

Sharine 闪灵 - 架构设计文档

项目概述

命名与介绍

项目的名称为 "闪灵 Sharine"，这个名称是富有深意的。在中文中，"闪灵" 象征着短视频创作者的灵感如同繁星一般闪烁，充满了无限的创意和可能。每一颗 "星星" 都代表着一个独特的想法，它们在创作者的心中闪耀，然后通过短视频的形式被展现出来，让更多的人能够看到这些闪烁的 "星星"。

在英文名称 "Sharine" 中，它是 "Sharing" 和 "Shining" 两个词的结合。"Sharing" 代表分享，意味着创作者们通过这个平台分享他们的创意和想法。"Shining" 代表闪耀，象征着每一个创作者都是一颗独特的 "星星"，他们的创意和想法在这个平台上闪耀。这个名称不仅与中文名称的读音相似，而且它的寓意也与中文名称相吻合。

"闪灵 Sharine" 这个名称既表达了平台对于创作者的尊重和期待，也展示出美好愿景，希望通过这个平台，让每一个有创意的想法都能被看到，被分享，让每一颗 "星星" 都能在这里闪耀。

背景与目标

闪灵短视频为七牛云 1024 编程马拉松活动参赛项目，此次活动的挑战在于，所有小组都需要在短短的十四天内，开发出一款基于 Web 端的短视频应用，并成功接入七牛云存储以及相关的视频 API。由于活动的赛题明确，因此必然会出现大量的同质化项目。

由于本项目主要由个人进行开发，因此不可能通过堆砌功能来脱颖而出，所以决定在闪灵短视频项目中作出一些创新，主要从界面设计和功能装配两个方面着手进行，采用原创的界面设计和相关美术元素，抛开繁琐复杂的业务功能，围绕着简约轻快的主题展开进行核心功能开发。

"闪灵短视频" 的目标是创建一款以用户体验为核心，操作简单，体验流畅的 Web 短视频应用，希望能够为用户带来独特且优质的应用体验。

项目分工

早期项目由 天辰破晓 三人团队进行开发，但由于队长和另外一个队员长期不参与项目贡献，在联系七牛云与二人沟通无果后，七牛云将项目由团队项目变更为个人项目。

该项目主要由 米可绫夏(MikkoAyaka) 个人负责界面UI美术设计、前后端开发测试部署、文档编写等内容。

技术栈

后端技术栈

在后端服务器开发中主要运用了下列技术栈：

IOC、AOP、DI、JDBC、ORM、JPQL、Json、Tomcat、Logger、Caching、Exception Handler、Resource Pooling、Maven、Git、Redis、MySQL

运用到的具体框架介绍如下：

Spring Boot 基于 Spring Context 等底层实现提供控制反转容器，基于 JavaBean 技术实现依赖注入，并支持面向切面编程，允许在应用中定义和使用切面以实现横切关注点的处理，如请求拦截器和日志管理。

Spring Data Web 集成 Tomcat 容器提供基于 Java Servlet 规范的 REST API 服务。

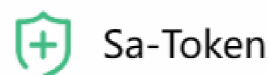
Spring Data Repository 提供了一种通用的、简化的方式来访问和操作不同类型的数据存储，包括关系型数据库、NoSQL 数据库，也允许开发者通过规范接口方法命名定义通用的数据操作方法，而不需要对接口做显式实现。

Spring Data Redis 简化了 Spring 对 Redis 的数据存取操作，也提供了与 Redis 相关的缓存管理器，以便于实现不同的缓存策略，如写穿透、读穿透、旁路缓存。

Spring Data JPA 则用于简化对关系型数据库的访问，基于 Java Persistence API 接口约定实现对象关系映射，用于简化关系型数据库增删改查等操作，开发者也可以使用相关注解定义实体类及相应表单，而省去了大量的 SQL 编写工作。

Qiniu SDK 是提供七牛云服务集成的开发工具包，允许开发者调用相关方法快速使用七牛云提供的 Web API 服务。在本项目中主要使用了 Qiniu SDK 关于视频/图片上传、文本/图像内容审核等部分功能。

