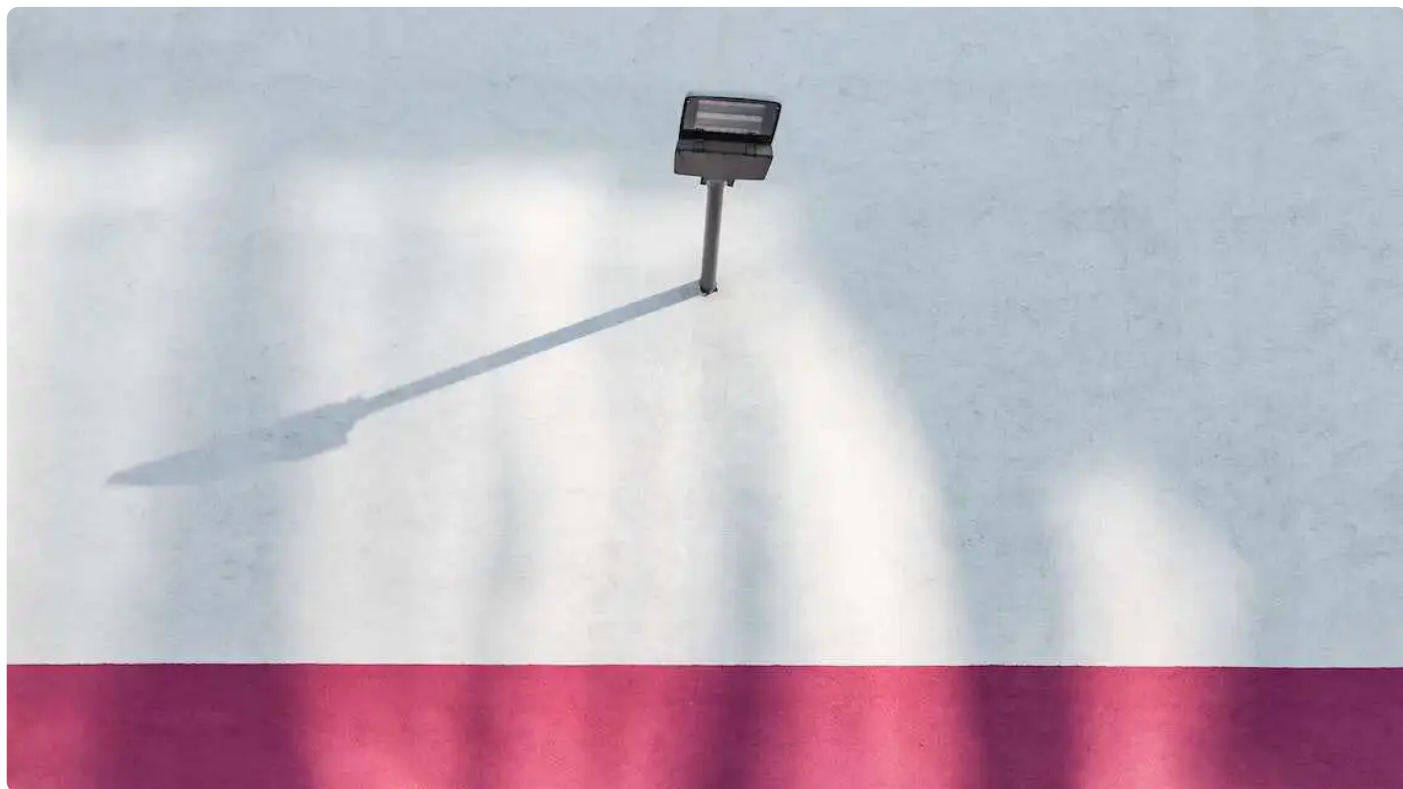


## 33 | 直播流程：如何设计搭建直播的核心流程？

2023-07-07 Barry 来自北京

《Python实战·从0到1搭建直播视频平台》



你好，我是 Barry。

前面的课程我们完成了后端接口的开发和项目的部署，也学习了网络相关知识，这些都为直播模块的建设打下了良好基础。这节课我们就来学习直播功能的核心流程搭建。

为了顺利推进后续开发，让你掌握独立开发直播系统的能力，我们需要先理清整个直播系统的开发流程和功能模块，还有在每一阶段我们需要实现哪些功能。最后我们再根据前面的梳理，确定直播后台模块的搭建方案。

