



- ▶ 姓名 : 毛静文 ( Momo )
- ▶ QQ : 707577045
- ▶ Github : <https://github.com/Momo707577045>
- ▶ 工作经历 : 曾从事H5游戏开发 , Web前端开发。  
现任贝聊web前端开发工程师 , 负责直播相关业务开发。

# H5视频播放页开发总结

By毛静文

# 内容定位：

- ▶ 开发中遇到的问题及其解决方案。
- ▶ 穿插部分知识点的延伸。
- ▶ 以解决方案为主线的知识点杂谈。



# 主要解决的问题

- ▶ 直播视频的播放适配。
- ▶ 软键盘遮挡输入框问题。
- ▶ 下拉无限加载的实现。

# 直播视频的播放适配

## 常规直播技术

- ▶ RTMP(Real Time Messaging Protocol)是Adobe公司推出的基于flash的实时视频消息传输协议。
- ▶ 服务端和用户端通过flash建立长链接，实时进行数据传递，对应的视频格式是flv。
- ▶ 视频延迟低，实时性高。



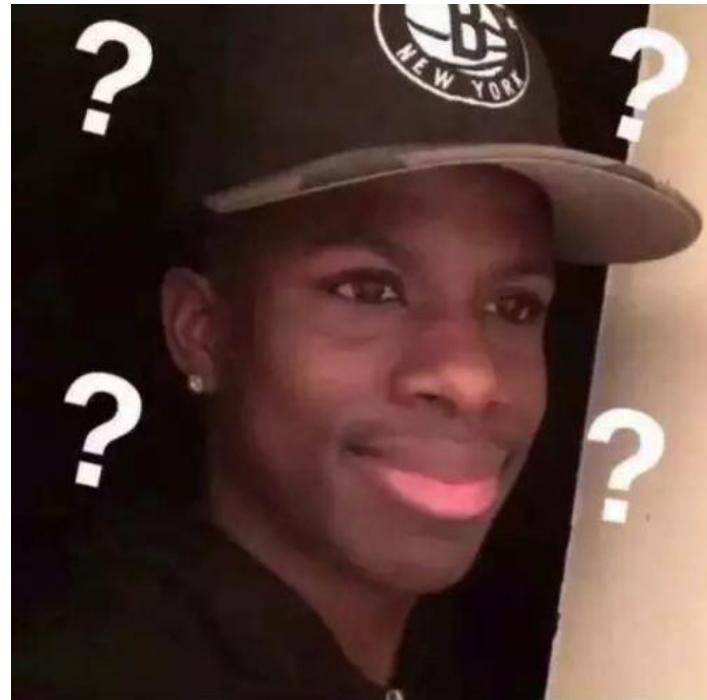
## 常规直播技术

- ▶ HLS(HTTP Live Streaming)是苹果公司为了抵制flash而实现的基于 HTTP 的流媒体传输协议。
- ▶ 对视频进行切片，用户端加载各个视频片段进行播放，对应的视频格式是m3u8。
- ▶ 实时性较低
- ▶ 可用CDN缓存。



## video标签对m3u8的支持情况

- ▶ video标签只支持在移动端及Safari浏览器下播放m3u8视频。



- ▶ HLS协议和Safari同属苹果公司的产物。
- ▶ iPhone在移动端的领头羊作用，其他移动端纷纷效仿。

## m3u8的播放适配方案

- ▶ 移动端和Safari使用H5的video标签播放。
- ▶ 其他情况下，借助flash进行播放。



# 第三方播放器组件问题

- ▶ 七牛推荐播放器：pili-mediaElement  
当用flash兼容m3u8适配时视频内容不能全屏播放。
- ▶ Sewise播放器  
单例模式，无法销毁视频。  
切换视频时，显示上一个视频最后一张的画面。



