_article\_list(1)

this.get\_type\_list()

},

filters: {

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

},

},

methods: {

// 分页的一页的大小

handleSizeChange(val) {

this.limit = val

this.get\_article\_list(1)

},

// 跳转到某一页

handleCurrentChange(val) {

this.get\_article\_list(val)

},

// 删除文章

deleteArticle(row) {

var msg =

'将要彻底删除文章 <' + row.articleTitle + '> ！不可恢复！是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteDeletedArticles(row.articleId).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

})

})

},

// 恢复文章

recoverArticle(row) {

var msg = '将要恢复文章 <' + row.articleTitle + '> 是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'success',

})

.then(() => {

batchRecoverArticles(row.articleId).then((response) => {

if (response.extend.suc === true) {

this.$message({

type: 'success',

message: '恢复成功！',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('恢复失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消',

})

})

},

// 获取文章列表

get\_article\_list(pn) {

this.loading = true

getAllDeleted(pn, this.limit, this.condition).then((response) => {

var pageInfo = response.extend.pageInfo

this.tableData = pageInfo.records

this.total = pageInfo.total

})

this.loading = false

},

// 获取类型列表

get\_type\_list() {

getAllType().then((response) => {

this.typeList = response.extend.pageInfo.records

})

},

// 多选

handleSelectionChange(val) {

this.multipleSelection = val

},

// 批量删除

batchDelete() {

var ids = ''

if (this.multipleSelection.length === 0) {

this.$message.error('请先选择')

return

}

// 字符串拼接

this.multipleSelection.forEach((item) => {

ids += item.articleId + '&'

})

this.$confirm('确定删除选中文章？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteDeletedArticles(ids).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

duration: 1000,

})

})

},

// 批量恢复

batchRecover() {

var ids = ''

if (this.multipleSelection.length === 0) {

this.$message.error('请先选择需要恢复的文章')

return

}

// 字符串拼接

this.multipleSelection.forEach((item) => {

ids += item.articleId + '&'

})

this.$confirm('确定恢复选中文章？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'success',

})

.then(() => {

batchRecoverArticles(ids).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '恢复成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('恢复失败，请稍后再试!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消恢复',

duration: 1000,

})

})

},

// 提交搜索

submit() {

this.condition = this.$refs.searchArticle.generateQueryString()

this.get\_article\_list(1)

},

// 文章列表重置

resetForm() {

this.condition = ''

this.get\_article\_list(1)

},

},

}

</script>

<style scoped>

.pagiation {

margin-top: 30px;

margin-left: 30px;

}

.btn-group {

margin: 20px;

}

.main {

margin: 20px;

}

.search-bar {

margin-bottom: 20px;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\article\personal-article.vue

<template>

<div class="main">

<!-- 搜索文章的搜索框 -->

<el-col :span="12">

<search-article

ref="searchArticle"

:typeList="typeList"

@submit="submit"

@reset="resetForm"

></search-article>

</el-col>

<!-- 文章数据表格 -->

<el-table

:data="tableData"

border

style="width: 100%"

v-loading="loading"

@selection-change="handleSelectionChange"

>

<el-table-column

type="selection"

width="55"

>

</el-table-column>

<el-table-column

prop="articleTitle"

label="标题"

width="280"

>

</el-table-column>

<el-table-column

label="类型"

width="120"

>

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.typeName }}</span>

</template>

</el-table-column>

<el-table-column

label="发表时间"

sortable

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.articlePostTime | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="articleEditTime"

label="修改时间"

sortable

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.articleEditTime | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="articleIsComment"

label="允许评论"

width="100"

>

<template slot-scope="scope">

<el-switch

:value="scope.row.articleIsComment === 1"

@change="handleIsCommentChange(scope.row)"

></el-switch>

</template>

</el-table-column>

<el-table-column

prop="articleViewCount"

label="浏览量"

sortable

width="100"

>

</el-table-column>

<el-table-column

prop="articleLikeCount"

label="点赞量"

sortable

width="100"

>

</el-table-column>

<el-table-column

prop="articleCommentCount"

label="评论量"

sortable

width="100"

>

</el-table-column>

<el-table-column

label="操作"

width="120"

>

<template slot-scope="scope">

<el-button

type="text"

size="small"

>

<router-link

:to="'/article?id=' + scope.row.articleId"

target="\_blank"

>查看</router-link>

</el-button>

<el-button

type="text"

size="small"

>

<router-link :to="'/write/write?id=' + scope.row.articleId">编辑</router-link>

</el-button>

<el-button

@click="deleteArticle(scope.row)"

type="text"

size="small"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<div class="btn-group">

<el-button

type="danger"

@click="batchDelete"

>删除选中</el-button>

</div>

<!-- 分页 -->

<div class="pagiation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="pn"

:page-sizes="[10, 20, 30, 40]"

:page-size="limit"

layout="total, sizes, prev, pager, next, jumper"

:total="total"

>

</el-pagination>

</div>

</div>

</template>

<script>

import {

getArticleByUser,

batchDeleteArticles,

toggleArticleIsComment,

} from '@/api/article'

import { getAllType } from '@/api/type'

import SearchArticle from '@/components/SearchArticle'

export default {

components: { SearchArticle },

data() {

return {

loading: false,

typeList: [],

tableData: [],

pn: 1,

limit: 10,

total: 0,

multipleSelection: [],

condition: '',

}

},

created() {

this.get\_article\_list(1)

this.get\_type\_list()

},

filters: {

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

},

},

methods: {

// 改变是否允许评论的状态

handleIsCommentChange(col) {

toggleArticleIsComment(col.articleId).then((res) => {

col.articleIsComment = 1 - col.articleIsComment

this.$message({

message: '成功',

type: 'success',

})

})

},

// 分页大小的改变

handleSizeChange(val) {

this.limit = val

this.get\_article\_list(1)

},

// 上一页

handleCurrentChange(val) {

this.get\_article\_list(val)

},

// 删除文章

deleteArticle(row) {

var msg = '将要删除文章 <' + row.articleTitle + '> , 是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteArticles(row.articleId).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

})

})

},

// 获取文章列表

get\_article\_list(pn) {

this.loading = true

getArticleByUser(pn, this.limit, this.condition).then((response) => {

var pageInfo = response.extend.pageInfo

this.tableData = pageInfo.records

this.total = pageInfo.total

})

this.loading = false

},

// 获取类型列表

get\_type\_list() {

getAllType().then((response) => {

this.typeList = response.extend.pageInfo.records

})

},

// 多选

handleSelectionChange(val) {

this.multipleSelection = val

},

// 批量删除

batchDelete() {

var ids = ''

if (this.multipleSelection.length === 0) {

this.$message.error('请先选择')

return

}

// 字符串拼接

this.multipleSelection.forEach((item) => {

ids += item.articleId + '&'

})

this.$confirm('确定删除选中文章？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteArticles(ids).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

duration: 1000,

})

})

},

// 提交搜索

submit() {

this.condition = this.$refs.searchArticle.generateQueryString()

this.get\_article\_list(1)

},

// 文章列表重置

resetForm() {

this.condition = ''

this.get\_article\_list(1)

},

},

}

</script>

<style scoped>

.pagiation {

margin-top: 30px;

margin-left: 30px;

}

.btn-group {

margin: 20px;

}

.main {

margin: 20px;

}

.search-bar {

margin-bottom: 20px;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\article\personal-deleted.vue

<template>

<div class="main">

<!-- 搜索框 -->

<el-col :span="12">

<!-- 组件 - 搜索文章的搜索框 -->

<search-article

ref="searchArticle"

:typeList="typeList"

@submit="submit"

@reset="resetForm"

></search-article>

</el-col>

<!-- 数据表格 -->

<el-table

:data="tableData"

border

style="width: 100%"

v-loading="loading"

@selection-change="handleSelectionChange"

>

<el-table-column

type="selection"

width="55"

>

</el-table-column>

<el-table-column

prop="articleTitle"

label="标题"

width="280"

>

</el-table-column>

<el-table-column

label="类型"

width="120"

>

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.typeName }}</span>

</template>

</el-table-column>

<el-table-column

label="发表时间"

sortable

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.articlePostTime | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="articleEditTime"

label="删除时间"

sortable

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.articleEditTime | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="articleViewCount"

label="浏览量"

sortable

width="100"

>

</el-table-column>

<el-table-column

prop="articleLikeCount"

label="点赞量"

sortable

width="100"

>

</el-table-column>

<el-table-column

prop="articleCommentCount"

label="评论量"

sortable

width="100"

>

</el-table-column>

<el-table-column

label="操作"

width="220"

>

<template slot-scope="scope">

<el-button

type="text"

size="small"

>

<router-link

:to="'/deleted-article?id=' + scope.row.articleId"

target="\_blank"

>查看</router-link>

</el-button>

<el-button

@click="deleteArticle(scope.row)"

type="text"

size="small"

style="color:#f00"

>彻底删除</el-button>

<el-button

@click="recoverArticle(scope.row)"

type="text"

size="small"

style="color:#5f3"

>恢复</el-button>

</template>

</el-table-column>

</el-table>

<div class="btn-group">

<el-button

type="danger"

@click="batchDelete"

>删除选中</el-button>

<el-button

type="success"

@click="batchRecover"

>恢复选中</el-button>

</div>

<!-- 分页 -->

<div class="pagiation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="pn"

:page-sizes="[10, 20, 30, 40]"

:page-size="limit"

layout="total, sizes, prev, pager, next, jumper"

:total="total"

>

</el-pagination>

</div>

</div>

</template>

<script>

import {

getArticleByUserDeleted,

batchDeleteDeletedArticles,

batchRecoverArticles,

} from '@/api/article'

import { getAllType } from '@/api/type'

import { parseTime } from '@/utils/index'

import SearchArticle from '@/components/SearchArticle'

export default {

components: { SearchArticle },

data() {

return {

loading: false,

typeList: [],

tableData: [],

pn: 1,

limit: 10,

total: 0,

multipleSelection: [],

condition: '',

}

},

created() {

this.get\_article\_list(1)

this.get\_type\_list()

},

filters: {

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

},

},

methods: {

// 页面大小变化

handleSizeChange(val) {

this.limit = val

this.get\_article\_list(1)

},

// 跳转

handleCurrentChange(val) {

this.get\_article\_list(val)

},

// 彻底删除文章

deleteArticle(row) {

var msg =

'将要彻底删除文章 <' + row.articleTitle + '> ！不可恢复！是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteDeletedArticles(row.articleId).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

})

})

},

// 批量恢复文章

recoverArticle(row) {

var msg = '将要恢复文章 <' + row.articleTitle + '> 是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'success',

})

.then(() => {

batchRecoverArticles(row.articleId).then((response) => {

if (response.extend.suc === true) {

this.$message({

type: 'success',

message: '恢复成功！',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('恢复失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消',

})

})

},

// 获取文章列表

get\_article\_list(pn) {

this.loading = true

getArticleByUserDeleted(pn, this.limit, this.condition).then(

(response) => {

var pageInfo = response.extend.pageInfo

this.tableData = pageInfo.records

this.total = pageInfo.total

}

)

this.loading = false

},

// 获取类型列表

get\_type\_list() {

getAllType().then((response) => {

this.typeList = response.extend.pageInfo.records

})

},

// 多选

handleSelectionChange(val) {

this.multipleSelection = val

},

// 批量删除

batchDelete() {

var ids = ''

if (this.multipleSelection.length === 0) {

this.$message.error('请先选择')

return

}

// 字符串拼接

this.multipleSelection.forEach((item) => {

ids += item.articleId + '&'

})

this.$confirm('确定删除选中文章？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteDeletedArticles(ids).then((response) => {

if (response.extend.suc) {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

duration: 1000,

})

})

},

// 批量恢复

batchRecover() {

var ids = ''

if (this.multipleSelection.length === 0) {

this.$message.error('请先选择需要恢复的文章')

return

}

// 字符串拼接

this.multipleSelection.forEach((item) => {

ids += item.articleId + '&'

})

this.$confirm('确定恢复选中文章？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'success',

})

.then(() => {

batchRecoverArticles(ids).then((response) => {

if (response.extend.suc === true) {

this.$message({

type: 'success',

message: '恢复成功!',

})

this.get\_article\_list(this.pn)

} else {

this.$message.error('恢复失败，请稍后再试!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消恢复',

duration: 1000,

})

})

},

// 提交搜索

submit() {

this.condition = this.$refs.searchArticle.generateQueryString()

this.get\_article\_list(1)

},

// 文章列表重置

resetForm() {

this.condition = ''

this.get\_article\_list(1)

},

},

}

</script>

<style scoped>

.pagiation {

margin-top: 30px;

margin-left: 30px;

}

.btn-group {

margin: 20px;

}

.main {

margin: 20px;

}

.search-bar {

margin-bottom: 20px;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\comment\all-comment.vue

<template>

<div class="main">

<!-- 搜索框 -->

<el-col :span="12">

<el-collapse accordion class="search-bar">

<el-collapse-item>

<template slot="title">

<i class="header-icon el-icon-search" style="font-size:20px;margin-right: 5px;"></i>筛选评论

</template>

<el-form ref="form" label-width="80px">

<el-form-item label="所在文章">

<el-input v-model="query.article\_title"></el-input>

</el-form-item>

<el-form-item label="作者">

<el-input v-model="query.comment\_author\_name"></el-input>

</el-form-item>

<el-form-item label="发表时间">

<el-col :span="11">

<el-date-picker type="date" placeholder="开始日期" v-model="query.start" style="width: 100%;"></el-date-picker>

</el-col>

<el-col :span="11">

<el-date-picker type="date" placeholder="结束日期" v-model="query.end" style="width: 100%;"></el-date-picker>