shikey.com转载分享

**直播系统开发流程**

## 直播系统 开发流程

1. 音视频  
数据采集

2. 服务器推流

3. 流媒体  
服务器

4. 拉流解码

5. 音视频解码

6. 音视频播放

极客时间

第一步是**音视频采集**。音视频采集是指将音视频数据从各种外界设备来源采集到系统中进行处理和传输的过程。常见的音频采集方式包括摄像头采集、麦克风采集、屏幕录制、视频和音频编辑软件，这些方式都可以帮我们收集流数据。

第二步是**推流**。推流是指将采集到的音视频数据通过服务器推送到直播平台或 CDN 网络中，供用户观看或存储。我们这就来看看推流的过程。

首先，我们需要配置推流参数。这一环节主要发生在推流前，用于需要配置相应的参数，例如视频分辨率和音频质量等。

其次，我们需要连接服务器，目的是连接到直播平台或 CDN 网络的服务器，从而实现音视频数据的准确推送。至于连接服务器的方式，我们可以在 HTTP 协议、RTMP 协议、HLS 协议等多种方式里按需选择。

接着，我们还需要进行推流控制，目的是控制推流的质量和稳定性。这个过程中我们可以通过调整推流码率、帧率、延迟等参数，并观测推流状态和网络状况，确保直播是稳定流畅的。

shikey.com 转载分享

另外，在推流过程中，为了保护直播内容的安全性和隐私性，我们还需要对推流数据进行加密和保护。常见的加密方式包括 HTTPS 协议、数字签名等。

shikey.com 转载分享

第三步就是**流媒体服务器**处理直播流的环节。你可能对流媒体服务器有些陌生，它是用来处理和分发流媒体数据的软件或硬件设备。

流媒体服务器能接收来自直播源的流媒体数据，并将其分发给用户，同时它还能够处理流媒体数据的传输、解码、播放等操作，保障直播顺利进行。流媒体可以把连续的音频和视频信息压

缩后放到网络服务上，供用户边下载边观看。

紧接着就是我们的第四步——**拉流**，其实就是用户从直播平台或 CDN 网络中获取直播流的过程。当然在拉流的过程中，用户需要通过 HTTP 协议、RTMP 协议、HLS 协议等多种方式连接服务器。

连接成功后，用户就可以从服务器获取直播流的地址、码率、分辨率等信息。这时候，用户使用相应的解码器和播放器对直播流进行解码和播放，播放的平台既可以是移动端。也可以是 PC 端。在拉流的过程中，我们同样要考虑网络稳定性、设备兼容性、带宽和延迟以及安全性等方面的问题。

拉流之后的第五步就是**音视频解码**，这是为了解码获取的直播流数据，确保播放器中能够呈现出音视频内容。完成初始化解码器、解压还原数据、处理和过滤数据等一系列操作之后，就能将解码后的音视频数据进行同步和输出，在播放器中呈现出流畅、高质量的音视频内容。

最后一步就是**音视频播放**。在音视频解码后，为了保证直播顺利进行，提升用户观看体验，播放器需要对解码后的音视频数据进行解码和渲染，并完成同步、流畅播放、互动、特效处理和错误处理等相关操作。音视频播放我们采用了 VideoJS 来实现，这一部分已经在第 15 讲和 16 讲中详细讲过，你可以回顾相关内容。

