

斐波那契 SDK 接入文档

目录

[斐波那契 SDK 接入文档](#)

[目录](#)

[更新日志](#)

[正文](#)

[1. SDK 使用步骤](#)

[1.1 步骤一：引入 JS 文件](#)

[1.2 步骤二：通过 init 接口初始化](#)

[2. SDK 接口](#)

[2.1 微信分享事件](#)

[2.2 按钮点击事件](#)

[2.3 自定义表单推送](#)

[2.4 翻页事件](#)

[2.5 操作结果事件](#)

[2.6 处理 url 参数](#)

更新日志

- 2017 年 02 月 18 日 完善基础接入步骤
- 2018 年 06 月 20 日 init 增加 apptype, unionid, userid 字段
- 2018 年 07 月 24 日 增加 pfid 权限 function 字段
- 2019 年 06 月 29 日 增加 saudid 字段处理逻辑
- 2020 年 02 月 21 日 增加 identitys、platform 字段处理多客户身份逻辑

正文

1. SDK 使用步骤

1.1 步骤一：引入 JS 文件

将下面的代码加在<body>中，在调用斐波那契 SDK 接口之前加载。

```
<script
  type="text/javascript"
  src="https://res.fibodata.com/data/fibosdk.min.js"
></script>
```

1.2 步骤二：通过 init 接口初始化

为了能够正常使用 SDK 接口，必须首先调用 init 接口进行初始化（建议接入微信网页授权，并在成功之后调用），

init 接口返回值为一个Promise对象。

注意事项：请确保编码格式为utf-8，以保证传入的授权信息不会出现乱码。

```
var fibo = fiboSDK.init({
  pfid: "", // （必须）斐波那契分配的平台id
  appid: "", // （必须，1-50位，数字，大小写字母，下划线，短横线）场景载体id，
  一般指作品id
  apptype: "xapp", // （必须）应用类型，xapp
  mpid: "", // （非必须参数，如果有微信授权，则必须填入）建议填绑定了兔展企业的
  公众号的appid
  openid: "", // （非必须参数，如果有微信授权，则必须填入）微信公众号获取的用户
  openid
  unionid: "", // （非必须参数，如果有微信授权且有unionid，则必须填入）微信公
  众号获取的用户unionid
  userInfo: {}, // （非必须，如果有微信授权并能拿到用户信息，则必须填入）微信
  公众号网页授权获取的用户信息JSON对象
  title: "", // （非必须）页面标题，因为在微信 pc 端内置浏览器使用 document.t
  itle 获取 title 会异常，所以建议直接传入
  identitys: [{identityType: "", identityValue: ""}], // （非必须）客户多身
  份类型标识，可以传入多组
  platform: '' // （非必须）接入平台来源类型
});

// init 接口的返回值为一个Promise对象，可以调用 then 方法实现相应的业务逻辑。
fibo.then(function(initData) {
  // 分享链接，将需要分享的链接传入fiboSDK.dealUrl()
  var shareLink = fiboSDK.dealUrl(location.href);
  ...
}, function(err){

});
```

2. SDK 接口

2.1 微信分享事件

使用场景：

1. 接入后可以看到作品的传播途径，微信朋友圈、微信朋友消息等各种方式所占比例
2. 作品在各个时间段的分享率
3. 传播的层级，各层级所占比例

如果需要监测作品在微信生态的传播，需要引入微信的 JS-SDK。具体的引入方式可以直接查看[微信 JS-SDK 说明文档](#)

在微信定义的几个分享接口中，对分享链接 link 和分享成功回调函数 success 接入斐波那契监测代码，例如分享到朋友圈：

```
wx.onMenuShareTimeline({
  title: "", // 分享标题
  // 分享链接，将需要分享的链接传入fiboSDK.dealUrl()
  // 如果 shareurl 是当前页面的链接，请务必保留链接上可能存在的特定参数，
  // 如：actid、stepid、cnl、cnltype、lc、sui
  // 如果 shareurl 是由程序生成的其它链接，请将当前页面链接上可能存在的
  // 参数（actid、stepid、cnl、cnltype、lc、sui）也拼接到此链接上
  link: fiboSDK.dealUrl(shareurl),
  imgUrl: "", // 分享图标
  success: function() {
    //用户确认分享后执行的回调函数，需要调用fiboSDK.share()并传入对应的分享渠道
    fiboSDK.share("timeline");
  },
  cancel: function() {
    // 用户取消分享后执行的回调函数
  }
});
```