SaToken 提供了一种轻量鉴权解决方案，数据基于 Session 进行存储，支持基于 Token 的认证和权限控制。



前端技术栈

在前端开发中，我们主要使用了以下框架：

Vue 是一种流行的前端 JavaScript 框架，它允许开发者使用声明式语法来构建用户界面。Vue 的核心库专注于视图层，使得它可以与其他库或现有项目一起使用。Vue 还提供了一种组件化开发模式，允许开发者将复杂的应用程序分解为小型、独立的、可复用的组件，从而提高开发效率和代码质量。

Vite 是一个现代化的前端构建工具，它使用 ES6 模块化特性，使得开发者可以使用原生 ES6 代码开发应用程序，而无需使用 Babel 等工具进行转译。Vite 提供了快速的冷启动、即时热更新和真正的按需编译等特性，大大提高了开发效率。

axios 是一个基于 Promise 的 HTTP 客户端，用于浏览器和 node.js。它提供了一个简单的 API 来实现 XMLHttpRequests，从浏览器中创建 XMLHttpRequests 到服务器，或者从 node.js 到服务器。

ant-design-vue 是 Ant Design 的 Vue 实现，它提供了一套完整的高质量 Vue 组件，使得开发者可以快速构建用户界面。这些组件遵循 Ant Design 的设计规范，提供了一致且高质量的用户体验。

pinia 是 Vue 应用程序的状态管理库，它提供了一种简洁且直观的 API 来管理应用程序的状态。与 Vuex 相比，pinia 提供了更简洁的 API、更好的 TypeScript 支持和更好的开发体验。

qiniu-js 是七牛云的前端 JavaScript SDK，它提供了七牛云的文件上传功能。本项目主要使用 qiniu-js 将文件上传到七牛云的存储空间，然后获取到文件的 URL 等属性，以通知后端服务器执行相应关系处理。

架构概述

功能列表

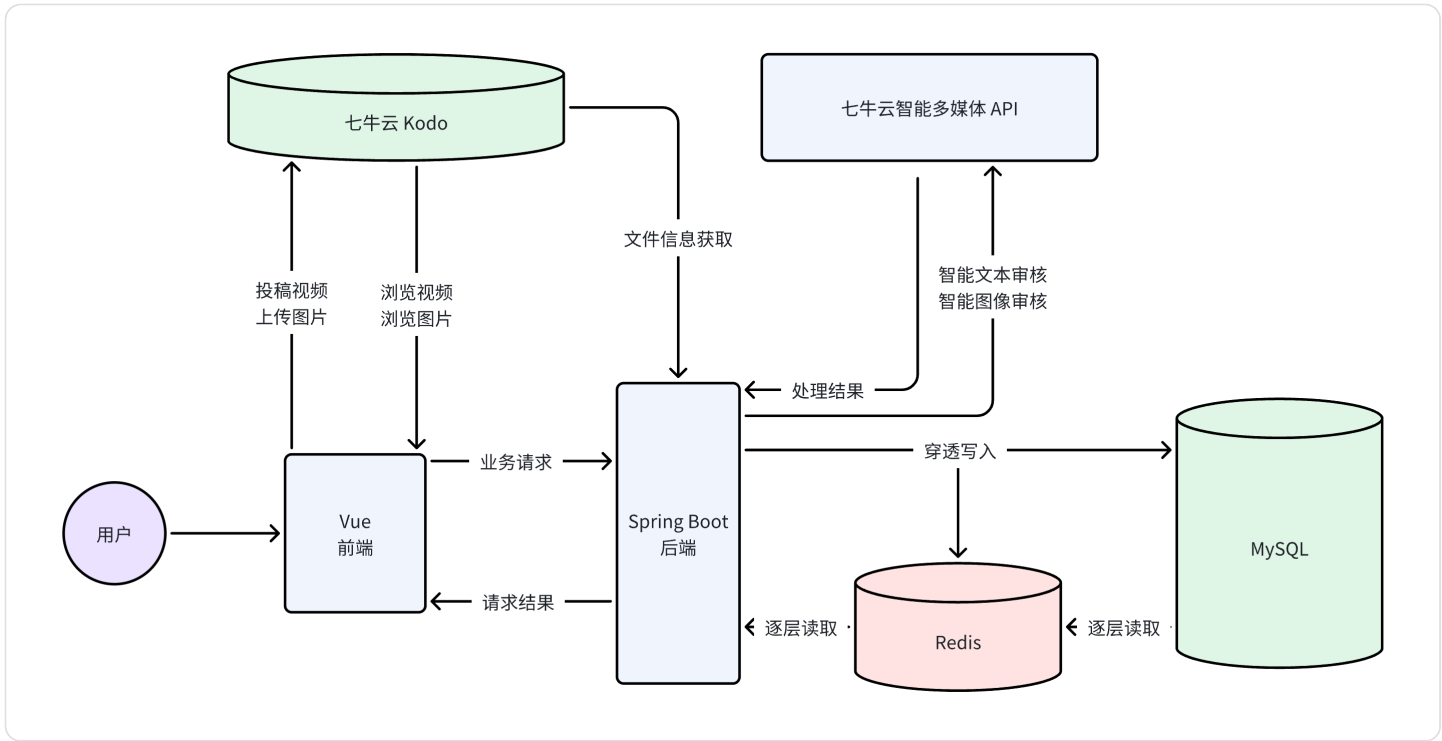
用户：邮箱注册、账号登录、自动登录、点赞、收藏、评论、独立订阅频道、粉丝数统计、播放量统计