相信经过梳理，整个直播系统的开发流程你已经比较清晰了。我们这就来看看直播系统应该具备哪些功能模块，每个模块又分别负责解决哪些问题。

## 功能模块梳理

直播系统可以分为流媒体模块、服务模块、管理模块这三大模块。

流媒体模块是直播系统中的技术核心，用于实现从音视频采集到直播渲染完整过程。服务模块主要用于满足和提升用户体验，完善平台服务体验。管理模块主要是管理系统和数据，也就是在前后端实现上的管理。

## 流媒体模块

在我们直播视频平台开发中，流媒体模块是最核心的功能模块，它是用来处理和分发流媒体数据的关键模块，前面提到的开发流程，就属于流媒体模块完成的功能。

流媒体模块能够为直播系统提供稳定的流媒体传输服务、保障用户的观看体验、处理直播延迟问题、提供安全和保护措施等。这些功能保障了直播系统高效、稳定、安全地运行，满足在 IOS、Android、PC 多场景直播的需求。

## 服务模块

服务模块是指为直播过程提供各种服务的子系统，比较重要的系统包括后面这几类。

**互动服务：**可以包括弹幕、评论、点赞、私信等功能，也可以添加抽奖、投票等互动活动。互动服务是直播系统里的顶梁柱，它的作用是增强直播间的活跃度和观看体验，为用户和主播提供互动空间。

**礼物服务：**为用户提供给主播刷礼物的功能（具体包括礼物选择、发送、特效处理和礼物接收功能），主播则可以通过平台内的一些兑换机制获得收益。这是我们日常观看直播最常见的功能了，礼物服务让直播更有特色，同时也是直播系统中的一种盈利模式。

**商城服务：**包括商品展示、购物车、支付以及订单管理和物流配送等功能。这是直播系统中的一种扩展功能，它为用户提供了在直播间购买商品或服务的渠道。

**推送服务：**通过消息推送的方式向平台内用户发送直播提醒、弹幕、评论等信息。推送服务主要包括消息的发送和接收、推送策略的配置等功能。这类服务也很重要，能为主播增强曝光，吸引更多的流量。

当然，除去这些常规的服务之外，我们还需要**安全保护服务**和**数据统计服务**。安全保护服务主要是在直播系统通过多种安全措施保护数据的安全和隐私，例如数据加密、权限管理、防火墙等。数据统计服务就是对直播系统通过数据统计和分析功能，可以让主播了解直播过程的观众数量、行为和喜好，不断优化自己的直播内容，提升直播效果和用户体验。

## 管理模块

管理模块比较好理解，就是对我们实现直播系统的**前端、后端、数据库、控制模块**的管理。

例如前端用 VideoJS 作为播放器来呈现直播内容。通过管理模块，前端能更方便地管理维护直播内容，例如设置直播的标题、分类、封面等。

后端可以通过管理模块设置用户的权限等级和访问权限，以控制不同用户的权限范围。这样能够确保只有授权的用户才能访问敏感功能和信息，增强平台的稳定性和安全性。

在音视频数据采集方面，我们使用 OBS 来采集，推流则采用 RTMP 来实现。另外，为了支持直播功能，数据库也要进一步设计，这些都是属于我们管理模块范畴。当然，每一模块的实现，我们会在开发阶段都逐步完成。

