**Mac 直播/点播服务器nginx+rtmp+hls 搭建**

**作者：arthur.liu**

[安装homebrow 2](#_Toc524446140)

[安装nginx服务器+rtmp模块 2](#_Toc524446141)

[安装ffmpeg 3](#_Toc524446142)

[配置nginx.conf 3](#_Toc524446143)

[基于rtmp的直播/点播 4](#_Toc524446144)

[直播 4](#_Toc524446145)

[配置 4](#_Toc524446146)

[rtmp 直播推流（本地资源推流 pc端，mac端推流） 4](#_Toc524446147)

[App 播放器 ijkplayer 播放 5](#_Toc524446148)

[电脑 vlc 播放 6](#_Toc524446149)

[点播 7](#_Toc524446150)

[配置 7](#_Toc524446151)

[App 播放器 ijkplayer 播放 8](#_Toc524446152)

[点播，直播区别 9](#_Toc524446153)

[电脑 vlc 播放 9](#_Toc524446154)

[基于hls的直播/点播 10](#_Toc524446155)

[直播 10](#_Toc524446156)

[配置 10](#_Toc524446157)

[Hls支持网页播放 12](#_Toc524446158)

[支持hls播放的浏览器 13](#_Toc524446159)

[Vlc播放 13](#_Toc524446160)

[Ijkplayer播放 14](#_Toc524446161)

[Hls直播特点 15](#_Toc524446162)

[点播 16](#_Toc524446163)

[ffmpeg将mp4转u3m8 16](#_Toc524446164)

[播放目录 16](#_Toc524446165)

[播放 16](#_Toc524446166)

[制作简单的视频网站 17](#_Toc524446167)

[视频列表 18](#_Toc524446168)

[播放界面 21](#_Toc524446169)

[页面传值 21](#_Toc524446170)

# 安装homebrow

Homebrew简称brew，是Mac OSX上的软件包管理工具，能在Mac中方便的安装软件或者卸载软件，可以说Homebrew就是mac下的apt-get、yum神器

/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

如果已经安装过，而想要卸载：

/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/uninstall)"

# 安装nginx服务器+rtmp模块

打开终端 依次复制

$ brew tap denji/nginx

brew install nginx-full --with-rtmp-module

* nginx安装所在位置  /usr/local/Cellar/nginx-full/1.10.1/bin/nginx
* nginx配置文件所在位置  /usr/local/etc/nginx/nginx.conf
* nginx服务器根目录所在位置  /usr/local/var/www
* 启动nginx服务
* nginx

# 安装ffmpeg

brew install ffmpeg

# 配置nginx.conf

用Xcode打开nginx.conf, 找到/usr/local/etc/nginx/nginx.conf 文件,拖入到Dock中的Xcode,就可以打开.

[复制代码](javascript:void(0);)

http {

……

}

#在http节点下面(也就是文件的尾部)加上rtmp配置：

rtmp {

server {

listen 1935;

application zbcs {

live on;

record off;

}

}

}

[复制代码](javascript:void(0);)

说明：

1. rtmp是协议名称
2. server 说明内部中是服务器相关配置
3. listen 监听的端口号, rtmp协议的默认端口号是1935
4. application 访问的应用路径是 zbcs
5. live on; 开启实时
6. record off; 不记录数据

# 基于rtmp的直播/点播

## 直播

### 配置

找到文件 /usr/local/etc/nginx/nginx.conf

文件末尾空白处 添加如下配置

rtmp {

server {

listen 1935;

ping 30s;

notify\_method get;

application rtmplive {

live on;

record off;

