易订货 OAuth

相关概念

OAuth2.0（开放授权）是一个开放标准，用户授权后，第三方应用无需获取用户的用户名和密码就可以访问该用户在某一网站上存储的私密的资源（如照片，视频，联系人列表）。

Access Token：用户身份验证和授权的凭证。第三方应用在调用易订货开放 API 之前，首先需要获取 Access Token。

使用易订货授权前准备

- 如果您已是易订货用户，请您访问易订货并使用易订货账号直接登录。
- 如果您还不是易订货用户，
  - 请您访问易订货，填写注册邮箱和密码并激活成为易订货用户。

- 您需要创建一个应用以获取 API Key（client_id）和 Secret Key（client_secret），申请方式发送邮件到 lgc@77ircloud.com，在邮件里面写上你的应用名称，应用回调地址，公司名称，公司网址，公司联系人，以及你本人的联系方式。易订货审核通过就可以使用 api

易订货支持的 OAuth 授权

目前，易订货 OAuth2.0 支持 3 种获取 Access Token 的流程和一种刷新获取 AccessToken 方式，第三方可根据需求选取合适的方式：

易订货授权的 Access Token 是有有效期的，这样会影响用户的体验和增加开发者的工作。所以平台提供了一种方式可以保证授权有效期为永久。

- **实现方式**：返回给第三方一个月有效期的 Access Token + 一年有效期的 Refresh Token。
- **实现原理**：Refresh Token 的作用就是在 Token 有效期截止前，刷新以获取新的 Access Token。

获取途径	授权流程	介绍	有效期
新获取	Authorization Code	又称 Web Server Flow，适用于所有有 Server 端配合的应用。	有效期一个月的 Access Token+有效期一年的 Refresh Token。
	Implicit Grant	又称 User-Agent Flow，适用于所有无 Server 端配合的应用（桌面客户端需要内嵌浏览器）。	有效期一个月的 Access Token。
	Client Credentials	即采用应用公钥、密钥获取 Access Token，适用于任何带 server 类型应用。 通过此授权方式获取 Access Token 仅可访问平台授权类的接口。	有效期一个月的 Access Token+有效期十年的 Refresh Token。
刷新	Refresh Token	Access Token 刷新方式，适用于所有有 Server 端配合的应用 。	十年刷新期限。

授权回调地址

为确保验证授权过程的安全，开发者必须在申请 app 的时候填写好应用所在的域名或 URL，用以 OAuth2.0 检验授权请求中的“redirect_uri”参数。以便保证 OAuth2.0 在回调过程中，会回调到安全域名。

- **站外有 Web Server 应用**
  - Web 应用
  - 有 Web Server 支持的非 Web 应用（例如：有 Web Server 支持的手机客户端、桌面客户端应用） 。
- **站外无 Web Server 应用**
  - 桌面客户端应用
  - 手机客户端应用
  - 基于浏览器脚本语言的应用（JavaScript、Flash、ActionScript）