</el-col>

</el-form-item>

<el-form-item label="赞同数量">

<el-radio-group v-model="query.rank\_like">

<el-radio label="无"></el-radio>

<el-radio label="升序"></el-radio>

<el-radio label="降序"></el-radio>

</el-radio-group>

</el-form-item>

<el-form-item>

<el-button type="primary" @click="submitSearch">搜索</el-button>

<el-button @click="resetForm">重置</el-button>

</el-form-item>

</el-form>

</el-collapse-item>

</el-collapse>

</el-col>

<el-table

:data="tableData"

border

style="width: 100%"

v-loading="loading"

@selection-change="handleSelectionChange">

<el-table-column

type="selection"

width="55">

</el-table-column>

<el-table-column

prop="articleTitle"

label="所在文章"

width="280">

</el-table-column>

<el-table-column

prop="commentAuthorName"

label="作者"

width="120">

</el-table-column>

<el-table-column

label="内容"

width="400">

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.commentContent | commentFormat }}</span>

</template>

</el-table-column>

<el-table-column

label="发表时间"

sortable

width="250">

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.commentDate | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="commentLikeCount"

label="赞同"

sortable

width="100">

</el-table-column>

<el-table-column

label="操作"

width="120">

<template slot-scope="scope">

<el-button @click="viewDetail(scope.row)" type="text" size="small">查看</el-button>

<el-button @click="deleteComment(scope.row)" type="text" size="small">删除</el-button>

</template>

</el-table-column>

</el-table>

<div class="btn-group">

<el-button type="danger" @click="batchDelete">删除选中</el-button>

</div>

<div class="pagiation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="pn"

:page-sizes="[10, 20, 30, 40]"

:page-size="limit"

layout="total, sizes, prev, pager, next, jumper"

:total="total">

</el-pagination>

</div>

<el-dialog title="评论详情" :visible.sync="commentDetailVisable" class="comment-detail">

<p> <span>所在文章</span> {{ commentDetail.articleTitle }}</p>

<p> <span>作者</span> {{ commentDetail.commentAuthorName }}</p>

<p> <span>内容</span> {{ commentDetail.commentContent }}</p>

<p> <span>发表日期</span> {{ commentDetail.commentDate | dateFormat }}</p>

<p> <span>赞同数</span> {{ commentDetail.commentLikeCount }}</p>

</el-dialog>

</div>

</template>

<script>

import { getAll, batchDeleteComments } from '@/api/comment'

import { parseTime } from '@/utils/index'

export default {

data() {

return {

loading: false,

tableData: [],

types: [],

pn: 1,

limit: 10,

total: 0,

multipleSelection: [],

condition: "",

query: {

comment\_author\_name: "",

article\_title: "",

start: "",

end: "",

rank\_like: "",

},

commentDetailVisable: false,

commentDetail: {}

}

},

created(){

this.get\_comment\_list(1)

},

filters: {

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

},

commentFormat(content){

if(content.length < 20){

return content

} else {

return content.substr(0, 20) + '... ...'

}

}

},

methods: {

viewDetail(row){

this.commentDetail = row

this.commentDetailVisable = !this.commentDetailVisable

},

handleSizeChange(val) {

this.limit = val

this.get\_comment\_list(1)

},

handleCurrentChange(val) {

this.get\_comment\_list(val)

},

deleteComment(row) {

var msg = '将要删除该评论 , 是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

batchDeleteComments(row.commentId).then(response => {

if(response.extend.suc === 'success'){

this.$message({

type: 'success',

message: '删除成功!'

});

this.get\_comment\_list(this.pn)

} else {

this.$message.error('删除失败!');

}

})

}).catch(() => {

this.$message({

type: 'info',

message: '已取消删除'

});

});

},

get\_comment\_list(pn){

this.loading = true

getAll(pn, this.limit, this.condition).then(response => {

var pageInfo = response.extend.pageInfo

this.tableData = pageInfo.records

this.total = pageInfo.total

})

this.loading = false

},

handleSelectionChange(val){

this.multipleSelection = val

},

batchDelete(){

var ids = ''

if(this.multipleSelection.length === 0){

this.$message.error('请先选择');

return

}

// 字符串拼接

this.multipleSelection.forEach(item => {

ids += item.commentId + '&'

});

this.$confirm('确定删除选中评论？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

batchDeleteComments(ids).then(response => {

if(response.extend.suc === 'success'){

this.$message({

type: 'success',

message: '删除成功!'

});

this.get\_comment\_list(this.pn)

} else {

this.$message.error('删除失败!');

}

})

}).catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

duration: 1000

});

});

},

submitSearch(){

this.generateQueryString();

this.get\_comment\_list(1);

},

resetForm(){

this.query.comment\_author\_name = ""

this.query.article\_title = ""

this.query.start = ""

this.query.end =""

this.query.rank\_like = ""

this.condition = ""

this.get\_comment\_list(1)

},

generateQueryString(){

let condition = ""

let query = this.query

if(query.comment\_author\_name !== ""){

condition += ("comment\_author\_name=" + query.comment\_author\_name

+ "--")

}

if(query.article\_title !== ""){

condition += ("article\_title=" + query.article\_title + "--")

}

if(query.start !== ""){

condition += ("start=" + parseTime(query.start) + "--")

}

if(query.end !== ""){

condition += ("end=" + parseTime(query.end) + "--")

}

if(query.rank\_like == '升序'){

condition += ("rank\_like=1--")

} else if (query.rank\_like == '降序'){

condition += ("rank\_like=0--")

}

condition = condition.lastIndexOf("--") === condition.length - 2 ?

condition.substr(0, condition.length - 2) :

condition

this.condition = condition

}

}

}

</script>

<style lang="scss" scoped>

.pagiation{

margin-top: 30px;

margin-left: 30px;

}

.btn-group{

margin: 20px;

}

.main{

margin: 20px;

}

.search-bar{

margin-bottom: 20px;

}

.comment-detail{

p{

font-size: 17px;

line-height: 30px;

}

span{

font-weight: 700;

}

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\comment\comment-posted.vue

<template>

<div class="main">

<!-- 搜索框 -->

<el-col :span="12">

<el-collapse accordion class="search-bar">

<el-collapse-item>

<template slot="title">

<i class="header-icon el-icon-search" style="font-size:20px;margin-right: 5px;"></i>筛选评论

</template>

<el-form ref="form" label-width="80px">

<el-form-item label="所在文章">

<el-input v-model="query.article\_title"></el-input>

</el-form-item>

<el-form-item label="发表时间">

<el-col :span="11">

<el-date-picker type="date" placeholder="开始日期" v-model="query.start" style="width: 100%;"></el-date-picker>

</el-col>

<el-col :span="11">

<el-date-picker type="date" placeholder="结束日期" v-model="query.end" style="width: 100%;"></el-date-picker>

</el-col>

</el-form-item>

<el-form-item label="赞同数量">

<el-radio-group v-model="query.rank\_like">

<el-radio label="无"></el-radio>

<el-radio label="升序"></el-radio>

<el-radio label="降序"></el-radio>

</el-radio-group>

</el-form-item>

<el-form-item>

<el-button type="primary" @click="submitSearch">搜索</el-button>

<el-button @click="resetForm">重置</el-button>

</el-form-item>

</el-form>

</el-collapse-item>

</el-collapse>

</el-col>

<el-table

:data="tableData"

border

style="width: 100%"

v-loading="loading"

@selection-change="handleSelectionChange">

<el-table-column

type="selection"

width="55">

</el-table-column>

<el-table-column

prop="articleTitle"

label="所在文章"

width="280">

</el-table-column>

<el-table-column

label="内容"

width="400">

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.commentContent | commentFormat }}</span>

</template>

</el-table-column>

<el-table-column

label="发表时间"

sortable

width="250">

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.commentDate | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="commentLikeCount"

label="赞同"

sortable

width="100">

</el-table-column>

<el-table-column

label="操作"

width="120">

<template slot-scope="scope">

<el-button @click="viewDetail(scope.row)" type="text" size="small">查看</el-button>

<el-button @click="deleteComment(scope.row)" type="text" size="small">删除</el-button>

</template>

</el-table-column>

</el-table>

<div class="btn-group">

<el-button type="danger" @click="batchDelete">删除选中</el-button>

</div>

<div class="pagiation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="pn"

:page-sizes="[10, 20, 30, 40]"

:page-size="limit"

layout="total, sizes, prev, pager, next, jumper"

:total="total">

</el-pagination>

</div>

<el-dialog title="评论详情" :visible.sync="commentDetailVisable" class="comment-detail">

<p> <span>所在文章</span> {{ commentDetail.articleTitle }}</p>

<p> <span>作者</span> {{ commentDetail.commentAuthorName }}</p>

<p> <span>内容</span> {{ commentDetail.commentContent }}</p>

<p> <span>发表日期</span> {{ commentDetail.commentDate | dateFormat }}</p>

<p> <span>赞同数</span> {{ commentDetail.commentLikeCount }}</p>

</el-dialog>

</div>

</template>

<script>

import { getCommentByUserPosted, batchDeleteComments } from '@/api/comment'

import { parseTime } from '@/utils/index'

export default {

data() {

return {

loading: false,

tableData: [],

types: [],

pn: 1,

limit: 10,

total: 0,

multipleSelection: [],

condition: "",

query: {

article\_title: "",

start: "",

end: "",

rank\_like: "",

},

commentDetailVisable: false,

commentDetail: {}

}

},

created(){

this.get\_comment\_list(1)

},

filters: {

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

},

commentFormat(content){

if(content.length < 20){

return content

} else {

return content.substr(0, 20) + '... ...'

}

}

},

methods: {

viewDetail(row){

this.commentDetail = row

this.commentDetailVisable = !this.commentDetailVisable

},

handleSizeChange(val) {

this.limit = val

this.get\_comment\_list(1)

},

handleCurrentChange(val) {

this.get\_comment\_list(val)

},

deleteComment(row) {

var msg = '将要删除该评论 , 是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

batchDeleteComments(row.commentId).then(response => {

if(response.extend.suc === 'success'){

this.$message({

type: 'success',

message: '删除成功!'

});

this.get\_comment\_list(this.pn)

} else {

this.$message.error('删除失败!');

}

})

}).catch(() => {

this.$message({

type: 'info',

message: '已取消删除'

});

});

},

get\_comment\_list(pn){

this.loading = true

getCommentByUserPosted(pn, this.limit, this.condition).then(response => {

var pageInfo = response.extend.pageInfo

this.tableData = pageInfo.records

this.total = pageInfo.total

})

this.loading = false

},

handleSelectionChange(val){

this.multipleSelection = val

},

batchDelete(){

var ids = ''

if(this.multipleSelection.length === 0){

this.$message.error('请先选择');

return

}

// 字符串拼接

this.multipleSelection.forEach(item => {

ids += item.commentId + '&'

});

this.$confirm('确定删除选中评论？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

batchDeleteComments(ids).then(response => {

if(response.extend.suc === 'success'){

this.$message({

type: 'success',

message: '删除成功!'

});

this.get\_comment\_list(this.pn)

} else {

this.$message.error('删除失败!');

}

})

}).catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

duration: 1000

});

});

},

submitSearch(){

this.generateQueryString();

this.get\_comment\_list(1);

},

resetForm(){

this.query.article\_title = ""

this.query.start = ""

this.query.end =""

this.query.rank\_like = ""

this.condition = ""

this.get\_comment\_list(1)

},

generateQueryString(){

let condition = ""

let query = this.query

if(query.article\_title !== ""){

condition += ("article\_title=" + query.article\_title + "--")

}

if(query.start !== ""){

condition += ("start=" + parseTime(query.start) + "--")

}

if(query.end !== ""){

condition += ("end=" + parseTime(query.end) + "--")

}

if(query.rank\_like == '升序'){

condition += ("rank\_like=1--")

} else if (query.rank\_like == '降序'){

condition += ("rank\_like=0--")

}

condition = condition.lastIndexOf("--") === condition.length - 2 ?

condition.substr(0, condition.length - 2) :

condition

this.condition = condition

}

}

}

</script>

<style lang="scss" scoped>

.pagiation{

margin-top: 30px;

margin-left: 30px;

}

.btn-group{

margin: 20px;

}

.main{

margin: 20px;

}

.search-bar{

margin-bottom: 20px;

}

.comment-detail{

p{

font-size: 17px;

line-height: 30px;

}

span{

font-weight: 700;

}

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\comment\comment-received.vue

<template>

<div class="main">

<!-- 搜索框 -->

<el-col :span="12">

<el-collapse accordion class="search-bar">

<el-collapse-item>

<template slot="title">

<i class="header-icon el-icon-search" style="font-size:20px;margin-right: 5px;"></i>筛选评论

</template>

<el-form ref="form" label-width="80px">

<el-form-item label="所在文章">

<el-input v-model="query.article\_title"></el-input>

</el-form-item>

<el-form-item label="作者">

<el-input v-model="query.comment\_author\_name"></el-input>

</el-form-item>

<el-form-item label="发表时间">

<el-col :span="11">

<el-date-picker type="date" placeholder="开始日期" v-model="query.start" style="width: 100%;"></el-date-picker>

</el-col>

<el-col :span="11">

<el-date-picker type="date" placeholder="结束日期" v-model="query.end" style="width: 100%;"></el-date-picker>

</el-col>

</el-form-item>

<el-form-item label="赞同数量">

<el-radio-group v-model="query.rank\_like">

<el-radio label="无"></el-radio>

<el-radio label="升序"></el-radio>

<el-radio label="降序"></el-radio>

</el-radio-group>

</el-form-item>

<el-form-item>

<el-button type="primary" @click="submitSearch">搜索</el-button>

<el-button @click="resetForm">重置</el-button>

</el-form-item>

</el-form>

</el-collapse-item>

</el-collapse>

</el-col>

<el-table

:data="tableData"

border

style="width: 100%"

v-loading="loading"

@selection-change="handleSelectionChange">

<el-table-column

type="selection"

width="55">

</el-table-column>

<el-table-column

prop="articleTitle"

label="所在文章"

width="280">

</el-table-column>

<el-table-column

prop="commentAuthorName"

label="作者"

width="120">

</el-table-column>

<el-table-column

label="内容"

width="400">

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.commentContent | commentFormat }}</span>

</template>

</el-table-column>

<el-table-column

label="发表时间"

sortable

width="250">

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.commentDate | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="commentLikeCount"

label="赞同"

sortable

width="100">

</el-table-column>

<el-table-column

label="操作"

width="120">

<template slot-scope="scope">

<el-button @click="viewDetail(scope.row)" type="text" size="small">查看</el-button>

<el-button @click="deleteComment(scope.row)" type="text" size="small">删除</el-button>

</template>

</el-table-column>

</el-table>

<div class="btn-group">

<el-button type="danger" @click="batchDelete">删除选中</el-button>

</div>

<div class="pagiation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="pn"

:page-sizes="[10, 20, 30, 40]"

:page-size="limit"

layout="total, sizes, prev, pager, next, jumper"

:total="total">

</el-pagination>

</div>

<el-dialog title="评论详情" :visible.sync="commentDetailVisable" class="comment-detail">

<p> <span>所在文章</span> {{ commentDetail.articleTitle }}</p>

<p> <span>作者</span> {{ commentDetail.commentAuthorName }}</p>

<p> <span>内容</span> {{ commentDetail.commentContent }}</p>

<p> <span>发表日期</span> {{ commentDetail.commentDate | dateFormat }}</p>

<p> <span>赞同数</span> {{ commentDetail.commentLikeCount }}</p>

</el-dialog>

</div>

</template>

<script>

import { getCommentByUserReceive, batchDeleteComments } from '@/api/comment'

import { parseTime } from '@/utils/index'

export default {

data() {

return {

loading: false,

tableData: [],

types: [],

pn: 1,

limit: 10,

total: 0,

multipleSelection: [],

condition: "",

query: {

comment\_author\_name: "",

article\_title: "",

start: "",

end: "",

rank\_like: "",

},

commentDetailVisable: false,

commentDetail: {}

}

},

created(){

this.get\_comment\_list(1)

},

filters: {

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

},

commentFormat(content){

if(content.length < 20){

return content

} else {

return content.substr(0, 20) + '... ...'