}

}

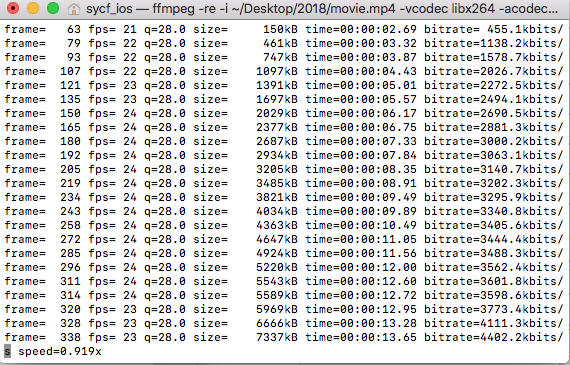
}

### rtmp 直播推流（本地资源推流 pc端，mac端推流）

本地资源推流 pc端，mac端推流

ffmpeg -re -i /Users/sycf\_ios/Desktop/2018/movie.mp4 -vcodec libx264 -acodec aac -strict -2 -f flv rtmp://localhost:1935/rtmplive/room

终端 执行如上，开始推流



### App 播放器 ijkplayer 播放

自定义Ijkplayer播放器：<https://github.com/fly0001/study_ijkplayer.git>

#pragma mark 界面布局

- (void)viewDidLoad {

[super viewDidLoad];

// Do any additional setup after loading the view.

//

self.view.frame=CGRectMake(0, 0, self.view.frame.size.width, self.view.frame.size.width/2);

NSURL \*url = [NSURL URLWithString:@"rtmp://localhost:1935/rtmplive/room"];

\_player = [[IJKFFMoviePlayerController alloc] initWithContentURL:url withOptions:nil];

UIView \*playerView = [\_player view];

playerView.frame = CGRectMake(0, 0, self.view.frame.size.width, self.view.frame.size.width/2);

[self.view addSubview:playerView];

[self.player prepareToPlay];

[self.player play];

[self createbottomview];

self.timer= [NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(updatetime) userInfo:nil repeats:YES];

UITapGestureRecognizer \*tapGesturRecognizer=[[UITapGestureRecognizer alloc]initWithTarget:self action:@selector(tapAction:)];

[playerView addGestureRecognizer:tapGesturRecognizer];

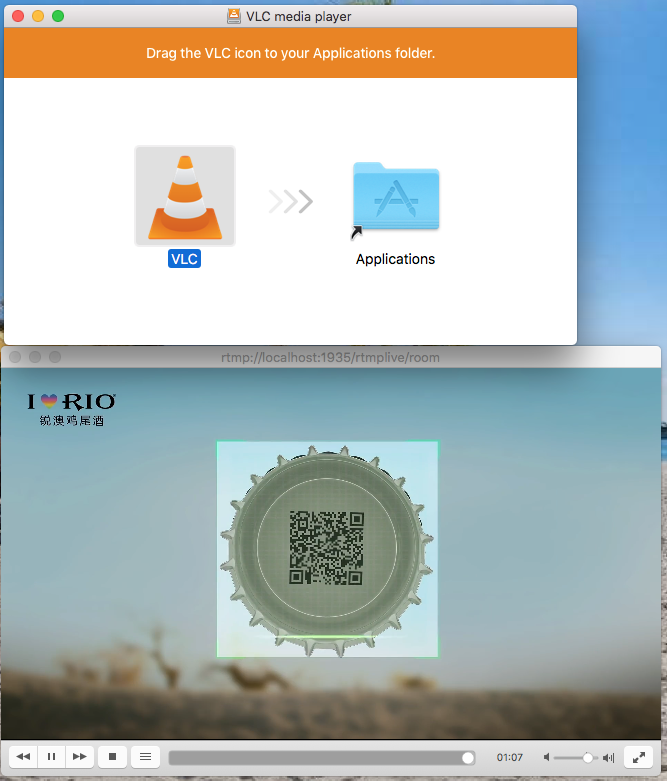
[self performSelector:@selector(hidebottom) withObject:nil afterDelay:10];

}



### 电脑 vlc 播放

播放地址 rtmp://localhost:1935/rtmplive/room 直播



## 点播

### 配置

找到文件 /usr/local/etc/nginx/nginx.conf

文件末尾空白处 添加如下配置

rtmp {

server {

listen 1935;

ping 30s;

notify\_method get;

application localmovie {

max\_connections 1024;

play /Users/sycf\_ios/Desktop/2018; #//视频文件存放位置。

}

}

}

### App 播放器 ijkplayer 播放

自定义Ijkplayer播放器：<https://github.com/fly0001/study_ijkplayer.git>

#pragma mark 界面布局

- (void)viewDidLoad {

[super viewDidLoad];

// Do any additional setup after loading the view.

