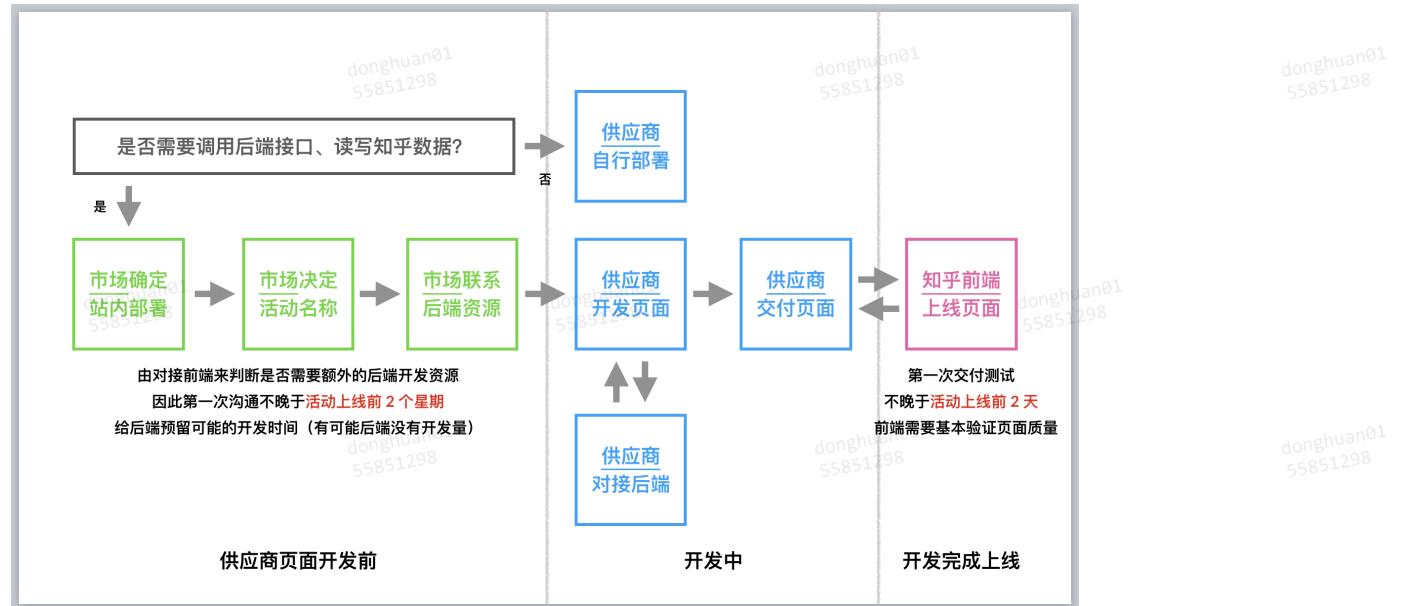


上线外包活动页（供应商阅读）

Created by 孙北吉-社区业务平台, last modified by 董欢-社区业务事业部 on Feb 18, 2025

本文介绍了由供应商制作的页面（或者行业内所称的 H5），在知乎 <https://event.zhihu.com> 上线与合作的过程。

知乎的老师可以将这份文档直接导出为 PDF 文件，交付供应商的老师作为参考。



- 合作流程
 - 开发中
 - 开发后与上线
- 对供应商的技术要求
 - 将引用的静态资源替换为 CDN 地址
 - 活动上线地址是同一域名下的子目录
 - 页面最终会部署在 HTTPS 协议下
 - 有 CSP 策略严格禁止引用知乎以外的资源
- 常见的活动页需求
 - 模拟活动页在知乎 App 内打开
 - 设置二次分享图片与文案
 - 常用端能力
 - 使用方式
 - 已开放端能力
 - 用户登录
 - 调用后端接口
 - 长按保存图片与识别二维码
 - 唤起知乎 App 内指定页面
 - 数据统计

合作流程

总体流程可以概括为以下 4 个阶段

1. 在 event-preview 调能力
2. 在对方服务器验收视觉效果
3. event 预部署、验收、测试，这个阶段不要泄露链接
4. event 上线，正式发布

开发中

独立和后端开发进行联调沟通

如果需要后端资源，且后端是独立资源，那么供应商老师需要单独、直接和后端老师进行沟通。

有时知乎前端也会承担提供后端接口的工作。

至少提前 2 天向知乎前端提供一版可以部署的版本

因为实际上线环境中：地址、CSP 策略、跨域等都会和本地开发有所不同，根据经验来看，**至少提前 2 天要向知乎前端提供一版**可以用来部署的版本比较稳妥。这一版本可以有很多设计细节并不确定。