}

}

},

methods: {

viewDetail(row){

this.commentDetail = row

this.commentDetailVisable = !this.commentDetailVisable

},

handleSizeChange(val) {

this.limit = val

this.get\_comment\_list(1)

},

handleCurrentChange(val) {

this.get\_comment\_list(val)

},

deleteComment(row) {

var msg = '将要删除该评论 , 是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

batchDeleteComments(row.commentId).then(response => {

if(response.extend.suc === 'success'){

this.$message({

type: 'success',

message: '删除成功!'

});

this.get\_comment\_list(this.pn)

} else {

this.$message.error('删除失败!');

}

})

}).catch(() => {

this.$message({

type: 'info',

message: '已取消删除'

});

});

},

get\_comment\_list(pn){

this.loading = true

getCommentByUserReceive(pn, this.limit, this.condition).then(response => {

var pageInfo = response.extend.pageInfo

this.tableData = pageInfo.records

this.total = pageInfo.total

})

this.loading = false

},

handleSelectionChange(val){

this.multipleSelection = val

},

batchDelete(){

var ids = ''

if(this.multipleSelection.length === 0){

this.$message.error('请先选择');

return

}

// 字符串拼接

this.multipleSelection.forEach(item => {

ids += item.commentId + '&'

});

this.$confirm('确定删除选中评论？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

batchDeleteComments(ids).then(response => {

if(response.extend.suc === 'success'){

this.$message({

type: 'success',

message: '删除成功!'

});

this.get\_comment\_list(this.pn)

} else {

this.$message.error('删除失败!');

}

})

}).catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

duration: 1000

});

});

},

submitSearch(){

this.generateQueryString();

this.get\_comment\_list(1);

},

resetForm(){

this.query.comment\_author\_name = ""

this.query.article\_title = ""

this.query.start = ""

this.query.end =""

this.query.rank\_like = ""

this.condition = ""

this.get\_comment\_list(1)

},

generateQueryString(){

let condition = ""

let query = this.query

if(query.comment\_author\_name !== ""){

condition += ("comment\_author\_name=" + query.comment\_author\_name

+ "--")

}

if(query.article\_title !== ""){

condition += ("article\_title=" + query.article\_title + "--")

}

if(query.start !== ""){

condition += ("start=" + parseTime(query.start) + "--")

}

if(query.end !== ""){

condition += ("end=" + parseTime(query.end) + "--")

}

if(query.rank\_like == '升序'){

condition += ("rank\_like=1--")

} else if (query.rank\_like == '降序'){

condition += ("rank\_like=0--")

}

condition = condition.lastIndexOf("--") === condition.length - 2 ?

condition.substr(0, condition.length - 2) :

condition

this.condition = condition

}

}

}

</script>

<style lang="scss" scoped>

.pagiation{

margin-top: 30px;

margin-left: 30px;

}

.btn-group{

margin: 20px;

}

.main{

margin: 20px;

}

.search-bar{

margin-bottom: 20px;

}

.comment-detail{

p{

font-size: 17px;

line-height: 30px;

}

span{

font-weight: 700;

}

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\dashboard\index.vue

<template>

<div class="dashboard-container">

<div class="personal-info">

<!-- 个人界面展示 -->

<el-card class="personal-info-sign">

<!-- 头像 -->

<el-col>

<el-avatar

:size="150"

:src="avatarUrl"

></el-avatar>

</el-col>

<el-col class="sign">

<!-- 用户名 -->

<div class="user-name">{{ name }}</div>

<!-- 签名 -->

<div class="user-signature">{{ user.data.userSignature }}</div>

</el-col>

</el-card>

<!-- 展示浏览、评论、点赞、收藏等数据 -->

<el-row class="statastic">

<el-col :span="4">

<!-- 总浏览量 -->

<el-card>

<svg

t="1633262231414"

class="icon"

viewBox="0 0 1024 1024"

version="1.1"

xmlns="http://www.w3.org/2000/svg"

p-id="3504"

width="80"

height="80"

>

<path

d="M734.634667 72.448C717.824 55.978667 685.226667 42.666667 661.674667 42.666667H204.8C162.389333 42.666667 128 79.189333 128 124.330667v775.338666C128 944.768 162.474667 981.333333 204.8 981.333333h614.4c42.410667 0 76.8-36.693333 76.8-81.578666V273.493333c0-23.68-13.781333-56.405333-30.378667-72.661333l-130.986666-128.426667zM341.333333 298.666667h341.333334a42.666667 42.666667 0 0 1 0 85.333333H341.333333a42.666667 42.666667 0 1 1 0-85.333333z m0 170.666666h170.666667a42.666667 42.666667 0 0 1 0 85.333334H341.333333a42.666667 42.666667 0 0 1 0-85.333334z"

p-id="3505"

fill="#13227a"

></path>

</svg>

<div class="count">

<span class="count-char">总文章</span>

<span class="count-number">{{ user.data.totalArticle | dataFormat}}</span>

</div>

</el-card>

</el-col>

<el-col :span="4">

<!-- 总浏览量 -->

<el-card>

<svg

t="1633243178182"

class="icon"

viewBox="0 0 1024 1024"

version="1.1"

xmlns="http://www.w3.org/2000/svg"

p-id="5920"

width="80"

height="80"

>

<path

d="M510.557138 204.354248c-302.839283 0-458.323497 306.150702-458.323497 306.150702s117.330242 306.189587 458.323497 306.189587c308.802088 0 458.300984-304.995389 458.300984-304.995389S818.167075 204.354248 510.557138 204.354248L510.557138 204.354248 510.557138 204.354248zM511.245823 701.866279c-110.729917 0-190.772928-83.72589-190.772928-191.364399 0-107.647719 80.049151-191.352119 190.772928-191.352119 110.723777 0 190.763718 83.697237 190.763718 191.352119C702.010565 618.140389 621.970624 701.866279 511.245823 701.866279L511.245823 701.866279 511.245823 701.866279zM511.245823 395.675668c-63.286372 0.145309-114.460892 53.321416-114.460892 114.827235 0 61.473073 51.175543 114.821095 114.460892 114.821095 63.282279 0 114.453728-53.352115 114.453728-114.821095C625.703645 448.975595 574.529125 395.556964 511.245823 395.675668L511.245823 395.675668 511.245823 395.675668z"

p-id="5921"

fill="#1296db"

></path>

</svg>

<div class="count">

<span class="count-char">总浏览</span>

<span class="count-number">{{ user.data.totalViewCount | dataFormat }}</span>

</div>

</el-card>

</el-col>

<el-col :span="4">

<!-- 评论量 -->

<el-card>

<svg

t="1633244239936"

class="icon"

viewBox="0 0 1024 1024"

version="1.1"

xmlns="http://www.w3.org/2000/svg"

p-id="7377"

width="80"

height="80"

>

<path

d="M832 128 192 128C139.072 128 96 171.072 96 224l0 512c0 52.928 43.072 96 96 96l210.752 0 86.624 86.624C495.616 924.864 503.808 928 512 928s16.384-3.136 22.624-9.376L621.248 832 832 832c52.928 0 96-43.072 96-96L928 224C928 171.072 884.928 128 832 128zM336 512C309.536 512 288 490.464 288 464S309.536 416 336 416s48 21.536 48 48S362.464 512 336 512zM528 512c-26.464 0-48-21.536-48-48s21.536-48 48-48 48 21.536 48 48S554.464 512 528 512zM720 512c-26.464 0-48-21.536-48-48s21.536-48 48-48 48 21.536 48 48S746.464 512 720 512z"

p-id="7378"

fill="#1dd76f"

></path>

</svg>

<div class="count">

<span class="count-char">总获评</span>

<span class="count-number">{{ user.data.totalCommentCount | dataFormat }}</span>

</div>

</el-card>

</el-col>

<el-col :span="4">

<!-- 点赞量 -->

<el-card>

<svg

t="1633244324852"

class="icon"

viewBox="0 0 1024 1024"

version="1.1"

xmlns="http://www.w3.org/2000/svg"

p-id="8333"

width="80"

height="80"

>

<path

d="M581.674667 170.666667c6.72 0 18.538667 1.429333 27.584 4.352 44.544 14.442667 70.186667 61.76 59.797333 109.12l-1.066667 4.437333-39.701333 148.906667h157.269333c19.093333 0 37.973333 7.082667 50.773334 21.248 14.293333 15.829333 19.861333 36.778667 15.616 57.109333l-1.066667 4.352-82.922667 295.253333a51.882667 51.882667 0 0 1-46.08 37.76l-3.84 0.128H298.666667V452.501333c84.949333-18.389333 209.194667-244.373333 209.194666-244.373333C525.525333 184.170667 546.944 170.666667 581.674667 170.666667zM234.666667 448v405.333333H170.666667V448h64z"

p-id="8334"

fill="#d81e06"

></path>

</svg>

<div class="count">

<span class="count-char">总获赞</span>

<span class="count-number">{{ user.data.totalLikeCount | dataFormat }}</span>

</div>

</el-card>

</el-col>

</el-row>

</div>

<!-- 最近发表的文章列表和收到的消息等 -->

<div class="message">

<el-row :gutter="12">

<!-- 最近发表的文章列表 -->

<el-col :span="8">

<el-card

class="not-exist"

v-if="user.article.length <= 0"

>

最近未发表文章

</el-card>

<el-card v-if="user.article.length > 0">

<h3>最近文章</h3>

<div

class="message-item"

v-for="(item, index) in user.article"

:key="index"

>

<span class="article">{{ item.articleTitle }}</span>

<span class="time">{{ item.articlePostTime | changeTime }}</span>

</div>

</el-card>

</el-col>

<!-- 收到的点赞 -->

<el-col :span="8">

<el-card

class="not-exist"

v-if="user.article.length <= 0"

>

最近未收到点赞

</el-card>

<el-card v-if="user.like.length > 0">

<h3>最近收到的赞</h3>

<div

class="message-item"

v-for="(item, index) in user.like"

:key="index"

>

<span class="name">{{item.userName}}</span> 点赞了

<span class="article">{{item.articleTitle}}</span>

<span class="time">{{ item.likeDate | changeTime }}</span>

</div>

</el-card>

</el-col>

<!-- 收到的评论 -->

<el-col :span="8">

<el-card

class="not-exist"

v-if="user.article.length <= 0"

>

最近未收到评论

</el-card>

<el-card v-if="user.comment.length > 0">

<h3>最近收到的评论</h3>

<div

class="message-item"

v-for="(item, index) in user.comment"

:key="index"

>

<span class="name">{{item.commentAuthorName}}</span> 评论了

<span class="article">{{item.articleTitle}}</span>

<span class="time">{{ item.commentDate | changeTime }}</span>

</div>

</el-card>

</el-col>

</el-row>

</div>

</div>

</template>

<script>

import { mapGetters } from 'vuex'

import { getUserTotalData } from '@/api/user'

import { getUserCommentRecently } from '@/api/comment'

import { getUserLikeRecently } from '@/api/like'

import { getUserArticleRecently } from '@/api/article'

import timeago from '@/assets/js/timeago.js'

export default {

name: 'Dashboard',

data() {

return {

avatarUrl: '',

user: {

data: {

articleAuthor: '',

userName: '',

userSignature: '代码改变世界',

totalArticle: 0,

totalViewCount: 0,

totalLikeCount: 0,

totalCommentCount: 0,

},

article: [],

like: [],

comment: [],

},

}

},

computed: {

...mapGetters(['name']),

},

filters:{

dataFormat(data){

if(null === data) return 0;

if(data > 1000){

return Number(data/1000) + "k"