//

self.view.frame=CGRectMake(0, 0, self.view.frame.size.width, self.view.frame.size.width/2);

//NSURL \*url = [NSURL URLWithString:@"rtmp://localhost:1935/rtmplive/room"]; 直播

NSURL \*url = [NSURL URLWithString:@"rtmp://localhost:1935/localmovie/movie.mp4"];//点播本地资源

\_player = [[IJKFFMoviePlayerController alloc] initWithContentURL:url withOptions:nil];

UIView \*playerView = [\_player view];

playerView.frame = CGRectMake(0, 0, self.view.frame.size.width, self.view.frame.size.width/2);

[self.view addSubview:playerView];

[self.player prepareToPlay];

[self.player play];

[self createbottomview];

self.timer= [NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(updatetime) userInfo:nil repeats:YES];

UITapGestureRecognizer \*tapGesturRecognizer=[[UITapGestureRecognizer alloc]initWithTarget:self action:@selector(tapAction:)];

[playerView addGestureRecognizer:tapGesturRecognizer];

[self performSelector:@selector(hidebottom) withObject:nil afterDelay:10];

}



### 点播，直播区别

视频是否有总时间

直播



点播



### 电脑 vlc 播放

播放地址：播放地址 rtmp://localhost:1935/localmovie/movie.mp4 点播



# 基于hls的直播/点播

## 直播

### 配置

找到文件 /usr/local/etc/nginx/nginx.conf

http {

server {

#这里加入

#HLS配置开始,这个配置为了`客户端`能够以http协议获取HLS的拉流

location /hls {

# Serve HLS fragments

types {

application/vnd.apple.mpegurl m3u8;

video/mp2t ts;

}

root html;

add\_header Cache-Control no-cache;

}

}

}

rtmp {

server {

listen 1935;

ping 30s;

notify\_method get;

chunk\_size 128;

publish\_time\_fix off;

#增加对HLS支持开始

application hls {

live on;

hls on;

hls\_path /usr/local/var/www/hls;

hls\_fragment 5s;

}

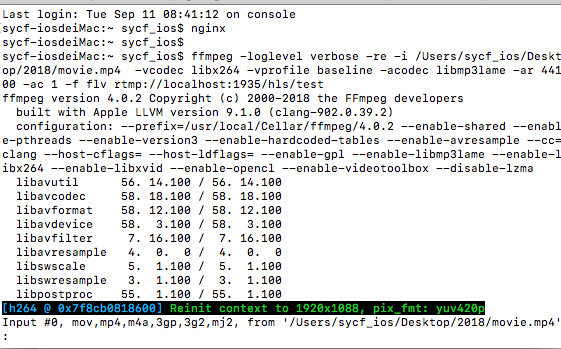
#增加对HLS支持结束

}

}

hls直播

ffmpeg -loglevel verbose -re -i /Users/sycf\_ios/Desktop/2018/movie.mp4 -vcodec libx264 -vprofile baseline -acodec libmp3lame -ar 44100 -ac 1 -f flv rtmp://localhost:1935/hls/test