开发后与上线

尽量减少不必要的上线需求

活动上线后几乎一定会有修改重新上线的需求。

知乎前端没有能力、也不会修改与维护供应商的代码。目前出现的修改活动代码的情形只局限在「修改非图片形式的文案」这一条上。

对供应商的技术要求

有四点需要注意，并会进行详细说明：

- 将引用的静态资源替换为 CDN 地址
- 活动上线地址是同一域名下的子目录
- 页面最终会部署在 HTTPS 协议下
- 有 CSP 策略严格禁止引用知乎外部资源

另外，前端不变更供应商代码，因此对供应商代码是否混淆压缩不作要求。原则上建议在不影响效果时压缩混淆。

将引用的静态资源替换为 CDN 地址

为了节约流量、提高响应速度。静态资源要求使用 <https://static.zhihu.com> 的 CDN 地址

要求：提前决定好活动名称

这将被用于 CDN 地址拼接中的第一级子目录。具体规则是：

上线在 <https://event.zhihu.com/foo/assets/style.rev.js> 可以通过 <https://static.zhihu.com/event/foo/assets/style.rev.js> 访问

要求：为静态资源生成 revision hash

因为 CDN 没有失效机制，所以要为静态资源文件名中增加版本 hash，比如 main.150c04697ca975963a3c.js

static.zhihu.com 上的静态资源会过滤 query，而且也不支持通过 query 进行穿透。比如通过 main.js?t=:timestamp 来获取到最新的静态文件版本。

活动上线地址是同一域名下的子目录

要求：引用静态资源时注意不要使用 / 开头的绝对路径

因为静态资源地址会被替换。这里可能不会遇到太大问题

要求：使用 localStorage 等本地储存信息时、需要添加 eventName 作为 namespace

所有 localStorage 的 key 需要统一用 eventName: 作为前缀，这是为了防止和其他活动冲突。

页面最终会部署在 HTTPS 协议下

要求：进行 API 请求或者引用静态文件时，使用 // 或者 https:// 作为前缀

因为 HTTPS 下无法请求 HTTP 协议内容

有 CSP 策略严格禁止引用知乎以外的资源

要求：将 jQuery 等第三方类库放到源码中

不要使用 unpkg 等其他 CDN 资源下的 jQuery、图标等静态资源

常见的活动页需求

模拟活动页在知乎 App 内打开

使用知乎 App 打开指定页面

在回答、文章或者想法内添加页面链接，就可以使用知乎 WebView 打开该页面，推荐在私信对话中填写链接最私密与方便。

这就是以后各种广告、推动等环境下打开活动页面的环境。

知乎 App 内的 UserAgent

使用上面的方法，访问 <https://sos.zhihu.com> 页面可以更加方便地查看到 UserAgent

iOS App 会增加 osee2unifiedRelease，Android App 会增加 Futureve，通过大小写严格进行匹配。

```
const isZhihu = Boolean(navigator.userAgent.match(/(osee2unifiedRelease|Futureve)/))
```

设置二次分享图片与文案

在微信内分享

因为页面部署在 zhihu.com 域名下，所以需要使用知乎的微信公众号 AppId 等进行微信内二次分享的设置。

示例代码如下。其中引入 zhstatic 脚本的代码必须放在 HTML 中 body 的尾部（因为涉及到对 document 的修改），后半部分实际设置逻辑可以在该脚本 load 后的任意地方执行：

```
<script src="https://zhstatic.zhihu.com/zWechat/zWechat.9b83b2d673c0bdd49070.js"></script>
<script>
var shareInfo = {
```

```

title: '没想到，我的室友竟然经历了.....', // 分享到朋友圈与微信好友时的标题
desc: '据说，看过的人 99% 都笑哭了', // 分享到微信好友的链接详情
imgUrl: 'https://zhstatic.zhihu.com/fly/duanzi/assets/images/share.jpg', // 分享到朋友圈与微信好友时的图片
link: location.href, // 分享的链接，可能会有增加追踪参数的需求
success: () => {
    // 微信 JS SDK 提供的分享成功回调，可以进行打点的操作
}
}

window.zWechat.init().then(function() {
    wx.onMenuShareTimeline(shareInfo) // 设置分享到朋友圈信息
    wx.onMenuShareAppMessage(shareInfo) // 设置分享到微信好友信息，如有需要，两者可以不同
})
</script>

