

移动应用钉钉登录开发指南

准备工作

移动应用钉钉登录是基于[OAuth2.0协议标准](https://oauth.net/2/)(https://oauth.net/2/) 构建的钉钉OAuth2.0授权登录系统。

在进行钉钉OAuth2.0授权登录接入之前，先邮件dingtalk-cooperation@list.alibaba-inc.com申请应用免登的一些关键字段，并获得相应的AppID和AppSecret，申请钉钉登录且通过审核后，可开始接入流程。邮件申请需要准备的物料如下：

名称：钉钉免登DEMO

描述：钉钉免登DEMO是钉钉为开发者提供的免登的参考APP。

ICON：小于48KB

Android包名称：com.dingtalk.com.test

Android签名：c70442be6e53b04ba8b22468d9ba2ac5

如何生成签名?(<https://open-doc.dingtalk.com/docs/doc.htm?spm=a219a.7629140.0.0.rAdTef&treeId=178&articleId=105495&docType=1>)

iOS BundleId：com.dingtalk.com.test

1. 目前移动应用上钉钉登录只提供原生的登录方式，需要用户安装钉钉客户端才能配合使用。
2. 对于Android应用，建议总是显示钉钉登录按钮，当用户手机没有安装钉钉客户端时，请引导用户下载安装钉钉客户端。
3. 对于iOS应用（`文档待补充,让ios同学补充后，再开放`）

授权流程说明

钉钉OAuth2.0授权登录让钉钉用户使用钉钉身份安全登录第三方应用或网站，在钉钉用户授权登录已接入钉钉OAuth2.0的第三方应用后，第三方可以获取到用户的接口调用凭证（*snstoken*），通过*snstoken*可以进行钉钉开放平台授权关系接口调用，从而可实现获取钉钉用户基本开放信息和帮助用户实现基础开放功能等。

钉钉OAuth2.0授权登录目前支持authorizationcode模式，适用于拥有server端的应用授权。该模式整体流程为：

1. 第三方发起钉钉授权登录请求，钉钉用户允许授权第三方应用后，钉钉会拉起应用或重定向到第三方网站，并且带上授权临时票据code参数；
2. 通过code参数加上AppID和AppSecret等，通过API换取accesstoken；
3. 通过accesstoken和code获取snstoken；
4. 通过snstoken进行接口调用，获取用户基本数据资源或帮助用户实现基本操作。

获取accesstoken时序图：

钉钉授权免登时序图(<http://img.blog.csdn.net/20180418211129282?watermark/2/text/aHR0cDovL2Jsbn2cuY3Nkbi5uZXQvSUVZVURFWUIOSkk=/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast>)

第一步请求Code

移动应用钉钉授权登录

开发者需要配合使用钉钉开放平台提供的SDK进行授权登录请求接入。正确接入SDK后并拥有相关授权域（scope）（即授权的范围，目前钉钉仅支持授权登录，获取用户的基本信息）权限后，开发者移动应用会在终端本地拉起钉钉应用进行授权登录，钉钉用户确认后钉钉将拉起开发者移动应用，并带上授权临时票据（code）。

iOS平台应用授权登录接入代码示例:

1. Info.plist配置查询钉钉授权登录scheme权限。LSApplicationQueriesSchemes中添加dingtalk, dingtalk-open, dingtalk-sso。dingtalk用于查询是否安装钉钉。dingtalk-open用于查询是否支持开放平台接口调用（不需要查询开放平台接口可不填）。dingtalk-sso用于查询是否支持授权登录。注：iOS系统限制LSApplicationQueriesSchemes最多只能有50个，超出的话会有一部分不生效。
2. Info.plist的URL Types中增加appId，用于钉钉App返回回调
3. app启动时注册appId。示例代码：

```
objective-c
```

```
// 在app启动时注册appId
```

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions

```
// 注册AppId;
```

```
DTOpenAPI registerApp:#该应用的appId#;
```

```
return YES;
```

```
}
```

1. 发起授权登录。示例代码：

```
objective-c
```

```
DTAuthorizeReq authReq = DTAuthorizeReq new;
```

```
authReq.bundleId = #该应用的bundleId#;
```

1. 处理回调。示例代码:

```
objective-c
```

```
// 在app delegate链接处理回调中调用钉钉回调链接处理方法
```

- (BOOL)application:(UIApplication *)application
openURL:(nonnull NSURL *)url

```
sourceApplication:(nullable NSString *)sourceApplication
```

```
annotation:(nonnull id)annotation
```

```
if (DTOpenAPI handleOpenURL:url delegate:self)
```

```
return YES;
```

```
}
```

```
return NO;
```

```
}
```

```
// delegate实现回调处理方法 onResp:
```

