



黑客派

程序员必备经验之从项目中来与到项目中去

作者: [zjhch123](#)

原文链接: <https://hacpai.com/article/1483553269734>

来源网站: 黑客派

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```
<h2 id="程序员必备经验之从项目中来与到项目中去">程序员必备经验之从项目中来与到项目中去</h2>
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
</script>
(adsbygoogle = window.adsbygoogle || []).push({});
</script>
<h2 id="0--前言">0. 前言</h2>
<p>很多朋友问我最近在忙什么？我很奇怪为什么他们能一眼就看出我在“忙”。后来仔细想想，这和我生活习惯很有关系。<br>平时自由散漫惯了，等到一有事情的时候就手忙脚乱，天天板着，看上去像天塌了一样。<br>哦不，至少对前几天的我来说，是“天就要塌了”。<br>多变的需，紧凑的时间，打的我措手不及。<br>恨不得一天掰成两天用，一台电脑劈成两台电脑用，一个人身成两个人用。</p>
<h2 id="1--从这个项目开始">1. 从这个项目开始</h2>
<p>几天前开始和小伙伴合伙做一个关于“新年计划”的项目。这是一个涉及到微信开发的项目，目前公众号是已认证的订阅号。</p>
<h2 id="2--多样化的需求是促使程序员进步的养料">2. 多样化的需求是促使程序员进步的养料</h2>
```

<p>程序员的确需要各种各样变换的需求来刺激自己的思维。各种各样看似“不可能”、“被 XX 限而无法实现”的需求，可能通过另一种方式或者其他形式的“黑科技”就可以实现。</p>

<h3 id="2-1-例一">2.1 例一</h3>

<p>客户需要实现的一个功能：在网页上获取到用户信息。这难倒了之前没有做过复杂微信开发的我在开发者文档里写着，订阅号没有“网页上授权获取用户信息”的权限，这可怎么办？
左思右，发现微信公众号里的关注用户具有唯一标识符"OpenId"，而后端又可以通过"OpenId"向微信服务获取到一些简单的用户信息，这样就可以绕过“授权”，直接获取到用户信息。
那么，现在问被简化成了如何获取到用户的 OpenId。用户的 OpenId 主要是在用户与公众号发生互动之后由腾讯服务器推送给后端的。那么需求可以再一次化简成“如何让用户与公众号发生互动”。
什么是互？腾讯这样定义互动“关注/取消关注/点击自定义菜单/向公众号发送消息”
关于“如何让用户与公众号发生互动”，就不仅仅得从一个程序员的角度去思考问题了。这里面需要考虑用户体验，用户习惯等各种问题。最终我们团队选择了“点击自定义菜单”和“关注”这两个互动。
此时，需求简化路线为：
“网页上获取到用户信息” -> “如何获取 OpenId” -> “让用户与公众号发生互动” -> “让用户点击自定义菜单/关注” -> “后端接收到微信的推送，根据 OpenId 获取用户信息”
而“根据 OpenId 获取用户信息”是明明白白的写在文档里的，非常容易实现。
这样一个看似无法实现的需求，就通过一步一步的分析得到了当前条件下最恰当的实现方式。</p>

<h3 id="2-2-例二">2.2 例二</h3>

<p>设计师的设计稿里有一个视频，是需要在首页，在原生 Web(不是弹出播放器)上自动播放。
然而微信有一个限制，不允许视频自动播放，视频的播放事件必须写在用户事件内。比方说，在 JS 码中</p>

```
<pre><code class="highlight-chroma">document.querySelector("video").play();
</code></pre>
```

<p>这样是失效的，视频不会播放。但是</p>

