账号风控平台

开发对接指南

腾讯云

文档目录

开发对接指南

概述

最近更新时间：2022-06-14 17:34:49

本指南介绍应用系统接入 CIAM 的方法及相应的 API。通过对接 CIAM，您的应用可以快速实现用户的登录、退出、注册等功能，并集成 CIAM 灵活、强大的配置与管理能力。

应用类型

基于常见的业务场景，CIAM 将应用分为以下几种类型：

Web 应用

运行在后端 Web 服务器上的应用系统，一般采用 Java, .NET, PHP, Node.js, Express 等语言或框架开发，用户通过浏览器访问应用。由于后端程序一般在受保护的服务器上运行，Web 应用能较好地存储应用密码等敏感信息，并保护 Token 等动态信息不泄露。

单页应用 (SPA)

直接运行在浏览器中的前端应用程序，一般采用 HTML、CSS 和 JavaScript 技术结合 React、Vue 和 Angular 等框架开发。单页应用可以直接向业务面 API 发起请求而无需经过后端程序转发，但由于程序和数据都存在于 user-agent (如浏览器) 中，单页应用不适合存储或处理需要受保护的敏感信息。

移动 App

安装和运行在用户设备 (如手机、平板电脑、PC、智能设备) 上的应用程序，一般采用专门的应用开发语言开发，如 Objective-C、Swift、 Kotlin 等。此类应用不适合存储应用密码等敏感信息，但一般能够保护 Token 等动态信息不泄露。

M2M 应用 (Machine to Machine)

运行在后端的应用程序 (如后端服务、命令行程序、守护进程)，一般通过调用其他的 API 来实现业务功能，无需用户参与。此类应用能够较好地保护敏感信息不泄露。

小程序应用

微信小程序应用，运行在移动端或桌面端微信 App 内。

接入概览

Web 应用、SPA 和 移动 App 可以作为 OIDC (OAuth) 标准客户端快速接入 CIAM，详见 使用认证门户登录 。OIDC 和 OAuth 协议在互联网的应用十分广泛，其相应的开发资源也非常丰富，各类应用一般都能找到开发库快速完成接入。

- 微信小程序应用需使用 CIAM 的小程序 SDK 来实现登录，操作详情请参见 接入微信小程序（SDK 模式） 。
- M2M 应用一般使用 客户端凭证模式获取 Access Token 后，携带 Access Token 访问相应的 API。

> **注意**
> 本指南的 API 通过 HTTPS 协议访问，访问地址前缀是您的用户目录域名，可以在 域名设置页面 查看。在本指南中，使用 `https://sample.portal.tencentciam.com` 作为访问地址前缀。

使用认证门户接入
使用认证门户登录
PKCE 授权码模式

最近更新时间：2024-06-19 10:46:21

接口描述

将用户（浏览器）重定向到此接口地址，发起登录。CIAM 会将用户重定向到认证门户进行登录认证。登录成功后，CIAM 将用户重定向到 `redirect_uri` 参数指定的地址。
如果发起登录时用户已经登录，则无需再次登录，CIAM 会直接将用户重定向到 `redirect_uri` 。

> **说明**
> - 根据 OAuth 协议的安全实践教程要求，本接口使用 PKCE 授权码模式。
> - 本节请求示例中使用的应用系统 Redirect URI 为 `https://example.com/callback` 。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
GET
```

请求路径

```
/oauth2/authorize
```

请求示例

```
GET /oauth2/authorize?
scope=openid&client_id=TENANT_CLIENT_ID&redirect_uri=https%3A%2F%2FTENANT.APP.DOMAIN%2Flogin%2Foauth2%2Fcod
e%2FTENANT_APP_ID&response_type=code&state=MOCK_STATE&code_challenge_method=S256&code_challenge=MOCK_CO
DE_CHALLENGE&auth_source_id=MOCK_USERNAME_PASSWORD_AUTH_SOURCE_ID HTTP/1.1
Host: sample.portal.tencentciam.com
```

请求参数

参数	可选	描述
scope	false	填固定值 `openid` 。
client_id	false	应用的 `client_id` 。可参考 应用管理页面 > **选定指定应用** > 单击**应用配置** > 对应的“Client Id”。
redirect_uri	false	授权成功后的重定向地址。需要与控制台配置的地址一致。
response_type	false	填固定值 `code` 。
state	true	应用随机生成的一个字符串，服务器会以 HTTP 响应参数的形式原样返回给应用。为防止 CSRF 攻击，建议携带此参数。
code_challenge_method	false	计算 PKCE code_challenge 的算法。目前仅支持固定值 `S256` 。
code_challenge	false	PKCE code_challenge ，计算方法请参考 RFC 7636 。
auth_source_id	true	使用特定的认证源登录。如果此参数为空，则显示默认登录页面。

正常响应示例

- 用户未登录，显示认证门户的默认登录页面。

```
HTTP/1.1 302 Found
Location: https://sample.portal.tencentciam.com/portal/login?p_state=MOCK_LOGIN_PORTAL_STATE
```

- 用户已登录，携带授权码和 state 参数重定向到应用回调地址。

```
HTTP/1.1 302 Found
Location: https://example.com/callback?
code=DVtNBg5XGqeu2IytLi6WOWwfh7pRc5jqI8vUb2K8k_2OryR2OsYN3260DwhlTDqEMtUSD1XN6gNuRDjYQ25nJX6H8MzfpI
xJHloi0tdtkXfRpV1ELhmw7behuwYraTlC&state=MOCK_STATE
```

> **说明**
> 应用回调地址拿到 code 参数后，需要调用 PKCE 模式 获取 Token 口获取 Access Token 和 ID Token，完成登录。

异常响应示例

- client_id 参数缺失或有误。

```
HTTP/1.1 400 Bad Request
```

- redirect_uri 参数与注册信息不匹配。

```
HTTP/1.1 400 Bad Request
```

- response_type 参数缺失或有误。

```
HTTP/1.1 400 Bad Request
```

- 不支持的 code_challenge_method。

```
HTTP/1.1 302 Found
Location: https://example.com/callback?
error=invalid_request&error_description=OAuth%202.0%20Parameter:%20code_challenge_method&error_uri=https://datat
racker.ietf.org/doc/html/rfc7636%23section-4.4.1&state=MOCK_STATE
```

普通授权码模式

最近更新时间：2023-08-14 14:39:42

接口描述

将用户（浏览器）重定向到此接口地址，发起登录。CIAM 会将用户重定向到认证门户进行登录认证。登录成功后，CIAM 将用户重定向到 `redirect_uri` 参数指定的地址。
如果发起登录时用户已经登录，则无需再次登录，CIAM 会直接将用户重定向到 `redirect_uri` 。

> **说明**
> - 出于安全因素，OAuth 协议不推荐使用此模式，请优先选择使用 PKCE 授权码模式 。
> - 本节请求示例中使用的应用系统 Redirect URI 为 `https://example.com/callback` 。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
GET
```

请求路径

```
/oauth2/authorize
```

请求示例

```
GET /oauth2/authorize?
scope=openid&client_id=TENANT_CLIENT_ID&redirect_uri=https%3A%2F%2Fexample.com%2Fcallback&response_type=code
&state=MOCK_STATE&auth_source_id=MOCK_USERNAME_PASSWORD_AUTH_SOURCE_ID HTTP/1.1
Host: sample.portal.tencentciam.com
```

请求参数

参数	可选	描述
scope	false	填固定值 openid。
client_id	false	应用的 client_id 。可参考 应用管理页面 > 选定指定应用 > 单击应用配置 > 对应的“Client Id” 。
redirect_uri	false	授权成功后的重定向地址。需要与控制台配置的地址一致。
response_type	false	填固定值 code。
state	true	应用随机生成的一个字符串，服务器会以 HTTP 响应参数的形式原样返回给应用。为防止 CSRF 攻击，建议携带此参数。
auth_source_id	true	使用特定的认证源登录。如果此参数为空，则显示默认登录页面。

正常响应示例

- 用户未登录，显示认证门户的默认登录页面。

```
HTTP/1.1 302 Found
Location: https://sample.portal.tencentciam.com/portal/login?p_state=MOCK_LOGIN_PORTAL_STATE
```

- 用户已登录，携带授权码和 state 参数重定向到应用回调地址。

```
HTTP/1.1 302 Found
Location: https://example.com/callback?code=3ZBaZRXqXZxdkQ1-qwj-
YLsWOhYJvQGzak_RrRmJQ5XHDFlXjeKiwlQBHp_PnYruhX1HuBL3OeZNcbrkWt41YBU-
xd6sHmoNEShRd9rWYYzpIPaQDWAKGptUtM-wRPbL&state=MOCK_STATE
```