```

分享到朋友圈：



donghuan01
55851298

donghuan01
55851298

分享给朋友：



donghuan01
55851298

donghuan01
55851298

在知乎 App 内分享

目前暂时无法设置知乎 App 内进行二次分享的图片与文案。分享到微信时会使用 document.title 作为分享的标题与描述，知乎 logo 作为图片

分享到朋友圈：1298



donghuan01
55851298

分享给朋友：



donghuan01
55851298

donghuan01
55851298

常用端能力

外包如果需要使用端能力，页面必须部署在 event.zhihu.com 和 event-preview.zhihu.com 下面

使用方式

通过 script 的方式引入知乎提供的 js sdk，目前最新版本：2.80.2

```
<script async src="https://unpkg.zhihu.com/zhihu-hybrid@2.80.2"></script>
```

引入后通过 window 上的 zhihuHybrid 对象进行相关操作，使用方式有两种，效果是一样的，返回 promise。推荐方式二

// 使用方式一

```
window.zhihuHybrid.dispatch(action,params)
```

// 使用方式二

```
window.zhihuHybrid(action).dispatch(params)
```

已开放端能力

base/supportAction

查询某个 action 是否被支持

参数与返回值

```
window.zhihuHybrid('base/supportAction').dispatch(params: Params): Promise<Result>
export type Params = {
    /** 模块名称 */
}
```

```

module: string
/** action 名称 */
action: string
}

export type Result = {
  /** 是否支持 */
  isSupported: boolean
}

```

示例

```
window.zhihuHybrid('base/supportAction').dispatch({module:'base',action:'supportAction'})
```

share/setShareInfo

设置分享文案

参数与返回值

```
window.zhihuHybrid('share/setShareInfo').dispatch(params: Params): Promise<Result>
```

```

type ZhihuMessageInfo = {
  /** 私信内容 */
  content?: string
  /** 分享链接 */
  link?: string
}

```

```

type WechatTimelineInfo = {
  /** 分享标题 */
  title: string
  /** 分享链接 */
  link?: string
  /** 分享图标 */
  imgUrl?: string
}

```

```

type WechatMessageInfo = {
  /** 分享标题 */
  title: string
  /** 分享描述 */
  desc?: string
  /** 分享链接 */
  link: string
  /** 分享图标 */
  imgUrl?: string
}

```

```

type QQInfo = {
  url?: string
  title?: string
  content?: string
  imageURL?: string
}

```

```

type QzoneInfo = {
  url?: string
  title?: string
  content?: string
  imageURL?: string
}

```

```

type WeiboInfo = {
  url?: string
  title?: string
  content: string
  imageURL?: string
}

```

```

type PinInfo = {
  url?: string
  content?: string
  imageURL?: string
}

```

```

type CopyLinkInfo = {
  /** 复制到剪贴板中的内容 */
  content?: string
}

```

```

type PosterShareInfo = {
  /** 图片 url */
  imageURL?: string
  /** 调起想法编辑器时预设的内容，是 JSON 字符串 */
  /** 例如: JSON.stringify({html:`<p><a class="hash_tag" data-topic-id="19776749">#话题名称#</a></p>`}) */
  /** html 的值参考 https://wiki.in.zhihu.com/pages/viewpage.action?pageId=331711456 */
  pinContent?: string
  /** 将页面作为分享海报 */
  web_poster?: {
    /** 页面 html，由前端提供 html 而不是由客户端通过页面 url 获取页面 html，因为这样生成海报图片更快 */
    web_html: string
    /** 海报尺寸 */
    web_size: {
      width: number
    }
  }
}

```

```

        height: number
    }
    /** 海报圆角 */
    corner_radius: number
}
}

export type Params = {
    /** 私信 */
    zhihuMessage?: ZhihuMessageInfo
    /** 微信朋友圈 */
    wechatTimeline?: WechatTimelineInfo
    /** 微信消息 */
    wechatMessage?: WechatMessageInfo
    /** QQ */
    QQ?: QQInfo
    /** QQ 空间 */
    Qzone?: QzoneInfo
    /** 微博 (iOS 分享到微博不支持带上配图) */
    weibo?: WeiboInfo
    /** 无需配置，配置了客户端 (iOS 全部版本、Android 10.42) 也不会出现该分享渠道，目前暂不支持分享到想法 */
    pin?: PinInfo
    /** 复制链接 (iOS 4.27、Android 4.28 新增) */
    copyLink?: CopyLinkInfo
    /** 分享图片 (iOS 6.63、Android 6.63 新增) */
    PosterShare?: PosterShareInfo
    /** 透传内容类型和 id 给客户端 (iOS 8.27、Android 8.27 新增) */
    type?: String
    id?: String
}

export type Result = void