# 播放器组件的封装

- ▶ 通过userAgent进行判断。
- ▶ 在移动端或Safari中使用video标签播放。
- ▶ 其余情况借助flash进行播放。
- ▶ 判断是否是Safari的正则：  
`/Safari/.test(navigator.userAgent);`
- ▶ 以上正则是否能成功区分  
Safari浏览器？



我觉得不行



我觉得还OK



## Safari匹配问题

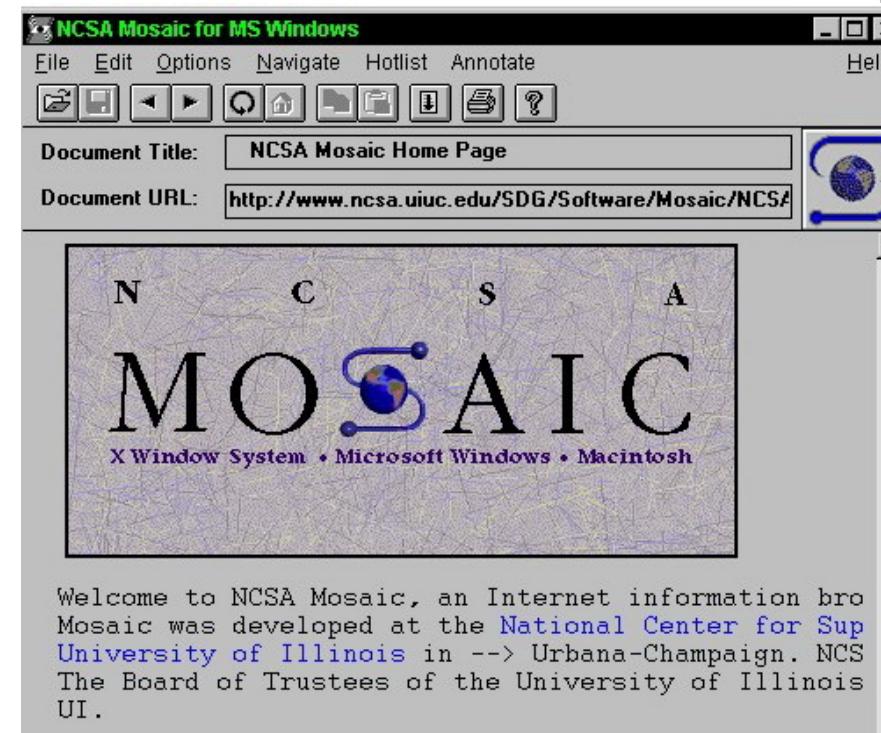
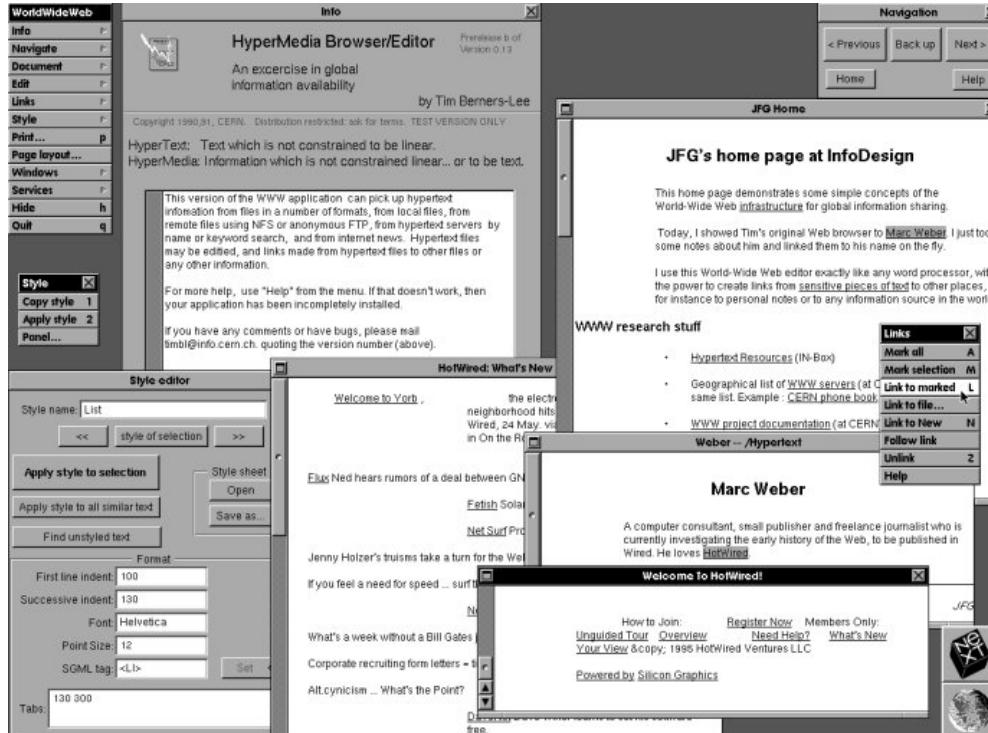
- ▶ /Safari/.test(navigator.userAgent);  
在Chrome上也能匹配。

```
> navigator.userAgent
< "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6)
  AppleWebKit/537.36 (KHTML, like Gecko)
  Chrome/60.0.3112.101 Safari/537.36"
```



