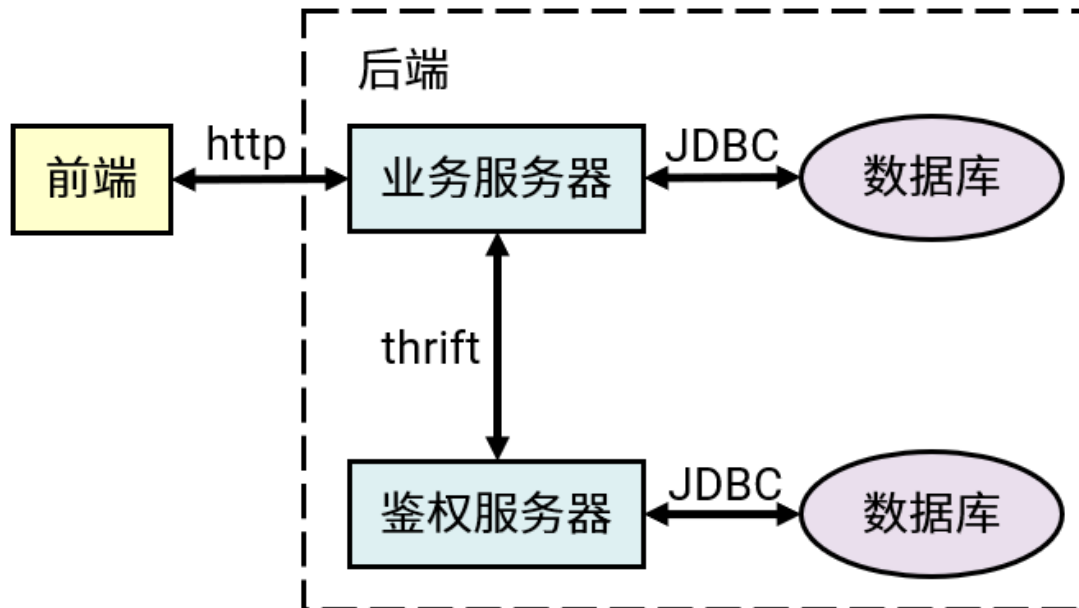


# 技术设计文档

## 1. 技术路线

- 系统架构图

◦



- 前端: Vue + ElementUI
- 后端: Java SpringBoot
- 数据库: MySQL
- 视频存储: 文件系统 + SHA-256哈希
- 分布式: 业务服务器 + 鉴权服务器
- 通信协议
  - 前端与后端通信使用http协议
  - 后端之间通信使用thrift
- 日志: AOP + LogBack
- Token: JWT + SHA-256
- 密码: 哈希 + 盐值

## 2. API文档

### 2.1 用户API

#### 2.1.1 用户注册 POST -> /api/v1/users

- 资源标识: `/api/v1/users`
- 操作: POST
- 权限: 无须权限

- 请求Payload

```
{
  "username": "exampleUser",
  "password": "password",
  "email": "example@email.com",
  "telephone": "10123456789"
}
```

- 响应Payload

```
{
  "code": 200,
  "message": "Success.",
  "data": {
    "id": "1",
    "username": "exampleUser",
    "password": "password",
    "email": "example@email.com",
    "telephone": "10123456789"
  }
}
```

### 2.1.2 用户登入 POST -> /api/v1/sessions

- 资源表示: /api/v1/sessions
- 操作: POST
- 权限: 无需权限
- 请求Payload

```
{
  "username": "exampleUser",
  "password": "password"
}
```

- 响应Payload

```
{
  "code": 200,
  "message": "Success.",
  "data": {
    "user": {
      "id": "1",
      "username": "exampleUser",
      "password": "password",
      "email": "example@email.com",
      "telephone": "10123456789",
    },
    "token": "abcdefg1234567890"
  }
}
```

```
}  
}
```

### 2.1.3 查询所有用户 GET -> /api/v1/users

- 资源表示: /api/v1/users
- 操作: GET
- 权限: Admin
- 请求Payload
  - 无
- 响应Payload

```
{  
  "code": 200,  
  "message": "Success.",  
  "data": [  
    {  
      "id": "1",  
      "username": "exampleUser",  
      "password": "password",  
      "email": "example@email.com",  
      "telephone": "10123456789",  
    }, {  
      ...  
    }  
  ]  
}
```