掌握了开发流程，也梳理好了功能模块，我们这就来设计直播系统后台的搭建方案。

## 直播后台搭建方案

后台搭建方案是我们后续实战环节的行动纲领，梳理清楚实现直播系统的每个环节，后面课程我们具体开发的时候才会更加高效、规范。

## OBS 来实现音视频采集

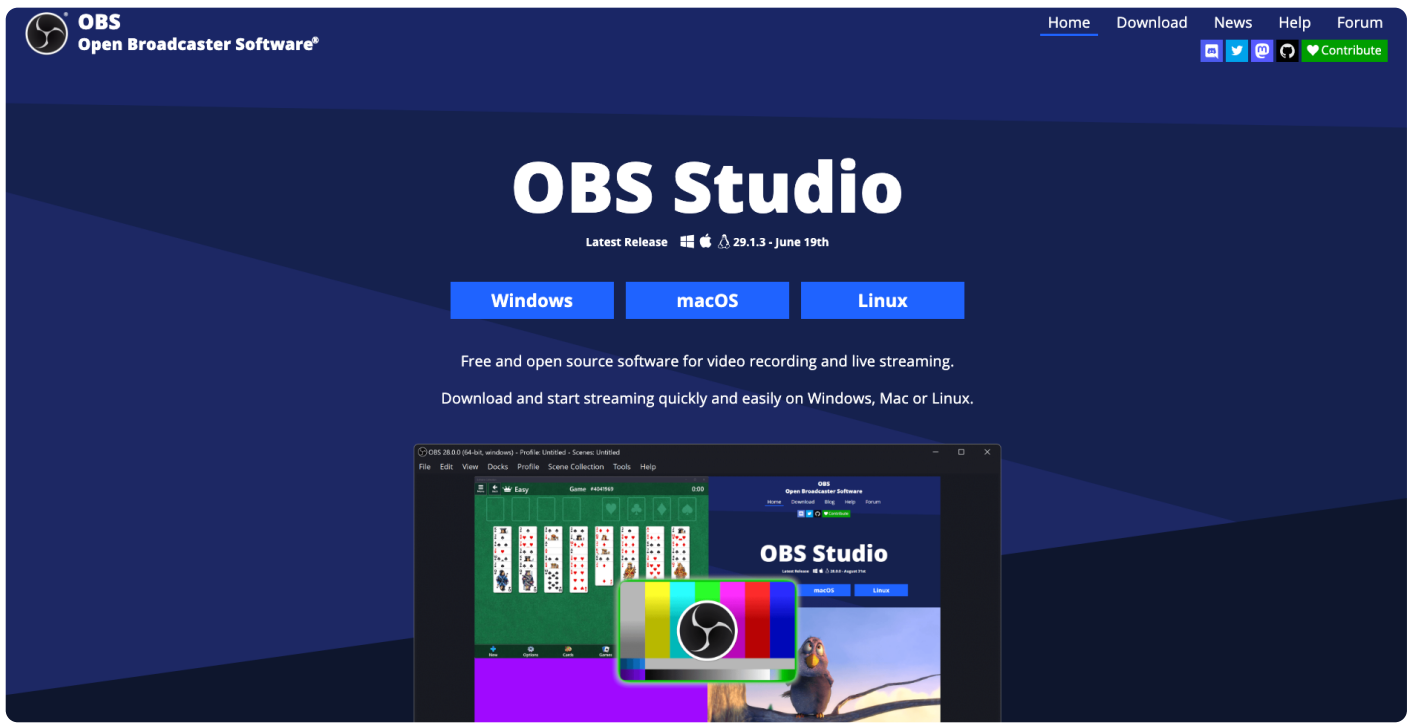
首先，我们选用 **OBS 来实现音视频采集**，你可以直接在 [🔗OBS 官网](#) 下载。

OBS 的全称是 Open Broadcaster Software，它是一款免费的开源跨平台流媒体和录影软件，支持多场景的录制和直播，也支持实时录制和播放音视频。

OBS 使用实时消息协议 RTMP，同时也可以与其他支持 RTMP 的软件无缝对接。这和我们之后推拉流的操作也可以巧妙结合。在系统中我们使用 OBS 去推流，前端使用 vue-video-player 播放器做流的承接，从而实现直播播放功能。

OBS 使用起来也比较简单，具体的版本你结合自己的系统来下载就可以了。





确定了音视频数据来源之后，对于**直播服务器**我们会使用 Nginx 和 nginx-rtmp-module 模块配置 RTMP 直播服务器。这一部分你先知道我们采取的实现方式，这是我们下节课的重点。