# userAgent野史

- ▶ 最早的浏览器名为Nexus，诞生于90年。该浏览器并不支持图片的显示。
- ▶ 93年，NCSA组织发明了第一款可显示图片的浏览器Mosaic(马赛克)。
- ▶ 为了区分是否支持图片显示，添加了userAgent字段。



## UserAgent野史

- ▶ 到了94年，网景公司(Netscape)研发了Mozilla浏览器。取自哥斯拉(Godzilla)谐音，也有“马赛克的终结者(Mosaic Killa)”的意思。
- ▶ 后来Mozilla改名为Netscape Navigator(网景航海家)，但 userAgent里字段名还是沿用Mozilla字段。



mozilla



Netscape

# UserAgent野史

- ▶ Netscape率先支持html标准框架，迅速占领市场份额。服务器端通过userAgent来判断是否返回支持html框架的文件。
- ▶ 后来出现了同样支持html标准框架的 IE 浏览器。为了获取 html标准框架的文件，在userAgent中加入了Mozilla字段。
- ▶ 由此Mozilla字段从普通的浏览器名，变成了浏览器支持 html标准框架的标示。



# UserAgent野史

- ▶ Safari在2003年发布，Chrome则是2008年才发布。
- ▶ Chrome和Safari的内核都是webkit，功能基本相同。
- ▶ Chrome为了得到写给Safari的页面，伪装成Safari。



## 匹配Safari

- ▶ 匹配规则：拥有Safari字段，并且没有Chrome字段。

- ▶ 常规的写法

```
/Safari/.test(navigator.userAgent)&&  
!/Chrome/.test(navigator.userAgent);
```

- ▶ 也可以利用navigator.vendor「浏览器提供商」字段。

- ▶ /Apple/.test(navigator.vendor)

```
> navigator.userAgent
```

```
< "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/601.7.7 (KHTML, like Gecko) Version/9.1.2 Safari/601.7.7" = $1
```

```
> navigator.vendor
```

```
"Apple Computer, Inc." = $2
```

## 抛弃flash的兼容方案

- ▶ HLS协议是苹果公司为了淘汰flash而实现直播技术。用flash进行兼容播放，有违HLS设计初衷。
- ▶ hls.js通过MSE(Media Source Extensions) 将m3u8视频在用户端进行格式转码(转换成mp4)，再放入video进行播放。



hls.js

# hls.js

- ▶ 目前MSE已普遍得到各大浏览器的支持。hls.js在非IE下主流浏览器(如，Chrome，Firefox，Edge，360等)也能完美运行。

```
<script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
<video id="video"></video>
<script>
  if(Hls.isSupported()) {
    var video = document.getElementById('video');
    var hls = new Hls();
    hls.loadSource('https://video-dev.github.io/streams/x36xhzz/x36xhzz.m3u8');
    hls.attachMedia(video);
    hls.on(Hls.Events.MANIFEST_PARSED,function() {
      video.play();
    });
  }
</script>
```