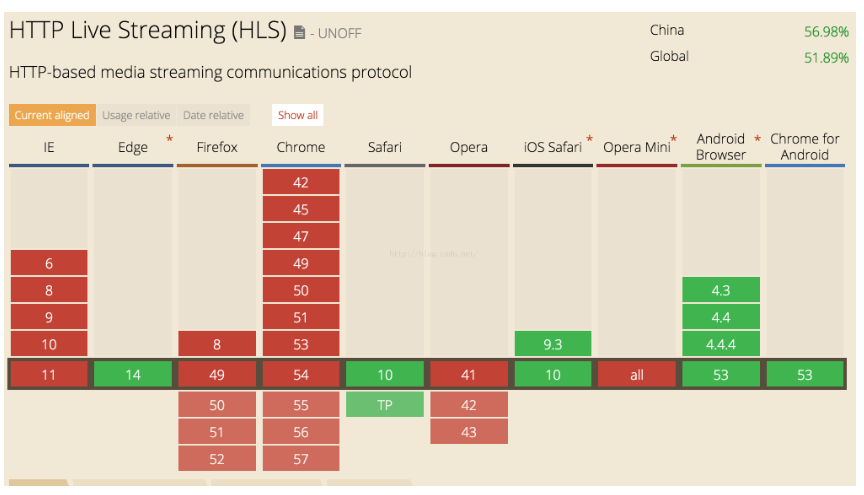


### Hls支持网页播放

播放地址<http://localhost:8080/hls/test.m3u8>



### 支持hls播放的浏览器



### Vlc播放

播放地址 rtmp://localhost:1935/hls/test 或者http://localhost:8080/hls/test.m3u8



### Ijkplayer播放

#pragma mark 界面布局

- (void)viewDidLoad {

[super viewDidLoad];

// Do any additional setup after loading the view.

//

self.view.frame=CGRectMake(0, 0, self.view.frame.size.width, self.view.frame.size.width/2);

//NSURL \*url = [NSURL URLWithString:@"rtmp://localhost:1935/rtmplive/room"]; 直播

// NSURL \*url = [NSURL URLWithString:@"rtmp://localhost:1935/localmovie/movie.mp4"];//点播本地资源

NSURL \*url = [NSURL URLWithString:@"rtmp://localhost:1935/hls/test"];//hls直播 播放地址：rtmp://localhost:1935/hls/test 或者 http://localhost:8080/hls/test.m3u8

\_player = [[IJKFFMoviePlayerController alloc] initWithContentURL:url withOptions:nil];

UIView \*playerView = [\_player view];

playerView.frame = CGRectMake(0, 0, self.view.frame.size.width, self.view.frame.size.width/2);

[self.view addSubview:playerView];

[self.player prepareToPlay];

[self.player play];

[self createbottomview];

self.timer= [NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(updatetime) userInfo:nil repeats:YES];

UITapGestureRecognizer \*tapGesturRecognizer=[[UITapGestureRecognizer alloc]initWithTarget:self action:@selector(tapAction:)];

[playerView addGestureRecognizer:tapGesturRecognizer];

[self performSelector:@selector(hidebottom) withObject:nil afterDelay:10];