> **说明**
> 应用回调地址拿到 `code` 参数后，需要调用 普通授权码模式获取 Token 接口获取 Access Token 和 ID Token，完成登录。

异常响应示例

- client_id 参数缺失或有误。

```
HTTP/1.1 400 Bad Request
```

- redirect_uri 参数与注册信息不匹配。

```
HTTP/1.1 400 Bad Request
```

- response_type 参数缺失或有误。

```
HTTP/1.1 400 Bad Request
```

退出认证门户登录

最近更新时间：2022-08-25 15:21:23

接口描述

将用户（浏览器）重定向到此接口地址，退出登录并清除认证门户登录态。退出后，CIAM 会将用户重定向到应用的退出回调地址 `logout_redirect_uri` 。

> **说明**
> - 此接口清除的是认证门户的登录态，用户在应用系统的登录态需要由应用自行清除。
> - 本节请求示例中使用的应用系统 Logout Redirect URI 为 `https://example.com/logout` 。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
GET
```

请求路径

```
/logout
```

请求示例

```
GET /logout?client_id=TENANT_CLIENT_ID&logout_redirect_uri=https%3A%2F%2Fexample.com%2Flogout HTTP/1.1
Host: sample.portal.tencentciam.com
```

请求参数

参数	可选	描述
client_id	false	应用的 `client_id` 。可参考 应用管理页面 > 选定指定应用 > 单击应用配置 > 对应的“Client Id”。
logout_redirect_uri	true	退出登录后的重定向地址，需要与应用参数配置中的 `Logout Redirect URI` 匹配。若应用仅配置了一条 `Logout Redirect URI` ，则此参数可以不传，系统将默认使用该条配置作为退出登录后的重定向地址。
state	true	应用随机生成的一个字符串，重定向时将原样返回。

正常响应示例

```
HTTP/1.1 302 Found
Location: https://example.com/logout
```

异常响应示例

client_id 参数缺失

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "error" : "invalid_request",
  "error_description" : "Client ID parameter not found"
}
```

使用认证门户注册

最近更新时间：2021-12-30 14:58:09

接口描述

将用户（浏览器）重定向到此接口地址，发起注册。如果应用开启了“注册后自动登录”，则注册成功后将自动重定向到应用回调地址，具体逻辑同 使用认证门户登录；如果未开启“注册后自动登录”，注册成功后重定向到 CIAM 认证门户登录页面。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
GET
```

请求路径

```
/oauth2/authorize
```

请求示例

```
GET /oauth2/authorize?
scope=openid&client_id=TENANT_CLIENT_ID&redirect_uri=https%3A%2F%2Fexample.com%2Fcallback&response_type=code
&state=MOCK_STATE&code_challenge_method=S256&code_challenge=MOCK_CODE_CHALLENGE&prompt=create HTTP/1.1
Host: sample.portal.tencentciam.com
```

请求参数

参数	可选	描述
scope	false	填固定值 `openid` 。
client_id	false	应用的 `client_id` 。可参考 应用管理页面 > **选定指定应用** > 单击**应用配置** > 对应的“Client Id”。
redirect_uri	false	授权成功后的重定向地址。需要与租户管理平台配置的地址一致。
response_type	false	填固定值 `code` 。
state	true	应用随机生成的一个字符串，服务器会以 HTTP 响应参数的形式原样返回给应用。为防止 CSRF 攻击，建议携带此参数。
code_challenge_method	false	计算 PKCE code_challenge 的算法。目前仅支持固定值 `S256` 。
code_challenge	false	PKCE code_challenge。
prompt	true	此参数存在且取值为 `create` 时代表发起注册请求。其他情况均会被解析为发起登录请求。

正常响应示例

重定向到认证门户用户注册页面。

```
HTTP/1.1 302 Found
Location: https://sample.portal.tencentciam.com/portal/signup?p_state=MOCK_LOGIN_PORTAL_STATE
```

异常响应示例

- client_id 参数缺失或有误。

```
HTTP/1.1 400 Bad Request
```

- redirect_uri 参数与注册信息不匹配。

```
HTTP/1.1 400 Bad Request
```

- response_type 参数缺失或有误。

```
HTTP/1.1 400 Bad Request
```

- 不支持的 code_challenge_method。

```
HTTP/1.1 302 Found
Location: https://example.com/callback?
error=invalid_request&error_description=OAuth%202.0%20Parameter:%20code_challenge_method&error_uri=https://datat
racker.ietf.org/doc/html/rfc7636%23section-4.4.1&state=MOCK_STATE
```

使用认证 API 接入
用户注册

最近更新时间：2022-03-28 11:02:21

接口描述

注册新用户。此接口适用于**应用系统自行开发注册功能的场景**，如果您的应用使用了 CIAM 认证门户，请参考 使用认证门户注册 。

调用此接口前，请确认已配置并启用了应用的注册流程。接口入参需要遵循注册流程中配置的业务规则。例如，注册流程配置了电话号码作为认证属性，用户昵称作为必填普通属性，则入参中必须包含电话号码和用户昵称这两个属性；**如果注册流程未配置电话号码作为认证属性，则入参中不能包含电话号码**。

注册信息中包含电话号码或邮箱地址时，需要先调用 发送 OTP 验证码 接口向用户发送验证码。

密码为可选参数，您可以根据具体业务情况决定是否要求用户设置密码。

- 如果用户仅通过短信 OTP、邮箱 OTP 或社交认证方式登录，可以不设置密码。
- 如需支持用户通过账号密码认证登录，则应设置密码。

如果需要设置密码，请确保应用的登录流程中**关联了账号密码认证源**，接口将根据该认证源的密码策略对传入密码进行校验，如果密码不满足策略要求则无法成功完成注册。

> **说明**
> - 此接口不支持设置用户组。注册成功的用户默认归属注册流程中配置的用户组。
> - 此接口不处理自动登录和实名认证逻辑。即注册流程中的自动登录和实名认证相关配置对此接口不生效。

支持的应用类型

Web 应用。

请求方法

```
POST
```

请求路径

```
/signup
```

请求 Content-Type

```
application/json
```

请求示例

- 使用用户名注册，并设置密码。

```
POST /signup HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
"username" : "MOCK_USERNAME",
"password" : "MOCK_PASSWORD"
}
```

- 使用邮箱和昵称注册，并设置密码。

```
POST /signup HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
"email" : "MOCK_USERNAME@example.com",
"email_otp_token" : "MOCK_EMAIL_OTP_TOKEN",
"email_otp" : "MOCK_EMAIL_OTP",
"password" : "MOCK_PASSWORD",
"nickname" : "MOCK_NICKNAME"
}
```

- **使用电话号码注册，不设置密码。**

```
POST /signup HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
"phone_number" : "13612345678",
"phone_number_otp_token" : "MOCK_PHONE_NUMBER_OTP_TOKEN",
"phone_number_otp" : "MOCK_PHONE_NUMBER_OTP"
}
```

请求头

名称	描述
Authorization	HTTP Basic 认证请求头，格式为 `Basic <credentials>`，其中 `Basic` 为固定字符串，`<credentials>` 的计算方式为 `base64(url_encode(client_id) + ":" + url_encode(client_secret))`，`Basic` 和 `<credentials>` 之间用一个空格隔开。

请求体 JSON 参数

JSON 路径	数据类型	描述
username	String	用户名，可以包含英文字母、数字和下划线，必须以字母开始，最长32个字符。
password	String	用户密码。如果设置，则必须符合应用关联的账号密码认证源的密码策略。
phone_number	String	用户的手机号，限国内三大运营商11位手机号。传递此参数时，须同时传递 `phone_number_otp_token` 和 `phone_number_otp` 两个参数。
phone_number_otp_token	String	发送短信验证码成功后服务端返回的 `otp_token`。
phone_number_otp	String	用户手机收到的 OTP 验证码。
email	String	用户的邮箱地址。传递此参数时，须同时传递 `email_otp_token` 和 `email_otp` 两个参数。
email_otp_token	String	发送邮箱验证码成功后服务端返回的 `otp_token`。
email_otp	String	用户邮箱收到的 OTP 验证码。

name	String	用户姓名。
nickname	String	用户昵称。
zoneinfo	String	用户时区，如 `Asia/Shanghai` 或 `Europe/Paris` 。
locale	String	用户 locale 信息，如 `zh-CN` 或 `en-US` 。

> **说明**
> 其他参数的取值为用户属性标识。属性标识可以在 属性自定义页面 的属性详情界面查看。

注册成功响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json

{
 "sub" : "MOCK_USER_ID"
}
```