平台提供了一种默认的 redirect uri 参数为 "oob"，回调后会返回 json 数据

授权权限列表

每一个 Access Token 代表“一个用户”授予“一个应用”的“一系列数据访问操作权限”，

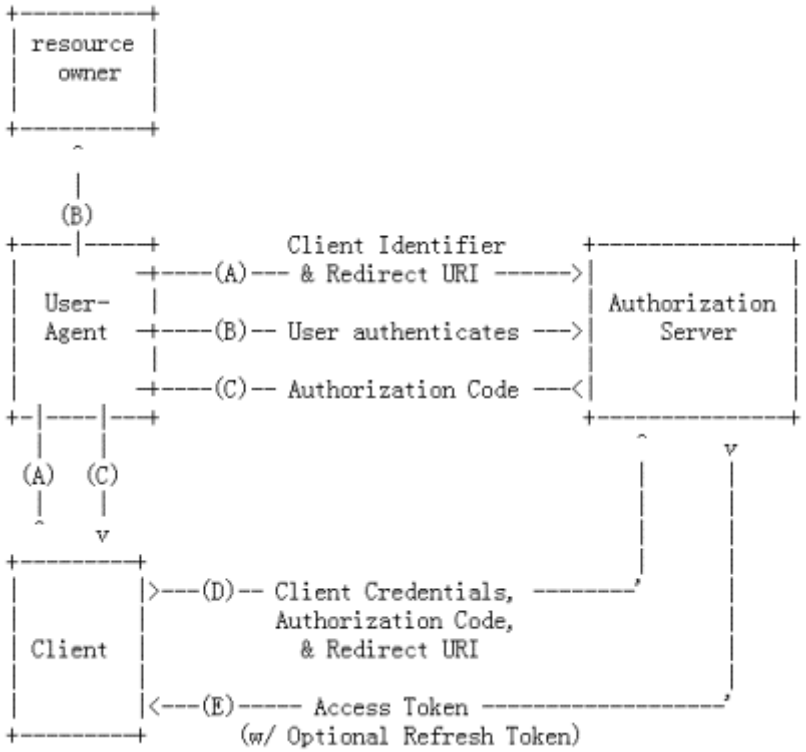
请参考：

授权相关权限	介绍
basic	用户基本权限,可以获取用户的基本信息,并且可以访问大部分公共的开放 API。
push	业务消息推送权限
report	报表数据权限
system	系统设置权限

Authorization Code 授权

简介

采用 Authorization Code 获取 Access Token 的授权验证流程又被称为 Web Server Flow，适用于所有有 Server 端的应用，如 Web/Wap 站点、有 Server 端的手机/桌面客户端应用等。其流程示意图如下：



对于应用而言，其流程由获取 Authorization Code 和通过 Authorization Code 获取 Access Token 这 2 步组成。

获取 Authorization Code

请求数据包格式

其获取方式是通过重定向用户浏览器（或手机/桌面应用中的浏览器组件）到“https://api.dinghuo123.com/v2/oauth2/authorize”地址上，并带上以下参数：

- **client_id**：必须参数，注册应用时获得的 API Key。
- **response_type**：必须参数，此值固定为“code”。
- **redirect_uri**：必须参数，授权后要回调的 URI，即接收 Authorization Code 的 URI。如果用户在授权过程中取消授权，会回调该 URI，并在 URI 末尾附上 error=access_denied 参数。对于无 Web Server 的应用，其值可以是“oob”，此时用户同意授权后，授权服务会将 Authorization Code 直接显示在响应页面的页面中及页面 title 中。非“oob”值的 redirect_uri 按照如下规则进

行匹配：（1）如果开发者在“授权安全设置”中配置了“授权回调地址”，则 redirect_uri 必须与“授权回调地址”中的某一个相匹配；（2）如果未配置“授权回调地址”，redirect_uri 所在域名必须与开发者注册应用时所提供的网站根域名列表或应用的站点地址（如果根域名列表没填写）的域名相匹配。

- **scope**：非必须参数，以空格分隔的权限列表，若不传递此参数，代表请求用户的默认权限。关于权限的具体信息请参考“权限列表”。
- **state**：非必须参数，用于保持请求和回调的状态，授权服务器在回调时（重定向用户浏览器到“redirect_uri”时），会在 Query Parameter 中原样回传该参数。OAuth2.0 标准协议建议，利用 state 参数来防止 CSRF 攻击。

例如：

```
https://api.dinghuo123.com/v2/oauth2/authorize?
   response_type=code&
   client_id=YOUR_CLIENT_ID&
   scope=basic&
   redirect_uri=http%3A%2F%2Fwww.dinghuo123.com
```

响应数据包格式

此时授权服务会根据应用传递参数的不同，为用户展现不同的授权页面。如果用户在此页面同意授权，授权服务则将重定向用户浏览器到应用所指定的“redirect_uri”，并附带上表示授权服务所分配的 Authorization Code 的 code 参数，以及 state 参数（如果请求 authorization code 时带了这个参数）。

例如：继续上面的例子，假设授权服务在用户同意授权后生成的 Authorization Code 为“a1a4b0b6dae19c35cd2d786fddb8e19f”，则授权服务将会返回如下响应包以重定向用户浏览器到“http://www.dinghuo123.com/ ”地址上：

```
HTTP/1.1 302 Found
Location: http://www.dinghuo123.com?code=a1a4b0b6da
e19c35cd2d786fddb8e19f
```

“code”参数可以在“redirect_uri”对应的应用后端程序中获取。

注意:

每一个 Authorization Code 的有效期为 10 分钟，并且只能使用一次，再次使用将无效。

通过 Authorization Code 获取 Access Token

请求数据包格式

通过上面第一步获得 Authorization Code 后，便可以用其换取一个 Access Token。获取方式是，应用在其服务端程序中发送请求（推荐使用 POST）到 易订货 OAuth2.0 授权服务的“https://api.dinghuo123.com/v2/oauth2/token”地址上，并带上以下 5 个必须参数：