}

return data

},

changeTime(val){

let time = new Date(val); //先将接收到的json格式的日期数据转换成可用的js对象日期

return new timeago().format(time, 'zh\_CN'); //转换成类似于几天前的格式

}

},

created() {

this.get\_user\_total\_data()

this.avatarUrl = this.$store.getters.avatar

},

methods: {

async get\_user\_total\_data() {

var \_this = this

await getUserTotalData(-1).then((res) => {

\_this.user.data = res.extend.data

})

getUserLikeRecently(\_this.user.data.articleAuthor, 1, 10).then(

(res) => {

\_this.user.like = res.extend.pageInfo.records

})

getUserCommentRecently(\_this.user.data.articleAuthor, 1, 10).then(

(res) => {

\_this.user.comment = res.extend.pageInfo.records

}

)

getUserArticleRecently(\_this.user.data.articleAuthor, 1, 10).then(

(res) => {

\_this.user.article = res.extend.pageInfo.records

}

)

},

},

}

</script>

<style lang="scss" scoped>

.dashboard {

background-color: rgb(240, 242, 245);

&-container {

padding: 20px;

.personal-info {

width: 100%;

text-align: center;

.personal-info-sign {

margin: 10px auto;

background-image: url('../../assets/imgs/personal-card.png');

.sign {

text-align: center;

color: white;

.user-name {

margin: 10px auto;

font-size: 40px;

}

.user-signature {

margin: 10px auto;

font-size: 18px;

}

}

}

.statastic {

display: flex;

justify-content: space-around;

}

.count {

.count-char {

font-size: 15px;

margin: 0 5px;

}

.count-number {

font-size: 25px;

font-weight: 550;

}

}

}

.message {

margin: 10px auto;

.message-item {

padding-top: 6px;

padding-bottom: 6px;

cursor: pointer;

.time {

float: right;

font-size: 13px;

color: #afa8a8;

}

.name {

font-weight: 600;

}

.article {

font-weight: 750;

}

&:hover {

background-color: rgb(236, 236, 236);

}

}

.not-exist{

padding: 20px;

text-align: center;

color: gray;

}

}

}

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\index\article.vue

<template>

<div id="app">

<!-- 导航栏-->

<el-row style="position: fixed; width:100%; z-index: 1">

<el-col

:span="14"

:offset="5"

>

<el-menu

default-active="0"

class="el-menu-demo"

mode="horizontal"

>

<el-menu-item>

<router-link to="/dashboard">

<span class="logo">Jancoyan</span>

</router-link>

</el-menu-item>

<el-menu-item

style="float: right"

@click="linkToDashBoard"

>

<el-avatar

size="large"

:src="avatarUrl"

>

登录

</el-avatar>

</el-menu-item>

</el-menu>

</el-col>

</el-row>

<!-- 页面主体-->

<el-row

:gutter="24"

style="padding-top: 60px"

>

<!-- 作者信息栏 和 文章目录导航-->

<el-col :span="5">

<!-- 文章信息卡片-->

<el-card

shadow="hover"

style="text-align: center;"

>

<div class="article-info">

<!-- 标题-->

<h2>{{ article.articleTitle }}</h2>

<div>{{ article.articlePostTime | dateFormat }}</div>

<el-divider></el-divider>

<!-- 文章信息栏-->

<div>

<div class="article-count">

<span class="count-number">{{ article.articleViewCount }}</span>

<span class="count-char">浏览</span>

</div>

<el-divider direction="vertical"></el-divider>

<div class="article-count">

<span class="count-number">{{ article.articleLikeCount }}</span>

<span class="count-char">获赞</span>

</div>

<el-divider direction="vertical"></el-divider>

<div class="article-count">

<span class="count-number">{{ article.articleCommentCount }}</span>

<span class="count-char">评论</span>

</div>

</div>

<el-divider></el-divider>

<!-- 文章操作栏-->

<div class="article-action">

<!-- 点赞-->

<div class="article-count">

<span

class="count-number"

@click="likeArticle"

>

<svg

t="1633675948290"

viewBox="0 0 1024 1024"

version="1.1"

xmlns="http://www.w3.org/2000/svg"

p-id="2028"

width="40"

height="40"

>

<path

d="M938.666667 362.666667A234.666667 234.666667 0 0 0 704 128 271.36 271.36 0 0 0 512 216.32 271.36 271.36 0 0 0 320 128 234.666667 234.666667 0 0 0 85.333333 362.666667c0 167.253333 202.666667 352 298.666667 448l97.28 97.28a32 32 0 0 0 22.613333 9.386666h16.213334a32 32 0 0 0 22.613333-9.386666L640 810.666667c96-96 298.666667-280.746667 298.666667-448z"

p-id="2029"

:fill="like.liked ? '#f00' : '#dbdbdb'"

></path>

</svg>

</span>

</div>

</div>

</div>

</el-card>

<!-- 文章目录导航-->

</el-col>

<el-col

:span="14"

style="padding: 0 60px"

>

<!-- 内容-->

<div

class="md-content"

v-html='article.articleHtml'

style="word-break:break-all;"

>

</div>

<el-divider>The End</el-divider>

<!-- 文章评论和发表评论-->

<!-- 作者关闭了评论-->

<el-card

shadow="never"

style="text-align: center;margin-top: 20px; padding: 20px; color: gray; border: none"

v-if="!article.articleIsComment"

>

作者关闭了评论

</el-card>

<!-- 评论区 -->

<el-col v-if="article.articleIsComment">

<!-- 评论的标题--评论 comment -->

<div class="comment-area-title">

<span class="comment-title-chinese">评论区</span>

<span class="comment-title-english">Comments</span>

</div>

<!-- 分页评论-->

<div class="comment-list">

<div

class="comment-item"

v-for="item in commentList"

:key="item.commentId"

>

<el-card shadow="never">

<!-- 评论内容-->

<div class="comment-content">

{{ item.commentContent }}

</div>

<!-- 评论信息-->

<div class="comment-info">

<!-- 作者-->

<span class="comment-info-item"><i class="el-icon-user"></i>

{{ item.commentAuthorName }}</span>

<!-- 评论时间-->

<span class="comment-info-item"><i class="el-icon-date"></i>

{{ item.commentDate | dateFormat }}</span>

<!-- 赞同的数量-->

<!-- <span class="comment-info-item"><i class="el-icon-caret-top"></i>

{{ item.commentLikeCount }}</span> -->

<!-- 回复和赞同-->

<!-- <el-row style="float: right">

<el-button

type="primary"

size="mini"

plain

@click="likeComment(item.commentId)"

>赞同</el-button>

<el-button size="mini">回复</el-button>

</el-row> -->

</div>

</el-card>

</div>

<div class="pagination">

<el-pagination

background

@current-change="commentPageChange"

layout="prev, pager, next"

:total="pagination.total"

:page-size="pagination.size"

v-if="commentList.length > 0"

>

</el-pagination>

</div>

</div>

<!-- 发表评论的表单-->

<div class="post-comment">

<el-form

:ref="form"

:model="form"

label-width="80px"

:rules="rules"

>

<el-form-item

label="昵称"

v-if="userlogin"

prop="name"

>

<el-input

v-model="form.name"

maxlength="20"

></el-input>

</el-form-item>

<el-form-item

label="邮箱"

v-if="userlogin"

prop="email"

>

<el-input

v-model="form.email"

maxlength="50"

></el-input>

</el-form-item>

<el-form-item

label="评论"

prop="content"

>

<el-input

type="textarea"

v-model="form.content"

autosize

></el-input>

</el-form-item>

<el-form-item>

<el-button

type="primary"

@click="onSubmit"

>评论</el-button>

</el-form-item>

</el-form>

</div>

</el-col>

</el-col>

<el-col :span="5">

<!-- 作者信息卡片-->

<el-card

shadow="hover"

style="text-align: center;"

>

<!-- 头像-->

<el-avatar

:size="80"

:src="authorAvatar"

></el-avatar>

<div class="user-info">

<!-- 用户名-->

<h3>{{ author.userName }}</h3>

<!-- 签名-->

<blockquote> {{ author.userSignature }} </blockquote>

<el-divider></el-divider>

<!-- 文章数量、获赞、收藏-->

<div>

<div class="user-count">

<span class="count-number">{{ author.totalArticle }}</span>

<span class="count-char">文章</span>

</div>

<el-divider direction="vertical"></el-divider>

<div class="user-count">

<span class="count-number">{{ author.totalLikeCount }}</span>

<span class="count-char">获赞</span>

</div>

<el-divider direction="vertical"></el-divider>

<div class="user-count">

<span class="count-number">{{ author.totalCommentCount }}</span>

<span class="count-char">获评</span>

</div>

</div>

</div>

</el-card>

<!-- 推荐文章列表-->

</el-col>

</el-row>

<!-- 返回顶部-->

<el-backtop></el-backtop>

</div>

</template>

<script>

import { getToken } from '@/utils/auth'

import { hljs } from '@/assets/js/highhight'

import {

getSingleArticle,

viewArticle,

likeArticle,

dislikeArticle,

} from '@/api/article'

import { getAuthorInfo } from '@/api/user'

import { getCommentByArticle, likeComment, postComment } from '@/api/comment'

import { isLiked } from '@/api/like'

export default {

data() {

var checkEmail = (rule, value, callback) => {

const mailReg = /^([a-zA-Z0-9\_-])+@([a-zA-Z0-9\_-])+(.[a-zA-Z0-9\_-])+/

if (!value) {

return callback(new Error('邮箱不能为空'))

}

setTimeout(() => {

if (mailReg.test(value)) {

callback()

} else {

callback(new Error('请输入正确的邮箱格式'))

}

}, 100)

}

return {

// 用户有没有登录

userlogin: false,

// 用户头像地址

avatarUrl: '',

// 评论表单

form: {

articleId: '',

name: '',

email: '',

content: '',

},

// 评论校验规则

rules: {

name: [

{ required: true, message: '请输入用户名', trigger: 'blur' },

{

min: 2,

max: 20,

message: '长度在 2 到 20 个字符',

trigger: 'blur',

},

],

email: [

{ required: true, message: '请输入邮箱', trigger: 'blur' },

{ validator: checkEmail, message: '格式不正确', trigger: 'blur' },

],

content: [{ required: true, message: '请输入内容', trigger: 'blur' }],

},

// 作者total信息

author: {},

// 文章信息

article: {},

// 评论列表

commentList: [],

// 评论分页

pagination: {

pn: 1,

total: 1,

size: 10,

},

// 文章id对象

query: {},

like: {

liked: false,

color: '#fffff',

},

}

},

created() {

this.renderPage()

this.userlogin = getToken() === undefined // 有没有登陆

this.query = this.$router.query

this.avatarUrl = this.$store.getters.avatar

},

filters: {

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

},

},

methods: {

// 将用户头像连接到后台

linkToDashBoard() {

// 判断用户是否登录

this.islogin = (undefined !== getToken())

if (this.islogin) this.$router.push('/dashboard')

else this.$router.push('/login')

},

async renderPage() {

var \_this = this

var query = this.$route.query

this.query = this.$route.query

// 文章信息

// 等待文章信息拉取结束

await getSingleArticle(query.id).then((response) => {

\_this.article = response.extend.article

})

this.addViewCount(query.id)

// 作者信息

getAuthorInfo(\_this.article.articleAuthor).then((response) => {

\_this.author = response.extend.data

// 作者头像

\_this.authorAvatar =

'http://101.201.64.102:8000/avatar/' + \_this.author.userName + '.png'

})

// 评论信息

this.get\_comment\_list(this.$route.query, 1)

// 作者是否对这个文章点过赞

isLiked(this.$route.query.id).then((res) => {

\_this.like.liked = res.extend.suc

})

hljs.highlightAll() // 渲染代码

},

get\_comment\_list(query, pn) {

var \_this = this

var id = query.id

getCommentByArticle(id, pn, 7).then((response) => {

var pageInfo = response.extend.pageInfo

\_this.pagination.total = pageInfo.total

\_this.pagination.pn = pageInfo.current

\_this.pagination.size = pageInfo.size

\_this.commentList = pageInfo.records

})

},

commentPageChange(currentPage) {

this.get\_comment\_list(this.$route.query, currentPage)

},

onSubmit() {

var \_this = this

// 提交评论

this.form.articleId = this.$route.query.id

this.$refs[this.form].validate((valid) => {

if (valid) {

postComment(\_this.form).then((res) => {

\_this.form.name = ''

\_this.form.email = ''

\_this.form.content = ''

\_this.$message({

showClose: true,

message: '成功',

type: 'success',

})

this.get\_comment\_list(\_this.$route.query, 1)

})

} else {

return false

}

})

},

likeComment(commentId) {

likeComment(commentId).then((res) => {})

},

likeArticle() {

var \_this = this

if (this.like.liked) {

dislikeArticle(this.article.articleId).then((res) => {

\_this.article.articleLikeCount -= 1

\_this.like.liked = false

})

} else {

likeArticle(this.article.articleId).then((res) => {

\_this.article.articleLikeCount += 1

\_this.like.liked = true

})

}

},

addViewCount(id) {

var \_this = this

viewArticle(id).then((res) => {

\_this.article.articleViewCount += 1

})

},

},

}

</script>

<style lang='scss' scoped>

@import '../../assets/css/highlight.css';

a {

text-decoration: none;

}

body {

background-color: white;

}

/\*导航栏\*/

.logo {

font-size: 30px;

line-height: 30px;

color: black;

}

/\*文章标题\*/

h1 {

text-align: center;

font-size: 40px;

margin: 40px 0;

font-weight: 400;

}

/\*用户和文章卡片信息\*/

.user-count,

.article-count {

display: inline-flex;

flex-direction: column;

.count-number {

font-size: 25px;

font-weight: 550;

}

.count-char {

font-size: 15px;

}

}

.like {

fill: #f00;

}

/\* 文章内容 \*/

.md-content {

line-height: 26px;

// blockquote {

// padding: 20px;

// background-color: #f7f6f3;

// }

// code {

// display: block;

// padding: 20px;

// background-color: #f7f6f3;

// }

// h2 {

// margin: 15px 0;

// font-weight: 600;

// font-size: 30px;

// }

// h3 {

// margin: 15px 0;

// font-weight: 550;

// font-size: 25px;