}



### Hls直播特点

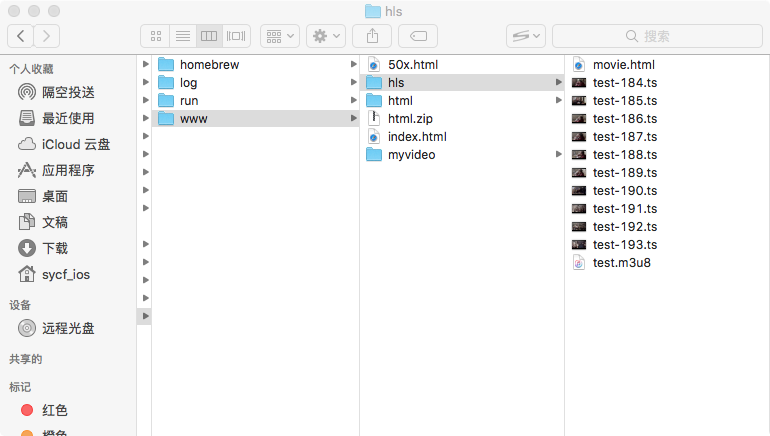
跨平台性好，支持网页播放，vlc，ijkplayer

网页播放 播放地址格式是<http://localhost:8080/hls/test.m3u8>

播放地址rtmp://localhost:1935/hls/test和

<http://localhost:8080/hls/test.m3u8>可以在vlc，ijkplayer播放，

目录/usr/local/var/www/hls。直播是 生成 m3u8文件及 大量ts文件。在这个目录 ts文件会随着推流直播被清空



## 点播

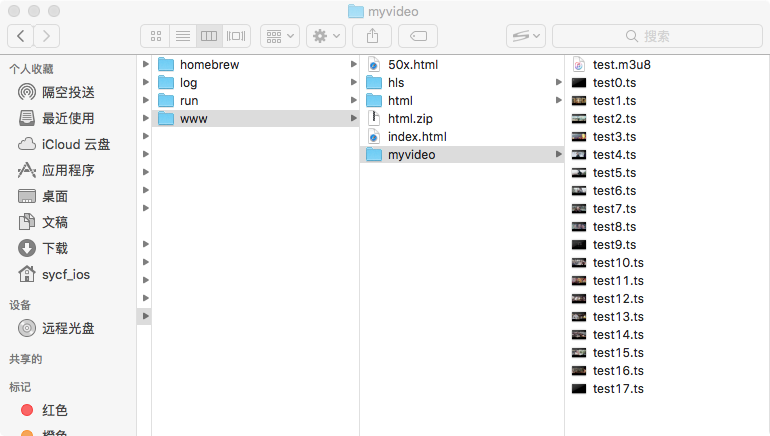
### ffmpeg将mp4转u3m8

打开终端，cd 到视频目录

ffmpeg -i zly.mp4 -c:v libx264 -c:a aac -strict -2 -f hls -hls\_time 15 -hls\_list\_size 0 test.m3u8