## 直播相关数据表的设计

明确了服务器的解决方案之后，接下来我们关注一下后端直播相关数据表的设计。

对于直播系统，我们需要创建两个数据库表——直播频道表 channel 和直播流表 stream。channel 表具体的类如何实现，你可以参考后面的代码。

shikey.com转载分享

复制代码

```
1
2 import uuid
3 from api import db
4 from .base import BaseModels
5
6 class Channel(BaseModels, db.Model):
7     __tablename__ = "Channel"
8     id = db.Column(db.Integer, primary_key=True)
9     user_id = db.Column(db.Integer)
10    stream_key = db.Column(db.String(255), unique=True)
11    channel_name = db.Column(db.String(255))
12    channel_loc = db.Column(db.String(255), unique=True)
```

shikey.com转载分享

```

13     topic = db.Column(db.Integer) # 类别
14     views = db.Column(db.Integer) # 总播放人数
15     current_viewers = db.Column(db.Integer) # 当前播放人数
16     record = db.Column(db.Boolean) # 是否可以录制回放
17     chat_enabled = db.Column(db.Boolean) # 是否可以聊天
18     image_location = db.Column(db.String(255))
19     description = db.Column(db.String(2048))
20     default_stream_name = db.Column(db.String(255)) # 录制回放的命名
21     protected = db.Column(db.Boolean)
22
23     def __init__(self, user_id, stream_key, channel_name, topic, record, chat_ena
24         self.user_id = user_id
25         self.stream_key = stream_key
26         self.channel_name = channel_name
27         self.channel_loc = str(uuid.uuid4())
28         self.topic = topic
29         self.views = 0
30         self.currentViewers = 0
31         self.record = record
32         self.chat_enabled = chat_enabled
33         self.description = description
34         self.defaultStreamName = ""
35         self.protected = False
36
37     def __repr__(self):
38         return '<id %r>' % self.user_id
39
40     def to_dict(self):
41         return {
42             'user_id': self.user_id,
43             'channel_name': self.channel_name,
44             'channel_loc': self.channel_loc,
45             'topic': self.topic,
46             'views': self.views,
47             'stream_key': self.stream_key,
48             'currentViewers': self.currentViewers,
49             'record': self.record,
50             'chat_enabled': self.chat_enabled,
51             'description': self.description,
52         }

```


前面这张表的作用就是管理直播频道，更准确地说就是创建频道房间。

我们对照代码来梳理一下过程。首先，用户直播前会选择自己感兴趣的直播频道，其中包含频道的 ID、用户 ID 以及频道名称等信息。我们只需要对应去设置表结构，用户创建完频道以后

就会生成唯一的 UUID。另外直播页面里会展示频道信息、用户信息、频道类型、用户人数等信息，表结构里也要设计好相关字段。

另外还有一些需要我们灵活定义的属性，比如用户进入直播间是否可以聊天，以及是否要生成回放。提前做好这些准备，后续我们功能开发时就可以灵活选择配置。

频道房间创建好以后，还需要有对应房间的直播流，这样才能让播放器显示直播内容。这里就需要用到 stream 表，这个表用来存储正在直播的流数据，如果直播结束了，这个表的数据也会随之删除。stream 表的内容如下所示，你可以看一下。