// }

}

/\*评论的标题\*/

.comment-area-title {

border-bottom: 1px solid black;

}

.comment-title-chinese {

font-size: 40px;

font-family: Arial, sans-serif;

}

.comment-title-english {

font: italic 22px/30px arial, sans-serif;

}

/\*评论的内容\*/

.comment-item {

margin: 5px 0;

}

.comment-info {

margin-top: 10px;

}

.comment-info-item {

color: gray;

margin-right: 20px;

}

.pagination {

margin: 20px 0;

text-align: center;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\index\deleted-article.vue

<template>

<div id="app">

<!-- 导航栏-->

<el-row style="position: fixed; width:100%; z-index: 1">

<el-col

:span="14"

:offset="5"

>

<el-menu

default-active="0"

class="el-menu-demo"

mode="horizontal"

>

<el-menu-item>

<router-link to="/dashboard">

<span class="logo">Jancoyan</span>

</router-link>

</el-menu-item>

<el-menu-item style="float: right">

<el-avatar src="http://localhost:8080/avatar/10800.jpg">

<a

href="http://localhost:9528"

target="\_blank"

></a>

</el-avatar>

</el-menu-item>

</el-menu>

</el-col>

</el-row>

<!-- 页面主体-->

<el-row

:gutter="24"

style="padding-top: 60px"

>

<!-- 作者信息栏 和 文章目录导航-->

<el-col :span="5">

<!-- 文章信息卡片-->

<el-card

shadow="hover"

style="text-align: center;"

>

<div class="article-info">

<!-- 标题-->

<h2>{{ article.articleTitle }}</h2>

<div>{{ article.articlePostTime | dateFormat }}</div>

<el-divider></el-divider>

<!-- 文章信息栏-->

<div>

<div class="article-count">

<span class="count-number">{{ article.articleViewCount }}</span>

<span class="count-char">浏览</span>

</div>

<el-divider direction="vertical"></el-divider>

<div class="article-count">

<span class="count-number">{{ article.articleLikeCount }}</span>

<span class="count-char">获赞</span>

</div>

<el-divider direction="vertical"></el-divider>

<div class="article-count">

<span class="count-number">{{ article.articleCommentCount }}</span>

<span class="count-char">评论</span>

</div>

</div>

<el-divider></el-divider>

<!-- 文章操作栏-->

<div class="article-action">

<!-- 点赞-->

<div class="article-count">

<span

class="count-number"

>

<svg

t="1633675948290"

viewBox="0 0 1024 1024"

version="1.1"

xmlns="http://www.w3.org/2000/svg"

p-id="2028"

width="40"

height="40"

>

<path

d="M938.666667 362.666667A234.666667 234.666667 0 0 0 704 128 271.36 271.36 0 0 0 512 216.32 271.36 271.36 0 0 0 320 128 234.666667 234.666667 0 0 0 85.333333 362.666667c0 167.253333 202.666667 352 298.666667 448l97.28 97.28a32 32 0 0 0 22.613333 9.386666h16.213334a32 32 0 0 0 22.613333-9.386666L640 810.666667c96-96 298.666667-280.746667 298.666667-448z"

p-id="2029"

:fill="like.liked ? '#f00' : '#dbdbdb'"

></path>

</svg>

</span>

</div>

</div>

</div>

</el-card>

<!-- 文章目录导航-->

</el-col>

<el-col

:span="14"

style="padding: 0 60px"

>

<!-- 内容-->

<div

class="md-content"

v-html='article.articleHtml'

style="word-break:break-all;"

>

</div>

<el-divider>The End</el-divider>

<!-- 文章评论和发表评论-->

<!-- 作者关闭了评论-->

<!-- 评论区 -->

<el-col v-if="article.articleIsComment">

<!-- 评论的标题--评论 comment -->

<div class="comment-area-title">

<span class="comment-title-chinese">评论区</span>

<span class="comment-title-english">Comments</span>

</div>

<!-- 分页评论-->

<div class="comment-list">

<div

class="comment-item"

v-for="item in commentList"

:key="item.commentId"

>

<el-card shadow="never">

<!-- 评论内容-->

<div class="comment-content">

{{ item.commentContent }}

</div>

<!-- 评论信息-->

<div class="comment-info">

<!-- 作者-->

<span class="comment-info-item"><i class="el-icon-user"></i>

{{ item.commentAuthorName }}</span>

<!-- 评论时间-->

<span class="comment-info-item"><i class="el-icon-date"></i>

{{ item.commentDate | dateFormat }}</span>

<!-- 赞同的数量-->

<!-- <span class="comment-info-item"><i class="el-icon-caret-top"></i>

{{ item.commentLikeCount }}</span> -->

<!-- 回复和赞同-->

<!-- <el-row style="float: right">

<el-button

type="primary"

size="mini"

plain

@click="likeComment(item.commentId)"

>赞同</el-button>

<el-button size="mini">回复</el-button>

</el-row> -->

</div>

</el-card>

</div>

<div class="pagination">

<el-pagination

background

@current-change="commentPageChange"

layout="prev, pager, next"

:total="pagination.total"

:page-size="pagination.size"

v-if="commentList.length > 0"

>

</el-pagination>

</div>

</div>

</el-col>

</el-col>

<el-col :span="5">

<!-- 作者信息卡片-->

<el-card

shadow="hover"

style="text-align: center;"

>

<!-- 头像-->

<el-avatar

:size="80"

:src="authorAvatar"

></el-avatar>

<div class="user-info">

<!-- 用户名-->

<h3>{{ author.userName }}</h3>

<!-- 签名-->

<blockquote> {{ author.userSignature }} </blockquote>

<el-divider></el-divider>

<!-- 文章数量、获赞、收藏-->

<div>

<div class="user-count">

<span class="count-number">{{ author.totalArticle }}</span>

<span class="count-char">文章</span>

</div>

<el-divider direction="vertical"></el-divider>

<div class="user-count">

<span class="count-number">{{ author.totalLikeCount }}</span>

<span class="count-char">获赞</span>

</div>

<el-divider direction="vertical"></el-divider>

<div class="user-count">

<span class="count-number">{{ author.totalCommentCount }}</span>

<span class="count-char">获评</span>

</div>

</div>

</div>

</el-card>

<!-- 推荐文章列表-->

</el-col>

</el-row>

<!-- 返回顶部-->

<el-backtop></el-backtop>

</div>

</template>

<script>

import { getToken } from '@/utils/auth'

import { hljs } from '@/assets/js/highhight'

import { getSingleArticleDeleted} from '@/api/article'

import { getAuthorInfo } from '@/api/user'

import { getCommentByDeletedArticle } from '@/api/comment'

import { isLiked } from '@/api/like'

export default {

data() {

return {

// 用户有没有登录

userlogin: false,

// 用户头像地址

authorAvatar: 'http://localhost:8080/avatar/10417.png',

// 作者total信息

author: {},

// 文章信息

article: {},

// 评论列表

commentList: [],

// 评论分页

pagination: {

pn: 1,

total: 1,

size: 10,

},

// 文章id对象

query: {},

like: {

liked: false,

color: '#fffff',

},

}

},

created() {

this.renderPage()

this.userlogin = getToken() === undefined // 有没有登陆

this.query = this.$router.query

},

filters: {

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

},

},

methods: {

async renderPage() {

var \_this = this

var query = this.$route.query

this.query = this.$route.query

// 文章信息

// 等待文章信息拉取结束

await getSingleArticleDeleted(query.id).then((response) => {

\_this.article = response.extend.article

})

// 作者信息

getAuthorInfo(\_this.article.articleAuthor).then((response) => {

\_this.author = response.extend.data

})

// 评论信息

this.get\_comment\_list(this.$route.query, 1)

// 作者是否对这个文章点过赞

isLiked(this.$route.query.id).then((res) => {

\_this.like.liked = res.extend.suc

})

hljs.highlightAll() // 渲染代码

},

get\_comment\_list(query, pn) {

var \_this = this

var id = query.id

getCommentByDeletedArticle(id, pn, 7).then((response) => {

var pageInfo = response.extend.pageInfo

\_this.pagination.total = pageInfo.total

\_this.pagination.pn = pageInfo.current

\_this.pagination.size = pageInfo.size

\_this.commentList = pageInfo.records

})

},

commentPageChange(currentPage) {

this.get\_comment\_list(this.$route.query, currentPage)

},

},

}

</script>

<style lang='scss' scoped>

@import '../../assets/css/highlight.css';

a {

text-decoration: none;

}

body {

background-color: white;

}

/\*导航栏\*/

.logo {

font-size: 30px;

line-height: 30px;

color: black;

}

/\*文章标题\*/

h1 {

text-align: center;

font-size: 40px;

margin: 40px 0;

font-weight: 400;

}

/\*用户和文章卡片信息\*/

.user-count,

.article-count {

display: inline-flex;

flex-direction: column;

.count-number {

font-size: 25px;

font-weight: 550;

}

.count-char {

font-size: 15px;

}

}

.like {

fill: #f00;

}

/\* 文章内容 \*/

.md-content {

line-height: 26px;

}

/\*评论的标题\*/

.comment-area-title {

border-bottom: 1px solid black;

}

.comment-title-chinese {

font-size: 40px;

font-family: Arial, sans-serif;

}

.comment-title-english {

font: italic 22px/30px arial, sans-serif;

}

/\*评论的内容\*/

.comment-item {

margin: 5px 0;

}

.comment-info {

margin-top: 10px;

}

.comment-info-item {

color: gray;

margin-right: 20px;

}

.pagination {

margin: 20px 0;

text-align: center;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\index\index.vue

<template>

<div class="app">

<!-- 导航栏-->

<vue-particles

class="particles"

color="#dedede"

shapeType="polygon"

:hoverEffect="false"

:clickEffect="false"

:linesWidth="2"

v-show="backgroundMode"

></vue-particles>

<!-- 导航栏 -->

<el-row

:gutter="20"

style="position: fixed; width:100%; z-index: 1"

>

<el-col

:span="16"

:offset="4"

>

<el-menu

default-active="0"

class="el-menu-demo"

mode="horizontal"

>

<el-menu-item>

<span class="logo">NICE</span>

</el-menu-item>

<!-- <el-menu-item index="0">首页</el-menu-item> -->

<!-- <el-menu-item index="1">读书笔记</el-menu-item> -->

<el-menu-item

style="float: right"

@click="linkToDashBoard"

>

<el-avatar

size="large"

:src="avatarUrl"

>

登录

</el-avatar>

</el-menu-item>

</el-menu>

</el-col>

</el-row>

<!-- 主要部分-->

<el-row :gutter="20">

<!-- 左栏搜索文章、文章列表、分页-->

<el-col

:span="10"

:offset="4"

class="left"

>

<!-- 组件 - 搜索文章的搜索框 -->

<search-article

ref="searchArticle"

:typeList="typeList"

@submit="submit"

@reset="resetForm"

></search-article>

<!-- 简洁开关-->

<el-switch

v-model="simpleMode"

active-color="#13ce66"

active-text="简洁模式"

inactive-text="标准模式"

>

</el-switch>

<!-- 动画开关 -->

<span style="margin:0 10px"></span>

<el-switch

v-model="backgroundMode"

active-color="#13ce66"

active-text="打开背景"

inactive-text="关闭背景"

>

</el-switch>

<!-- 文章列表动态更新 -->

<div

id="article-list"

v-for="item in articleList"

:key="item.articleId"

>

<router-link

:to="base\_article\_url + item.articleId"

target="\_blank"

>

<el-card

class="box-card"

shadow="hover"

>

<div style="font-weight: 700;">

<!-- 置顶标签 -->

<el-tag

size="mini"

effect="dark"

type="danger"

v-if="item.articleRank"

>置顶</el-tag>

{{ item.articleTitle }}

</div>

<el-divider

v-if="!simpleMode"

class="el-divider"

><i class="el-icon-star-off"></i></el-divider>

<div v-if="!simpleMode">{{ item.articleSummary }}</div>

<el-divider

v-if="!simpleMode"

class="el-divider"

><i class="el-icon-star-off"></i></el-divider>

<div>

<span><i class="el-icon-user"></i> {{ item.userName }} </span>

<el-divider direction="vertical"></el-divider>

<span><i class="el-icon-time"></i> {{ item.articlePostTime | dateFormat }}</span>

<el-divider direction="vertical"></el-divider>

<span><i class="el-icon-view"></i> {{ item.articleViewCount }}</span>

</div>

</el-card>

</router-link>

</div>

<!-- 分页 -->

<div class="pagigation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="currentPage"

:page-sizes="[10, 20, 30, 40]"

:page-size="page\_size"

layout="total, sizes, prev, pager, next, jumper"

:total="total"

>

</el-pagination>

</div>

</el-col>

<!-- 右栏分类等-->

<el-col

:span="5"

:offset="1"

class="right"

>

<!-- 文章类型分类 -->

<el-card

class="box-card"

id="article\_type\_list"

>

<div

slot="header"

class="clearfix"

>

<span style="font-weight: 600">文章分类</span>

</div>

<div

v-for="item in typeList"

:key="item.typeId"

>

<a @click="get\_article\_by\_type(item.typeId)">{{ item.typeName }}</a>

</div>

</el-card>

</el-col>

</el-row>

<el-backtop></el-backtop>

</div>

</template>

<script>

import { getIndexArticleList } from '@/api/article'

import { getAllType } from '@/api/type'

import { getToken } from '@/utils/auth'

import SearchArticle from '@/components/SearchArticle'

export default {

components: { SearchArticle },

data() {

return {

islogin: false,

// 简洁模式

simpleMode: false,

// 背景开关

backgroundMode: true,

avatarUrl: '',

base\_article\_url: '/article?id=',

condition: '',

// 文章列表

articleList: [],

// 类型列表

typeList: [],

// 当前的页面

currentPage: 1,

// 每一页多少

page\_size: 10,

// 总数

total: 0,

}

},

filters: {

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

},

},

created: function () {

// 进行数据请求，拿到数据

this.get\_article\_list(1)

this.get\_type\_list()

this.avatarUrl = this.$store.getters.avatar

},

methods: {

// 将用户头像连接到后台

linkToDashBoard() {

// 判断用户是否登录

this.islogin = undefined !== getToken()

if (this.islogin) this.$router.push('/dashboard')

else this.$router.push('/login')