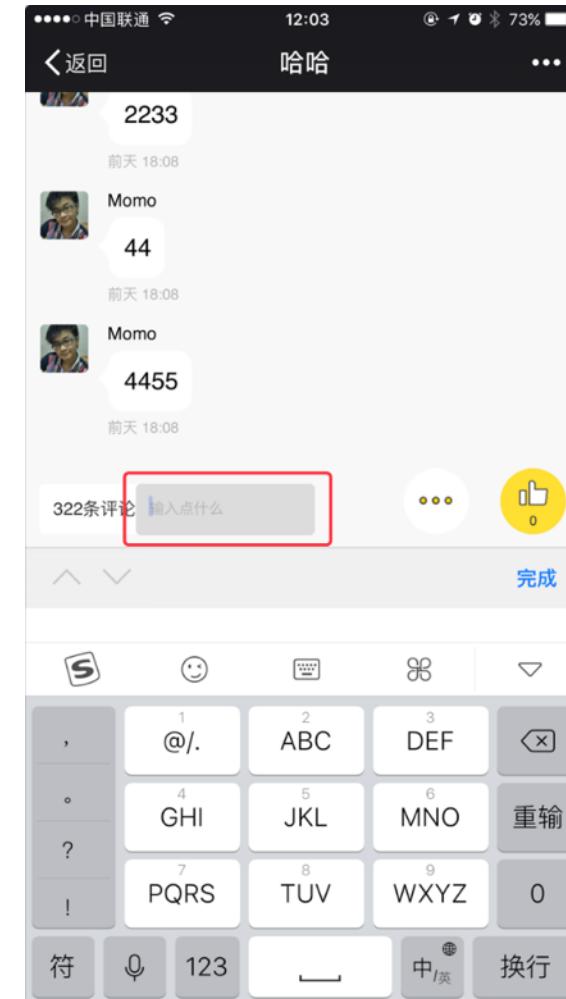
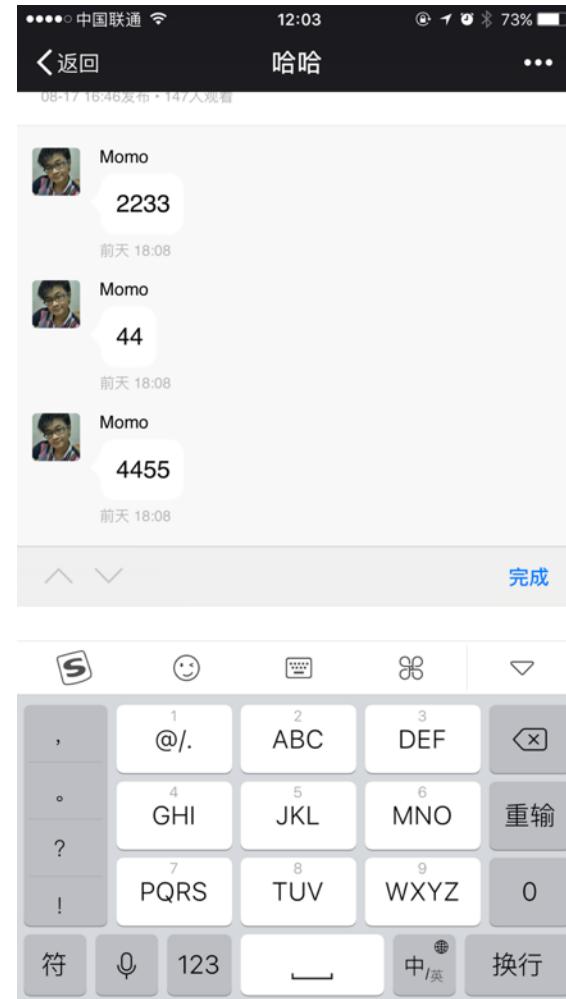
