利用HTML5 Web Audio API给网页JS交互增加声音

这篇文章发布于 2017年06月10日，星期六，23:41，归类于 JS API。 阅读 47296 次, 今日 37 次 32 条评论

一、庞然的HTML5 Web Audio API

首先务必要弄清这一点，本文这里所说的HTML5 Web Audio API和HTML5 `<audio>` 元素完全不是一个东西，其体量也完全不是一个等级的，HTML5 Web Audio API接口的丰富程度和体量可以和HTML canvas API相提并论，其能实现的功能也非常令人瞠目。

HTML5 Web Audio API可以让我们无中生有创造声音，而且是各种音调的声音，换句话说，我们通过JavaScript就会创建一个完整的音乐出来，类似卡农将有固定的音乐，或是通过一些随机算法创建随机的音乐都都可以由前端开发工程师来完成（例如这个使用Web Audio API创建多个经典游戏背景音乐的案例），如果再配合一点人工智能的东西，我们说不定可以造出一个人工智能音乐生成工具。

当然其功能绝不仅限于这一点，我们还可以：

- 对简单或复杂的声音进行混合；
- 精确控制声音的密度和节奏；
- 内置淡入/淡出，颗粒噪点，音调控制等效果；
- 灵活的处理在音频流的声道，使它们成为拆分和合并；
- 处理从 `<audio>` 音频或 `<video>` 视频的媒体元素的音频源；
- 使用MediaStream的getUserMedia()方法事实处理现场输入的音频，例如变声；
- 立体音效，可以支持多种3D游戏和沉浸式环境；
- 利用卷积引擎，创建各类线性音效，例如小/大房间、大教堂、音乐厅、洞穴、隧道、走廊、森林、圆形剧场等。尤其适用于创建高质量的房间效果。
- 高效的实时的时域和频分析，配合CSS3或Canvas或webGL可以实现音乐可视化支持；
- 以及其他多种音频波形算法控制，几乎就是把一个音频编辑类软件搬到web上。

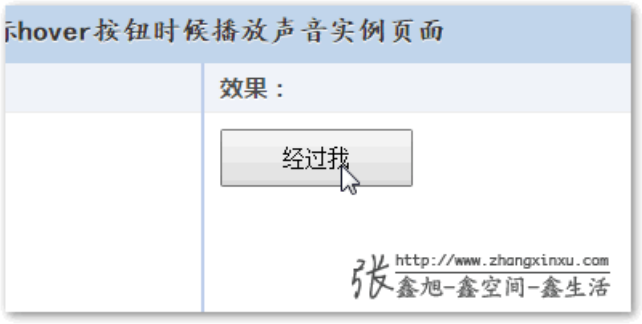
但是对大多数的Web开发人员，我们并不需要和音频打交道这么深，因为毕竟不是做音乐网站的，此时，那些很高级的HTML5 Web Audio API功能就离我们实际工作的价值有些远，如果贸然花大量时间去深入学习其实是一件投入产出比很低的事情。当然，我这种担心实际上是多余的，“贸然花大量时间去深入学习？”搞反了吧，其实真实的现状是开发人员对此鸟都不鸟，看都不看。一来因为毕竟不火嘛，二来资料少看不懂嘛，三来自己项目很少有声音打交道，学了也白学。

考虑到这种现状，如果本文内容洋洋洒洒讲各种api，估计很多人瞟两眼就直观把页面关掉了，所以接下来我们从一个最简单最小的而且非常实用的案例来慢慢解开HTML5 Web Audio API的面纱。

二、HTML5 Web Audio API给网页交互增加声音

先来看实例效果，您可以狠狠地点击这里：鼠标hover按钮无中生有播放声音demo

当我们鼠标hover下图所示的按钮的时候，在非IE浏览器下，就会播放出类似电子琴琴键按下的声音，而且每次hover的时候，音调都会发生有规律的变化：



下面这句话是重点，demo页面中播放的声音是硬件自动生成的，不是调用现成的mp3音频播放出来的。

换种说法就是：无需任何音频文件，只需要几行JS代码，我们就能让网页发出各种各样的音调。

10年前，flash炫酷网页风兴起的时候，交互音效是非常普遍的，可以说是flash酷站必备元素。只是后来flash衰落了，而web对于多媒体资源的支持长久以来是比较弱的，于是Web网站全部都是安安静静的。在移动端，H5广告营销性质的网页都是音频用的比较多，但大多都是作为背景音乐使用的，交互时候其实用的并不多。我想，可能与大多数开发并不知道HTML5 Web Audio API可以直接弄出声音，知识的广度局限了技术的选型。