```

share/showShareDialog

唤起分享引导弹窗

依赖 share/setShareInfo 中设置的信息

参数与返回值

```

window.zhihuHybrid('share/showShareDialog').dispatch(params: Params): Promise<Result>
export type Params = {
    /** 分享弹窗标题 */
    title: string
    /** 分享弹窗描述 */
    description: string
    /** 分享弹窗头图 */
    imgUrl: string
    /** 分享渠道 */
    channels: (
        | 'wechatMessage'
        | 'wechatTimeline'
        | 'weibo'
        | 'QQ'
        | 'Qzone'
        | 'pin'
    )[]
}

export type Result = void

```

share/shareTo

直接触发分享

参数与返回值

```

window.zhihuHybrid('share/shareTo').dispatch(params: Params): Promise<Result>
export type Params = {
    /** 分享方式 */
    type:
        | 'zhihuMessage'
        | 'wechatTimeline'
        | 'wechatMessage'
        | 'QQ'
        | 'Qzone'
        | 'weibo'
        | 'pin'
        | 'PosterShare'
    /** sub_types 只在 type 为 PosterShare 时生效 */
    sub_types?: (
        | 'zhdb' // 转发到想法
        | 'wechatMessage'
        | 'wechatTimeline'
        | 'QQ'
        | 'weibo'
        | 'saveAlbum'
    )[]
}

export type Result = void

```

share/showShareActionSheet

唤起分享面板

参数与返回值

```
window.zhihuHybrid('share/showShareActionSheet').dispatch(params: Params): PromiseObservable<Result>
export type Params = void
export type Result = void
```

share/actionSheetShareClick

唤起分享面板

参数与返回值

```
'window.zhihuHybrid('share/actionSheetShareClick').listen((params: Params) => void): Function
export type Params = {
  /** 类型 */
  type: string
  type: 'QQ'
  type: 'QQ 空间'
  type: '新浪微博'
  type: '微信好友'
  type: '朋友圈'
  type: '海报分享+微信好友'
  type: '海报分享+朋友圈'
  type: '海报分享+QQ'
  type: '海报分享+分享到想法'
  type: '海报分享+新浪微博'
  type: '知乎私信'
}
```

base/downloadImage

下载图片

参数与返回值

```
window.zhihuHybrid('base/downloadImage').dispatch(params: Params): PromiseObservable<Result>
export type Params = {
  /** 图片 url */
  url: string
}
export type Result = void
```

用户登录

判断登录状态与获取用户信息

通过请求 GET <https://www.zhihu.com/api/v4/me> 来判断当前是否登录是登录状态，注意发送 ajax 请求时要设置 withCredentials 为 true 带上 cookie。

目前这个接口请求有跨域问题，可能需要伴随活动进行处理。

zepto 代码:

```
$.ajax({url: 'https://www.zhihu.com/api/v4/me', xhrFields: {withCredentials: true}})
```

知乎 App 外

使用跳转到登录页并返回的方式进行登录：

```
location.href = `https://www.zhihu.com/signin?next=${encodeURIComponent(location.href)}`
```

知乎 App 内

目前没有办法在知乎 App 内由活动页主动唤起进行登录操作。

调用后端接口

和对接后端一起解决跨域问题

<https://event.zhihu.com> 的线上环境和绝大部分知乎后端 API 都处于跨域状态。

- 在开发过程中遇到的跨域问题（比如本地运行的开发环境 localhost:3000 与 API 跨域），由供应商开发自行解决
- 在上线后遇到的跨域问题，由供应商和对接后端一起协商解决。

建议对接后端采用严格的跨域策略，来限制接口只被活动页正常调用。

长按保存图片与识别二维码

使用 canvas 制作可以保存的图片

知乎站内其他地址的图片，比如头像、回答插图等位于 pic.zhimimg.com 域名下。这些知乎站内图片的 CDN 地址均设置了跨域头（Access-Control-Allow-Origin 为 *）。

因此 canvas 在插入了这些图片后不会成为 tainted canvas、也支持 toBlob()、toDataURL() 与 getImageData() 进行导出，进而支持设置在 img 上长按保存头像等设计。