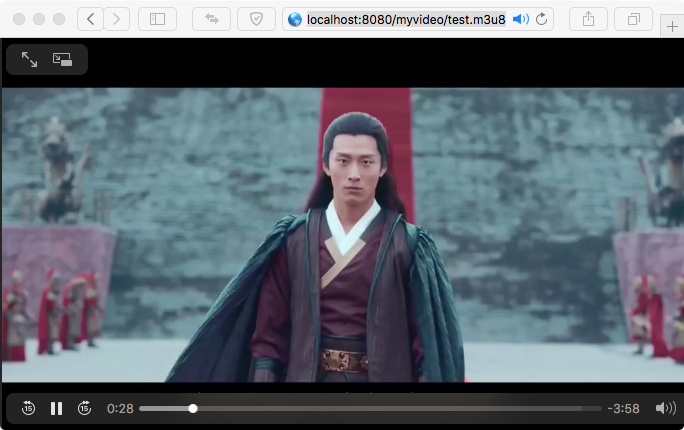
### 播放目录

在/usr/local/var/www/ 目录 新建文件夹 如myvideo 把 生成的test.m3u8 及ts文件 存放这里



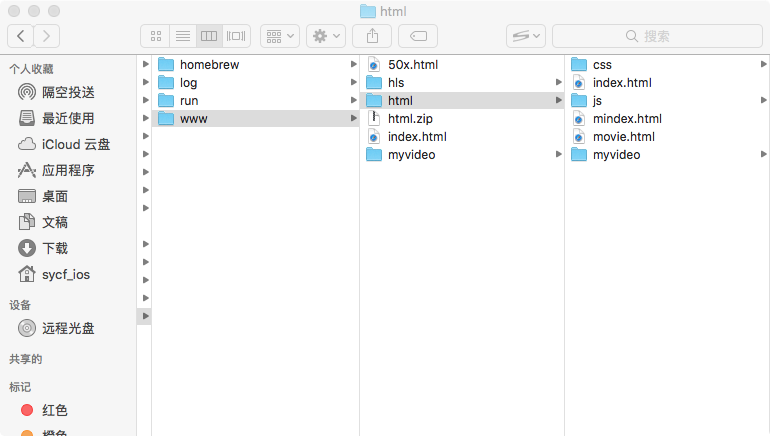
### 播放

在safari浏览器 vlc ，ijkplayer就是可以通过如下地址 播放了



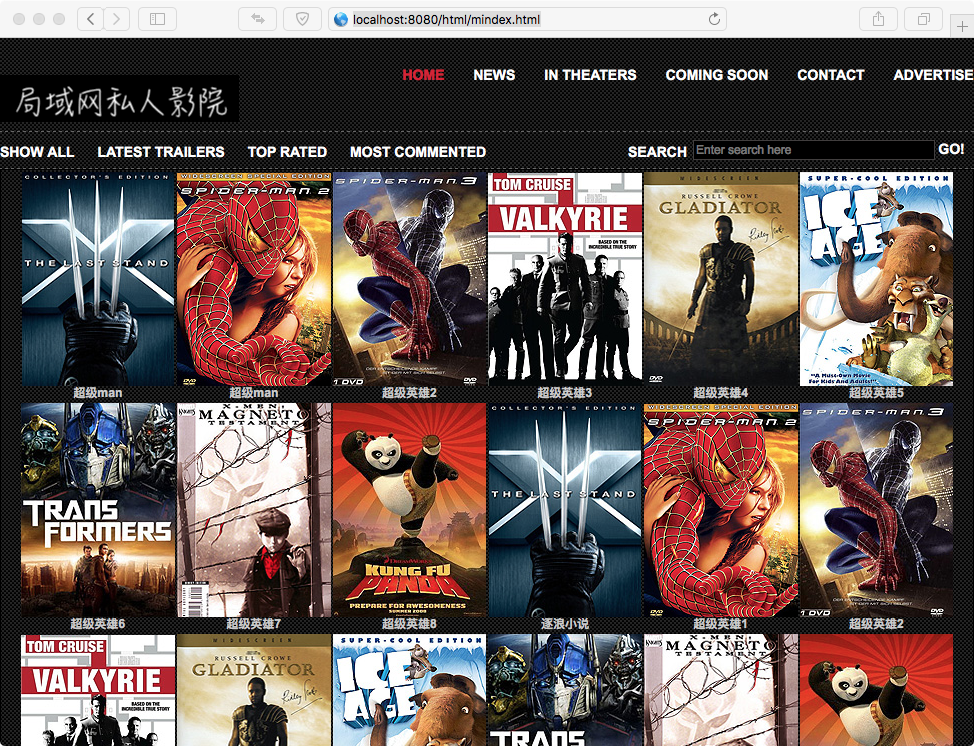
# 制作简单的视频网站

在/usr/local/var/www/ 目录新建html目录存放 html文件及视频文件



终端开启 nignx

Safary浏览器打开：<http://localhost:8080/html/mindex.html>





## 视频列表

目前采用静态数据，视频列表实现如下

<script>

var website = [

{"site":"超级man","url":"/html/movie.html?film=myvideo","pic":"css/images/movie1.jpg"},

{"site":"超级man","url":"/html/movie.html?film=myvideo","pic":"css/images/movie2.jpg"},

{"site":"超级英雄2","url":"/html/movie.html?film=myvideo","pic":"css/images/movie3.jpg"},

{"site":"超级英雄3","url":"http://www.qidian.com","pic":"css/images/movie4.jpg"},

{"site":"超级英雄4","url":"http://www.qidian.com","pic":"css/images/movie5.jpg"},

{"site":"超级英雄5","url":"http://www.qidian.com","pic":"css/images/movie6.jpg"},

{"site":"超级英雄6","url":"http://www.qidian.com","pic":"css/images/movie7.jpg"},

{"site":"超级英雄7","url":"http://www.qidian.com","pic":"css/images/movie8.jpg"},

{"site":"超级英雄8","url":"http://www.qidian.com","pic":"css/images/movie9.jpg"},

{"site":"逐浪小说","url":"/html/movie.html?film=myvideo","pic":"css/images/movie1.jpg"},