于是我就想到，HTML5 Web Audio API提及的人不多，使用并不火，并不一定是因为这个技术本身的局限或者使用场景不多，而是具有较大影响力的人，社区或企业并没有在这一块身先士卒进行推动。很多企业的官网、产品宣传页，广告营销性质的专题活动非常讲求炫酷，音视频层出不穷，很显然（参考flash时代），交互音效也其实是非常欢迎的，只是技术眼界的局限认为需要专门的mp3/wav音频调用成本高而不使用罢了。

HTML5 Web Audio API可以直接产生声音，并且可以随意控制音调，时长以及淡入淡出等效果，更强大更灵活。而这个功能点是HTML5 Web Audio API中最简单最基本也是最实用的，没有理由不给web开发带来交互音效热潮。

三、网页交互音效AudioContext，AudioParam等API介绍

下面就是声音出现的JS相关代码：

```
1.  window.AudioContext = window.AudioContext || window.webkitAudioContext;
2.  var audioCtx = new AudioContext();
3.  var oscillator = audioCtx.createOscillator();
4.  var gainNode = audioCtx.createGain();
5.  oscillator.connect(gainNode);
6.  gainNode.connect(audioCtx.destination);
7.  oscillator.type = 'sine';
8.  oscillator.frequency.value = 196.00;
9.  gainNode.gain.setValueAtTime(0, audioCtx.currentTime);
10. gainNode.gain.linearRampToValueAtTime(1, audioCtx.currentTime + 0.01);
11. oscillator.start(audioCtx.currentTime);
12. gainNode.gain.exponentialRampToValueAtTime(0.001, audioCtx.currentTime + 1);
13. oscillator.stop(audioCtx.currentTime + 1);
```

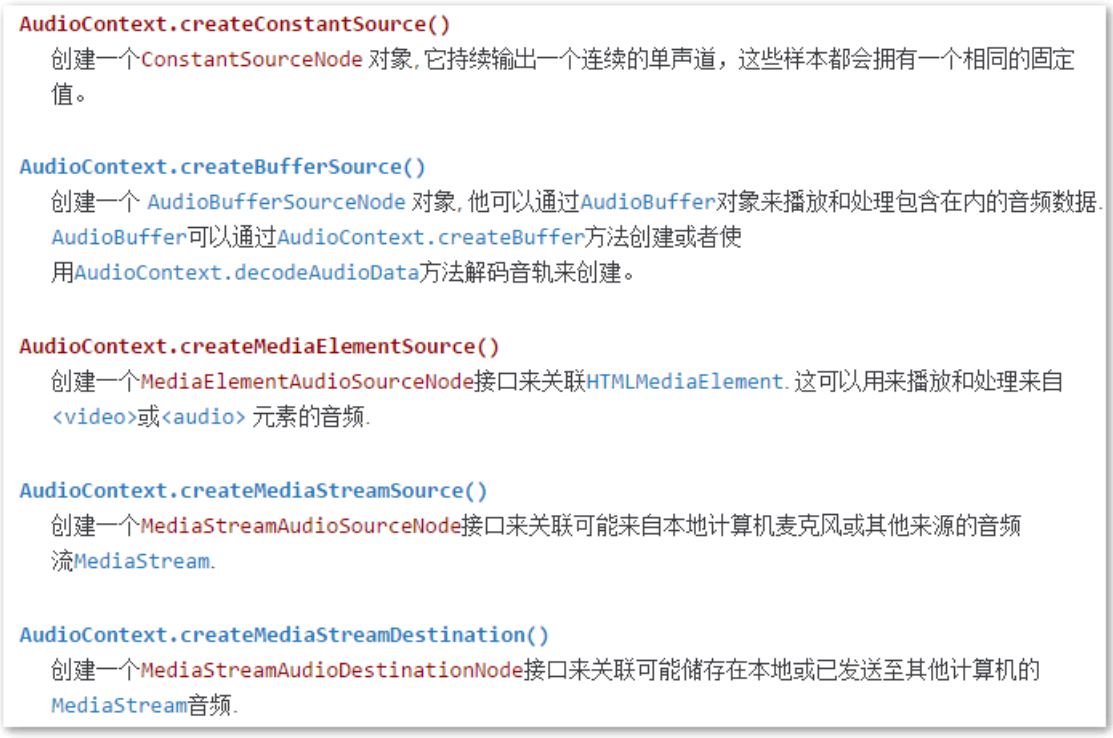
总共13行代码，虽然行数不多，但其中出现的API数量以及其中的含义还是有点厚度的。