- (void)onResp:(DTBaseResp *)resp

```
//授权登录回调参数为DTAuthorizeResp, accessCode为授权码
```

```
if (resp isKindOfClass:[DTAuthorizeResp class])
```

```

DTAuthorizeResp authResp = (DTAuthorizeResp )resp;

NSString accessCode = authResp.accessCode;

// 将授权码交给服务端做SSO

}

}

```

Android平台应用授权登录接入代码示例：

```

// send oauth request

SendAuth.Req req = new SendAuth.Req();

req.scope = SendAuth.Req.SNSLOGIN;

req.state = "test";

iddShareApi.sendReq(req);

}

```

入参说明

参数	是否必须	说明
----	------	----

- appid	是	应用唯一标识，在钉钉开放平台提交应用审核通过后获得
-----------	---	---------------------------

| scope | 是 | 应用授权作用域，如获取用户个人信息则填写(SendAuth.Req.SNSLOGIN); |

| state | 否 | 用于保持请求和回调的状态，授权请求后原样带回给第三方。该参数可用于防止csrf攻击（跨站请求伪造攻击），建议第三方带上该参数，可设置为简单的随机数加session进行校验 |

返回示例：

在DDShareActivity中回调的onResp（baseResp）的方法中，解析钉钉返回的授权临时票据（code）

示例如下(具体实现细节请参照DDShareDemo中的DDShareActivity)

```
@Override

public void onResp(BaseResp baseResp)

int errCode = baseResp.mErrCode;

Log.d("dd" , "errorCode=====>" +errCode);

String errMsg = baseResp.mErrStr;

Log.d("dd" , "errMsg=====>" +errMsg);

switch (errCode)

case BaseResp.ErrCode.ERR_OK:

if(baseResp.getType() == ShareConstant.COMMANDSENDAUTHV2 && (baseResp instanceof
SendAuth.Resp))

SendAuth.Resp resp = (SendAuth.Resp) baseResp;

Toast.makeText(this, "授权成功，授权码为:" +resp.code, Toast.LENGTH_SHORT).show();

}else

Toast.makeText(this, "分享成功", Toast.LENGTH_SHORT).show();
```

```

}

break;

}

finish();

}

```

备注： DDShareActivity的路径，需要配置为第三方 App的
PackageName.ddshare. DDShareActivity

可拉起钉钉打开授权登录页：

```
!(http://img.blog.csdn.net/20180418214942777?watermark/2/text/
aHR0cDovL2Jsbn2cuY3Nkbi5uZXQvSUVZVURFWUIOSkk=/font/5a6L5L2T/fontsize/400/fill/
I0JBQkFCMA==/dissolve/70/gravity/SouthEast)
```

返回说明

用户点击授权后，钉钉客户端会被拉起，跳转至授权界面，用户在该界面点击允许或取消，SDK通过SendAuth的Resp返回数据给调用方。

| 返回值 | 说明 |

| :-: | :-: |

| ErrCode | *ERROK* = 0(用户同意) *ERRAUTHDENIED* = -4（用户拒绝授权） *ERRUSERCANCEL* = -2（用户取消） |

| code | 用户换取accesstoken的code，仅在ErrCode为0时有效 |

| state | 第三方程序发送时用来标识其请求的唯一性的标志，由第三方程序调用sendReq时传入，由钉钉终端回传，state字符串长度不能超过1K |

第二步：获取`accesstoken`

请求以下链接获取`accesstoken`：

`https://oapi.dingtalk.com/sns/gettoken?appid=APPID&appsecret=APPSECRET`

参数说明

参数	是否必须	说明
----	------	----

-	是	应用唯一标识，在钉钉开放平台提交应用审核通过后获得
---	---	---------------------------

appid	是	应用唯一标识，在钉钉开放平台提交应用审核通过后获得
-------	---	---------------------------

appsecret	是	应用密钥AppSecret，在钉钉开放平台提交应用审核通过后获得
-----------	---	----------------------------------