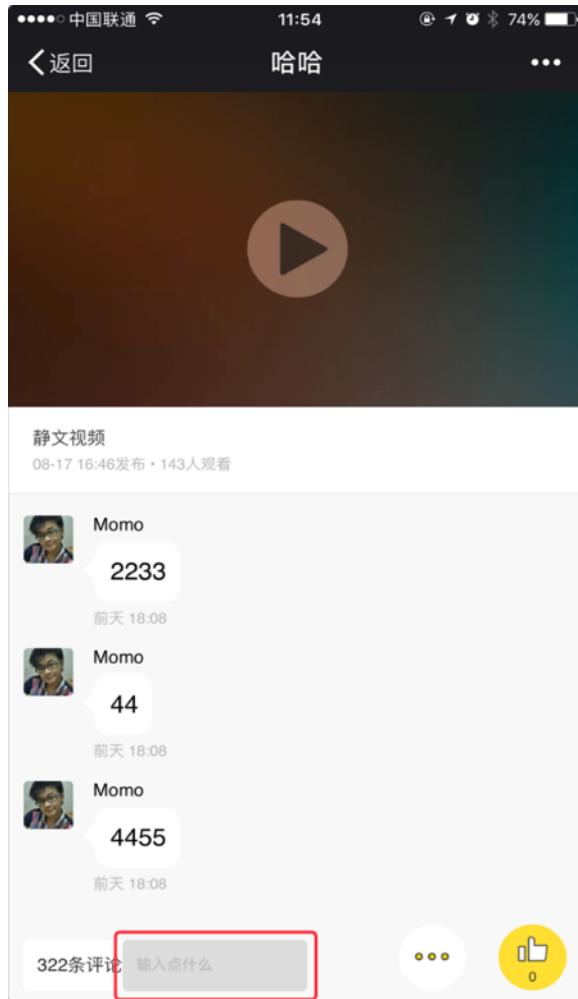
## 最终的适配方案

