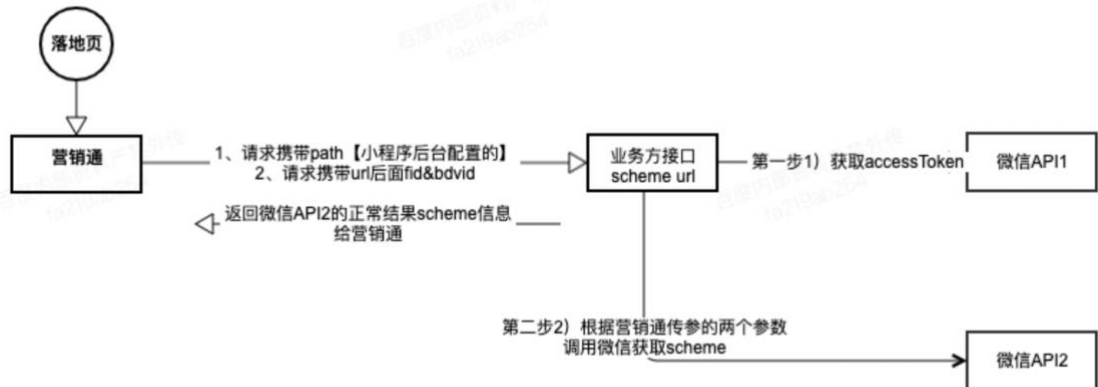


营销通微信组件全场景唤起小程序说明文档

营销通全场景唤起小程序需要依赖业务方对接微信小程序 API 开发，提供接口给营销通平台调用。具体解决方案如下图：



微信 API1:

<https://developers.weixin.qq.com/miniprogram/dev/api-backend/open-api/access-token/auth.getAccessToken.html>

微信 API2:

<https://developers.weixin.qq.com/miniprogram/dev/api-backend/open-api/url-scheme/urlscheme.generate.html>

注意：对接微信 API2 时，微信文档中有个参数 `jump_wxa` 对象，该对象的取值，依赖营销通请求时携带的 `path` 以及 `query` 传入。

`jump_wxa.path` 是指营销通后台，新建小程序信息的 `path` 地址（地址中如果有问号，就是问号前部分字符串），如下图：

新增小程序授权

“全场景调起”可在所有浏览器内调起微信小程序，需开发接口与百度对接，如您不具备技术开发能力，可选择“百度APP内调起”填写固定内容授权小程序，仅支持百度APP内调起微信小程序

调起小程序类型 全场景调起 百度APP内调起

请务必提前阅读[说明文档](#)

* 小程序名称 0/40

* 小程序原始ID

path地址

* 获取scheme URL地址

jump_wxa.query 是指跳转到小程序地址中，业务方希望接收到的参数信息。目前有的业务方希望获取 fid&bdvid，这部分信息都会作为 query 词在调用业务方的 urlScheme 地址的时候传递过去，业务方将营销通平台传过来 path，query 直接赋值到需要调用微信 API2 接口中作为传参。

同时：为防止恶意用户攻击获取 urlScheme 信息，建议接入时，提供给营销通平台的 urlScheme 地址保证有以下逻辑的验签机制。如果验签不通过。则不返回数据，如果验签通过，则返回对应结果信息，具体验签机制如下（**为了您的网站安全，请一定保证有验签机制获取数据**）：

1、请求说明

请求方式	请求数据类型	响应数据类型	接口状态
------	--------	--------	------

POST (application/json)	JSON	JSON	有效
名称	获取 scheme 接口		
接口地址	https://example.com/getScheme (示例地址, 请用自己的地址配置在营销通后台)		

参数名称	参数类型	是否必须传	默认值	备注
originId	string	是		小程序原始 Id
currentTime	long	是		请求当前时间戳 (毫秒)
sign	string	是		签名
path	string	是	“ ”	调用微信 API2 需要的 jump_wxa.path 【没有加入验签】
query	string	是	“ ”	调用微信 API2 需要的 jump_wxa.query 【没有加入验签】

2、签名及校验规则说明：

```
String str=" currentTime=123&originId=123"
```

```
sign= DigestUtils.md5Hex(str.toLowerCase())
```

参数 currentTime 以及 originId 参与验签，对字符串进行 MD5 加密，接口中对 sign 进行签名验证，并对 currentTime 当前时间限制，请求时间和 currentTime 不能相差超过 5s；

3、返回结构体表格：

参数名称	参数格式	是否必有	描述
code	number	是	请求编码, 0:成功
msg	string	是	异常信息

result	Object		数据
scheme	string	是	微信 api2 返回的 openlink

4、补充说明

4.1 加密验签 sign 生成代码 java 示例:

```
private String getSign(String originId) {

    String sign = "";

    Long time = System.currentTimeMillis();

    String str = "currentTime=" + time + "&originId=" + originId;

    sign = DigestUtils.md5Hex(str.toLowerCase());

    return sign;

}
```

originId: 小程序原始 id

将此方法返回的 sign 与营销通请求的 sign 进行一致性对比。如果不匹配，则验签失败，不用返回数据。

4.2 获取 urlScheme 地址必填，返回格式一定要参照“3、返回结构体表格”，返回结果示例:

```
{
    "code": 0,
    "msg": "success",
```

```
"result": {  
  "scheme": "weixin://dl/business/?t=KoPsO5525Ai"  
}
```