响应参数

字段	数据类型	描述
sub	String	用户唯一标识。

注册失败响应示例

- 应用注册流程未启用。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "misconfigured",
"error_description" : "Sign up flow of the application is not enabled."
}
```

- 入参缺少注册流程配置的认证属性或必填普通属性。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Missing required sign-up attribute(s)."
}
```

- 入参包含注册流程未配置的认证属性或普通属性。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Unconfigured sign-up attribute(s) found."
```

```
}
```

- **入参包含未知属性。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Unknown attribute(s) found."
}
```

- **用户名格式不合法。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_username"
}
```

- **用户名已存在。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_username"
}
```

- **电话号码格式不合法。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_phone_number"
}
```

- **电话号码已存在。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_phone_number"
}
```

- `phone_number_otp_token` **错误或已过期，或注册时使用的参数与发送验证码时不一致（例如：手机号不同）。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_phone_number_otp_token"
}
```

- `phone_number_otp` **错误或已过期。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_phone_number_otp"
}
```

- **邮箱格式不合法。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_email"
}
```

- **邮箱已存在。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_email"
}
```

- `email_otp_token` **错误或已过期，或注册时使用的参数与发送验证码时不一致（例如：邮箱不同）。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp_token"
}
```

- `email_otp` **错误或已过期。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp"
}
```

- **入参中传入了密码，但应用登录流程中未关联账号密码认证源。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "misconfigured",
"error_description" : "No password auth source is associated with the application."
}
```

- **密码不满足策略要求。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
```

```
{
"error" : "invalid_password"
}
```

账号密码认证

最近更新时间：2021-12-30 14:58:15

接口描述

校验用户的用户名和密码，获取 Access Token 和 ID Token，完成登录。此接口对应 OAuth 2.0 协议的 Resource Owner Password Credentials 模式。

> **说明**
> 由于用户密码将在用户终端与应用之间传递，请务必使用高度可信的应用调用此接口，并妥善处理密码的传输（例如确保使用了 HTTPS 协议）。在条件允许的情况下，建议优先使用 认证门户登录 。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
POST
```

请求路径

```
/oauth2/token
```

请求 Content-Type

```
application/x-www-form-urlencoded
```

请求示例

```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded
grant_type=password&client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&auth_source_id=MOCK_USERNA
ME_PASSWORD_AUTH_SOURCE_ID&username=MOCK_USERNAME&password=MOCK_PASSWORD&scope=openid
```

请求参数

参数	可选	描述
grant_type	false	填固定值 `password` 。
client_id	false	应用的 `client_id` 。可参考 应用管理页面 > **选定指定应用** > 单击**应用配置** > 对应的“Client Id”。
client_secret	true	应用的 `client_secret` 。可参考 应用管理页面 > **选定指定应用** > 单击**应用配置** > 对应的“client_secret”。 • Web 应用须传递此参数。 • 单页应用和移动 App 不传递此参数。
auth_source_id	false	账号密码认证源 ID。可在控制台的 通用认证源页面 查看。
username	false	用户名。需要使用账号密码认证源配置的认证源属性，如用户名称、电话号码、邮箱地址。
password	false	用户密码。

scope	true	可省略。如传递，则填固定值 `openid` 。

正常响应示例

认证成功

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" :
"eyJraWQiOiI1MzQyOGU3ZS1kOTJiLTQ3OTAtOGIwMC0wMmEyZjc4NjUxNzMiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJN
T0NLX1VTRVJfSUQiLCJhdWQiOiJURU5BTlRfQ0xJRU5UX0lEIiwibmJmIjoxNjQwNTg4ODI2LCJzY29wZSI6WyJvcGVuaWQiXSwiaXNzIjoi
aHR0cHM6XC9cL3NhbXBsZS5wb3J0YWwudGVuY2VudGNpYW0uY29tIiwiZXhwIjoxNjQwNTg5MTI2LCJpYXQiOjE2NDA1ODg4MjYsI
mp0aSI6IjU0ODZhNWRhLTE0YTYtNDNkZC04MWNkLTEwNmRiMDVhMjJmZiJ9.cd2hoQEVZLGcRaqhDLs-
W8m9IN2pDT4XzluxYz4ulNwHTfBTu_1NamlH0kmd0KDhzBMRAl6YXHVDVOXZEV47C1uoYDATnwQetUsg8eJObzlPEVZ81ovU-
eRm7I0lQ_MVI9MC7jMcdOKvEDJCOJtnwtorkQpqEPYovrxctNvvClqLBsDVbUebC2-iUGwz4vWE-
DL17M39rqtmwXv6C21ovv6Pe9BSUUAj_CzHSV-
ZSF7fjyYjnlTEOLjilFNsFD9Ow1czNFqnwxkc7dwugOGJ7qM30zuTgSme76K3tLbol8fKO-
CUKglZO9mWIXUWizhePTzd3CKGE4C7gUHR40EjNw",
  "refresh_token" :
"7uqTlthTQrzlZx8joT20chQbakZp81_iv39GTyCpsEyYpWoquNhuB3s6qEQHGeu2RBpaZcavFf9UOdgtUaCjB42D478MISnj6qBY52q
3Pd5eNBBcXZ68oqMkFVidvgha",
  "scope" : "openid",
  "id_token" :
"eyJraWQiOiI1MzQyOGU3ZS1kOTJiLTQ3OTAtOGIwMC0wMmEyZjc4NjUxNzMiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJN
T0NLX1VTRVJfSUQiLCJhdWQiOiJURU5BTlRfQ0xJRU5UX0lEIiwiYXpwIjoiVEVOQU5UX0NMSUVOVF9JRCIsImlzcyI6Imh0dHBzOlwvXC
9zYW1wbGUucG9ydGFsLnRlbmNlbnRjaWFtLmNvbSIsImV4cCI6MTY0MDU5MDYyNiwiaWF0IjoxNjQwNTg4ODI2LCJqdGkiOiI0Mzg0
OTUwYS0zOTFlLTQ3MDgtYjFjYS05ZjVkNzQzN2Q3ZDIifQ.kaquINkB64dRAaXPG8KLBCpZ6hwNxtA5ruXpwYgEzFHppXQMqPRQQ-
lwn0659lpy8B2CqjRlar_A1xS518Uua3ItkUtoJQXclxLDSGRatk6mZ6q7XgOHvALsLhQlYQjn36kXdnOjipPVB124y-
o20UHwUC27nUH0e9GuYxwX3p56684NB4xJXUfEL2yl9Zt8rMEpD1I1hI5exg75ZP7MCSjzoNgr7fmpGuDNWWhRALh6sY4Dv5BSgE
9B0No0xwFOLnAV4NMsvimJg5rBlJXepOhW4JfwZ3TnzXtDUk3oeajhRxPtjbkxt5NrSYMxUwBggkcoi4eOfVpo8crmQNrSVQ",
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

响应参数

参数	数据类型	描述
access_token	String	OAuth 2.0 Access Token (JWT)。
token_type	String	Token 类型，目前返回的是固定值 `Bearer` 。
expires_in	Number	Access Token 有效期，单位秒。
scope	String	Access Token scope。
refresh_token	String	OAuth 2.0 Refresh Token。
id_token	String	OIDC ID Token (JWT)。

异常响应示例

- 用户名密码错误。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Wrong username or password"
```

```
}
```

- 用户状态异常（如被锁定或冻结）。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Abnormal user status"
}
```

- 使用了认证源不支持的属性作为用户名（例如：username 传入了邮箱地址，但账号密码认证源未配置邮箱地址为认证源属性）。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Unsupported username identifier"
}
```

- 认证源不是应用的首选认证源或关联认证源。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Auth source and application not associated"
}
```

短信和邮箱 OTP 认证

最近更新时间：2023-08-14 15:50:22

接口描述

校验短信或邮箱 OTP 验证码，获取 Access Token 和 ID Token，完成登录。调用此接口前，需要先通过 发送 OTP 验证码 接口向用户发送验证码。

> **说明**
> 可以通过传递 `auto_signup=true` 参数来支持自动注册用户。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
POST
```