- ▶ 通过userAgent进行判断。
- ▶ 在移动端或Safari浏览器中使用video标签播放。
- ▶ 在其余的非 IE 浏览器端使用hls.js进行兼容播放。
- ▶ 在 IE 浏览器借助flash进行播放。



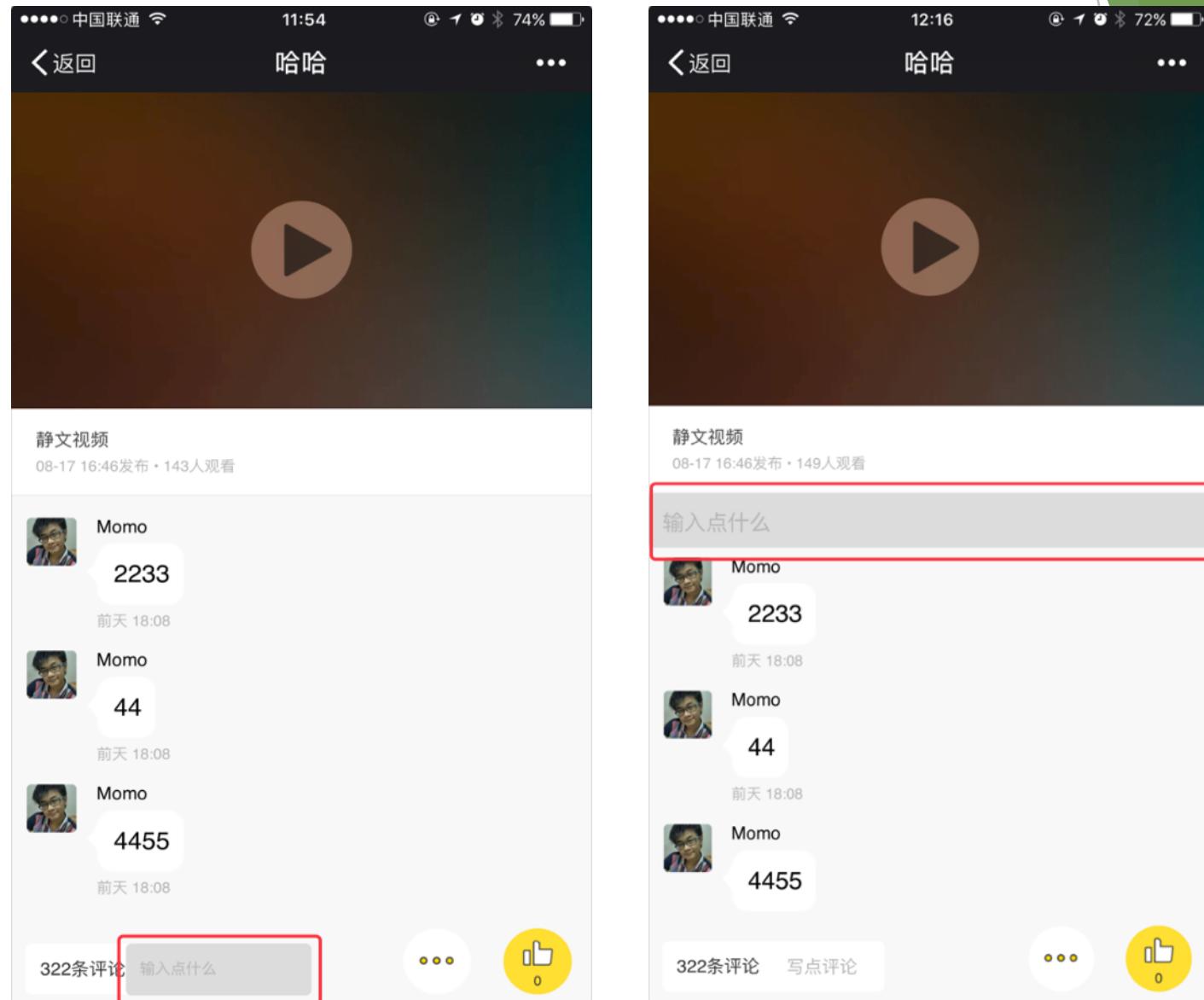
# 软键盘遮挡输入框问题

# 软键盘遮挡输入框



# 常规解决方案

- ▶ 将评论框置于底部，在软键盘弹出后调用scrollIntoView()方法将显示容器滚动至底部。
- ▶ 将输入框置于屏幕上半部分。



# scrollIntoView与scrollIntoViewIfNeeded

函数	scrollIntoView	scrollIntoViewIfNeeded
触发条件	调用即触发	仅当元素显示不完全时调用触发
默认值	true	true
功能	传值为true，窗口会尽可能滚动到自身顶部与元素顶部平齐	传值为true，元素将在其所在滚动区的可视区域中居中对齐
	传值为false，窗口会尽可能滚动到自身底部与元素底部平齐	传值为false，元素将与其所在滚动区的可视区域最近的边缘对齐
兼容性	除IE6,7以外，其余浏览器均支持	webkit内核的浏览器支持，如chrome，safari，opara