我们一行一行来看。

1. window.AudioContext = window.AudioContext || window.webkitAudioContext
没啥好说的，向前兼容一下老的 `webkit` 浏览器。

2. var audioCtx = new AudioContext()

`AudioContext` 是HTML5 Web Audio最基础估计也是最常用的API了，其角色和地位可以和Canvas的 `canvas.getContext` 一拼。顾名思义，指定一个音频上下文，包含大量的属性和方法，而且这些方法名又多又长，根本就记不住，随便截个图感受一下  ：



所以，必须要有文档：MDN-AudioContext

但，不要怕，对于简单的音调播放，我们只需要关心下面几个属性和方法就好了，上面那么多高级的东西都是留给需要深入和音频打交道的开发人员用的。

3. var oscillator = audioCtx.createOscillator();

`createOscillator()` 是 `AudioContext` 的一个方法，创建一个OscillatorNode。 `OscillatorNode` 表示一个周期性波形，比如一个正弦波（就是上下弯弯的那个波），大家应该都知道声音本质其实就是震动，是和波形紧密关联的，因此波形不一样，最终的声音也就不一样，例如有的波形是“哔哔哔…”，有的波形出来的是“嘟呜呜…”。波还受频率影响，最终表现为不一样的音调高低。因此， `OscillatorNode` 就有 `frequency` 和 `type` 下面2个即将登场的属性。

总之，这里的变量 `oscillator` 基本上来说创造了一个音调。

4. var gainNode = audioCtx.createGain()

`createGain()` 也是 `AudioContext` 的一个方法，创建一个GainNode，它可以控制音频的总音量。也就是说 `createOscillator()` 控制音调， `createGain()` 控制音量，互相配合，黄金搭档。

`GainNode` 只有一个简单的只读属性，名为 `gain` ，完整书写 `GainNode.gain` ，本身没什么，但是 `GainNode.gain` 的返回值可是个大家伙，是个 *`a-rate`* 类型的AudioParam。

`AudioParam` 总共有2个类型，一个是 *`a-rate`* 还有一个是 *`k-rate`* ，前者 `AudioParam` 为音频信号每个样品帧的当前声音参数值；后者的 `AudioParam` 使用整个块（即128个样本帧）相同的初始音频参数值。

`AudioParam` 包含诸多属性和方法，都是与音量控制相关的。

5. oscillator.connect(gainNode)

音调和音量关联。

6. gainNode.connect(audioCtx.destination)

这里的`audioCtx.destination`返回AudioDestinationNode对象，`AudioDestinationNode`接口表示音频图形在特定情况下的最终输出地址 – 通常为扬声器。话句话说就是和音频设备关联。

这个关联为audio context中所有节点的最终节点，看下面的工作流图：

7. oscillator.type = 'sine'

我们的声音波形指定为`'sine'`，也就是正弦波，还支持其他3种波形，为`'square'`方波，`'triangle'`三角波以及`'sawtooth'`锯齿波。形状如下图（图来自css-tricks）

下面这个Codepen可以感受4中波形带来的哔哔啵啵的声音：

以上四种波形是内置的，我们还可以使用`setPeriodicWave()`方法自定义波形。

8. oscillator.frequency.value = 196.00

设置波形的频率，实际上，可以理解为设置声音的音调，也就是设置最后的声音是“多瑞米发嗦啦西”其中的一个。数值越小，声音越低沉，越大提琴；数值越大，声音越清脆，越小提琴。

demo页面中的音调完整范围如下：

```
[196.00, 220.00, 246.94, 261.63, 293.66, 329.63, 349.23, 392.00, 440.00, 493.88, 523.25,
587.33, 659.25, 698.46, 783.99, 880.00, 987.77, 1046.50]
```

