

# C/C++Linux服务器开发

## 高级架构师课程

三年课程沉淀

五次精益求精

十年行业积累

百个实战项目

十万内容受众

讲师:darren/326873713



扫一扫 升职加薪

班主任:柚子/2690491738

# 讲师介绍--专业来自专注和实力



**King老师**

系统架构师，曾供职著名创业公司系统架构师，微软亚洲研究院、创维集团全球研发中心。国内第一代商业Paas平台开发者。著有多个软件专利，参与多个开源软件维护。在全球化，高可用的物联网云平台架构与智能硬件设计方面有丰富的研发与实战经验。



**Darren老师**

曾供职于国内知名半导体公司（珠海扬智/深圳联发科），曾在某互联网公司担任音视频通话项目经理。主要从事音视频驱动、多媒体中间件、流媒体服务器的开发，开发过即时通讯+音视频通话的大型项目，在音视频、C/C++/GO Linux服务器领域有丰富的实战经验。



## 2 架构分析和API设计

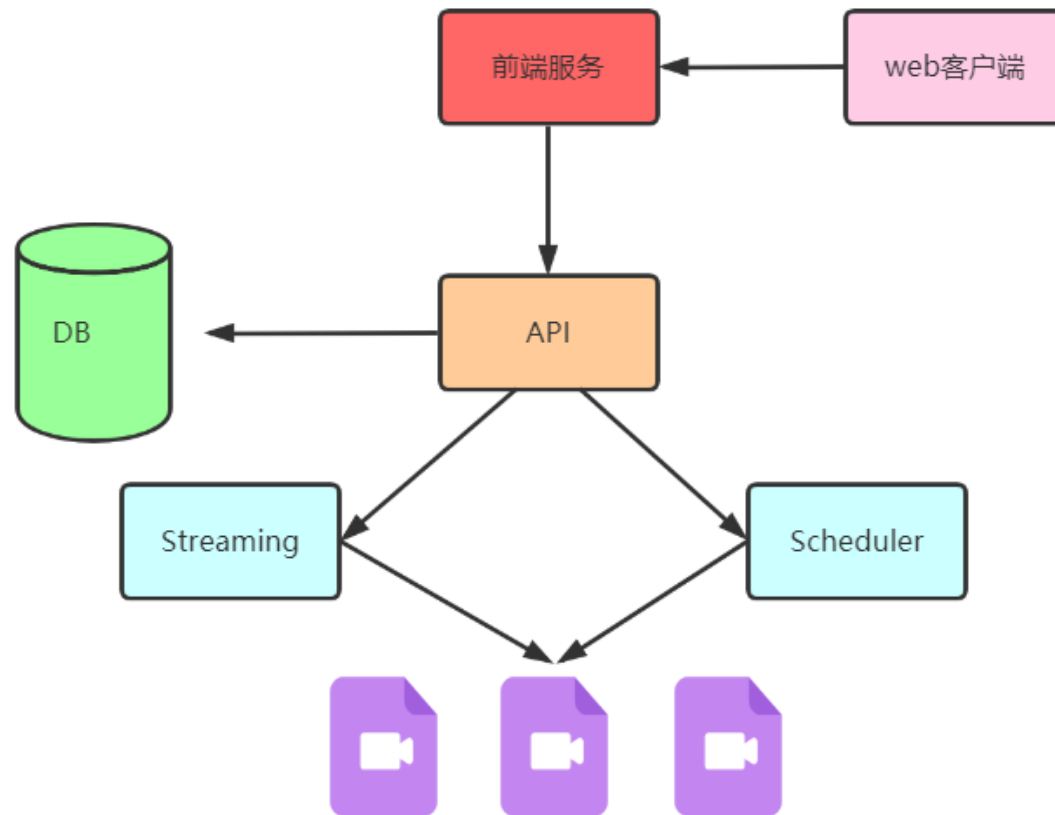
1. 技术要点分析
2. REST API设计
3. API设计实战
4. 端口开放

# go流媒体网站技术要点

- 前后端分离的系统架构设计
- RESTful风格API设计与实现
- Go实现web服务
- 系统的服务化解耦
- go的channel和并发模型的实践
- 使用go原生template完成web UI的实现