# 异常表现

- ▶ scrollIntoView在iOS下，当传入值为false，且当输入框位于上半屏进行点击输入时，输入框会被滚动到软键盘下方。



# 额外延伸

为什么scrollIntoView转入为true，且位于屏幕底部点击输入时，能显示在页面底部？



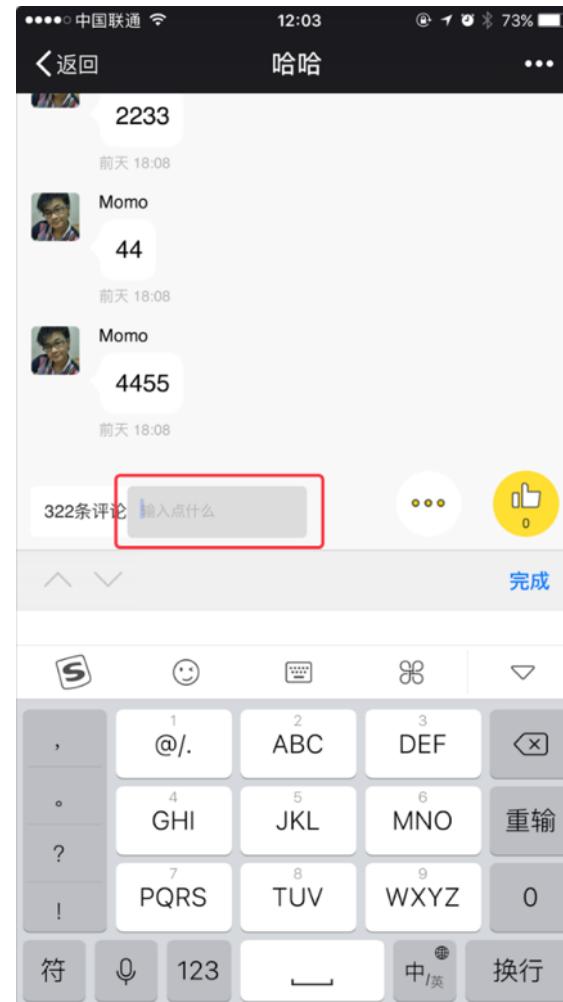
# 异常表现

- ▶ scrollIntoViewIfNeeded  
传值true并不一定与显示容器居中，只要完全不显示时，才会居中。部分显示则会与边缘对齐。



## 方案一的缺陷：

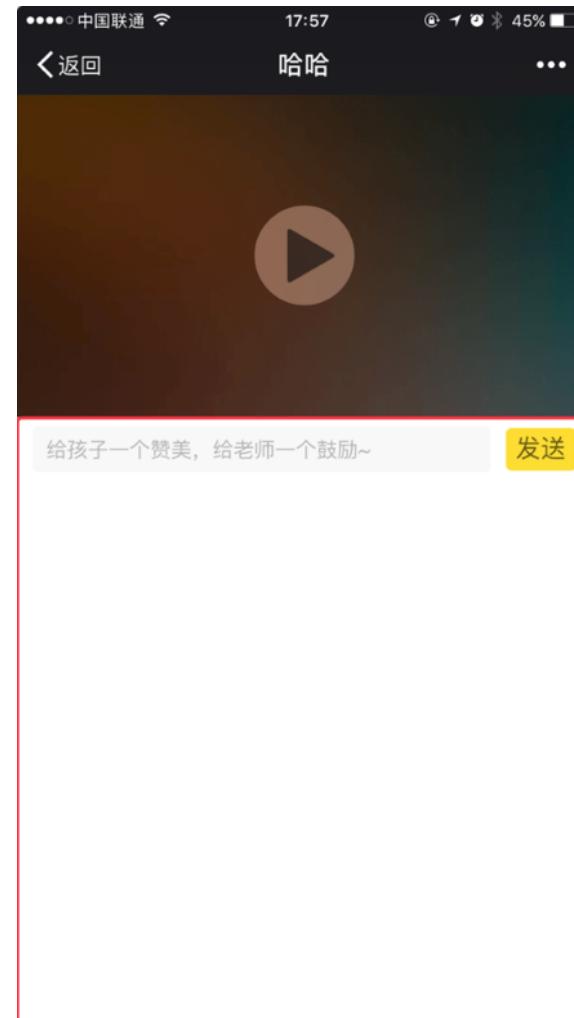
- ▶ 页面只能滚动到底部，在输入过程中，无法显示页面上半部分内容。可能由于进行评论错过了视频精彩内容。