{"site":"超级英雄1","url":"/html/movie.html?film=myvideo","pic":"css/images/movie2.jpg"},

{"site":"超级英雄2","url":"/html/movie.html?film=myvideo","pic":"css/images/movie3.jpg"},

{"site":"超级英雄3","url":"http://www.qidian.com","pic":"css/images/movie4.jpg"},

{"site":"超级英雄4","url":"http://www.qidian.com","pic":"css/images/movie5.jpg"},

{"site":"超级英雄5","url":"http://www.qidian.com","pic":"css/images/movie6.jpg"},

{"site":"超级英雄6","url":"http://www.qidian.com","pic":"css/images/movie7.jpg"},

{"site":"超级英雄7","url":"http://www.qidian.com","pic":"css/images/movie8.jpg"},

{"site":"超级英雄8","url":"http://www.qidian.com","pic":"css/images/movie9.jpg"},

{"site":"逐浪小说","url":"/html/movie.html?film=myvideo","pic":"css/images/movie1.jpg"},

{"site":"超级英雄1","url":"/html/movie.html?film=myvideo","pic":"css/images/movie2.jpg"},

{"site":"超级英雄2","url":"/html/movie.html?film=myvideo","pic":"css/images/movie3.jpg"},

{"site":"超级英雄3","url":"http://www.qidian.com","pic":"css/images/movie4.jpg"},

{"site":"超级英雄4","url":"http://www.qidian.com","pic":"css/images/movie5.jpg"},

{"site":"超级英雄5","url":"http://www.qidian.com","pic":"css/images/movie6.jpg"},

{"site":"超级英雄6","url":"http://www.qidian.com","pic":"css/images/movie7.jpg"},

{"site":"超级英雄7","url":"http://www.qidian.com","pic":"css/images/movie8.jpg"},

{"site":"超级英雄8","url":"http://www.qidian.com","pic":"css/images/movie9.jpg"}

];

var col = 6;//表格列数

var td = website.length%col==0?website.length:Math.ceil(website.length/col)\*col;

var \_i = 0;

var html ="<table ><tr>";

for(var i=0;i<td;i++){

if(website[i]){

html += '<th><a href="'+website[i].url+'"><img src="'+website[i].pic+'" alt="movie" /></a><p>'+website[i].site+'</p></th>';

}else{

html += '<th></th>';

}

if((i+1)%col==0 && i>0){

html+="</tr><tr>";

}

}

html +="</tr></table>"

document.getElementById('siteTable').innerHTML = html;

</script>

## 播放界面

<style>

video {

object-fit:fill;

width:640px;

height:320px;

}

</style>

<video id="videoid" src="" controls="controls">

</video>

## 页面传值

mindex.html中

<a href="'+website[i].url+'">

"url":"/html/movie.html?film=myvideo" 其中film=myvideo 为点击要播放的视频路径

Movie.html中获取参数，播放对应路径的视频

<script>

function getRequest1(){

var str=location.search; // ?userName=zhanghao$userId=123

if(str){

var url=str.substr(1),arr=url.split('&'),len=arr.length,i= 0,request={};

// ["userName=zhanghao", "userId=123"]

for(;i<len;i++){

request[arr[i].split('=')[0]]=arr[i].split('=')[1];

// ["userName", "zhanghao"] ["userId", "123"]

}

return request; //{userName: "zhanghao", userId: "123"}

}else{

alert('没有传递参数');

}

}

var Request=getRequest1();//可以查看返回的对象，根据对象的参数获取其中的键值对

var name=Request.film;//获取某个属性的值

alert(name);

document.getElementById("videoid").src = name+"/test.m3u8";

</script>