请求路径

```
/oauth2/token
```

请求 Content-Type

```
application/json
```

请求示例

短信 OTP 登录

```
POST /oauth2/token HTTP/1.1
Content-Type: application/json
Host: sample.portal.tencentciam.com

{
 "grant_type" : "http://tencentciam.com/oauth2/grant-type/otp/sms",
 "client_id" : "TENANT_CLIENT_ID",
 "client_secret" : "TENANT_CLIENT_SECRET",
 "auth_source_id" : "MOCK_SMS_OTP_AUTH_SOURCE_ID",
 "phone_number" : "13612345678",
 "otp_token" : "MOCK_OTP_TOKEN",
 "otp" : "123456"
}
```

邮箱 OTP 登录

```
POST /oauth2/token HTTP/1.1
Content-Type: application/json
Host: sample.portal.tencentciam.com

{
 "grant_type" : "http://tencentciam.com/oauth2/grant-type/otp/email",
 "client_id" : "TENANT_CLIENT_ID",
 "client_secret" : "TENANT_CLIENT_SECRET",
 "auth_source_id" : "MOCK_EMAIL_OTP_AUTH_SOURCE_ID",
 "email" : "MOCK_USERNAME@example.com",
 "otp_token" : "MOCK_EMAIL_OTP_TOKEN",
```

```
  "otp" : "123456"
}
```

请求体 JSON 参数

JSON 路径	数据类型	描述
grant_type	String	• 短信 OTP 登录输入：http://tencentciam.com/oauth2/grant-type/otp/sms • 邮箱 OTP 登录输入：http://tencentciam.com/oauth2/grant-type/otp/email
client_id	String	应用的 client_id 。需要与发送验证码时使用的一致。
client_secret	String	应用的 client_secret 。Web 应用须传递此参数。单页应用和移动 App 不传递此参数。
auth_source_id	String	短信 OTP 或邮箱 OTP 认证源 ID。需要与发送验证码时使用的一致。
phone_number	String	用户的手机号。需要与发送验证码时使用的一致。短信 OTP 登录时传递此参数。
email	String	用户的邮箱地址。需要与发送验证码时使用的一致。邮箱 OTP 登录时传递此参数。
otp_token	String	发送验证码成功后服务端返回的 otp_token 。
otp	String	用户手机或邮箱收到的 OTP 验证码。
auto_signup	Boolean	如需支持自动注册用户，则此参数传 true，否则可以不传。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" :
"eyJraWQiOiJmZTQ4YTJjYS1lNGU3LTQyMGEtOThjOS01OGM5NmI2NzUwZjIiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNT
0NLX1VTRVJfSUQiLCJhdWQiOiJURU5BTlRfQ0xJRU5UX0lEIiwibmJmIjoxNjQ3NDIwMDM1LCJzY29wZSI6WyJvcGVuaWQiXSwiaXNzIjoi
aHR0cHM6XC9cL3NhbXBsZS5wb3J0YWwudGVuY2VudGNpYW0uY29tIiwiZXhwIjoxNjQ3NDIwMzM1LCJpYXQiOjE2NDc0MjAwMzUsI
mp0aSI6ImMxZmE5Yzk4LThhZjQtNDA1Zi1iOWFhLTdiNTU2MjY1NDljNSJ9.FzvKdLeIgNeYKwQeixIGKX2JPkZ9tJ43fnwuaruLY85RQj9
cMedm9eSU4Ft_h7NJkwH-
eBTmSybg7174RsQ98yOaW77u2flQwxm0xZCx74kY2dOZOf3YhRJwVLVhocMtLC1NrrP3phJSVfYYzClS_ppTnSHcGZhiVzW57YgolTr
0EeuOMucmt1jh_I76kDreo_B5UhV95sRqP_R5FMVBLpGvlAD3TPVCMs3zQETlgHHyq2UE9YBnkNBLK9RzxknRZ0XSnUMxpcPCod4
e7Q7S87QqML2S_3AbcmJlPY5q0D-XTqzyjvS2QByUOUQNOX6pEH4Pe7fV6phVrfXh0lenDQ",
  "refresh_token" : "B-72VlkQa3jQNuo9Xbbl-muoh4w7nYu-7Q3Wb-
qmPgyftN1CgXPov2aWsOBWeelOlVHjVxxHxbOa21Oz0CtlgsIz1LMZ_HG7eLxF-qk6hiRcFzPOcSI8PBsCdd3QXaEd",
  "scope" : "openid",
  "id_token" :
"eyJraWQiOiJmZTQ4YTJjYS1lNGU3LTQyMGEtOThjOS01OGM5NmI2NzUwZjIiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNT
0NLX1VTRVJfSUQiLCJhdWQiOiJURU5BTlRfQ0xJRU5UX0lEIiwiYXpwIjoiVEVOQU5UX0NMSUVOVF9JRCIsImlzcyI6Imh0dHBzOlwvXC9z
YW1wbGUucG9ydGFsLnRlbmNlbnRjaWFtLmNvbSIsImV4cCI6MTY0NzQyMTgzNSwiaWF0IjoxNjQ3NDIwMDM1LCJqdGkiOiI5MGFiMz
ljMi00NjQzLTQwYTEtODdmOC0yN2Q5ODkzOTExMDQifQ.ZqgRcJae_XEUd1XIbu2_pzdgnJCtEehLoCTTHJhEvewOeEnUlfYkRMrpfZ_
hYSVsaWDZy0zdqntWmpmN57eJuw-nfwaUGBUjc1e3KgyvY9jr5vo4zll5O2NJYYMwP8uwwCFsqWjbNl1cl-
dVPu6pIGAvPWBx_Hm1C0vMsPICv61KE7I4bGi_XCSQ--CQjvjzE8ly4I7Z1jCfVl9f4Aybve2HJkuD-
m73nZsgluAGOANXvBLcYi1bj4ncXt9Ybk45Gt_vtlCOY9Ab-N6STm4omtKuxyMQUfy7Rv-
9RXBuvDFldDl6tpENxch1N0V027FdtdWk_JOk9mq97rqI-LycPA",
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

响应参数

字段	数据类型	描述
access_token	String	OAuth 2.0 Access Token (JWT)。

token_type	String	Token 类型，目前返回的是固定值 `Bearer` 。
expires_in	Number	Access Token 有效期，单位秒。
scope	String	Access Token scope。
refresh_token	String	OAuth 2.0 Refresh Token。
id_token	String	OIDC ID Token (JWT)。

异常响应示例

- otp_token 错误或已过期。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Unknown or expired otp_token"
}
```

- otp 错误或已过期。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Unknown or expired OTP"
}
```

- 使用的参数与发送验证码时不一致（例如：手机号不同）。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Mismatched OTP token and OTP sending parameters"
}
```

- 找不到手机号或邮箱对应的用户（不允许自动注册用户的情况下）。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "User not found"
}
```

- 手机号或邮箱对应的用户状态异常（如被锁定或冻结）。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Abnormal user status"
```

```
}
```

- **认证源不是应用的首选认证源或关联认证源。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Auth source and application not associated"
}
```

微信授权登录

最近更新时间：2022-12-05 10:15:49

接口描述

网站、微信公众号或移动 App 使用微信登录获取用户授权的临时票据 code 后，调用此接口验证 code，获取 CIAM 的 Access Token 和 ID Token，完成登录。
CIAM 会使用微信的 unionid 或 openid 关联本地用户 (优先使用 unionid)，如果用户不存在则自动创建新用户。如果登录成功，CIAM 会根据认证源的属性映射配置将微信侧返回的用户信息映射到本地用户数据上。

> **说明**
> - 在使用此接口前，建议您阅读相应的微信登录官方文档，了解详细的登录流程：网站应用微信登录 、 微信公众号网页授权 、 移动应用微信登录 。
> - 微信公众号和移动 App 如需获取用户的昵称、头像等信息，请确保以 `snsapi_userinfo` 为 `scope` 发起授权。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
POST
```

请求路径

```
/oauth2/token
```

请求 Content-Type

```
application/json
```

请求示例

```
POST /oauth2/token HTTP/1.1
Content-Type: application/json
Host: sample.portal.tencentciam.com