## 方案二：将输入框置于屏幕上半部分



点击评论按钮



弹出输入层



编写评论

# 同步输入层与软键盘

- ▶ 如何在显示输入层的同时调起软键盘？
- ▶ 显示输入层后，对输入框调用focus操作。



# 同步输入层与软键盘

- ▶ 如何监听软键盘收起操作？



- ▶ 监听input的blur事件。  
在安卓下，需要额外监听  
resize事件判断软键盘收  
起操作。



# 下拉无限加载

# 上拉无限加载

- ▶ 滚动区滚动到底部。
- ▶ 触发数据加载。
- ▶ 从后端获取数据。
- ▶ 拿到数据后直接将数据添加到显示容器的底部。



# 下拉无限加载

- ▶ 滚动区滚动到顶部。
- ▶ 触发数据加载。
- ▶ 从后端获取数据。
- ▶ 将数据添加到显示容器的顶部。
- ▶ 计算并设置滚动区scrollTop的值。



# 为什么需要设置scrollTop值？

怎么肥四！



- ▶ 因为滚动区是根据scrollTop值决定显示的位置的。
- ▶ 直接往显示容器头部添加数据会导致显示数据跳跃。
- ▶ 显示容器好比show(array[2])。
- ▶ 尾部添加好比array.push(data)。
- ▶ 头部添加好比array.unshift(data)。



# 下拉无限加载

- ▶ 滚动区滚动到顶部，
- ▶ 记录scrollHeight的值(记为scrollHeightBeforeAdd)
- ▶ 触发数据加载
- ▶ 从后端获取数据
- ▶ 将数据添加到显示容器的顶部。
- ▶ 再次获取scrollHeight的值(记为scrollHeightAfterAdd)
- ▶ 修改scrollTop值:
- ▶  $\text{scrollTop} += \text{scrollHeightAfterAdd} - \text{scrollHeightBeforeAdd};$

## 插入数据的时机

- ▶ 在iOS下需要在滚动容器的CSS中添加-webkit-overflow-scrolling: touch;样式来实现顺滑滚动。
- ▶ 当页面处于滚动趋势中，将无法设置其scrollTop的值。
- ▶ 页面定位失败，数据显示跳跃。
- ▶ 监听显示容器滚动趋势，待滚动趋势结束后，再往显示容器添加数据。



编程5分钟  
改bug两小时

# 间接监听滚动趋势结束

## ▶ 监听滚动趋势

```
onScroll() {
    loadMoreStore.commit('setScrolling', true);
    clearTimeout(this.scrollTimer);
    this.scrollTimer = setTimeout(() => {
        loadMoreStore.commit('setScrolling', false);
    }, 1000);
},
```

## ▶ 添加数据

```
loadMore() {
    ajax.send('./getData', { params: params }).then((res) => {
        let timer = setInterval(() => {
            if (!loadMoreStore.state.isScrolling) {
                clearInterval(timer);
                setData(res); // 显示数据
            }
        }, 100);
    });
}
```

# 下拉无限加载

- ▶ 滚动区滚动到顶部，
- ▶ 记录scrollHeight的值(记为scrollHeightBeforeAdd)
- ▶ 触发数据加载
- ▶ 从后端获取数据
- ▶ 等待显示容器滚动趋势结束
- ▶ 将数据添加到显示容器的顶部
- ▶ 再次获取scrollHeight的值(记为scrollHeightAfterAdd)
- ▶ 修改scrollTop值:  
`scrollTop += scrollHeightAfterAdd – scrollHeightBeforeAdd;`

终于讲完了

谢谢大家！

钢铁侠 蜘蛛侠 蝙蝠侠 雷峰侠此刻灵魂附体