# 总体架构





# 什么是前后端解耦

- 前后端解耦是时下流行的Web网站架构
- 前端页面和服务通过普通的Web引擎渲染
- 后端数据通过渲染后的脚本调用后处理呈现





# 前后端解耦的优势

- 解放生产力
- 松耦合的架构更灵活，部署更方便，更符合微服务的设计特性
- 性能的提升、可靠性的提升



# ■ 前后端解耦的缺点

- 工作量大
- 前后端分离带来团队成本以及学习成本
- 系统复杂度加大







# API

- REST是Representational State Transfer（表现层状态转移）的缩写
- 常用的行为（查看（view），创建（create），编辑（edit）和删除（delete））都可以直接映射到HTTP 中已实现的GET,POST,PUT和DELETE方法。
- 通常使用Json作为数据封装格式
- 统一接口
- 无状态
- 可缓存





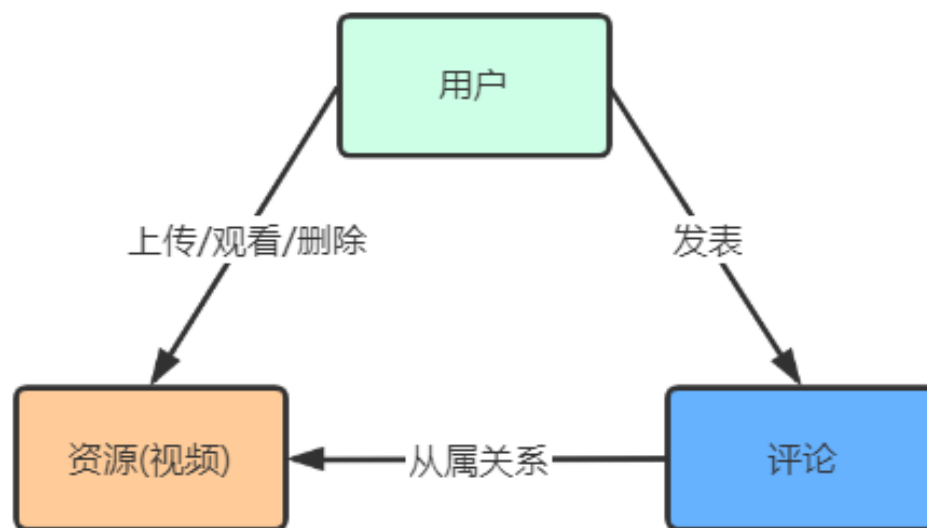
# API 设计原则

- 以URL（统一资源定位符）风格设计API
- 通过不同的Method（GET/POST/PUT/DELETE）来区分对资源的CURD
- 返回码（Status code）符合HTTP资源描述的规定





# API 设计





# API设计：用户

行为	URL	方法	状态码
注册用户	/user	POST	200, 201,400,500
用户登录	/user/:username	POST	200,400
获取用户信息	/user/:username	GET	200,400,401,403,500





## API设计：视频

行为	URL	方法	状态码
新增视频	/user/:username/videos	POST	
获取所有视频	/user/:username/videos	GET	
删除视频	/user/:username/videos/:vid-id	DELETE	





## API设计：评论

行为	URL	方法	状态码
增加评论	/videos/:vid-id/comments	POST	
显示所有的评论	//videos/:vid-id/comments	GET	





# 数据库设计-用户

```
CREATE TABLE `video_server`.`users` (  
  `id` int unsigned primary key auto_increment,  
  `login_name` varchar(64) unique key,  
  `pwd` text  
);
```





# 数据库设计-视频

```
CREATE TABLE `video_server`.`video_info` (  
  `id` varchar(64) NOT NULL,  
  `author_id` int(10) NULL,  
  `name` text NULL,  
  `display_ctime` text NULL,  
  `create_time` datetime NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
);
```







# 数据库设计-评论

```
CREATE TABLE `video_server`.`comments` (  
  `id` varchar(64) NOT NULL,  
  `video_id` varchar(64) NULL,  
  `author_id` int(10) NULL,  
  `content` text NULL,  
  `time` datetime(0) NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
);
```





# 数据库设计-会话

```
CREATE TABLE `video_server`.`sessions` (  
  `session_id` varchar(244) NOT NULL,  
  `TTL` tinytext NULL,  
  `login_name` text NULL,  
  PRIMARY KEY (`session_id`)  
);
```



# 数据库设计-待删除视频表

```
CREATE TABLE `video_server`.`video_del_rec` (  
  `video_id` varchar(64) NOT NULL,  
  PRIMARY KEY (`video_id`)  
);
```



# 端口开放

■ api 10000

■ scheduler 10001

■ streamserver 10002

■ web 10003