4.3 测试接口，完成信息填写后可进行接口测试，测试接口是结合客户输入的地址，小程序原始 id，以及 path，去请求 scheme URL，获取信息。请求结果会有以下三种情况：

情况 1，成功对接：可以正常获取到 scheme。则会显示下图提

示



情况 2，对接失败：访问不通，无法访问（请求业务方配置地址不通）时，需业务方研发同学确认 urlScheme 地址信息是否配置正确



情况 3，对接失败（如果不是情况 1 或者 2 那只能是情况 3）：

可以正常访问通，但是获取 scheme 失败，则提示显示的直接就是

【业务方接口返回的原样数据信息】，需业务技术根据此接口文档

仔细自查接口逻辑【以下图片只是一个示例】



5 对接微信生成 urlScheme 示例代码:

5.1 请求接收对象

```
public class WechatSchemeReq {
```

```
    private String path;
```

```
    private String query;
```

```
    private String originId;
```

```
    private Long currentTime;
```

```
    private String sign;
```

```
}
```

5.2 返回对象

```
public class SchemeDto {
```

```
    private int code;
```

```
    private Result result;
```

```
private String msg;

@Data

public static class Result {

    private String scheme;

}

}
```

5.3 请求微信 API 需要对象

```
public class JumpInfo {

    private String path;

    private String query;

}
```

5.4 对接微信代码示例，仅做参考

```
public SchemeDto getSchemeTest(@RequestBody WechatSchemeReq req) {

    SchemeDto schemeDto = new SchemeDto();

    // 第一步 校验时间是否合法
```



```
Long now = System.currentTimeMillis();

if (req.getCurrentTime() == null || (now - req.getCurrentTime()) > 5000) {

    schemeDto.setCode(1001);

    schemeDto.setMsg("time expire");

    return schemeDto;

}

// 第二步 校验 originId 是否是业务方自己的

String originId = "gh_2290219a636"; // 此处是业务方自己的小程序原始 id

if (!originId.equals(req.getOriginId())) {

    schemeDto.setCode(1002);

    schemeDto.setMsg("origin error");

    return schemeDto;

}

// 第三步 校验 sign 是否合法

String str="currentTime="+req.getCurrentTime()+"&originId="+req.getOriginId();

String sign= DigestUtils.md5Hex(str.toLowerCase());

if (!sign.equals(req.getSign())) {

    schemeDto.setCode(1003);

    schemeDto.setMsg("sign error");

    return schemeDto;

}
```

// 第四步 调用微信 API1 获取 accessToken

```
String accessToken = "";
```

```
try {
```

```
    Map<String, String> httpParamMap = new HashMap<>();
```

```
    httpParamMap.put("grant_type", "client_credential");
```

```
    httpParamMap.put("appid", "wxfe9c714dd54f3058"); // 请填写业务方自己的
```

```
    httpParamMap.put("secret", "032a8e691d9492078103eb39e514b0b11"); // 请填写业务方自己
```

的

```
    String response = HttpUtils.getRequest("https://api.weixin.qq.com/cgi-bin/token",
```

```
    httpParamMap);
```

```
    JSONObject object = JSON.parseObject(response);
```

```
    if (object.containsKey("access_token")) {
```

```
        accessToken = object.getString("access_token");
```

```
    }
```

```
    } catch (Exception e) {
```

```
        log.error("sendMessage error.", e);
```

```
    }
```

// 第五步 调用微信 API2 获取 scheme

```
String host = "https://api.weixin.qq.com/wxa/generatescheme?access_token=" + accessToken; //
```

accessToken 为第四步调用微信的结果

```
Map<String, Object> map = new HashMap<>();
```

```

JumpInfo jumpInfo = new JumpInfo();

jumpInfo.setPath(req.getPath());

jumpInfo.setQuery(req.getQuery());

map.put("jump_wxa", jumpInfo);

map.put("expire_time", (System.currentTimeMillis() / 1000) + 24 * 60 * 60);

map.put("is_expire", true);

String result = HttpUtils.sendJsonHttpPost(host, JSONObject.toJSONString(map));

JSONObject resultJsonObject = JSON.parseObject(result);

if (resultJsonObject != null &&

    resultJsonObject.containsKey("errcode") && resultJsonObject.get("errcode").equals(0) {

    if (resultJsonObject.get("openlink") != null) {

        String scheme = resultJsonObject.get("openlink").toString();

        schemeDto.setCode(0);

        schemeDto.setMsg("success");

        SchemeDto.Result r = new SchemeDto.Result();

        r.setScheme(scheme);

        schemeDto.setResult(r);

        return schemeDto;

    }

}

```

//如果走到这里 没有拿到微信的结果 辛苦分析您微信结果的返回

```
schemeDto.setCode(1005);
```

```
schemeDto.setMsg("error");
```

```
return schemeDto;
```

```
}
```