 复制代码

```
1 from uuid import uuid4
2
3 from api import db
4 from .channel import Channel
5
6 class Stream(db.Model):
7     __tablename__ = "Stream"
8     id = db.Column(db.Integer, primary_key=True)
9     uuid = db.Column(db.String(255))
10    channel_id = db.Column(db.Integer)
11    stream_key = db.Column(db.String(255))
12    stream_name = db.Column(db.String(255))
13    topic = db.Column(db.Integer)
14    current_viewers = db.Column(db.Integer)
15    total_viewers = db.Column(db.Integer)
16
17    def __init__(self, stream_key, stream_name, channel_id, topic):
18        self.uuid = str(uuid4())
19        self.stream_key = stream_key
20        self.stream_name = stream_name
21        self.channel_id = channel_id
22        self.current_viewers = 0
23        self.total_viewers = 0
24        self.topic = topic
25
26    def __repr__(self):
27        return '<id %r>' % self.id
28
29    def add_viewer(self):
30        self.current_viewers = self.current_viewers + 1
31        db.session.commit()
32
```



```

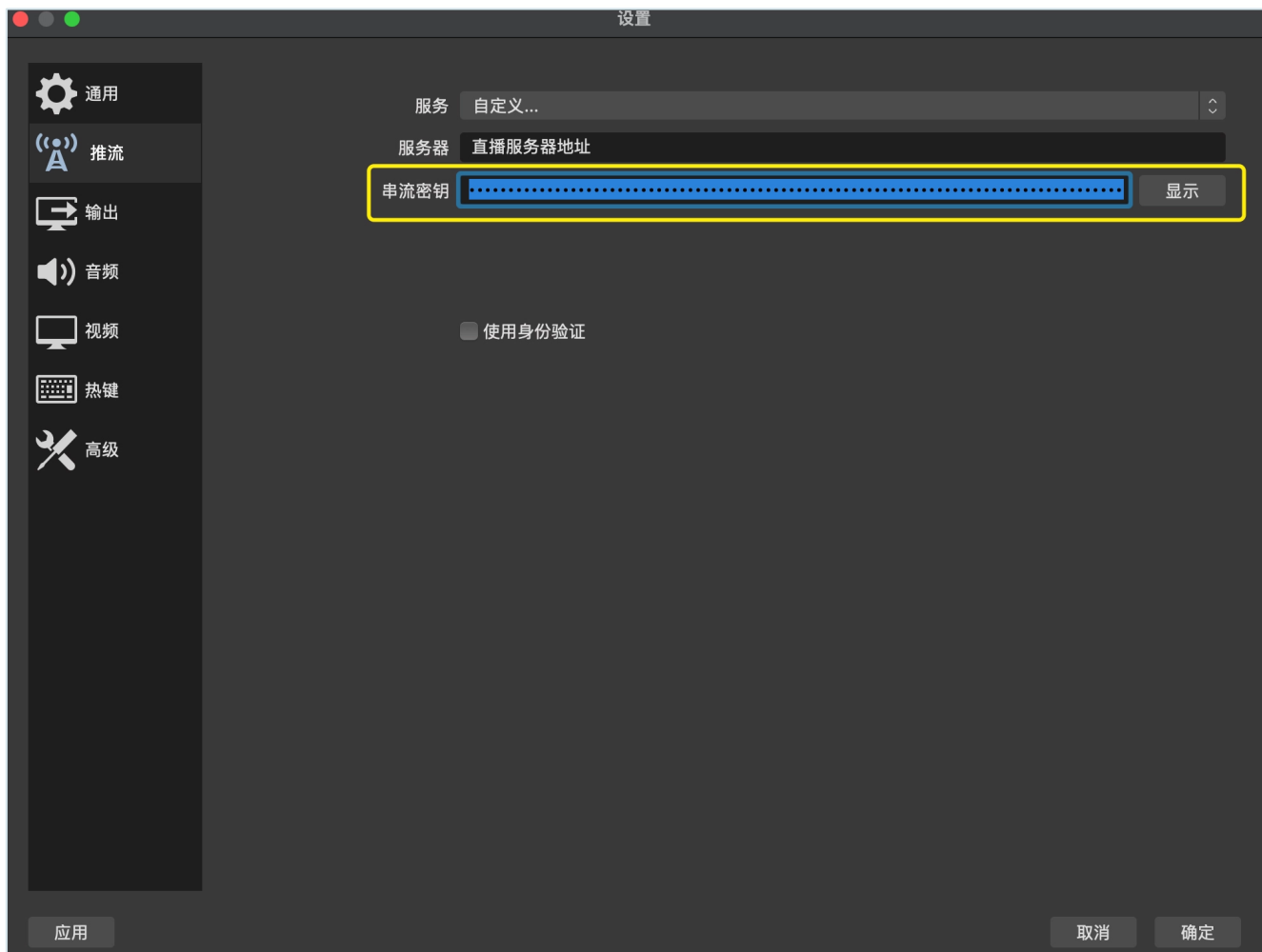
33     def remove_viewer(self):
34         self.current_viewers = self.current_viewers - 1
35         db.session.commit()
36
37     def serialize(self):
38         # 如果设计到推流分为不同的方案 可以设置参数处理
39         channel = Channel.query.get(self.channel_id)
40         channel_loc = channel.channel_loc
41         user_id = channel.user_id
42         stream_url = '/live/' + channel_loc + '/index.m3u8'
43         return {
44             'id': self.id,
45             'uuid': self.uuid,
46             'channel_id': self.channel_id,
47             'channel_loc': channel_loc,
48             'user_id': user_id,
49             'stream_page': '/view/' + channel_loc + '/',
50             'stream_url': stream_url,
51             'stream_name': self.stream_name,
52             'thumbnail': '/stream-thumb/' + channel_loc + '.png',
53             'gif_location': '/stream-thumb/' + channel_loc + '.gif',
54             'topic': self.topic,
55             'current_viewers': self.current_viewers,
56             'total_viewers': self.total_viewers,
57         }
58