},

// 分页大小的改变

handleSizeChange(val) {

this.page\_size = val

this.get\_article\_list(1)

},

// 当前页面的改变

handleCurrentChange(val) {

this.get\_article\_list(val)

},

// 获取文章列表

get\_article\_list(pn) {

getIndexArticleList(pn, this.page\_size, this.condition).then(

(response) => {

var pageInfo = response.extend.pageInfo

if (pageInfo.total === 0) {

this.$message.error('没有相关文章~')

return

}

this.articleList = pageInfo.records

this.page\_size = pageInfo.size

this.total = pageInfo.total

}

)

},

// 获取类型列表

get\_type\_list(pn) {

getAllType().then((response) => {

var pageInfo = response.extend.pageInfo

this.typeList = pageInfo.records

})

},

// 提交搜索

submit() {

this.condition = this.$refs.searchArticle.generateQueryString()

this.get\_article\_list(1)

},

// 文章列表重置

resetForm(){

this.condition = ''

this.get\_article\_list(1)

},

// 从类型获取文章(搜)

get\_article\_by\_type(typeId) {

this.condition = 'type=' + String(typeId)

this.get\_article\_list(1)

},

},

}

</script>

<style lang='scss' scoped>

\*{

/\* 设置网页的字体的基调 \*/

font-family: 'Nunito', sans-serif;

/\* 设置所有盒子的展示样式 \*/

box-sizing: border-box;

/\* outline和border都是把所有元素的轮廓取消 \*/

outline: none; border: none;

/\* 字体样式 \*/

text-decoration: none;

/\* 设置所有的变化都是线性的持续0.2秒的 \*/

transition: all .2s linear;

}

a {

text-decoration: none;

}

.particles {

width: 100%;

height: 100vh;

position: fixed;

}

.app {

padding-bottom: 40px;

}

/\* 头像右浮动 \*/

.user-avatar {

float: right;

}

.el-divider {

margin: 10px;

}

.logo {

font-size: 30px;

line-height: 30px;

color: black;

}

.el-card\_\_body {

padding-bottom: 8px;

}

#article-list {

margin-top: 20px;

margin-bottom: 20px;

}

.search-bar {

margin-top: 60px;

margin-bottom: 20px;

}

#article\_type\_list {

margin-top: 80px;

width: 300px;

position: fixed;

a {

color: black;

text-decoration: none;

display: block;

height: 30px;

font-size: 16px;

padding: 10px;

&:hover {

font-weight: 600;

background-color: #ebeff5;

}

}

}

@media (max-width: 991px) {

.right{

display: none;

}

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\login\index.vue

<template>

<div class="login">

<vue-particles

class="particles"

color="#fff"

linesColor="#fff"

shapeType="polygon"

:hoverEffect="true"

:clickEffect="false"

:linesWidth="2"

:lineOpacity="0.6"

></vue-particles>

<div class="container">

<el-tabs

:value="currPanel"

:stretch="true"

type="border-card"

v-loading="loading"

element-loading-text="登陆中"

>

<!-- 登录界面 -->

<el-tab-pane

label="登录"

name="first"

>

<el-form

label-width="80px"

:model="login"

class="login-form"

>

<el-form-item label="用户名">

<el-input v-model="login.username"></el-input>

</el-form-item>

<el-form-item label="密码">

<el-input

type="password"

v-model="login.password"

@keyup.enter.native="handleLogin"

></el-input>

</el-form-item>

<el-button

type="primary"

@click="handleLogin"

>登录</el-button>

</el-form>

</el-tab-pane>

<!-- 注册界面 -->

<el-tab-pane

label="注册"

name="second"

>

<el-form

label-width="80px"

:model="register"

class="register"

:rules="rules"

:ref="register"

>

<el-form-item

label="用户名"

prop="username"

>

<el-input

type="name"

v-model="register.username"

></el-input>

</el-form-item>

<el-form-item

label="密码"

prop="password"

>

<el-input

type="password"

v-model="register.password"

></el-input>

</el-form-item>

<el-form-item

label="确认密码"

prop="password2"

>

<el-input

type="password"

v-model="register.password2"

></el-input>

</el-form-item>

<el-form-item

label="验证码"

prop="code"

>

<el-col :span="12">

<el-input

type="text"

v-model="register.code"

@keyup.enter.native="handleRegister"

>

</el-input>

</el-col>

<el-image

:src="verifyCode"

@click="generateVerifyCode"

></el-image>

</el-form-item>

<el-button

type="primary"

@click="handleRegister"

>注册</el-button>

</el-form>

</el-tab-pane>

</el-tabs>

</div>

</div>

</template>

<script>

import { checkUserNameUnique, register } from '@/api/user'

export default {

name: 'Login',

data() {

var checkUserName = (rule, value, callback) => {

if (!value) {

return callback(new Error('用户名不能为空'))

}

checkUserNameUnique(value).then((res) => {

if (res.extend.unique) {

callback()

} else {

return callback(new Error('用户名已被注册'))

}

})

}

var checkPassAgain = (rule, value, callback) => {

if (!value) {

return callback(new Error('请输入密码'))

}

if (value !== this.register.password) {

return callback(new Error('两次密码不一致'))

}

callback()

}

return {

currPanel: 'first',

login: {

username: '',

password: '',

},

register: {

username: '',

password: '',

password2: '',

code: '',

},

rules: {

username: [

{ required: true, validator: checkUserName, trigger: 'blur' },

],

password: [{ required: true, message: '请输入密码', trigger: 'blur' }],

password2: [

{ required: true, validator: checkPassAgain, trigger: 'blur' },

],

code: [{ required: true, message: '请输入验证码', trigger: 'blur' }],

},

loading: false,

registerVisiable: false,

verifyCode: '',

}

},

created() {

this.generateVerifyCode()

},

methods: {

generateVerifyCode() {

var url = 'http://101.201.64.102:8000/user/getverifycode?' + Math.random()

// var url = 'http://localhost:8000/user/getverifycode?' + Math.random()

this.verifyCode = url

},

handleLogin() {

this.loading = true

this.$store

.dispatch('user/login', this.login)

.then((res) => {

// 登录成功之后显示登录成功，然后跳转到首页

this.$message({

message: '登录成功',

type: 'success',

})

setTimeout(() => {

this.$router.push({ path: '/' })

}, 1000)

this.loading = false

})

.catch(() => {

this.loading = false

})

},

async handleRegister() {

var \_this = this

this.$refs[this.register].validate((valid) => {

if (valid) {

// 提交注册表单

register(this.register).then((res) => {

if (res.extend.success) {

// 成功

this.$message({

message: '注册成功',

type: 'success',

})

// 跳转登录

\_this.login.username = \_this.register.username

\_this.login.password = \_this.register.password

\_this.register.username = ''

\_this.register.password = ''

\_this.register.password2 = ''

\_this.register.code = ''

\_this.currPanel = 'first'

this.generateVerifyCode()

} else {

// 失败

this.generateVerifyCode()

}

})

} else {

// 告知注册失败

return false

}

})

},

},

}

</script>

<style lang="scss" scoped>

.particles {

width: 100%;

height: 90vh;

position: absolute;

}

.login {

width: 100vw;

height: 100vh;

background-image: linear-gradient(#0ed2f7, #0ed2f7);

display: flex;

justify-content: center;

align-items: center;

}

.container {

margin-bottom: 5%;

width: 400px;

float: right;

.el-button {

width: 80%;

margin: 0 17%;

}

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\personal\change-password.vue

<template>

<div class="change-password">

<el-row>

<el-col

:span="12"

:offset="6"

>

<el-form

label-width="80px"

:model="password"

class="password"

:rules="rules"

:ref="password"

>

<el-form-item

label="旧密码"

prop="old"

>

<el-input

type="password"

v-model="password.old"

></el-input>

</el-form-item>

<el-form-item

label="新密码"

prop="password"

>

<el-input

type="password"

v-model="password.password"

></el-input>

</el-form-item>

<el-form-item

label="确认密码"

prop="password2"

>

<el-input

type="password"

v-model="password.password2"

></el-input>

</el-form-item>

<el-button

type="primary"

@click="changePwd"

>修改密码</el-button>

</el-form>

</el-col>

</el-row>

</div>

</template>

<script>

import { changePassword,logout } from '@/api/user'

export default {

data() {

var checkPassAgain = (rule, value, callback) => {

if (!value) {

return callback(new Error('请输入密码'))

}

if (value !== this.password.password) {

return callback(new Error('两次密码不一致'))

}

callback()

}

return {

password: {

old: '',

password: '',

password2: '',

},

rules: {

old: [{ required: true, message: '请输入密码', trigger: 'blur' }],

password: [{ required: true, message: '请输入密码', trigger: 'blur' }],

password2: [

{ required: true, validator: checkPassAgain, trigger: 'blur' },

],

},

}

},

methods: {

changePwd() {

var \_this = this

this.$refs[this.password].validate((valid) => {

if (valid) {

changePassword(

'',

\_this.password.old,

\_this.password.password2

).then((res) => {

let suc = res.extend.suc

if(suc){

// 修改成功

this.$message({

message: '修改成功，请重新登陆',

type: 'success',

duration: 3000

})

// 先退出登录，跳转登录页面

logout().then(res => {

if(res.extend.suc){

// 退出成功，跳转登录页面

\_this.$router.push('/login')

}

})

} else {

this.$message.error('旧密码不正确')

}

})

} else {

// 告知注册失败

return false

}

})

},

},

}

</script>

<style scoped>

.change-password {

padding: 20px;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\personal\component\CropperImage.vue

<template>

<div class="cropper-content">

<div class="cropper-box">

<div class="cropper">

<vue-cropper

ref="cropper"

:img="option.img"

:outputSize="option.outputSize"

:outputType="option.outputType"

:info="option.info"

:canScale="option.canScale"

:autoCrop="option.autoCrop"

:autoCropWidth="option.autoCropWidth"

:autoCropHeight="option.autoCropHeight"

:fixed="option.fixed"

:fixedNumber="option.fixedNumber"

:full="option.full"

:fixedBox="option.fixedBox"

:canMove="option.canMove"

:canMoveBox="option.canMoveBox"

:original="option.original"

:centerBox="option.centerBox"

:height="option.height"

:infoTrue="option.infoTrue"

:maxImgSize="option.maxImgSize"

:enlarge="option.enlarge"

:mode="option.mode"

@realTime="realTime"

@imgLoad="imgLoad"

>

</vue-cropper>

</div>

<!--底部操作工具按钮-->

<div class="footer-btn">

<div class="scope-btn">

<label

class="btn"

for="uploads"

>选择图片</label>

<input

type="file"

id="uploads"

style="position:absolute; clip:rect(0 0 0 0);"

accept="image/png, image/jpeg, image/gif, image/jpg"

@change="selectImg($event)"

>

<el-button

size="mini"

type="danger"

plain

icon="el-icon-zoom-in"

@click="changeScale(1)"

>放大</el-button>

<el-button

size="mini"

type="danger"

plain

icon="el-icon-zoom-out"

@click="changeScale(-1)"

>缩小</el-button>

<el-button

size="mini"

type="danger"

plain

@click="rotateLeft"

>↺ 左旋转</el-button>

<el-button

size="mini"

type="danger"

plain

@click="rotateRight"

>↻ 右旋转</el-button>

</div>

<div class="upload-btn">

<el-button

size="mini"

type="success"

@click="uploadImg('blob')"

>上传 <i class="el-icon-upload"></i></el-button>

</div>

</div>

</div>

<!--预览效果图-->

<div class="show-preview">

<div

:style="previews.div"

class="preview"

>

<img

:src="previews.url"

:style="previews.img"

>

</div>

</div>

</div>

</template>

<script>

import { VueCropper } from 'vue-cropper'

import { getToken } from '@/utils/auth'

export default {

name: 'CropperImage',

components: {

VueCropper,

},

props: ['Name'],

data() {

return {

name: this.Name,

previews: {},

option: {

img: '', //裁剪图片的地址

outputSize: 1, //裁剪生成图片的质量(可选0.1 - 1)

outputType: 'png', //裁剪生成图片的格式（jpeg || png || webp）

info: true, //图片大小信息

canScale: true, //图片是否允许滚轮缩放

autoCrop: true, //是否默认生成截图框

autoCropWidth: 256, //默认生成截图框宽度

autoCropHeight: 256, //默认生成截图框高度

fixed: true, //是否开启截图框宽高固定比例

fixedNumber: [1, 1], //截图框的宽高比例

full: false, //false按原比例裁切图片，不失真

fixedBox: true, //固定截图框大小，不允许改变

canMove: false, //上传图片是否可以移动

canMoveBox: true, //截图框能否拖动

original: false, //上传图片按照原始比例渲染

centerBox: false, //截图框是否被限制在图片里面

height: true, //是否按照设备的dpr 输出等比例图片

infoTrue: false, //true为展示真实输出图片宽高，false展示看到的截图框宽高

maxImgSize: 2000, //限制图片最大宽度和高度

enlarge: 1, //图片根据截图框输出比例倍数

mode: '256px 256px', //图片默认渲染方式

},

}

},

methods: {

//初始化函数

imgLoad(msg) {},

//图片缩放

changeScale(num) {

num = num || 1

this.$refs.cropper.changeScale(num)

},

//向左旋转

rotateLeft() {

this.$refs.cropper.rotateLeft()

},

//向右旋转

rotateRight() {

this.$refs.cropper.rotateRight()