### 2.1.4 查询单个用户 GET -> /api/v1/users/{uuid}

- 资源表示: /api/v1/users/{id}
- 操作: GET
- 权限: Admin
- 参数

```
"id": "1"
```

- 请求Payload
  - 无
- 响应Payload

```
{
  "code": 200,
  "message": "Success.",
  "data": {
    "id": "1",
    "username": "exampleUser",
    "password": "password",
    "email": "example@email.com",
    "telephone": "10123456789"
  }
}
```

## 2.2 视频API

### 2.2.1 添加视频 POST -> /api/v1/videos

- 资源表示: /api/v1/videos
- 操作: POST
- 权限: User
- 请求Payload

```
{
  "file": "video.mp4",
  "title": "examplevideo",
  "token": "***",
}
```

- 响应Payload

```
{
  "code": 200,
  "message": "Success.",
  "data": {
    "id": "1",
    "title": "examplevideo",
    "likes": 0,
    "filepath": "video.mp4",
    "user_id": "1"
  }
}
```

### 2.2.2 查询所有视频 GET -> /api/v1/videos

- 资源表示: /api/v1/videos
- 操作: GET
- 权限: User
- 请求Payload

```
{
  "page": 1
}
```

- 响应Payload

```
{
  "code": 200,
  "message": "Success.",
  "data": [
    {
      "id": "1",
      "title": "examplevideo",
      "likes": 0,
      "filepath": "video.mp4",
      "user_id": "1"
    }, {
      ...
    }
  ]
}
```

### 2.2.3 查询单个视频 GET -> /api/v1/videos/{uuid}

- 资源表示: /api/v1/videos/{id}
- 操作: GET
- 权限: User
- 参数

```
"id": "1"
```

- 请求Payload
  - 无
- 响应Payload

```
{
  "code": 200,
  "message": "Success.",
  "data": {
    "id": "1",
    "title": "examplevideo",
    "likes": 0,
    "filepath": "video.mp4",
    "user_id": "1"
  }
}
```

## 2.2.4 删除单个视频 DELETE -> /api/v1/videos/{uuid}

- 资源表示: `/api/v1/videos/{id}`
- 操作: DELETE
- 权限: User
- 参数

```
"id": "1"
```

- 请求Payload
  - 无
- 响应Payload

```
{  
  "code": 200,  
  "message": "Success."  
}
```

## 2.2.5 获取我的视频 GET -> /api/v1/playable\_videos/{uuid}

- 资源表示: `/api/v1/playable_videos/{id}`
- 操作: GET
- 权限: User
- 参数

```
"id": "1"
```

- 请求Payload
  - 无
- 响应Payload

```
{  
  "code": 200,  
  "message": "Success."  
}
```

## 2.2.6 给视频点个赞 POST -> /api/v1/videos/{uuid}/likes

- 资源表示: `/api/v1/videos/{id}/likes`
- 操作: POST
- 权限: User
- 参数

```
"id": "1"
```

- 请求Payload
  - 无
- 响应Payload

```
{  
  "code": 200,  
  "message": "Success."  
}
```

### 2.2.7 查询视频点赞数 GET -> /api/v1/videos/{uuid}/likes

- 资源表示: `/api/v1/videos/{id}/likes`
- 操作: GET
- 权限: User
- 参数

```
"id": "1"
```

- 请求Payload
  - 无
- 响应Payload

```
{  
  "code": 200,  
  "message": "Success.",  
  "data": 1  
}
```

### 2.2.8 获取推荐视频 GET -> /api/v1/videos/recommendations

- 资源表示: `/api/v1/videos/recommendations`
- 操作: GET
- 权限: User
- 请求Payload
  - 无
- 响应Payload

```
{  
  "code": 200,  
  "message": "Success.",  
  "data": [  
    {
```

```
        "id": "1",
        "title": "examplevideo",
        "likes": 0,
        "filepath": "video.mp4",
        "user_id": "1"
      }, {
        ...
      }
    ]
  }
}
```

## 3. 数据库设计

---

### 3.1 数据库表

- autho: 用户鉴权表
  - uuid: 用户唯一标识 (主键)
  - password: 用户密码
- user: 用户信息表
  - uuid: 用户唯一标识 (主键)
  - username: 用户名
  - password: 用户密码
  - email: 用户邮箱
  - telephone: 用户电话
- video: 视频信息表
  - uuid: 视频唯一标识 (主键)
  - title: 视频标题
  - filepath: 视频文件路径
  - likes: 视频点赞数
  - userUuid: 用户唯一标识 (外键)
  - hash: 视频hash值

### 3.2 数据库表关系

- user -> video: 一对多
- user -> autho: 一对一

### 3.3 数据库表结构

- 见 `docs/tiktok.sql`

## 4. 后端类图

---