```

可以看到，这个表包含的字段主要有 uuid、频道房间 IDchannel\_id、直播流的 Key，stream\_key。这里我单独说一下 stream\_url，这就是串流密钥，需要我们直接在 OBS 中配置，配置 OBS 串流密钥的截图如下所示。串流密钥的生成则需要我们通过后端功能接口来实现。

shikey.com转载分享

shikey.com转载分享



到这里，我们明确了直播后台搭建需要的工具，还有表设计和表关系了，也明确了每一个环节的实现方法，后续我们结合具体接口功能来配合前端操作就可以实现直播功能了。

## 总结

又到了课程的尾声，我们来回顾一下这节课的内容。

直播系统的开发流程中，音视频数据的收集到客户端直播播放，每一个环节都必不可少。你需要明确每个环节重点需要实现的功能，这样后续开发才会事半功倍。

接下来的功能模块梳理也很重要，具体就是流媒体模块、服务模块、管理模块这三部分。其中最核心的部分是流媒体模块，提升用户体验的部分是服务模块，提高效率、管理系统的则是管理模块。几个模块相互协作，才能确保直播稳定流畅，为用户提供更优质的服务体验。

最后，我们确定了直播后台搭建的方案。选用 OBS 采集音视频数据来推流，然后在前端通过 VideoJS 呈现直播内容。我们会使用 Nginx 和 nginx-rtmp-module 模块配置 RTMP 直播服务器。

为了满足直播模块的需求，还要新增两个表。channel 表用于管理频道房间生成，stream 表用于生成和管理推流码。我们先创建好这两个表的结构，后续课程里我们还会通过功能接口来开发相关功能。

下节课，我们将会用 Nginx 搭建直播服务器，敬请期待。

## 思考题

课程中我们在学习服务模块的时候，了解了较为常用的服务系统，你可以继续发散思维，想一想为了增强直播体验，还可以新增哪些服务呢？

欢迎你在留言区和我交流互动，也推荐你把这节课的内容分享给身边更多朋友。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

## 精选留言 (1)



**peter**

2023-07-09 来自北京

Q1: OBS可以用在安卓平台，也可以用在网页端吗？

本文的前端是用vue开发，即网页端，没有移动端。安卓平台可以用OBS，那么，网页也可以用OBS吗？

Q2: 直播流表并不包含流数据，只是记录流数据的位置，流数据单独存在其他地方，对吗？

作者回复: 1、OBS是开源直播录制软件，OBS可以用于安卓平台和网页端。如果你想志发起的直播能否在安卓和网页端使用，那就要看播放器了。

2、直播流表是一种用于管理直播流的表格，其中记录了每个直播流的地址、状态等信息，但并不包含实际的直播流数据。实际的直播流数据是通过其他方式传输的，比如RTMP、HLS等协议。这个在后面的课程都会讲，记得收听。



shikey.com转载分享

shikey.com转载分享