分享渠道和需要传入 fiboSDK.share()的值对应如下：

分享渠道	对应的值
朋友圈	timeline
朋友	friend
QQ	qq
腾讯微博	txweibo
QQ 空间	qzone

2.2 按钮点击事件

使用场景：通过接入按钮点击事件，可以获取到按钮的点击次数和转化率。

对于需要记录点击次数的按钮或者 a 标签里的跳转链接，需要在待点击的元素上绑定 click 事件并调用 `fibosdk.btnClick(buttonId,buttonName)`，其中参数 `buttonId` 为按钮识别 id，`buttonName` 是该按钮的名称，例如：

```
<a
  href="http://www.baidu.com"
  onclick="fibosdk.btnClick('button1', '跳到百度')"
></a>
```

或者：

```
<script>
  function buttonClick() {
    // 登录流程...
    fibosdk.btnClick("button2", "登录按钮");
  }
</script>
<button onclick="buttonClick()">登录</button>
```

如果需要统计多个不同按钮的点击次数，`buttonId` 必须不同。如果两个不同的按钮传入同一个 `buttonId`，会算成同一个按钮的点击事件。

为了防止 `buttonId` 与其他作品的冲突，导致统计混乱，建议使用“appid-buttonId”的形式作为 `buttonId` 传入。

2.3 自定义表单推送

使用场景：推送自定义表单后，可以及时获取收集到的表单数据。

对于需要提交表单信息的作品，可以推送表单信息，表单的格式必须是键值对。

注意事项：表单的键值都必须是字符串。

使用 `saveFormInfo(formInfo, version)` 接口。

```
var form = { "姓名": "张三", "手机": "13512345678" };

fibosdk.saveFormInfo(form, "1");
}
```

2.4 翻页事件

使用场景：通过在翻页及离开页面时接入 SDK 方法，可获取各页面停留时间，流失情况统计，得到用户最有兴趣的页码和最没兴趣的页码。

对于有分页的 H5 作品，在每次翻页时调用 `switchPage()` 方法，例如：

```
fiboSDK.switchPage(currentPage, targetPage, totalPage);
```

其中 currentPage 是当前页码，targetPage 是目标页码，totalPage 是总页数。currentPage 需要从 1 开始，否则方法调用失败。

在退出页面时调用 exitPage()方法，若非浏览器行为跳转页面（如手动更改 location.href 的指向），则需手动调用 exitPage，例如：

```
window.onbeforeunload = function() {  
    fiboSDK.exitPage(currentPage);  
};
```

其中 currentPage 是当前页码。

2.5 操作结果事件

使用场景：用户在页面内完成某项操作，或调用外部接口完成操作，当下无法马上记录此操作带来的结果/状态变化，需要等服务器返回数据后，前端得到数据再传输给斐波。

例如：邀请有奖场景中，用户参与活动填写并提交表单后，后端验证并返回获奖信息时，前端调用此事件

```
var opInfo = { operation: "getAward" };  
fiboSDK.opState(opInfo);
```

其中 opInfo 是存储操作结果信息的容器对象，支持任意传入任意键值对（方便后期的字段扩展），但具体要传到 fibo 的有效字段由 opState 的代码内部进行确定。目前确定的有效字段有：

- operation
- ownerOpenid

2.6 处理 url 参数

使用场景：用户在分享时，处理分享链接，添加或修改 url 中的 lc、sui、cni、cnltype 等参数。

注意事项：在调用此方法前，一定要保证已经成功的调用 fiboSDK.init 接口完成了SDK的初始化，如果分享链接是当前页面的链接，请务必保留链接上可能存在的特定参数，如：actid、stepid、cni、cnltype、lc、sui；

如果分享链接是由程序生成的其它链接，请将当前页面链接上可能存在的参数（actid、stepid、cni、cnltype、lc、sui）也拼接到此链接上。

- lc 从 0 开始(int) 层级参数
- cni 渠道 id (使用渠道分发时，会在链接中)
- sui 分享来源者的 viewid，由前端 sdk 从作品 url 中获取
- 请务必保留原链接上可能存在的特定参数，如：actid、stepid、cni、cnltype、lc、sui

例如：在处理作品的分享链接时

```
// shareLink = http://www.xxx.com/a/b?lc=0&sui=2&cnl=123&cnltype=CHANNEL  
var link = fiboSDK.dealUrl(shareLink);  
// 设置为分享链接  
// ...
```