每次鼠标hover我都会变换一个频率，这就是为什么每次经过按钮就会听到有规律的音调变化。

9. gainNode.gain.setValueAtTime(0, audioCtx.currentTime)

先来了解下`audioCtx.currentTime`，其值是以双精度浮点型数字返回硬件调用的秒数。只读，也就是说这个时间你是无法改变的，无论是暂停或者重置都不可以。当我们使用类似`new AudioContext()`创建`AudioContext`的时候，这个时间就可以从`0`开始走了，然后就像离弦的箭，一直不回头。

`gainNode.gain`这个前面介绍过了，返回的是`AudioParam`，所以这里实际上执行的是`AudioParam.setValueAtTime()`方法，此方法支持两个参数，一个是音量（范围0~1），一个时间，这里`setValueAtTime(0, audioCtx.currentTime)`表示当下就把音量设为`0`，也就是没声音，因为快速鼠标hover时候，之前声音可能还没播放结束，需要从`0`开始。

10. gainNode.gain.linearRampToValueAtTime(1, audioCtx.currentTime + 0.01)

`linearRampToValueAtTime()` 方法表示音量在某时间线性变化到某值，这里 `linearRampToValueAtTime(1, audioCtx.currentTime + 0.01)` 实际上表示的是在 `0.01` 秒的时间内，声音音量线性从 `0` 到 `1`。

11. oscillator.start(audioCtx.currentTime)

开始出声啦……

12. gainNode.gain.exponentialRampToValueAtTime(0.001, audioCtx.currentTime + 1)

`exponentialRampToValueAtTime()` 方法表示音量在某时间指数变化到某值，这里的 `exponentialRampToValueAtTime(0.001, audioCtx.currentTime + 1)` 实际上表示的是在 `1.00` 秒的时间内，音量由之前的 `1` 以指数曲线的速度降到极低的 `0.001` 音量。

我们把10, 12行代码联系起来，可能就是这样一个曲线（自己手绘的，凑合看吧）：

这样处理，声音消失就有“念念不忘，必有回响”的效果，就很有琴键按下去声音的感觉。

13. oscillator.stop(audioCtx.currentTime + 1)

1秒后，声音关闭。

最终，我们把源码和注释重新整合下就有：

```
// 创建音频上下文
var audioCtx = new AudioContext();
// 创建音调控制对象
var oscillator = audioCtx.createOscillator();
// 创建音量控制对象
var gainNode = audioCtx.createGain();
// 音调音量关联
oscillator.connect(gainNode);
// 音量和设备关联
gainNode.connect(audioCtx.destination);
// 音调类型指定为正弦波
oscillator.type = 'sine';
// 设置音调频率
oscillator.frequency.value = 196.00;
// 先把当前音量设为0
gainNode.gain.setValueAtTime(0, audioCtx.currentTime);
// 0.01秒时间内音量从刚刚的0变成1，线性变化
gainNode.gain.linearRampToValueAtTime(1, audioCtx.currentTime + 0.01);
// 声音走起
oscillator.start(audioCtx.currentTime);
// 1秒时间内音量从刚刚的1变成0.001，指数变化
gainNode.gain.exponentialRampToValueAtTime(0.001, audioCtx.currentTime + 1);
// 1秒后停止声音
oscillator.stop(audioCtx.currentTime + 1);
```

这下就好理解多了！

四、结束语

本文展示的交互音效仅仅是HTML5 Web Audio API众多特性中的冰山一角，但已经足够让广告性质类的网页熠熠生辉，甚至传统web站点在一些特殊场合也是可以尝试使用的。

最后，介绍几个和web audio相关的JS库，Pizzicato.js、webaudiox.js、howler.js、WAD、Tone.js，适合复杂音频处理场景。

HTML5 Web Audio API Chrome，Safari以及Firefox浏览器都是支持的，兼容性还是不错的。

参考文章：

- MDN – AudioContext
- 那些 audio api的事 (一) AudioContext
- Introduction to Web Audio API

感谢阅读！

> 《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？点击这里。有话要说？点击这里。

« -webkit-text-stroke文字描边CSS属性及展开

我是如何实现electron的在线升级热更新功能的？ »