- **grant_type**：必须参数，此值固定为“authorization_code”；
- **code**：必须参数，通过上面第一步所获得的 Authorization Code；
- **client_id**：必须参数，应用的 API Key；
- **client_secret**：必须参数，'应用的 Secret Key；
- **redirect_uri**：必须参数，该值必须与获取 Authorization Code 时传递的“redirect_uri”保持一致。

例如：

```
https://api.dinghuo123.com/v2/oauth2/token?
   grant_type=authorization_code&
   code=a1a4b0b6dae19c35cd2d786fddb8e19f&
   client_id=YOUR_CLIENT_ID&
   client_secret=YOUR_CLIENT_SECRET&
   redirect_uri=http%3A%2F%2Fwww.dinghuo123.com
```

响应数据包格式

若参数无误，服务器将返回一段 JSON 文本，包含以下参数：

- **access_token**：要获取的 Access Token；
- **expires_in**：Access Token 的有效期，以秒为单位；
- **refresh_token**：用于刷新 Access Token 的 Refresh Token,所有应用都会返回该参数；（**1 年的有效期**）
- **scope**：Access Token 最终的访问范围，即用户实际授予的权限列表；

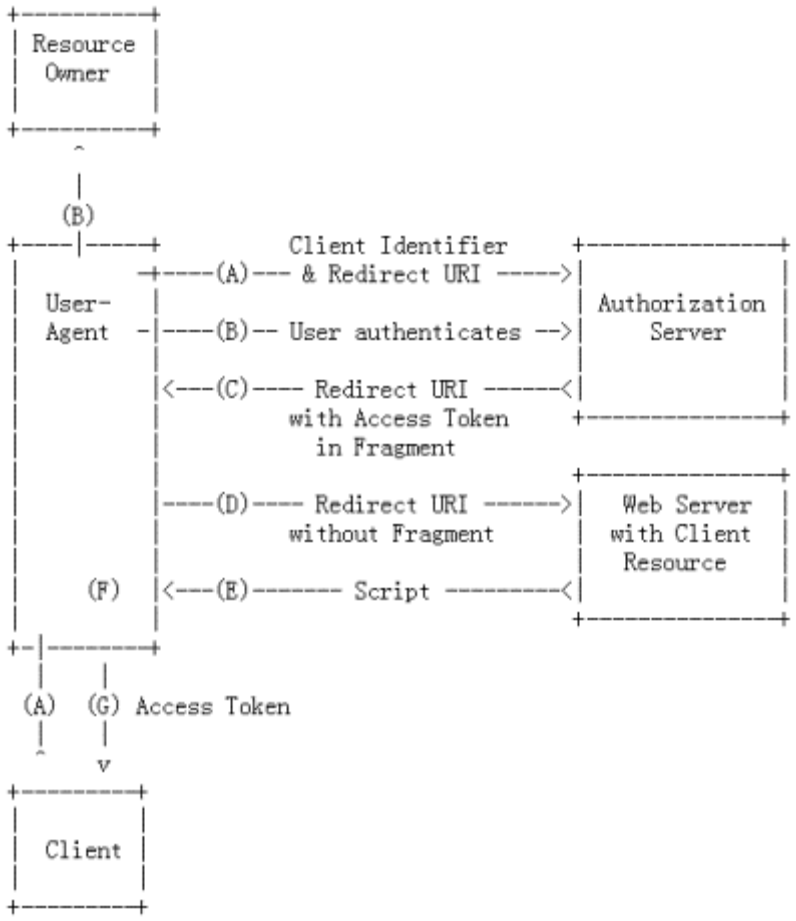
例如：

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{"code":200,"message":"操作成功","data":{"access_tok