{
  "grant_type" : "http://tencentciam.com/oauth2/grant-type/social/wechat/code",
  "client_id" : "TENANT_CLIENT_ID",
  "client_secret" : "TENANT_CLIENT_SECRET",
  "auth_source_id" : "MOCK_WECHAT_AUTH_SOURCE_ID",
  "code" : "MOCK_CODE"
}
```

请求体 JSON 参数

JSON 路径	数据类型	描述
grant_type	String	填固定值 `http://tencentciam.com/oauth2/grant-type/social/wechat/code` 。
client_id	String	应用的 client_id。
client_secret	String	应用的 client_secret。Web 应用须传递此参数；单页应用和移动App 不传递此参数。
code	String	在微信端获取用户授权后得到的授权临时票据 `code` 。

auth_source_id	String	根据实际需求，选择如下任意一种认证源 ID： • PC 微信登录认证源 ID • 微信网页登录认证源 ID • 移动 App 微信登录认证源 ID
getUserInfo	Boolean	如果除了用户的微信 openid 和 unionid 外还需获取昵称、头像等用户信息，则此参数传 `true`，否则可以不传递此参数。 注意：此参数传 `true` 时，如果您使用的是微信公众号或移动 App，请确保以 `snsapi_userinfo` 为 `scope` 发起授权。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" :
"eyJraWQiOiI3NjVkMjImYi05ODE1LTRmZGQtYmU5NS00NzJhOTYzZmQ5YmYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJ
NT0NLX1VTRVJfSUQiLCJhdWQiOiJURU5BTlRfQ0xJRU5UX0lEIiwibmJmIjoxNjQyMDYwMjY0LCJzY29wZSI6WyJvcGVuaWQiXSwiaXNzIj
oiaHR0cHM6XC9cL3NhbXBsZS5wb3J0YWwudGVuY2VudGNpYW0uY29tIiwiZXhwIjoxNjQyMDYwNTY0LCJpYXQiOjE2NDIwNjAyNjQsI
mp0aSI6IjNlYjA3NGE1LTIwMzUtNDk1NS1hYjBmLTM1MjM3MmZjNzFjMiJ9.In1nErmTjH9Vqkd1iqM6mf58276jSbonENi5kvkakDfB-
2Mc08OCm_rn4NoW0DoBu9ojXaI5_g-
DY5_XexznCmgfRKGw2Ey6GNXYFkHiGliEdwTBcfNKfgEZN7V3LuQUcHTlPWj_U5SHumGqwcvOcZPvEJuXMpBPWdZKDeJisMpBqdK
mmEsWgX2hXhzgRIC3j7fNLoE6b-pZEFrmWbfYXbjq0BLhPpbMV8dZtrkfnuTVWh1P6EjagHXjZreSlx6fwqVRvNrmUqA4-
b_CcoJw57g9zDLfWfOdMtSQShlKOSsy_dFVivGsDI9bXiFIUq9Zdah_2qNCU8t4CCDM0TqYug",
  "refresh_token" : "4gCXN_3PuKIXG1MdhqxJWhHGpeC_sjphxwx6AqMsyENU-Hj2rEf94q1x0hkmg5oEEXw39z688sHfox1W-N-
idNvmLpdASAxwGo_Z4zEvyueOOAUApZEJQoXPfu9oCu45",
  "scope" : "openid",
  "id_token" :
"eyJraWQiOiI3NjVkMjImYi05ODE1LTRmZGQtYmU5NS00NzJhOTYzZmQ5YmYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJ
NT0NLX1VTRVJfSUQiLCJhdWQiOiJURU5BTlRfQ0xJRU5UX0lEIiwiYXpwIjoiVEVOQU5UX0NMSUVOVF9JRCIsImlzcyI6Imh0dHBzOlwvX
C9zYW1wbGUucG9ydGFsLnRlbmNlbnRjaWFtLmNvbSIsImV4cCI6MTY0MjA2MjA2NCwiaWF0IjoxNjQyMDYwMjY0LCJqdGkiOiJmYWR
jY2Q4Yi0xMjg1LTRiYTQtOTU0OS02NWVmMTJhYjllOGQifQ.XFhK8OpbqKcEwk_bNUHSSYTKuQpVxQOOgBWrKIK2Gh78ou4Q2l7w-
jXu7urRP_lFiOXuHss13vEByA9aHt54ZJyPSUiRkNOxDDXNyj6JkQlrK2nnFYAPlF9Hjob-
Be8NmcM486l_SMJBGgCNpoXtPwKekkAuYOzomqU2lVrNyKjKK7sTB-
3D6EVqC470eYX6xqHG4ASO8lGLvJMYeRLzbrumf95rsXaWC0iAJj0zxDq66ozqH2R3qj0aggxPR4mfie6VLGUvFv1pOeHDaYiLf6jTrfeI
9FGDS7xdosqPuNGjISjn1GjfvbfJ1r-pQBf-eV53i5zrCwkOSFTwKbSFJQ",
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

响应参数

参数	数据类型	描述
access_token	String	OAuth 2.0 Access Token (JWT)。
token_type	String	Token 类型，目前返回的是固定值 `Bearer`。
expires_in	Number	Access Token 有效期，单位秒。
scope	String	Access Token scope。
refresh_token	String	OAuth 2.0 Refresh Token。
id_token	String	OIDC ID Token (JWT)。

异常响应示例

- code 无效。

```
HTTP/1.1 400 Bad Request
```

```
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "WeChat error: 40029"
}
```

- **用户状态异常（如被锁定或冻结）。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Abnormal user status"
}
```

- **认证源不存在或未启用。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Auth source not found or disabled"
}
```

- **认证源不是 `微信小程序内登录` 认证源。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Wrong auth source type"
}
```

- **认证源未关联应用。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Auth source and application not associated"
}
```

微信小程序登录（API 模式）

最近更新时间：2022-12-05 10:15:32

接口描述

微信小程序应用通过 `wx.login()` 获取微信端的临时登录凭证 `code` 后，调用此接口验证 `code`，获取 CIAM 的 Access Token 和 ID Token，完成登录。
CIAM 会使用微信的 `unionid` 或 `openid` 关联本地用户 (优先使用 `unionid`)，如果用户不存在则自动创建新用户。如果登录成功，CIAM 会根据认证源的属性映射配置将微信侧返回的用户信息映射到本地用户数据上。

> **说明**
> - 您还可以通过 SDK 方式更加简单快捷地接入微信小程序，详情请参见 接入微信小程序（SDK 模式）。
> - 在使用此接口前，建议您阅读 微信小程序登录官方文档 了解小程序登录的详细流程。

支持的应用类型

小程序应用。

请求方法

```
POST
```

请求路径

```
/oauth2/token
```

请求 Content-Type

```
application/json
```

请求示例

```
POST /oauth2/token HTTP/1.1
Content-Type: application/json
Host: sample.portal.tencentciam.com