},

//实时预览函数

realTime(data) {

this.previews = data

},

//选择图片

selectImg(e) {

let file = e.target.files[0]

if (!/\.(jpg|jpeg|png|JPG|PNG)$/.test(e.target.value)) {

this.$message({

message: '图片类型要求：jpeg、jpg、png',

type: 'error',

})

return false

}

//转化为blob

let reader = new FileReader()

reader.onload = (e) => {

let data

if (typeof e.target.result === 'object') {

data = window.URL.createObjectURL(new Blob([e.target.result]))

} else {

data = e.target.result

}

this.option.img = data

}

//转化为base64

reader.readAsDataURL(file)

},

//上传图片

uploadImg(type) {

let \_this = this

if (type === 'blob') {

//获取截图的blob数据

this.$refs.cropper.getCropBlob(async (data) => {

let formData = new FormData()

formData.append('file', data, 'file.png')

//调用

const ajax = new XMLHttpRequest()

ajax.open('POST', 'http://101.201.64.102:8000/user/upload/avatar', true)

// ajax.open('POST', 'http://localhost:8000/user/upload/avatar', true)

ajax.setRequestHeader('token', getToken())

ajax.send(formData)

ajax.onreadystatechange = function () {

if(ajax.readyState === 4){

\_this.$emit('uploadImgSuccess', ajax.responseText)

}

}

})

}

},

},

}

</script>

<style scoped lang="scss">

.cropper-content {

display: flex;

display: -webkit-flex;

justify-content: flex-end;

.cropper-box {

flex: 1;

width: 100%;

.cropper {

width: auto;

height: 300px;

}

}

.show-preview {

flex: 1;

-webkit-flex: 1;

display: flex;

display: -webkit-flex;

justify-content: center;

.preview {

overflow: hidden;

border: 1px solid #67c23a;

background: #cccccc;

}

}

}

.footer-btn {

margin-top: 30px;

display: flex;

display: -webkit-flex;

justify-content: flex-end;

.scope-btn {

display: flex;

display: -webkit-flex;

justify-content: space-between;

padding-right: 10px;

}

.upload-btn {

flex: 1;

-webkit-flex: 1;

display: flex;

display: -webkit-flex;

justify-content: center;

}

.btn {

outline: none;

display: inline-block;

line-height: 1;

white-space: nowrap;

cursor: pointer;

-webkit-appearance: none;

text-align: center;

-webkit-box-sizing: border-box;

box-sizing: border-box;

outline: 0;

-webkit-transition: 0.1s;

transition: 0.1s;

font-weight: 500;

padding: 8px 15px;

font-size: 12px;

border-radius: 3px;

color: #fff;

background-color: #409eff;

border-color: #409eff;

margin-right: 10px;

}

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\personal\login-log.vue

<template>

<div class="login-log">

<el-col

:span="14"

:offset="5"

>

<el-table

:data="login\_list"

style="width: 100%"

>

<el-table-column

label="登录日期"

width="250"

>

<template slot-scope="scope">

<i class="el-icon-time"></i>

<span style="margin-left: 10px">{{ scope.row.loginDate | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="loginAddress"

label="地址"

width="180"

>

</el-table-column>

<el-table-column

prop="loginIp"

label="ip"

>

</el-table-column>

<el-table-column

prop="osName"

label="系统"

>

</el-table-column>

<el-table-column

prop="browserName"

label="浏览器"

>

</el-table-column>

</el-table>

</el-col>

</div>

</template>

<script>

import { getLoginLog } from '@/api/login'

export default {

data() {

return {

login\_list: [],

}

},

filters: {

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

},

},

created() {

this.get\_login\_log()

},

methods: {

get\_login\_log() {

getLoginLog().then((res) => {

this.login\_list = res.extend.pageInfo.records

})

},

},

}

</script>

<style scoped>

.login-log {

background-color: white;

margin: 20px;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\personal\personal-setting.vue

<template>

<div class="personal-settings">

<el-row :gutter="10">

<el-col

:span="18"

:offset="3"

>

<!-- 展示信息: 头像、用户名、签名 -->

<el-card class="show-info">

<!-- 用户头像 -->

<el-col :span="5">

<el-avatar

:size="120"

:src="userAvatar"

></el-avatar>

</el-col>

<!-- 用户信息 -->

<el-col :span="19">

<el-form :disabled="!editBtn.user">

<el-form-item label="用户名">

<el-input v-model="user.userName"></el-input>

</el-form-item>

<el-form-item label="签名">

<el-input v-model="user.userSignature"></el-input>

</el-form-item>

</el-form>

<el-button

type="primary"

@click="submit\_user"

v-show="editBtn.user"

>提交</el-button>

<el-button

type="primary"

v-show="editBtn.user"

@click="changeAvatarVisiable = true"

>修改头像</el-button>

<el-button

v-show="editBtn.user"

@click="editBtn.user = false;get\_user\_info()"

>取消</el-button>

<el-button

class="edit-btn"

v-show="!editBtn.user"

@click="editBtn.user = true"

>编辑</el-button>

</el-col>

</el-card>

<!-- 个人信息：真实姓名、性别、邮箱、电话、地区、生日 -->

<el-card>

<h4>个人信息</h4>

<el-form

ref="form"

:model="userInfo"

label-width="80px"

:disabled="!editBtn.personal"

>

<el-form-item label="真实姓名">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userRealName"></el-input>

</el-col>

</el-form-item>

<el-form-item label="性别">

<el-select

v-model="userInfo.userSex"

placeholder="选择"

>

<el-option

label="男"

value="1"

></el-option>

<el-option

label="女"

value="0"

></el-option>

<el-option

label="未知"

value="-1"

></el-option>

</el-select>

</el-form-item>

<el-form-item label="邮箱">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userEmail"></el-input>

</el-col>

</el-form-item>

<el-form-item label="电话">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userTelephone"></el-input>

</el-col>

</el-form-item>

<el-form-item label="地区">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userRegion"></el-input>

</el-col>

</el-form-item>

<el-form-item label="生日">

<el-col :span="11">

<el-date-picker

type="date"

placeholder="选择日期"

v-model="userInfo.userBirthdate"

></el-date-picker>

</el-col>

</el-form-item>

<el-form-item>

</el-form-item>

</el-form>

<el-button

type="primary"

@click="submit\_user\_info"

v-show="editBtn.personal"

>提交</el-button>

<el-button

v-show="editBtn.personal"

@click="editBtn.personal = false;get\_user\_info()"

>取消</el-button>

<el-button

class="edit-btn"

v-show="!editBtn.personal"

@click="editBtn.personal = true"

>编辑</el-button>

</el-card>

<!-- 教育信息：学校、学位、入学时间、最高学历 -->

<el-card>

<h4>教育信息</h4>

<el-form

ref="form"

:model="userInfo"

label-width="80px"

:disabled="!editBtn.education"

>

<el-form-item label="毕业院校">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userSchool"></el-input>

</el-col>

</el-form-item>

<el-form-item label="专业">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userMajor"></el-input>

</el-col>

</el-form-item>

<el-form-item label="入学时间">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userEnterSchoolDate"></el-input>

</el-col>

</el-form-item>

<el-form-item label="最高学历">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userAcademicDegree"></el-input>

</el-col>

</el-form-item>

<el-form-item>

</el-form-item>

</el-form>

<el-button

type="primary"

@click="submit\_user\_info"

v-if="editBtn.education"

>提交</el-button>

<el-button

v-if="editBtn.education"

@click="editBtn.education = false;get\_user\_info()"

>取消</el-button>

<el-button

class="edit-btn"

v-if="!editBtn.education"

@click="editBtn.education = true"

>编辑</el-button>

</el-card>

<!-- 工作信息：公司、职位、领域 -->

<el-card>

<h4>工作信息</h4>

<el-form

ref="form"

:model="userInfo"

label-width="80px"

:disabled="!editBtn.work"

>

<el-form-item label="公司">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userCompany"></el-input>

</el-col>

</el-form-item>

<el-form-item label="职位">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userPosition"></el-input>

</el-col>

</el-form-item>

<el-form-item label="领域">

<el-col

:span="structure.formSpan"

:offset="structure.formOffset"

>

<el-input v-model="userInfo.userField"></el-input>

</el-col>

</el-form-item>

</el-form>

<el-button

type="primary"

@click="submit\_user\_info"

v-if="editBtn.work"

>提交</el-button>

<el-button

v-if="editBtn.work"

@click="editBtn.work = false;get\_user\_info()"

>取消</el-button>

<el-button

class="edit-btn"

v-if="!editBtn.work"

@click="editBtn.work = true"

>编辑</el-button>

</el-card>

</el-col>

</el-row>

<el-dialog

title="裁剪头像"

:visible.sync="changeAvatarVisiable"

width="950px"

center

>

<cropper-image

:Name="cropperName"

@uploadImgSuccess="handleUploadSuccess"

ref="child"

>

</cropper-image>

</el-dialog>

</div>

</template>

<script>

import CropperImage from './component/CropperImage'

import {

getUserDetailInfo,

getUserInfo,

updateUser,

updateUserInfo,

} from '@/api/user'

export default {

data() {

return {

structure: {

formSpan: 10,

formOffset: 0,

},

editBtn: {

sign: false,

education: false,

personal: false,

work: false,

user: false,

},

changeAvatarVisiable: false,

user: {},

userInfo: {},

userAvatar: '',

//裁切图片参数

cropperModel: false,

cropperName: 'CropperImage',

imgName: '',

imgVisible: false,

}

},

components: {

'cropper-image': CropperImage,

},

created() {

this.get\_user\_info()

},

methods: {

get\_user\_info() {

var \_this = this

getUserDetailInfo().then((res) => {

\_this.userInfo = res.extend.info

\_this.userAvatar =

'http://localhost:8000/avatar/' + \_this.user.userName + '.png'

})

getUserInfo().then((res) => {

\_this.user = res.extend.user

})

},

submit\_user\_info() {

updateUserInfo(this.userInfo).then((res) => {

if (res.extend.suc) {

// 成功

// 修改成功

this.$message({

message: '修改成功',

type: 'success',

duration: 3000,

})

this.editBtn.personal = false

this.editBtn.education = false

this.editBtn.work = false

this.get\_user\_info()

} else {

// 修改成功

this.$message({

message: '修改失败',

type: 'fail',

duration: 3000,

})

}

})

},

submit\_user() {

updateUser(this.user.userName, this.user.userSignature).then((res) => {

if (res.extend.suc) {

// 成功

// 修改成功

this.$message({

message: '修改成功',

type: 'success',

duration: 3000,

})

this.editBtn.personal = false

this.editBtn.education = false

this.editBtn.work = false

this.editBtn.user = false

this.get\_user\_info()

} else {

// 修改成功

this.$message({

message: '修改失败',

type: 'fail',

duration: 3000,

})

}

})

},

//图片上传成功后

handleUploadSuccess(data) {

if (JSON.parse(data).extend.suc) {

this.$message({

message: '修改成功',

type: 'success',

})

this.cropperModel = false

this.changeAvatarVisiable = false

// 重新获取用户头像

this.get\_user\_info()

} else {

this.$message.error('修改失败');

this.cropperModel = false

this.changeAvatarVisiable = false

}

},

},

}

</script>

<style lang="scss" scoped>

.personal-settings {

padding: 10px;

.el-card {

margin-top: 10px;

}

// 展示头像、用户名、签名

.show-info {

padding-bottom: 15px;

.user-name {

font-size: 25px;

font-weight: 600;

margin: 20px;

}

.signature {

font-size: 18px;

margin: 20px;

}

}

// 编辑按钮

.edit-btn {

float: right;

margin-bottom: 8px;

}

}

///////////////////////////////////////////

.upload-list-cover {

position: absolute;

top: 0;

bottom: 0;

left: 0;

right: 0;

display: flex;

flex-wrap: wrap;

justify-content: space-between;

padding: 0 40px;

align-items: center;

background: rgba(0, 0, 0, 0.6);

opacity: 0;

transition: opacity 1s;

}

.cover\_icon {

font-size: 30px;

}

.upload-btn {

display: -webkit-box;

display: -ms-flexbox;

display: flex;

-ms-flex-wrap: wrap;

flex-wrap: wrap;

-webkit-box-pack: center;

-ms-flex-pack: center;

justify-content: center;

-webkit-box-align: center;

-ms-flex-align: center;

align-items: center;

border: 1px solid #cccccc;

border-radius: 5px;

overflow: hidden;

box-shadow: 0 0 1px #cccccc;

}

.upload-btn:hover {

border: 1px solid #69b7ed;

}

.upload-btn i {

margin: 5px;

}

</style>

R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\settings\process.vue

<template>

<div class="timeline">

<el-col span="12" offset="4">

<el-timeline>

<el-timeline-item timestamp="2021/10/01" placement="top">

<el-card>

<h4>JancoBlog v3.0 发布</h4>

<p>前端：使用 Vue.js + ElementUI 开发</p>

<p>后端：使用SpringBoot框架开发</p>

</el-card>

</el-timeline-item>

<el-timeline-item timestamp="2021/07/10" placement="top">

<el-card>

<h4>JancoBlog v2.0 发布</h4>

<p>前端：使用LayUI组件库</p>

<p>后端：使用SpringBoot框架开发</p>

</el-card>

</el-timeline-item>

<el-timeline-item timestamp="2021/05/06" placement="top">

<el-card>

<h4>JancoBlog v1.0 发布</h4>

<p>前端：使用任何框架，纯手写</p>

<p>后端：使用SSM（Spring/SpringMVC/MyBatis）框架开发</p>

</el-card>

</el-timeline-item>

</el-timeline>

</el-col>

</div>

</template>

<style scoped>

.timeline{

margin: 20px;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\user\user.vue

<template>

<div class="main">

<!-- 搜索框和添加用户 -->

<el-col :span="12">

<!-- 搜索框 -->

<el-collapse

accordion

class="search-bar"

>

<!-- 搜索 -->

<el-collapse-item>

<template slot="title">

<i

class="header-icon el-icon-search"

style="font-size:20px;margin-right: 5px;"

></i>筛选用户

</template>

<el-form

ref="form"

label-width="80px"