视频：播放、暂停、音量调整、上下键或滚轮翻页、自由投稿(七牛云智能审核)

分区：视频分区、分区热度统计、最常浏览分区统计

其它：关键词搜索、分页查询

整体架构图



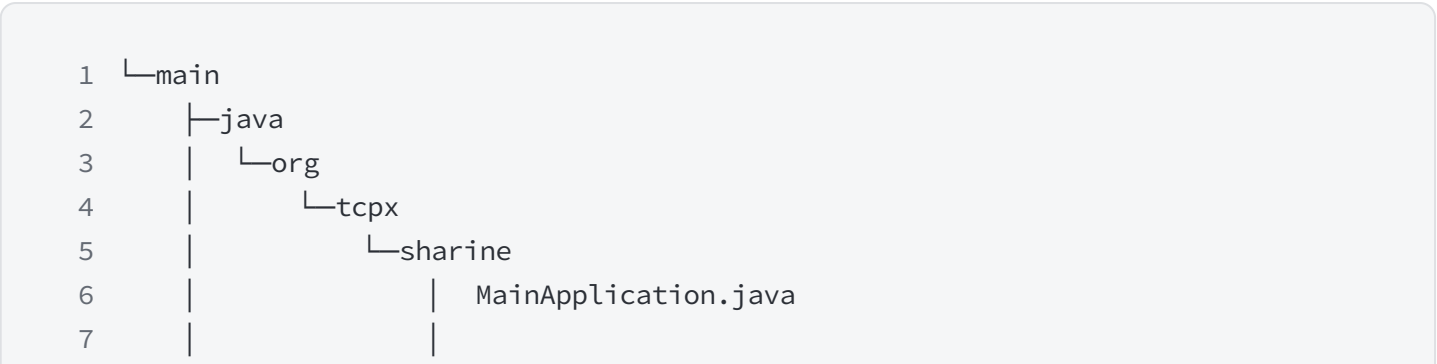
项目整体架构图

数据库设计



后端架构

代码层次图

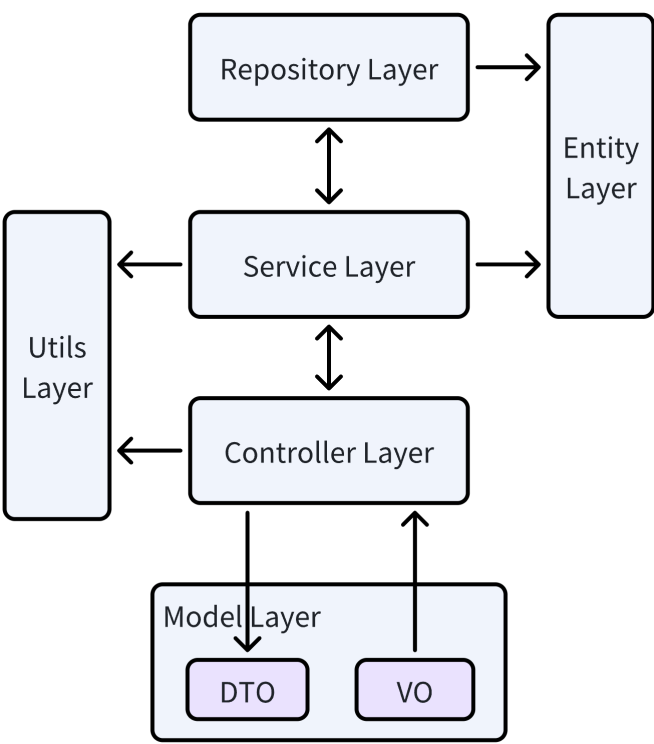


8			└─config
9			CacheConfig.java
10			GlobalCorsConfig.java
11			
12			└─constants
13			DatabaseConst.java
14			RedisPrefixConst.java
15			UserConst.java
16			
17			└─controller
18			BaseController.java
19			CategoryController.java
20			QiniuController.java
21			UserController.java
22			VideoController.java
23			
24			└─dto
25			CommentDTO.java
26			ConditionDTO.java
27			QiniuAuthorizationRequest.java
28			QiniuBasePack.java
29			QiniuCensorPack.java
30			QiniuUptokenRequest.java
31			QiniuUptokenResponse.java
32			UploadVideoDTO.java
33			UserLoginDTO.java
34			UserPass.java
35			
36			└─entity
37			Bookmark.java
38			Category.java
39			Comment.java
40			Favorite.java
41			User.java
42			UserRelation.java
43			Video.java
44			VideoCategoryRelation.java
45			
46			└─enums
47			QiniuFileType.java
48			StatusCodeEnum.java
49			UserRelationEnum.java
50			VideoTypeEnum.java
51			
52			└─exception
53			ErrorException.java
54			WarnException.java

```
55 |
56 | └─repository
57 |     BookmarkRepository.java
58 |     CategoryRepository.java
59 |     CommentRepository.java
60 |     FavoriteRepository.java
61 |     UserRelationRepository.java