{
  "grant_type" : "http://tencentciam.com/oauth2/grant-type/social/wechat/jscode",
  "client_id" : "WECHAT_MINI_PROGRAM_CLIENT_ID",
  "auth_source_id" : "WECHAT_MINI_PROGRAM_AUTH_SOURCE_ID",
  "js_code" : "MOCK_JS_CODE"
}
```

请求体 JSON 参数

JSON 路径	数据类型	描述
grant_type	String	填固定值 `http://tencentciam.com/oauth2/grant-type/social/wechat/jscode`。
client_id	String	小程序应用的 client_id。
auth_source_id	String	微信小程序内登录认证源 ID。
js_code	String	微信小程序 wx.login() 接口返回的临时登录凭证 code。

getUserInfo	Boolean	如需获取昵称、头像等用户信息，则此参数传 `true`，否则可以不传递此参数。此参数传 `true` 时，需要同时传递 `encryptedData` 和 `iv` 两个参数。
encryptedData	String	wx.getUserProfile() 接口返回的用户信息加密数据 encryptedData。
iv	String	wx.getUserProfile() 接口返回的加密算法初始向量 iv。
getPhoneNumber	Boolean	如需获取用户手机号，则此参数传 `true`，否则可以不传递此参数。此参数传 `true` 时，需要同时传递 `phone_code` 参数。
phone_code	String	微信小程序获取手机号接口返回的动态令牌 code。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" :
"eyJraWQiOiI3NjVkMjImYi05ODE1LTRmZGQtYmU5NS00NzJhOTYzZmQ5YmYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJ
NT0NLX1VTRVJfSUQiLCJhdWQiOiJXRUNIQVRfTUlOSV9QUk9HUkFNX0NMSUVOVF9JRCIsIm5iZiI6MTY0MjA2MDI2Mywic2NvcGUiOls
ib3BlbmlkIl0sImlzcyI6Imh0dHBzOlwvXC9zYW1wbGUucG9ydGFsLnRlbmNlbnRjaWFtLmNvbSIsImV4cCI6MTY0MjA2MDU2MywiaW
F0IjoxNjQyMDYwMjYzLCJqdGkiOiI1YzQ2YmM5Mi0yYzg0LTQxOTEtOWZmMC1kNjdjYmIwMTk3MDIifQ.wMRbMeeGxtrX--
unHY9Qi8X7anz_STKseO0dApY9bC5-
gh0_76QmNHKFTibU0XZeMg1yEq0ewApBN4tlTVJ1SzYeNgxNfWGGiAxFxKzHBS06Tli1p0ByiFl-
gPccVb4S0Bn4T6p1KDmQSchV07O4PB3gjSjLKmqHT4qVO2OfCm47kfSxn31QsCF9BoJdIBJ43ic8hk2LFJ5gTSPBeZM8Dr5BcaUbScl
JpLulQNuwmax7gQnw7jpeDGAGOPVih-YbRgXST-
es0TIhUToNt0V_p9ytw1Ck4PxSbvUy6emNH70oopoNXLazc6vpX1bbBweQ7oaDfXkoYq5ZrV8On2qwPA",
  "scope" : "openid",
  "id_token" :
"eyJraWQiOiI3NjVkMjImYi05ODE1LTRmZGQtYmU5NS00NzJhOTYzZmQ5YmYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJ
NT0NLX1VTRVJfSUQiLCJhdWQiOiJXRUNIQVRfTUlOSV9QUk9HUkFNX0NMSUVOVF9JRCIsImF6cCI6IldFQ0hBVF9NSU5JX1BST0dSQU
1fQ0xJRU5UX0lEIiwiaXNzIjoiaHR0cHM6XC9cL3NhbXBsZS5wb3J0YWwudGVuY2VudGNpYW0uY29tIiwiZXhwIjoxNjQyMDYyMDYzLC
JpYXQiOjE2NDIwNjAyNjMsImp0aSI6IjA2OGNhZGVlLTAzYTMtNDVmZC05ZWJhLTU4MjZiZmU1ZDljOCJ9.D-
MxjT5D2qB77vtQClDRnetfI4ZuRyXgm82FwlP4t25gNDlcAsNKc3dIfVH-
gR46J6Dy3p8B1RohSzoMWPvqx0RbusBaMSR4GgeYxjlHQ5cXG2NIPoxZMTZ6LU5skDNlgI3zf7Ma1lfwkMDBVjNLjDJMM4LtlntJ9tK5
WD1ti8lBuJmeCtAnatDY_5_fB4_zBhg14LEc3LSMuVTupxe08QTe0csw_lREswsVN3ugBBY7ejMmCxxqdu_CngTDbZ-
JyIq4dAzwKlWTz0T5BpQDX3gjDyrOjONjX7kzocH2t6qB5TSukv4eliNIJncDzkCV9fm-N1xSBqlTRxzys6x_5A",
  "token_type" : "Bearer",
  "expires_in" : 299
}
```

响应参数

参数	数据类型	描述
access_token	String	OAuth 2.0 Access Token (JWT)。
token_type	String	Token 类型，目前返回的是固定值 `Bearer`。
expires_in	Number	Access Token 有效期，单位秒。
scope	String	Access Token scope。
refresh_token	String	OAuth 2.0 Refresh Token。
id_token	String	OIDC ID Token (JWT)。

异常响应示例

- js_code 无效。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "WeChat error: 40029"
}
```

- 用户状态异常（如被锁定或冻结）。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant",
"error_description" : "Abnormal user status"
}
```

- 认证源不存在或未启用。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Auth source not found or disabled"
}
```

- 认证源不是 `微信小程序内登录` 认证源。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Wrong auth source type"
}
```

- 认证源未关联应用。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_auth_source",
"error_description" : "Auth source and application not associated"
}
```

- 指定了获取用户信息，但缺少必要的参数。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Missing required parameter: encryptedData"
}
```