```
<pre><code class="highlight-chroma">document.querySelector("button").addEventListener("click", function() {
  document.querySelector("video").play()
});
</code></pre>
```

<p>这样是可以的，视频会播放。
那么，怎么将这个思路运用到项目里呢？
我们团队参了各种 H5 页面，发现许多 H5 页面都具有一个“点击开始”的按钮，点击之后会进行网页的各种动处理。此时，我们的灵感也随之浮现。
我们可以设计一个友好的“loading page”，假装正在理数据，处理完成后需要用户点击一下再继续展开页面。
总的来说，就是设计一个“友好”的

钮，欺骗用户去点击。用户点击之后只会以为自己进入了一个页面，而根本不会意识到，这个按钮其实是一个让视频播放的按钮。
 // plus: 用户的流量也随着这次点击哗哗的扣着~
 然而，这做的体验并不是很好。因为微信的限制，视频不能预先加载(其实微信这点做的很好，是为用户的流量虑的)，而是只有触发播放事件之后才会开始加载并播放。而对于我们的项目来说，我们需要视频能很畅的即时播放出来，这怎么办？
 经过分析，我们发现，在微信架构内，暂停的视频是会继续加的。这就给了我们很大的启示。我们的设计团队马上拿出了一套方案：“在 loading page 上点击了个按钮之后，先播放一段 3.7s 的 gif 动画，再开始播放视频。视频在这 3.7s 内进行加载”。
 终，我的实现代码大概是这样的。</p>

```
<script async src="https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js"></script>
<!-- 黑客派PC帖子内嵌-展示 -->
<ins class="adsbygoogle" style="display:block" data-ad-client="ca-pub-5357405790190342" data-ad-slot="8316640078" data-ad-format="auto" data-full-width-responsive="true"></ins>
<script>
  (adsbygoogle = window.adsbygoogle || []).push({});
</script>
<pre><code class="highlight-chroma">document.querySelector("button").addEventListener("click", function() {
  var video = document.querySelector("video")
  video.play();
  video.pause();
  // gif.play() 伪代码，表示gif播放
  // 加载一些资源
  setTimeout(function() {
    // 处理一些动效
    video.play();
  }, 3700)
});
</code></pre>
```

<p>看上去非常 zz...但是的确实现了预加载，在网络良好的情况下，视频也能较流畅的播放。
 求变化链是：
 “视频自动原生 Web 播放” -> “展示一个 page，用户点击按钮之后视频开始播放” -> “展示一个 page，用户点击之后播放 gif，此时视频开始加载，3.7s 之后开始播放视频”
 可以看出，设计/需求 并不是死的，很多被限制的功能完全可以通过另一种方式(设计/诱导/“欺骗”)完成。
 </p>

<blockquote>
 <p>例三例四例五...省略</p>
</blockquote>

<h2 id="3--及时吸取项目经验是程序员升级的阶梯">3. 及时吸取项目经验是程序员升级的阶梯</h2>

<p>项目是做完了就扔的吗？显然不是。从一个项目里我们往往可以学到很多。
 比方说，你学了 Java 的 SSH 三大框架，就能马上上手一个后台管理系统了吗？明显不能。因为此时你的项目经验足，换句话说，坑踩的少。而 往往坑踩的越少的人，越能挖坑。此时就需要你小项目开始着手，一步一步开始向大项目前进。
 而很多人说小项目学的东西少。我不这么认为做小项目往往是练手的时候，此时不应该把它当做是一个“练手小项目”，而应该把他当做一个“将会有 500W 人来使用的巨型项目”来制作，小到接口设计，大到架构设计，里面都藏满了无尽的学问坑)，等着你去挖掘(跳)。
 而在这个项目里，我确实确实的学到了许多关于微信开发的知识。之只做过一次基于订阅号的弹幕墙系统，从未做过认证订阅号的微信网页设计，所以在接项目的时候还有点担心有点紧张。
 然而在开发过程中，坑踩的多了，就会发现

 噫 无非就那样儿~
 </p>

<p>===</p>

<p>评论区伙伴有提到几种解决微信的播放限制的方案...但是无一例外全部失败。
 如果大家能

到更多的解决方案，不妨一起讨论一下! </p>