返回说明

正确的返回：

```
json
```

```
"accesstoken": "070c171a26d633d1b631dxxxxxxx",
```

```
"errcode": 0,
```

```
"errmsg": "ok"
```

```
}
```


| 参数 | 说明 |

| :: | ::

| *access_token* | 接口调用凭证 |

| *errcode* | 错误码 |

| *errmsg* | 错误信息

注意：

1. AppSecret 是应用接口使用密钥，泄漏后将可能导致应用数据泄漏、应用的用户数据泄漏等高风险后果；存储在客户端，极有可能被恶意窃取（如反编译获取Appsecret）；
2. *access_token* 为用户授权第三方应用发起接口调用的凭证（相当于用户登录态），存储在客户端，可能出现恶意获取*access_token* 后导致的用户数据泄漏、用户钉钉相关接口功能被恶意发起等行为；
3. *refresh_token* 为用户授权第三方应用的长效凭证，仅用于刷新*access_token*，但泄漏后相当于*access_token* 泄漏，风险同上。

建议将Appsecret、用户数据（如*access_token*）放在App云端服务器，由云端中转接口调用请求。

第三步：通过第三方应用的*access_token* + code 获取持久授权码

获取*access_token*后，进行接口调用，有以下前提：

*access_token*有效且未超时；

钉钉用户已授权给第三方应用code。

接口调用方法可查阅 [获取用户授权的持久授权码](https://open-doc.dingtalk.com/docs/doc.htm?spm=a219a.7629140.0.0.GwVwxk&treld=385&articleId=104968&docType=1#s1)(https://open-doc.dingtalk.com/docs/doc.htm?spm=a219a.7629140.0.0.GwVwxk&treld=385&articleId=104968&docType=1#s1)

正确返回示例如下：

json

```
"errcode": 0,  
  
"errmsg": "ok",  
  
"openid": "liSii8KCxxxxx",  
  
"persistentcode": "dsa-d-asdasdadHIBlinoninINln-ssdasd",  
  
"unionid": "7Huu46kk"  
  
}
```

返回数据说明

| 参数 | 说明 |

| :-: | :-: |

| openid | 用户在当前开放应用内的唯一标识

|

| unionid | 用户在当前钉钉开放平台账号范围内的唯一标识，同一个钉钉开放平台账号可以包含多个开放应用，同时也包含ISV的套件应用及企业应用

|

| persistentcode | 用户给开放应用授权的持久授权码，此码目前无过期时间

第四步：通过持久授权码获取用户授权的SNSTOKEN

接口调用方法可查阅 [获取用户授权的SNSTOKEN](https://open-doc.dingtalk.com/docs/doc.htm?spm=a219a.7629140.0.0.GwVwxk&treeld=385&articleId=104968&docType=1#s1)(https://open-doc.dingtalk.com/docs/doc.htm?spm=a219a.7629140.0.0.GwVwxk&treeld=385&articleId=104968&docType=1#s1)

正确返回示例如下：

json

```
"errcode": 0,  
"errmsg": "ok",  
"expiresin": 7200,  
"snstoken": "c76dsc87ds6c876sd87csdcxxxxx"  
}
```

返回数据说明

参数	说明
----	----

expiresin	snstoken的过期时间
-----------	---------------

snstoken	用户授权的token
----------	------------

第五步: 通过SNSTOKEN获取用户授权的个人信息

接口调用方法可查阅 [获取用户授权的个人信息](https://open-doc.dingtalk.com/docs/doc.htm?spm=a219a.7629140.0.0.GwVwxk&treeld=385&articleId=104968&docType=1#s1)(https://open-doc.dingtalk.com/docs/doc.htm?spm=a219a.7629140.0.0.GwVwxk&treeld=385&articleId=104968&docType=1#s1)

正确返回示例如下:

json

```
"errcode": 0,  
"errmsg": "ok",  
"userinfo":  
  "maskedMobile": "1301234",  
  "nick": "张三",  
  "openid": "liSii8KCxxxxx",  
  "unionid": "7Huu46kk",  
  "dingId": "dingId"  
}  
}
```

返回数据说明

| 参数 | 说明 |

| :-: | :-: |

| nick | 用户在钉钉上面的昵称

|

| openid | 用户在当前开放应用内的唯一标识 |

| unionid | 用户在当前开放应用所属的钉钉开放平台账号内的唯一标识 |

| dingId | 钉钉Id |

FAQ

1. 什么是授权临时票据 (code) ?

答：第三方通过code进行获取`accesstoken`的时候需要用到，`code`的超时时间为10分钟，一个`code`只能成功换取一次`accesstoken`即失效。`code`的临时性和一次保障了钉钉授权登录的安全性。第三方可通过使用https和state参数，进一步加强自身授权登录的安全性。

2. 开放平台移动应用钉钉登陆目前是否收费？

答：“钉钉登录”和第三方网站共享钉钉庞大的用户价值，同时为钉钉用户提供更便捷服务和更优质内容，实现双向共赢，目前不收取任何费用。