- 指定了获取用户手机号，但缺少必要的参数

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Missing required parameter: phone_code"
}
```

接入微信小程序（SDK 模式）

概述

最近更新时间：2022-06-14 17:35:53

账号风控平台小程序 SDK（ ciam-miniapp-sdk ）适用于在微信小程序环境下使用，以 ciam-js-sdk 为基类，对微信小程序环境做了适配与实现。在小程序环境下，可以使用通过微信授权获取用户手机号、 使用微信授权登录、使用微信授权的手机号登录等方法。小程序 SDK 的相关操作如下：

- 准备工作
- 平台配置
- SDK 集成开发

准备工作

最近更新时间：2023-08-14 15:50:22

操作场景

为了在小程序中使用账号风控平台 SDK，需要先在微信开放平台申请一个小程序，并在 账号风控平台控制台 内填入该小程序的配置。

操作步骤

步骤1：在微信开放平台完成小程序应用申请

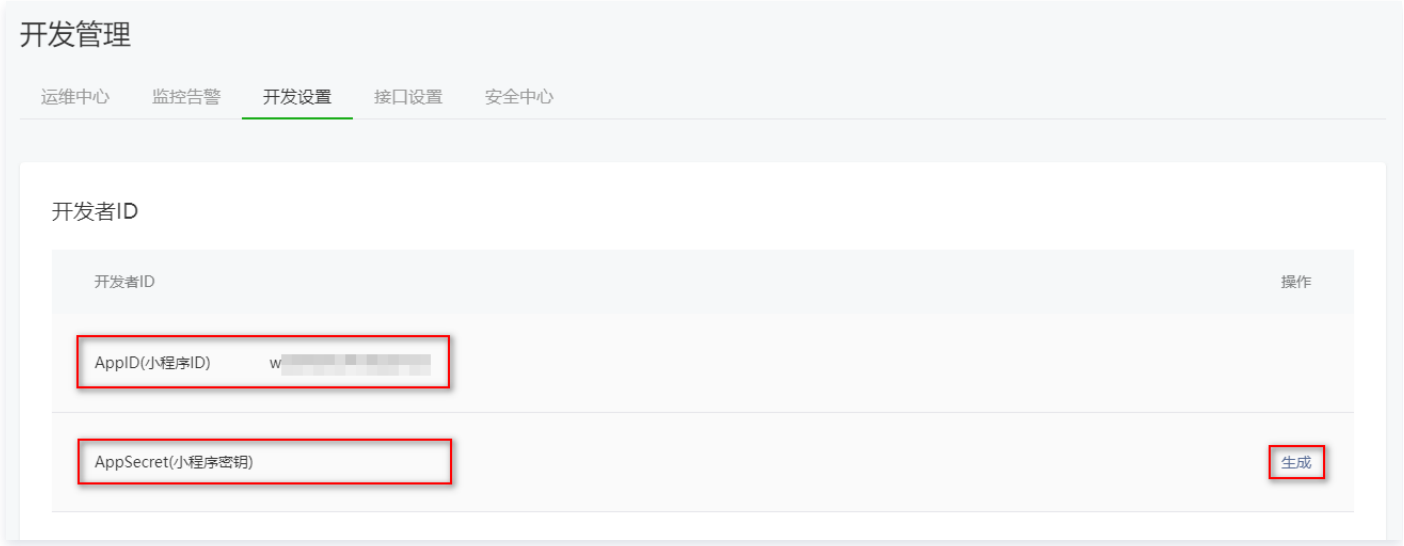
进入 微信开放平台 注册一个微信小程序账号。

> **说明**
> 如果您需要获取用户手机号，需要通过微信认证。

步骤2：获取微信小程序 AppId 和 AppSecret

1. 进入开发管理页面，单击**开发设置**，进入开发设置页面。
2. 在开发者 ID 模块中，获取微信小程序 AppId 和 AppSecret。

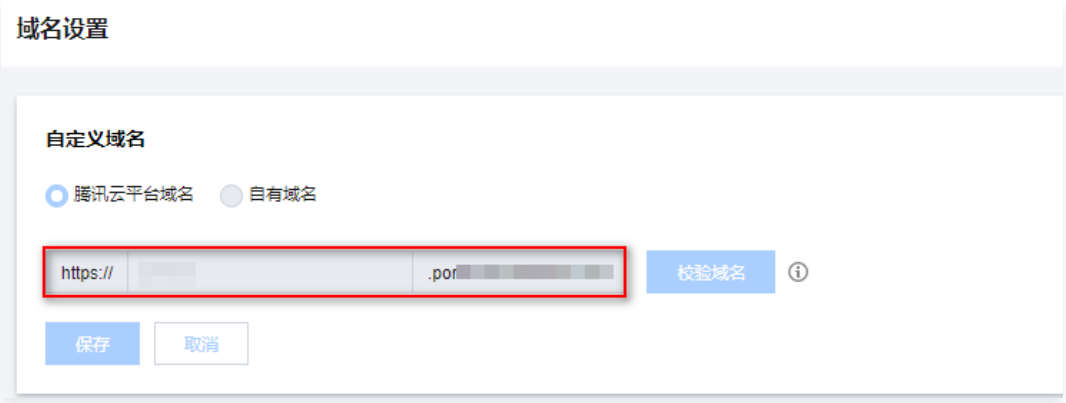
> **说明**
> 首次生成 AppSecret ，单击**生成**，使用管理员微信扫描二维码生成 AppSecret。



步骤3：配置微信的服务器域名和业务域名

配置服务器域名

1. 登录 账号风控平台控制台，在左侧导航栏，选择**个性化设置** > **域名设置**,进入域名设置页面。
2. 在域名设置页面，获取腾讯云平台域名。



3. 进入开发管理页面，单击**开发设置**，进入开发设置页面。
4. 在服务器域名模块中，单击**开启配置**，按照页面提示，将已获取的域名配置到微信公众平台的服务器域名中。

配置业务域名（可选）

1. 进入开发管理页面，单击**开发设置**，进入开发设置页面。
2. 在业务域名配置模块中，单击**下载校验文件**，将下载的文件上传至 CIAM，详情请参见 平台配置 。

步骤4：将微信小程序绑定到微信开放者平台（可选）

如果需要获取 UnionID 字段信息，需要将微信小程序绑定至微信开放者平台中。


微信·开放平台
首页
资源中心
管理中心
帐号中心
通知
退出
移动应用
网站应用
公众帐号
小程序
第三方平台
应用类目
绑定小程序
相同主体：上限50个，绑定次数不限。
不同主体：上限5个，本月还可以绑定5次。
小程序
帐号类型
操作
CIAM
CIAM demo
小程序
查看

平台配置

最近更新时间：2022-11-02 15:56:14

操作场景

为了在小程序中使用账号风控平台 SDK，需要先在微信开放平台申请一个小程序，并在 账号风控平台控制台 内填入该小程序的配置。

操作步骤

步骤1：添加和配置小程序认证源

1. 登录 账号风控平台控制台，在左侧导航栏，选择**认证管理** > **社交认证源**,进入社交认证源页面。
2. 在社交认证源页面，单击**新建认证源**，进入新建认证源页面。
3. 在新建认证源页面，选择微信小程序内登录，单击**下一步**。

4. 在新建认证源页面，填写所需内容，单击**确定**。

> **说明**
> 其中 AppId 、 AppSecret 和业务域名校验文件，敬请参考 准备工作 。

步骤2：添加和配置小程序应用

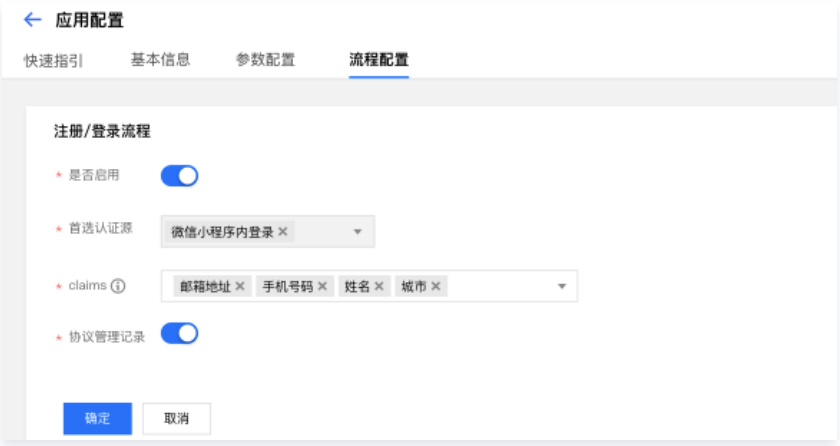
1. 登录 账号风控平台控制台，在左侧导航栏选择**应用管理**，进入应用管理页面。
2. 在应用管理页面，单击操作列的**新建应用**，弹出新建应用弹窗。
3. 在新建应用弹窗中，输入所需内容后，单击**确定**，即可创建新应用。

> **注意**
> 页面标*的为必填项。

4. 在应用管理页面，选择已创建的应用，单击操作列的**配置**，进入应用配置的基本信息页面。

> **说明**
> 应用状态为关闭，才能进行配置和删除操作。

5. 单击**参数配置**，切换到流程配置的参数配置页面。
6. 在参数配置页面，选择登录/注册流程模块，单击**编辑**，将首选认证源配置为小程序认证源，配置 claims 参数。



7. 单击**确定**，保存配置。

SDK 集成开发
准备开发配置

最近更新时间：2022-06-14 17:35:49

操作场景

本文档介绍将账号风控平台的 SDK 集成到小程序应用中，需要安装的工具。

安装和配置开发者工具

开发者工具 1.06.2201240 及以上，小程序支持使用 npm 安装第三方包，详情请见 npm 支持 文档 。

开发者工具本地配置

设置小程序基础库版本 2.21.2或以上， 用户授权使用 wx.getUserProfile 接口，详情可访问 getUserProfile 勾选使用 npm。

安装小程序 SDK

最近更新时间：2023-08-14 15:50:22

操作背景

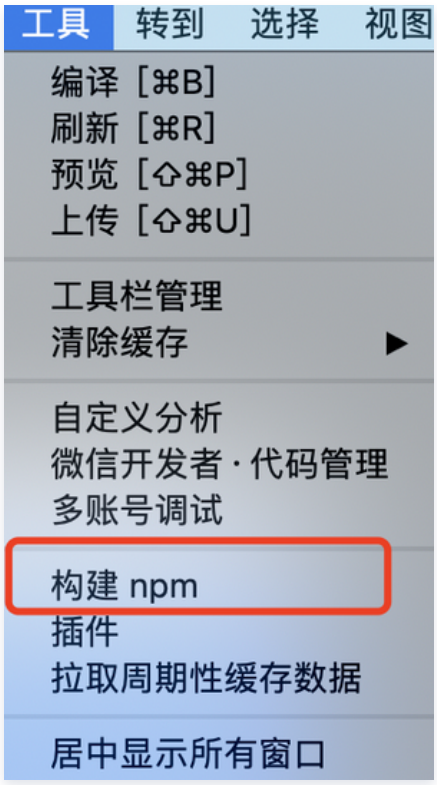
本文档介绍如何在账号风控平台中安装小程序 SDK。

操作步骤

1. 打开小程序项目工程，在工程根目录下输入以下命令：

```
yarn add ciam-miniapp-sdk
// 添加babel编译
yarn add regenerator-runtime -SD
```

2. 安装成功后在小程序开发者工具中，单击**工具** > **构建 npm**，成功构建 npm 后，即可调用 SDK 中的方法 。



使用 SDK 完成集成开发

最近更新时间：2022-01-24 19:43:46

操作背景

本文档介绍如何将账号风控平台的 SDK 集成到小程序应用中。

> **说明**
> 推荐下载 小程序 Demo 实现快速开发。

相关示例

SDK 实例化

```
import { WXAppletClient }  from 'ciam-miniapp-sdk';
const authSDK = new WXAppletClient({
    clientId: 'your-clientId',
    authSourceId: 'your-authSourceId',
    userDomain: 'your-userDomain',
       agreements: [
     {
       "name": "用户协议|隐私协议|附加协议|...", // 必传
       "url": "https://qq.com", // 必传，最大长度1024
       "version": "1.0" // 必传，最大长度10
     },
     {
       "name": "用户协议|隐私协议|附加协议|...", // 必传
       "url": "https://baidu.com", // 必传，最大长度1024
       "version": "2.0.0" // 必传，最大长度10
     }
    ]
   });
```

初始化参数说明

参数名	类型	是否必填	长度限制	描述
clientId	string	是	-	管理端添加的小程序应用 ID
authSourceId	string	是	-	管理端添加的小程序认证源 ID
userDomain	string	是	-	租户域名（自定义域名获取）
agreements	array[Agreement]	否	-	如协议流程开启则开发者需要上传协议签署

Agreement 参数说明

参数名	类型	是否必填	长度限制	描述
name	string	是	100	协议名称，如用户协议、隐私协议、附加协议
url	string	是	1024	协议链接地址
version	string	是	10	协议版本号

静默授权登录 loginCode

```
async bindLoginBySlience() {
   try {
     await authSDK.loginCode();
   }
```

```
  catch(err) {
    console.log('bindLoginBySlience error', err)
  }
  const userInfo = authSDK.getUser();
  if (userInfo) {
      ...
    }
},
```

参数说明

无

返回值

类型为 Promise[User|null]，User 结构如下：