62 |     UserRepository.java
63 |     VideoCategoryRepository.java
64 |     VideoRepository.java
65 |
66 | └─service
67 |     BookmarkService.java
68 |     CategoryService.java
69 |     CommentService.java
70 |     EmailService.java
71 |     FavoriteService.java
72 |     RedisService.java
73 |     SubscribeChannelService.java
74 |     UserRelationService.java
75 |     UserService.java
76 |     VideoCategoryService.java
77 |     VideoService.java
78 |     ViewCountService.java
79 |
80 | └─utils
81 |     BeanCopyUtils.java
82 |     EncryptionUtil.java
83 |     HTMLUtils.java
84 |     IOC.java
85 |     IpUtils.java
86 |     JwtUtil.java
87 |     QiniuUtils.java
88 |     RegexpUtils.java
89 |     SensitiveUtils.java
90 |     StringUtils.java
91 |
92 | └─vo
93 |     CategoryVO.java
94 |     CommentVO.java
95 |     PageVO.java
96 |     UserDetailVO.java
97 |     UserSimpleVO.java
98 |     VideoVO.java
99 |
100 | └─resources
101 |     application.yml
```

```
102 | sensitive-words.txt
103 |
104 |└─config
105 |     application-dev.yml
106 |     application-prod.yml
```

分层架构图



后端分层架构图(箭头描述依赖方向)

接口说明

用户收藏("/bookmark")

接口地址	接口类型	接口描述	接口参数
/[{videoid}]	POST	用户对视频进行收藏	Long videoid
/[{videoid}]	GET	查询用户是否已经收藏了视频	Long videoid
/[{videoid}]	DELETE	用户取消收藏视频	Long videoid

视频分区("/categories")

--	--	--	--

接口地址	接口类型	接口描述	接口参数
/all	GET	获取所有分区信息(不包含分区视频，仅包含分区名，图标URL等)	