猜你喜欢

- 使用wavesurfer.js显示mp3 audio音频的波形图
- 二次元live2d看板娘效果中的web前端技术
- HTML5 Battery电池状态相关API简介
- 翻译-你必须知道的28个HTML5特征、窍门和技术
- 翻译-10件Flash可以做而HTML5做不了的事情
- HTML5终极备忘大全（图片版+文字版）
- 基于HTML5 audio元素播放声音jQuery小插件
- HTML5 drag & drop 拖拽与拖放简介
- 观点：不要太依赖JavaScript库
- 渐进使用HTML5语言识别, so easy!
- 从天猫某活动视频不必要的3次请求说起

标签： API, audio, AudioContext, AudioParam, createGain, createOscillator, HTML5, 音效

发表评论（目前32条评论）

名称 (必须)

邮件地址(不会被公开) (必须)

网站

提交评论

1. **Nick说道：**

2018年12月30日 11:40

chrome 71 版本 默认关闭，需要用户有发生交互才能发出声音
https://developers.google.com/web/updates/2017/09/autoplay-policy-changes#webaudio

回复

2. **unnue.com说道：**

2018年11月26日 21:48

厉害。

点击评论者的头像，会直接当前页打开，建议新页面打开额。
也看个人习惯 ctrl + 点击。
来自小网站 unnue.com

回复

3. **稻草人说道：**

2018年09月25日 15:52

你好，现在的录音是基于两个按钮，《开始》《停止》来结束一段录音的，我们现在想实现一种类似长连接的方式的录音，即点击开始录音（有点像打开录音模式），保持录音状态，说一句，过了一会，又说一句，过了一会又说一句，能不能分别获取第一句，第二句，第三句的内容，即监测某句话是否说完了然后分别只获取发送这一句话的流内容呢？希望可以指导一下，谢谢啦

回复

4. **api挺强大说道：**

2018年04月17日 17:19

为什么有时候划过没声音(浏览器的标题上有个小喇叭的提示已经出来了)

回复

5. **ctj说道：**

2018年01月29日 17:45

我现在项目也有用到H5的这个录音，但是不知道为什么mac下的某些火狐会出来不能录音【或者说是录出的无声音频】的问题。
这是个在线的一个录音的地址，这个地址也有问题
https://webaudiodemos.appspot.com/AudioRecorder/index.html
一直没有找到是什么原因造成的，希望可以指导下，谢谢

回复

6. **小罗说道：**

2018年01月14日 14:13

我想说大神您真是什么都懂，最近在研究这个，一搜您的文就出来了，不说了API走起

回复

7. **jiafu**说道：
2017年12月7日 17:06

web还有这种操作，长见识了

回复

8. 来学习学习说道：
2017年11月2日 14:50

流劈

回复

9. 夜晚风景说道：
2017年10月15日 18:59

看到你的文章都是堪称经典！就是想问问有没有网页输入音频的办法！如果有的话希望能写一下给我们引路的文章！

回复

10. 小刚说道：
2017年07月22日 20:35

web这厉害了

回复

11. 小杰说道：
2017年07月7日 16:38

请问可以录音吗?

回复

12. **jcomey**说道：
2017年06月29日 15:03

gainNode.gain.exponentialRampToValueAtTime(0.00000000000000000000000000000000000001, audioCtx.currentTime + 1);试试~什么效果 。哈哈哈哈哈

回复

13. 刀刀说道：
2017年06月27日 18:01

哈哈哈。。可以在页面里写个钢琴自己玩了

回复

14. **PHP程序员雷雪松**说道：
2017年06月27日 16:07

HTML5功能还是很强大的！

回复

15. 当年明月说道：
2017年06月24日 18:30

学习了

回复

16. **wingmeng说道：**
2017年06月23日 09:42

Web上还能这么玩？继美学、数学后，Web又涉足音乐领域，感觉Web要一统天下了，哈哈

回复

17. 盖大楼说道：
2017年06月19日 14:39

下一期能不能写个关于 html嵌套规则的 文章

回复

18. 卷子说道：
2017年06月16日 15:47

第一次看到，长见识了，但是这个api能兼容到ie几？

回复

> 某某某说道：
> 2018年04月26日 17:57
>
> 好像IE全体不支持
>
> 回复

19. **nicholasurey说道：**
2017年06月15日 13:38

FF中并没有钢琴声啊。。。

回复

> **nicholasurey说道：**
> 2017年06月15日 13:47
>
> 是事件没触发。。。。
>
> 回复

20. 刘东奇说道：
2017年06月14日 10:00

是不是可以做一个html版的节奏大师？

回复

21. 盖大楼说道：
2017年06月14日 08:52

博大精深

回复

22. **###win7killer说道：**
2017年06月13日 14:18

https://win7killer.github.io/can_demo/demo/voice.html
借着大神的代码撸了一个钢琴键的demo，移动端还没处理。。。