但是你需要为 img 特别指定 crossOrigin 属性为 anonymous（对于 img 标签是 crossorigin attribute），具体参考：https://developer.mozilla.org/en-US/docs/Web/HTML/CORS_enabled_image

知乎 App 外
视浏览器情况而定

知乎 App 内

知乎 App 没有长按识别二维码功能。

知乎 Android App 没有长按保存图片功能，即使是使用 img 标签声明的图片。

知乎 iOS App 有长按保存图片功能，需要用 img 而不是 background-image 方式显示。

唤起知乎 App 内指定页面

知乎 App 外

跳转到 <https://oia.zhihu.com> 开头的 URL 即可自动根据平台（兼容微信）来唤起 App。具体链接需要和对接知乎前端确定，例如：

```
location.href = 'https://oia.zhihu.com/answers/522333613'
```

注意链接中的大小写。

知乎 App 内

直接跳转到对应页面的网页版 URL 就会被客户端拦截进入到指定页面。具体链接需要和对接知乎前端确定，例如：

```
location.href = 'https://www.zhihu.com/question/300652747/answer/522333613'
```

如果简单处理、在 App 内也跳转到 oia.zhihu.com 会导致连续发生两次跳转，体验不是很好。建议先判 UA 环境再进行对应跳转

数据统计

使用第三方平台进行页面数据统计

直接在页面中嵌入有关平台的统计代码即可。注意 event.zhihu.com 设置了 CSP 策略，所以引入的第三方统计代码可能不会生效，需要和对接知乎前端一起调试。

目前支持的第三方统计平台包括：

- 百度 (hm.baidu.com)
- 腾讯 MTA (<https://mta.qq.com/>)

使用知乎 ZA 平台进行页面数据统计

知乎 ZA 平台 npm 包地址：<https://www.npmjs.com/package/za-js-sdk>

ZA 是知乎内类似 Google Analytics 的一套打点系统，可以记录 PV、事件等等。

从 npm 引入（推荐）

```
import za from 'za-js-sdk'
```

通过 script 标签引入

```
<script src="https://unpkg.zhimg.com/za-js-sdk@4.17.2/dist/loader.js"></script>
```

使用方式

```
// 一定要覆盖一下
const za = new window.zap.Client({useProto3: true})

// 页面曝光
za.trackPageShow()

// 元素曝光
za.trackShow({
  elementLocation: {
    content: {
      type: 'Answer',
      token: '223'
    },
    moduleId: '21',
    moduleIndex: 0
  },
  {
    link: {url: 'www.zhihu.com'}
  }
})
```

```

    }

// za.trackShow(ViewInfo, ExtraInfo)

// 交互点击
za.trackEvent({
  action: 'openUrl',
  elementLocation: {
    content: {
      type: 'Zvideo',
      token: '112'
    },
    moduleId: 'id1',
    moduleIndex: 0,
    text: '文',
    type: 'Button',
  },
  {
    link: {url: 'www.zhihu.com'}
  }
)
// za.trackEvent(ViewInfo, ExtraInfo)

```

ViewInfo.action 说明

action	string	交互意图是什么。例如鼓掌、点赞等 Unknown OpenUrl Close PageDisappear ModuleDisappear Upvote UnUpvote Downvote UnDownvote Thank Pin UnThank NoHelp UnNoHelp Follow UnFollow Co UnBlockUser Applaud UnApplaud Danmaku Save Upload Post Video ClubPost JumpOutside Purchase Download Grade UnPurchase ReadFinished UnIgnore Signin SignUp Forward Gift Statement ReferView

ViewInfo.elementLocation 说明

content	type	string (Type)	业务场景类型，如回答、视频实体、文章、想法等
	id	string	使用 id 还是 token 详见：视频实体、想法、直播用 id，其他用 token
	token	string	
block	/	string	旧 - 用于区分不同区块（可使用 moduleId、moduleIndex 代替）
moduleId	/	string	元素模块标识，可以为任意值
moduleIndex	/	number	元素模块标识，从 0 开始
text		string	
type		string	必填.Unknown Button Text Image Page Card Audio Video Block Popup File Input

ExtraInfo 说明

名称	类型	必填	说明
link.url	string	false	元素标识，可以为任意网址

一个demo**查看埋点**

当前页面控制台输入 `localStorage.setItem('zap:logenabled', '1')`，刷新页面

No labels

欢迎查阅 [《公共技术文档》](#) | [《知乎技术雷达》](#)



donghuan01
55851298

donghuan01
55851298