用户点赞("/favorite")

接口地址	接口类型	接口描述	接口参数
/videoId	POST	用户对视频进行点赞	Long videoId
/videoId	GET	查询用户是否已经点赞了视频	Long videoId
/videoId	DELETE	用户取消点赞视频	Long videoId

七牛云服务("/qiniu")

接口地址	接口类型	接口描述	参数类型
/generateToken	GET	获取七牛指定 bucket 上传 Token	String fileName
/authorization	POST	获取七牛 Authorization 签名	String url , String body
/downloadUrl/{fileKey}	GET	获取文件下载 URL	String fileKey

用户("/users")

接口地址	接口类型	接口描述	参数类型
/login	POST	用户进行登录	UserPass userPass
/register	POST	用户进行注册	UserPass userPass
/changePassword	POST	用户修改密码	UserPass userPass
/sendCode	POST	用户请求邮箱验证码	UserPass userPass
/userId	GET	查询指定用户的详细信息	Long userId

视频("/videos")

接口地址	接口类型	接口描述	参数类型

/verify	POST	用户投稿完成以后调用该接口通知后端进行数据验证和关系绑定	UploadVideoDTO dto
/[{videoid}]	GET	获取视频信息	Long videoid
/[{videoid}]/comments	GET	获取视频评论信息	Long videoid
/[{videoid}]/comments	POST	发送视频评论信息	CommentDTO dto
/[{videoid}]/addViewCount	POST	记录用户播放	Long videoid
	GET	根据条件分页查询视频信息	ConditionDTO dto
/favorite	GET	获取用户个人点赞的视频列表	
/bookmark	GET	获取用户个人收藏的视频列表	
/subscribe	GET	获取用户的订阅频道视频列表	
/upload/{userId}	GET	获取用户投稿视频列表	Long userId

自定义状态码规范

自定义状态码	状态描述
0	操作成功
1	没有操作权限
2	系统异常
3	操作失败
4	参数格式不正确
5	数据已存在
6	数据不存在
7	登陆错误
8	密码错误
9	下标越界
10	解压错误

11	创建文件夹错误
12	写入文件错误
13	评测时异常
14	不支持该语言
15	编辑错误
16	正在运行
17	不在运行时间
18	未登录
19	验证失败
20	计划开发中，暂未实现
21	信息审核不通过