回复

野原说道：
2017年08月29日 16:57

你真强大！！！

回复

> ivy说道：
> 2018年03月21日 17:40
>
> 想问一下， voiceList里面的内容，是怎么拿到这么精确的数字的啊？有什么转换的网站吗？
>
> 回复
>
> > 米每秒说道：
> > 2018年04月26日 18:10
> >
> > 手算。。。纯八度相差二分之一

何满子说道：
2018年01月9日 10:18

叼炸天

回复

23. 墨、水瓶说道：
2017年06月12日 11:11

这个点太前卫了，目前还没遇到这样站点。
PC端的站点运用的话，没有太明显的效果，多数用户不会连接声音输出设备。
移动端可能会好点吧 。

回复

24. 不二很纯洁说道：
2017年06月12日 09:47

之前只是做过可视化的，对创建音乐的没怎么弄明白，大神就是大神啊，清晰明了

回复

25. zhzh5724说道：
2017年06月12日 08:56

画个声音出来，
啥时候能画视频前端就一统天下了

回复

26. 曾小乱说道：
2017年06月11日 08:37

这个好酷，我要拿来改进一下我的网站

回复

最新文章
» 常见的CSS图形绘制合集
» 粉丝群第1期CSS小测点评与答疑
» 分享三个纯CSS实现26个英文字母的案例
» 小tips: 纯CSS实现打字动画效果

» CSS/CSS3 box-decoration-break属性简介
» CSS :placeholder-shown伪类实现Material Design占位符交互效果
» 从天猫某活动视频不必要的3次请求说起
» CSS vector-effect与SVG stroke描边缩放
» CSS ::backdrop伪元素是干嘛用的?
» 周知：CSS -webkit-伪元素选择器不再导致整行无效

今日热门

» 常见的CSS图形绘制合集 (190)
» 未来必热：SVG Sprite技术介绍 (119)
» 粉丝群第1期CSS小测点评与答疑 (115)
» HTML5终极备忘大全（图片版+文字版） (93)
» 让所有浏览器支持HTML5 video视频标签 (86)
» Selectivizr-让IE6~8支持CSS3伪类和属性选择器 (82)
» CSS3下的147个颜色名称及对应颜色值 (79)
» 视区相关单位vw, vh..简介以及可实际应用场景 (76)
» 写给自己看的display: flex布局教程 (76)
» 小tips: 纯CSS实现打字动画效果 (76)

今年热议

» 《CSS世界》女主角诚寻靠谱一起奋斗之人 (76)
» 不借助Echarts等图形框架原生JS快速实现折线图效果 (64)
» 看，for..in和for..of在那里吵架！ (60)
» 是时候好好安利下LuLu UI框架了！ (47)
» 原来浏览器原生支持JS Base64编码解码 (35)
» 妙法攻略：渐变虚框及边框滚动动画的纯CSS实现 (33)
» 炫酷H5中序列图片视频化播放的高性能实现 (31)
» CSS scroll-behavior和JS scrollIntoView让页面滚动平滑 (30)
» windows系统下批量删除OS X系统.DS_Store文件 (26)
» 写给自己看的display: flex布局教程 (26)

猜你喜欢

- 使用wavesurfer.js显示mp3 audio音频的波形图
- 二次元live2d看板娘效果中的web前端技术
- HTML5 Battery电池状态相关API简介
- 翻译-你必须知道的28个HTML5特征、窍门和技术
- 翻译-10件Flash可以做而HTML5做不了的事情
- HTML5终极备忘大全（图片版+文字版）
- 基于HTML5 audio元素播放声音jQuery小插件
- HTML5 drag & drop 拖拽与拖放简介
- 观点：不要太依赖JavaScript库
- 渐进使用HTML5语言识别, so easy!
- 从天猫某活动视频不必要的3次请求说起