>

<el-form-item label="用户名">

<el-input v-model="query.user\_name"></el-input>

</el-form-item>

<el-form-item label="注册时间">

<el-col :span="11">

<el-date-picker

type="date"

placeholder="开始日期"

v-model="query.start"

style="width: 100%;"

></el-date-picker>

</el-col>

<el-col :span="11">

<el-date-picker

type="date"

placeholder="结束日期"

v-model="query.end"

style="width: 100%;"

></el-date-picker>

</el-col>

</el-form-item>

<el-form-item>

<el-button

type="primary"

@click="submitSearch"

>搜索</el-button>

<el-button @click="resetForm">重置</el-button>

</el-form-item>

</el-form>

</el-collapse-item>

<!-- 添加用户 -->

<el-collapse-item>

<template slot="title">

<i

class="header-icon el-icon-plus"

style="font-size:20px;margin-right: 5px;"

></i>添加用户

</template>

<el-form

label-width="80px"

:model="register"

class="register"

:rules="rules"

:ref="register"

>

<el-form-item

label="用户名"

prop="username"

>

<el-input

type="name"

v-model="register.username"

></el-input>

</el-form-item>

<el-form-item

label="密码"

prop="password"

>

<el-input

type="password"

v-model="register.password"

></el-input>

</el-form-item>

<el-form-item

label="确认密码"

prop="password2"

>

<el-input

type="password"

v-model="register.password2"

></el-input>

</el-form-item>

<el-button

type="primary"

@click="handleRegister"

>添加</el-button>

</el-form>

</el-collapse-item>

</el-collapse>

</el-col>

<!-- 用户表格 -->

<el-table

:data="tableData"

border

style="width: 100%"

v-loading="loading"

@selection-change="handleSelectionChange"

>

<el-table-column

type="selection"

width="55"

>

</el-table-column>

<el-table-column

prop="userName"

label="用户名"

width="150"

>

</el-table-column>

<el-table-column

label="注册日期"

width="200"

>

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.userCreateDate | dateFormat }}</span>

</template>

</el-table-column>

<!-- <el-table-column

label="上次登录"

width="200"

>

<template slot-scope="scope">

<span style="margin-left: 10px">{{ scope.row.userLastLoginDate | dateFormat }}</span>

</template>

</el-table-column>

<el-table-column

prop="userIp"

label="IP"

width="150"

>

</el-table-column> -->

<el-table-column

prop="userRole"

label="权限"

sortable

width="100"

>

</el-table-column>

<el-table-column

label="操作"

width="220"

>

<template slot-scope="scope">

<!-- <el-button

@click="viewDetail(scope.row)"

type="text"

size="small"

>详细信息</el-button> -->

<el-button

@click="changePwdDialog(scope.row)"

type="text"

size="small"

>修改密码</el-button>

<el-button

@click="deleteUser(scope.row)"

type="text"

size="small"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<div class="btn-group">

<el-button

type="danger"

@click="batchDelete"

>删除选中</el-button>

</div>

<!-- 分页 -->

<div class="pagiation">

<el-pagination

@size-change="handleSizeChange"

@current-change="handleCurrentChange"

:current-page="pn"

:page-sizes="[10, 20, 30, 40]"

:page-size="limit"

layout="total, sizes, prev, pager, next, jumper"

:total="total"

>

</el-pagination>

</div>

<el-dialog title="修改密码" :visible.sync="changePwdVisable">

<el-form

label-width="80px"

:model="password"

class="password"

:rules="rules"

:ref="password"

>

<el-form-item

label="新密码"

prop="password"

>

<el-input

type="password"

v-model="password.password"

></el-input>

</el-form-item>

<el-button

type="primary"

@click="changePwd"

>修改密码</el-button>

</el-form>

</el-dialog>

</div>

</template>

<script>

import { getAll, batchDeleteUsers, checkUserNameUnique, addUser, changePassword } from '@/api/user'

import { parseTime } from '@/utils/index'

export default {

data() {

var checkUserName = (rule, value, callback) => {

if (!value) {

return callback(new Error('用户名不能为空'))

}

checkUserNameUnique(value).then((res) => {

if (res.extend.unique) {

callback()

} else {

return callback(new Error('用户名已被注册'))

}

})

}

var checkPassAgain = (rule, value, callback) => {

if (!value) {

return callback(new Error('请输入密码'))

}

if (value !== this.register.password) {

return callback(new Error('两次密码不一致'))

}

callback()

}

return {

loading: false,

tableData: [],

types: [],

pn: 1,

limit: 10,

total: 0,

multipleSelection: [],

condition: '',

query: {

user\_name: '',

start: '',

end: '',

},

commentDetailVisable: false,

changePwdVisable: false,

commentDatail: {},

register: {

username: '',

password: '',

},

password: {

id: '',

password: '',

},

rules: {

username: [

{ required: true, validator: checkUserName, trigger: 'blur' },

],

password: [{ required: true, message: '请输入密码', trigger: 'blur' }],

password2: [

{ required: true, validator: checkPassAgain, trigger: 'blur' },

],

}

}

},

created() {

this.get\_user\_list(1)

},

filters: {

dateFormat(date) {

if (date === null) return ''

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

},

},

methods: {

changePwdDialog(col){

this.password.id = col.userId

this.changePwdVisable = true

},

changePwd() {

var \_this = this

this.$refs[this.password].validate((valid) => {

if (valid) {

changePassword(

\_this.password.id,

'',

\_this.password.password

).then((res) => {

let suc = res.extend.suc

if(suc){

// 修改成功

this.$message({

message: '修改成功',

type: 'success',

duration: 3000

})

\_this.password.password = ''

\_this.changePwdVisable = false

}

})

} else {

// 告知注册失败

return false

}

})

},

viewDetail(row) {

// this.commentDetailVisable = false

},

handleSizeChange(val) {

this.limit = val

this.get\_user\_list(1)

},

handleCurrentChange(val) {

this.get\_user\_list(val)

},

deleteUser(row) {

var msg = '将要删除该用户 , 是否继续?'

this.$confirm(msg, '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteUsers(row.userId).then((response) => {

if (response.extend.suc === 'success') {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_user\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

})

})

},

get\_user\_list(pn) {

this.loading = true

getAll(pn, this.limit, this.condition).then((response) => {

var pageInfo = response.extend.pageInfo

this.tableData = pageInfo.records

this.total = pageInfo.total

})

this.loading = false

},

handleSelectionChange(val) {

this.multipleSelection = val

},

batchDelete() {

var ids = ''

if (this.multipleSelection.length === 0) {

this.$message.error('请先选择')

return

}

// 字符串拼接

this.multipleSelection.forEach((item) => {

ids += item.userId + '&'

})

this.$confirm('确定删除选中用户？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning',

})

.then(() => {

batchDeleteUsers(ids).then((response) => {

if (response.extend.suc === 'success') {

this.$message({

type: 'success',

message: '删除成功!',

})

this.get\_user\_list(this.pn)

} else {

this.$message.error('删除失败!')

}

})

})

.catch(() => {

this.$message({

type: 'info',

message: '已取消删除',

duration: 1000,

})

})

},

submitSearch() {

this.generateQueryString()

this.get\_user\_list(1)

},

resetForm() {

this.query.user\_name = ''

this.query.start = ''

this.query.end = ''

this.condition = ''

this.get\_user\_list(1)

},

generateQueryString() {

let condition = ''

let query = this.query

if (query.user\_name !== '') {

condition += 'user\_name=' + query.user\_name + '--'

}

if (query.start !== '') {

condition += 'start=' + parseTime(query.start) + '--'

}

if (query.end !== '') {

condition += 'end=' + parseTime(query.end) + '--'

}

condition =

condition.lastIndexOf('--') === condition.length - 2

? condition.substr(0, condition.length - 2)

: condition

this.condition = condition

},

handleRegister() {

var \_this = this

this.$refs[this.register].validate((valid) => {

if (valid) {

// 提交注册表单

addUser(this.register).then((res) => {

if (res.extend.success) {

// 成功

this.$message({

message: '注册成功',

type: 'success',

})

// 跳转登录

\_this.register.username = ''

\_this.register.password = ''

\_this.register.password2 = ''

\_this.currPanel = 'first'

} else {

// 失败

this.$message.error('注册失败')

}

})

} else {

// 告知注册失败

return false

}

})

},

},

}

</script>

<style scoped>

.pagiation {

margin-top: 30px;

margin-left: 30px;

}

.btn-group {

margin: 20px;

}

.main {

margin: 20px;

}

.search-bar {

margin-bottom: 20px;

}

</style>R:\GITHUB\JancoBlog\BackEnd\jancoblogback\src\views\write\write.vue

<template>

<div class="write">

<el-form

:ref="post\_article"

label-width="80px"

:rules="rules"

:model="post\_article"

>

<el-form-item

label="标题"

prop="article\_title"

>

<el-input

v-model="post\_article.article\_title"

maxlength="50"

></el-input>

</el-form-item>

<el-form-item

label="摘要"

prop="article\_summary"

>

<el-input

v-model="post\_article.article\_summary"

placeholder="如果不写，就会截取内容的前100字符"

maxlength="150"

></el-input>

</el-form-item>

<el-form-item label="类型" prop="article\_type">

<el-select

v-model="post\_article.article\_type"

filterable

placeholder="选择或搜索"

>

<el-option

v-for="item in typeList"

:key="item.typeId"

:label="item.typeName"

:value="item.typeId"

>

</el-option>

</el-select>

</el-form-item>

<el-form-item>

<el-switch

v-model="post\_article.is\_comment"

active-text="开启评论"

inactive-text="关闭评论"

>

</el-switch>

</el-form-item>

</el-form>

<div class="editor-container">

<markdown-editor

ref="markdownEditor"

v-model="post\_article.article\_md"

:options="{hideModeSwitch:true,previewStyle:'tab'}"

height="580px"

/>

</div>

<el-button

type="primary"

style="margin: 20px; float:right; width: 200px"

@click="postArticle"

v-if="!edit"

>发表</el-button>

<el-button

type="primary"

style="margin: 20px; float:right; width: 200px"

@click="saveEdit"

v-if="edit"

>保存修改</el-button>

</div>

</template>

<script>

import MarkdownEditor from '@/components/MarkdownEditor'

import { getAllType } from '@/api/type'

import { postArticle, getArticleEdit, updateArticle } from '@/api/article'

export default {

name: 'MarkdownDemo',

components: { MarkdownEditor },

data() {

var checkTitle = (rule, value, callback) => {

if (!value) {

return callback(new Error('请输入标题'))

}

if (value.length > 50) {

return callback(new Error('长度不能大于50个字'))

}

callback()

}

var checkSummary = (rule, value, callback) => {

if (value.length > 150) {

return callback(new Error('长度不能大于150个字'))

}

callback()

}

return {

typeList: [],

edit: false,

post\_article:{

article\_id: '',

article\_title: '',

article\_summary: '',

article\_md: '',

article\_type: '',

is\_comment: true,

},

editId: '',

html: '',

rules: {

article\_title: [

{ required: true, validator: checkTitle, trigger: 'blur' },

],

article\_summary: [

{ required: true, validator: checkSummary, trigger: 'blur' },

],

article\_type: [

{ required: true, message: '请选择类型', trigger: 'blur' },

],

},

}

},

mounted() {

this.get\_article\_edit(this.$route.query)

this.get\_type\_list()

},

methods: {

get\_article\_edit(id){

if(undefined == id.id){

// 普通的写文章

this.edit = false

return

}

this.edit = true

var \_this = this

getArticleEdit(id.id).then(res => {

let article = res.extend.article

\_this.post\_article.article\_title = article.articleTitle

\_this.post\_article.article\_summary = article.articleSummary

\_this.post\_article.article\_type = article.articleType

\_this.post\_article.is\_comment = article.articleIsComment

\_this.post\_article.id = article.articleId

\_this.post\_article.article\_md = article.articleMd

})

},

getHtml() {

this.html = this.$refs.markdownEditor.getHtml()

},

get\_type\_list() {

getAllType().then((response) => {

this.typeList = response.extend.pageInfo.records

})

},

postArticle() {

var \_this = this

this.getHtml()

if (this.html.length < 50) {

this.$message.error('字数太少了，多写一点吧')

return

}

this.$refs[this.post\_article].validate((valid) => {

if (valid) {

postArticle(

this.post\_article.article\_title,

this.post\_article.article\_type,

this.post\_article.article\_summary,

this.post\_article.is\_comment,

this.post\_article.article\_md,

this.html

).then((response) => {

var id = response.extend.id

var msg = '发表成功'

this.$message({

dangerouslyUseHTMLString: true,

message: msg,

type: 'success',

duration: 3000,

})

\_this.post\_article.article\_title = ''

\_this.post\_article.article\_type = ''

\_this.post\_article.article\_summary = ''

\_this.post\_article.is\_comment = true

\_this.post\_article.article\_md = ''

\_this.html = ''

})

} else {

return false

}

})

},

saveEdit() {

var \_this = this

this.getHtml()

if (this.html.length < 50) {

this.$message.error('字数太少了，多写一点吧')

return

}

this.$refs[this.post\_article].validate((valid) => {

if (valid) {

updateArticle(

this.post\_article.id,

this.post\_article.article\_title,

this.post\_article.article\_type,

this.post\_article.article\_summary,

this.post\_article.is\_comment,

this.post\_article.article\_md,

this.html

).then((response) => {

var msg = '发表成功'

this.$message({

dangerouslyUseHTMLString: true,

message: msg,

type: 'success',

duration: 3000,

})

\_this.post\_article.article\_title = ''

\_this.post\_article.article\_type = ''

\_this.post\_article.article\_summary = ''

\_this.post\_article.is\_comment = true

\_this.post\_article.article\_md = ''

\_this.html = ''

this.$router.push('/write/write')

this.edit = false

})

} else {

return false

}

})

},

},

}

</script>

<style scoped>

.write {

margin: 20px;

}

</style>