前端架构

代码层次图

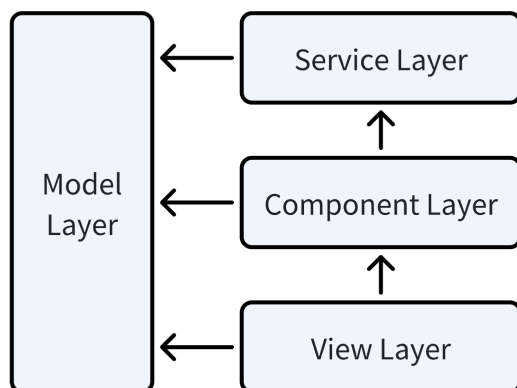
```
1  .
2  |   App.vue
3  |   main.js
4  |
5  |—assets
6  |   |   logo.png
7  |   |   no-data.png
8  |   |   not-login.png
9  |   |   test-video-cover.png
10 |   |   ui-icon.zip
11 |   |   vue.svg
12 |   |   wait-img.png
13 |   |
14 |   |—category-icon
15 |   |   alltype-category.png
16 |   |   funny-category.png
17 |   |   game-category.png
18 |   |   live-category.png
19 |   |   music-category.png
20 |   |   tech-category.png
```

```
21 | |
22 | |─font
23 | |   SourceHanSansCN-Bold.otf
24 | |   SourceHanSansCN-ExtraLight.otf
25 | |   SourceHanSansCN-Heavy.otf
26 | |   SourceHanSansCN-Light.otf
27 | |   SourceHanSansCN-Medium.otf
28 | |   SourceHanSansCN-Normal.otf
29 | |   SourceHanSansCN-Regular.otf
30 | |
31 | |─ui-icon
32 | |   bookmark-icon.png
33 | |   boy-icon.png
34 | |   date-icon.png
35 | |   dropdown-icon.png
36 | |   favorite-navbar-icon.png
37 | |   filled-bookmark-icon.png
38 | |   filled-like-icon.png
39 | |   girl-icon.png
40 | |   home-navbar-icon.png
41 | |   left-icon.png
42 | |   like-icon.png
43 | |   lock-icon.png
44 | |   person-navbar-icon.png
45 | |   right-icon.png
46 | |   search-icon.png
47 | |   share-icon.png
48 | |   upload-navbar-icon.png
49 | |   view-count.png
50 | |
51 | |─commons
52 | |   global-var.less
53 | |   global.css
54 | |
55 | |─components
56 | |   BasicButton.vue
57 | |   BasicCard.vue
58 | |   CategoryBar.vue
59 | |   CategoryDropdownButton.vue
60 | |   CategoryItem.vue
61 | |   CategorysContainer.vue
62 | |   CommentItem.vue
63 | |   DetailVideoCard.vue
64 | |   Divider.vue
65 | |   LoginButton.vue
66 | |   Logo.vue
67 | |   NavBar.vue
```

```
68 |     NavBarItem.vue
69 |     PersonCard.vue
70 |     PersonVideoBar.vue
71 |     ProgressBar.vue
72 |     SearchBar.vue
73 |     SimpleVideoCard.vue
74 |     ToggleButton.vue
75 |     TopChart.vue
76 |
77 | └─layout
78 |     LayoutOverview.vue
79 |
80 | └─models
81 |     Category.ts
82 |     CategoryFrequency.ts
83 |     ResponsePack.ts
84 |     User.ts
85 |     Video.ts
86 |     VideoComment.ts
87 |     VideoType.ts
88 |
89 | └─router
90 |     index.js
91 |
92 | └─services
93 |     AccountService.ts
94 |     CategoryService.ts
95 |     QiniuService.ts
96 |     UserService.ts
97 |     VideoService.ts
98 |
99 | └─store
100 |     system.ts
101 |
102 | └─utilities
103 |     HttpUtility.ts
104 |     ResourceUtility.ts
105 |
106 | └─views
107 |     └─favorite
108 |         Favorite.vue
109 |
110 |     └─home
111 |         └─Home.vue
112 |
113 |         └─cpns
114 |             HomeCategory.vue
```

```
115      |           HomeContent.vue
116      |
117      |├person
118      |   Person.vue
119      |
120      |├upload
121      |   Upload.vue
122      |
123      |└video
124      |   Video.vue
```

分层架构图



前端分层架构图(箭头描述依赖方向)

安全性

接口请求限制

后端服务器根据用户访问权限可将接口分为以下两类：公开接口、保护接口。以下是接口对应的安全保护措施：

公开接口：仅采用简单的 QPM Limit 策略限制每 IP 每分钟请求阈值。

保护接口：采用 QPM Limit 策略 + SaToken 登录鉴权。

隐私数据保护

用户密码将使用 BCrypt 算法进行哈希单向加密后保存至数据库，以防止数据库失窃后造成严重隐私数据泄露。

异常拦截

定义通用的 `ExceptionHandler` 异常处理方法以接管控制器抛出的异常，避免将堆栈信息等服务器细节泄露到前端。

性能优化

由于本项目不涉及较多的实时运算，因此在性能优化方面主要围绕数据库进行优化设计，核心优化策略即采用缓存数据库分担 MySQL 主要数据库压力。

在项目中引入 Redis 以提供以下功能：

热点数据缓存：配合 Spring Data Redis 相关注解，在 JPA 仓库中声明缓存逻辑，将经常访问的数据缓存在 Redis 中，这样可以减少对 MySQL 数据库的访问，提高数据访问速度。

订阅消息频道：由于 MySQL 并不直接支持对数组型数据的存取，因此使用 Redis 的队列存储数据结构为每一位用户维护长度有限的订阅消息频道。

持久化配置：虽然 Redis 主要是内存数据库，但是也支持数据持久化，可以将用户消息频道等数据保存到磁盘中。这样在 Redis 重启后，可以从磁盘中恢复数据。