en":"ca52163e2d9217e971e03cfa1e94cdd1","expires_in
":2592000,"scope":"basic","refresh_token":"bf0a7a90
ad384c72de13e9d3f9034d60","create_time":14174239365
90}}
```

Implicit Grant 授权

简介

采用 Implicit Grant 方式获取 Access Token 的授权验证流程又被称为 User-Agent Flow，适用于所有无 Server 端配合的应用（由于应用往往位于一个 User Agent 里，如浏览器里面，因此这类应用在某些平台下又被称为 Client-Side Application），如手机/桌面客户端程序、浏览器插件等，以及基于 JavaScript 等脚本客户端脚本语言实现的应用，他们的一个共同特点是，应用无法妥善保管其应用密钥（App Secret Key），如果采取 Authorization Code 模式，则会存在泄漏其应用密钥的可能性。其流程示意图如下：



对于应用而言，其流程只有一步，即直接获取 Access Token。

获取 Access Token

请求数据包格式

为了获取 Access Token，应用需要将用户浏览器（或手机/桌面应用中的浏览器组件）到易订货 OAuth2.0 授权服务的“https://api.dinghuo123.com/v2/oauth2/authorize”地址上，并带上以下参数：

- **client_id**：必须参数。注册应用时获得的 API Key。

- **response_type**：必须参数。此值固定为“token”。
- **redirect_uri**：必须参数。授权后要回调的 URI，即接受 Access Token 的 URI。如果用户在授权过程中取消授权，会回调该 URI，并在 URI 末尾附上 error=access_denied 参数。对于无 Web Server 的应用，其值可以是“oob”，授权后会回调 OAuth 提供的一个默认页面。如果 redirect_uri 不为"oob"，则 redirect_uri 指向的页面必须与开发者在申请时候中所填写的"授权回调地址"相匹配。

- **scope**：非必须参数。以空格分隔的用户权限列表，若不传递此参数，代表请求用户的默认权限。关于权限的具体信息请参考“权限列表”。
- **state**：非必须参数。用于保持请求和回调的状态，授权服务器在回调时（重定向用户浏览器到“redirect_uri”时），会在 Fragment 中原样回传该参数。

例如：“client_id”为应用要请求某个用户的默认权限和 basic 访问权限，并在授权后需跳转到“http://www.dinghuo123.com/”，同时希望在弹出窗口中展现用户登录授权界面，则应用需要重定向用户浏览器到如下 URL：

```
https://api.dinghuo123.com/v2/oauth2/authorize?
      response_type=token&
      client_id=YOUR_CLIENT_ID&
      redirect_uri=http%3A%2F%2Fwww.dinghuo123.com
&scope=basic&state=1
```

响应数据包格式

若用户登录并接受授权，授权服务将重定向用户浏览器到“redirect_uri”。如果开发者传递的"redirect_uri"为"oob"，浏览器将被重定向到 OAuth 默认提供的一个页面"https://api.dinghuo123.com/common/oauth_success.html"。并在 Fragment 中追加如下参数：

- **access_token**：要获取的 Access Token；
- **expires_in**：Access Token 的有效期，以秒为单位；
- **scope**：Access Token 最终的访问范围，即用户实际授予的权限列表
- **state**：如果请求获取 Access Token 时带有 state 参数，则将该参数原样返回。

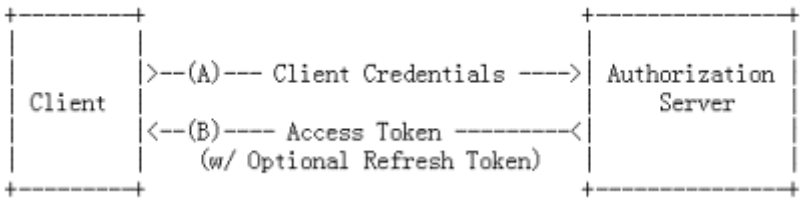
返回例子：

```
HTTP/1.1 302 Found
Location: http://www.dinghuo123.com/#state=1&expire
s_in=2592000&access_token=6ddb8345d8c7249911e07f40b
333fd24&scope=basic
```

Client Credentials 授权

简介

采用 Client Credentials 方式，即应用公钥、密钥方式获取 Access Token，适用于任何类型应用，但通过它所获取的 Access Token 只能用于访问与用户无关的 Open API，并且需要开发者提前向易订货申请，成功对接后方能使用。其流程示意图如下：



对于应用而言，其流程只有一步，即直接获取 Access Token。

注意：获得这种方式的需要申请

获取 Access Token

请求数据包格式

使用 Client Credentials 获取 Access Token 需要应用在其服务端发送请求（推荐用 POST 方法）到易订货 OAuth2.0 授权服务的“https://api.dinghuo123.com/v2/oauth2/token”地址上，并带上以下参数：

- **grant_type**：必须参数，固定为“client_credentials”；
- **client_id**：必须参数，应用的 API Key；
- **client_secret**：必须参数，应用的 Secret Key;
- **scope**：非必须参数。以空格分隔的权限列表

例如：

```
https://api.dinghuo123.com/v2/oauth2/token?
    grant_type=client_credentials&
    client_id=YOUR_CLIENT_ID&
    client_secret=YOUR_CLIENT_SECRET&
    scope = basic
```

响应数据包格式

若参数无误，服务器将返回一段 JSON 文本，包含以下参数：

- **access_token**：要获取的 Access Token；

- **expires_in**：Access Token 的有效期，以秒为单位
- **refresh_token**：用于刷新 Access Token 的 Refresh Token,所有应用都会返回该参数；（**1 年的有效期**）
- **scope**：Access Token 最终的访问范围，即用户实际授予的权限列表

例如：

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{"code":200,"message":"操作成功","data":{"access_tok
en":"d9305d9ed6d91d1a0a8fb25de967ba03","expires_in
":2592000,"scope":"basic","refresh_token":"0d7e1971
82c92ceaf429f2fc1a04f613","create_time":13898632204
11}}
```

Refresh Token

简介

对于从开放平台申请到允许永久授权权限的应用，无论其采用 Authorization Code、Reource Owner Password Credentials、Client Credentials 中的哪一个去获取 Access Token，都会拿到一个有效期为个月的 Access Token 和有效期为 1 年的 Refresh Token 对于这些应用，只要用户在 1 年内登陆应用，应用就可以使用 Refresh Token 刷新以获得新的 Access Token（新的 Refresh Token 也会同时下发），从而达到只要用户不连续 1 年未登陆过你的应用就不需要重新登陆的目的。

获取 Access Token

请求数据包格式

使用 Refresh Token 刷新以获得新的 Access Token，需要应用在其服务端发送请求（推荐用 POST 方法）到易订货 OAuth2.0 授权服务的“https://api.dinghuo.com/v2/oauth2/token”地址上，并带上以下参数：

- **grant_type**：必须参数，固定为“refresh_token”；
- **refresh_token**：必须参数，用于刷新 Access Token 用的 Refresh Token；
- **client_id**：必须参数，应用的 API Key；
- **client_secret**：必须参数，应用的 Secret Key;
- **scope**：非必须参数。以空格分隔的权限列表，若不传递此参数，代表请求的数据访问操作权限与上次获取 Access Token 时一致。通过 Refresh Token 刷新 Access Token 时所要求的 scope 权限范围必须小于等于上次获取 Access Token 时授予的权限范围。

例如：

```
https://api.dinghuo123.com/v2/oauth2/token?
    grant_type=refresh_token&
    refresh_token=0d7e197182c92ceaf429f2fc1a04f613&
    client_id=YOUR_CLIENT_ID&
    client_secret=YOUR_CLIENT_SECRET&
    &scope=basic
```

响应数据包格式

若参数无误，服务器将返回一段 JSON 文本，包含以下参数：

- **access_token**：要获取的 Access Token；
- **expires_in**：Access Token 的有效期，以秒为单位；

- **refresh_token**：用于刷新 Access Token 的 Refresh Token,并不是所有应用都会返回该参数；（1 年的有效期）
- **scope**：Access Token 最终的访问范围，即用户实际授予的权限列表

例如：

```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-store

{"code":200,"message":"操作成功","data":{"access_tok
en":"d9305d9ed6d91d1a0a8fb25de967ba03","expires_in
":2592000,"scope":"basic","refresh_token":"0d7e1971
82c92ceaf429f2fc1a04f613","create_time":13898632204
11}}
```

注意：

关于 Client Credentials 模式：
可以直接使用下面 URL 不需要显式地登录完成获取 token

```
https://api.dinghuo123.com/v2/oauth2/token?
    grant_type=client_credentials&
    client_id=YOUR_CLIENT_ID&
    client_secret=YOUR_CLIENT_SECRET&
    scope=basic&userName=YOUR_NAME&password=YOUR_PASS
```

在原来的 url 加上 userName 和 password 两个参数，就不会强制要求登录，这样授权操作就包括登陆用户名密码认证了，直接返回 token 等信息

关于体验（需要申请）
内部使用的模式：try_token

```
https://api.dinghuo123.com/v2/oauth2/token?
    grant_type=try_token&
    client_id=YOUR_CLIENT_ID&
    client_secret=YOUR_CLIENT_SECRET&
    scope=basic
```

这样向服务器请求就会随机返回一个 access_token，具有一天的有效期，用于体验，这个随机生成的 token 是绑定到某个具体的用户，所以这个 url 请求作用就相当于 Client Credentials 模式，包括完成登陆、oauth，并且不需要指定用户名和密码，用户是用服务器随机获取的