```
{
    sub: "69bc2a4d-e575-41cf-bbc6-996e0911e2ec",
    username: "xxx",
    name: "xxx",
    gender: "female",
    phoneNumber: "+86-13612345678",
    email: "xxx@qq.com",
    nickname: "xxxx",
    wechatUnionId: "xxxxx",
    wechatOpenid: "xxxxx"
}
```

用户信息授权登录 loginUser

> **注意**
> 该方法需要页面产生点击事件（例如：button 上 bindtap 的回调中）后才可调用。详情参见：获取用户信息 。

```
  async bindLoginByUserInfo() {
    try {
      await authSDK.loginUser();
    }
    catch(err) {
      console.log('bindLoginByUserInfo error', err)
    }
    const userInfo = authSDK.getUser();
    if (userInfo) {
      ...
    }
},
```

参数说明

无

返回值

类型为 Promise[User|null]，User 结构如下：

```
  {
    sub: "69bc2a4d-e575-41cf-bbc6-996e0911e2ec",
    username: "xxx",
    name: "xxx",
    gender: "female",
    phoneNumber: "+86-13612345678",
    email: "xxx@qq.com",
    nickname: "xxxx",
    wechatUnionId: "xxxxx",
    wechatOpenid: "xxxxx"
```

```
}
```

用户手机号授权登录 loginPhone

> **注意**
> - 该方法需要页面产生点击事件（例如：button 上 bindtap 的回调中）后才可调用。详情参见： 获取手机号 。
> - 该方法已更新为官方最新版本的获取手机号接口，开发时请检查开发者工具是否更新以及小程序基础库版本是否在2.21.2以上，暂不支持旧版本调用。
> - 官方文档详情参见： 获取手机号 。

```
async bindLoginByPhone(e) {

  const {code} = e.detail;
  if (!code) {
    console.error('未获取到code,请检查开发者工具是否更新以及小程序基础库版本是否在2.21.2以上');
    return;
  }
  try {
    await authSDK.loginPhone(code);
  }
  catch(err) {
    console.log('bindLoginByPhone error', err)
  }

  const userInfo = authSDK.getUser();

  if (userInfo) {
    ...
  }

 },
```

参数说明

无

返回值

类型为 Promise[User|null]，User 结构如下:

```
  {
    sub: "69bc2a4d-e575-41cf-bbc6-996e0911e2ec",
    username: "xxx",
    name: "xxx",
    gender: "female",
    phoneNumber: "+86-13612345678",
    email: "xxx@qq.com",
    nickname: "xxxx",
    wechatUnionId: "xxxxx",
    wechatOpenid: "xxxxx"
}
```

获取登录状态 checkLoginState

```
  const isLogin = authSDK.checkLoginState();
    if (!isLogin) {
      return;
  }
```

参数说明

无

返回值
类型为 Boolean，返回 true/false。

退出登录 logout

```
await authSDK.logout();
```

参数说明
无
返回值
类型为 Boolean，返回 true/false。

发送 OTP 验证码

最近更新时间：2022-09-27 16:55:01

接口描述

向用户发送短信或邮箱 OTP 验证码，用于登录、注册或更新用户信息。

支持的应用类型

Web 应用、M2M 应用。

请求方法

```
POST
```

请求路径

```
/otp/send
```

请求 Content-Type

```
application/json
```

请求示例

- 短信 OTP 登录场景，发送短信验证码用于登录。

```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
"usage" : "login",
"phone_number" : "13612345678",
"auth_source_id" : "MOCK_SMS_OTP_AUTH_SOURCE_ID"
}
```

- 邮箱 OTP 登录场景，发送邮箱验证码用于登录。

```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "login",
"email" : "MOCK_USERNAME@example.com",
"auth_source_id" : "MOCK_EMAIL_OTP_AUTH_SOURCE_ID"
}
```

- 用户注册场景，发送短信验证码用于绑定手机。

```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
```

```
Host: sample.portal.tencentciam.com

{
"usage" : "signup",
"phone_number" : "13612345678"
}
```

- **用户注册场景，发送邮箱验证码用于绑定邮箱。**

```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "signup",
"email" : "MOCK_USERNAME@example.com"
}
```

- **更新用户信息场景，发送短信验证码绑定或更新手机号。**

```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "update_userinfo",
"phone_number" : "13612345678"
}
```

- **重置密码场景，发送邮箱验证码。**

```
POST /otp/send HTTP/1.1
Content-Type: application/json
Authorization: Basic Q0xJRU5UXzRfSUQ6Q0xJRU5UXzRfU0VDUkVU
Host: sample.portal.tencentciam.com

{
"usage" : "reset_password",
"email" : "MOCK_USERNAME@example.com"
}
```

请求头

名称	描述
Authorization	HTTP Basic 认证请求头，格式为 `Basic <credentials>`，其中 `Basic` 为固定字符串，`<credentials>` 的计算方式为 `base64(url_encode(client_id) + ":" + url_encode(client_secret))`，`Basic` 和 `<credentials>` 之间用一个空格隔开。

请求体 JSON 参数

JSON 路径	数据类型	描述
usage	String	OTP 验证码的使用场景。 • 短信和邮箱 OTP 登录场景输入 `login`。 • 用户注册场景输入 `signup`。 • 更新用户信息场景输入 `update_userinfo`。

		• 重置用户密码场景输入 `reset_password` 。 • 如果没有输入参数，默认代表登录场景。
phone_number	String	用户的手机号，限国内三大运营商11位手机号。发送短信 OTP 验证码时传递此参数。
email	String	用户的邮箱地址。发送邮箱 OTP 验证码时传递此参数。
auth_source_id	String	短信 OTP 或邮箱 OTP 认证源 ID。可在控制台的通用认证源列表页面查看。短信和邮箱 OTP 登录场景传递此参数，系统将使用认证源配置的验证码长度和有效期。其他场景不传递此参数，系统默认使用6位数字验证码，有效期60秒。

正常响应示例

验证码发送成功。

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "otp_token" : "MOCK_OTP_TOKEN"
}
```

响应参数

字段	数据类型	描述
otp_token	String	OTP token，后续验证 OTP 时携带使用。有效期5分钟。

异常响应示例

- 手机号格式有误。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_phone_number"
}
```

- 邮箱格式有误。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_email"
}
```

- 因短信额度不足无法发送短信，一般是由于免费短信额度已用尽，需要到控制台配置 短信模板 。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "insufficient_sms_quota"
}
```

- 因邮箱额度不足无法发送邮件，一般是由于免费邮箱额度已用尽，需要到控制台配置 邮箱模板 。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "insufficient_email_quota"
}
```

- 邮箱地址不存在或在黑名单中。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_email"
}
```

- 验证码发送失败。

```
HTTP/1.1 503 Service Unavailable
Content-Type: application/json;charset=UTF-8

{
"error" : "temporarily_unavailable",
"error_description" : "Failed to send OTP. Please try again later."
}
```

- 注册场景，邮箱被使用。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
  "error": "email_is_used"
}
```

- 注册场景，手机号被使用。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8


{
  "error": "phone_number_is_used"
}
```

- 向单个手机号发送短信频率超限。
  - 如果使用的是自购短信服务，可自行到 短信控制台 调整短信频率限制策略。
  - 如果使用的是免费短信额度，频率限制为：对同一手机号，每个自然日内发送短信条数不超过50条；相同内容短信对同一手机号，30秒内发送短信条数不超过1条。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "sms_rate_limit_exceeded",
"error_description" : "SMS rate limit exceeded for same phone number"
}
```

获取 Token
PKCE 授权码模式

最近更新时间：2022-03-28 11:05:14

接口描述

应用系统通过 PKCE 授权码模式获得认证门户返回的 `code` 之后，调用此接口获取 Access Token 和 ID Token，完成登录。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
POST
```

请求路径

```
/oauth2/token
```

请求 Content-Type

```
application/x-www-form-urlencoded
```

请求示例

```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

client_id=TENANT_CLIENT_ID&grant_type=authorization_code&code=MOCK_CODE&redirect_uri=https%3A%2F%2Fexample.com%2Fcallback&code_verifier=MOCK_CODE_VERIFIER
```

请求参数

参数	可选	描述
client_id	false	应用的 `client_id` 。需要与获取授权时使用的一致。
grant_type	false	填固定值 `authorization_code` 。
code	false	获取授权时返回的授权码。
redirect_uri	false	授权成功后的重定向地址。需要与获取授权时指定的地址一致。
code_verifier	false	PKCE code_verifier 。需要与获取授权时用于生成 code_challenge 的 code_verifier 一致。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" :
"eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNT0NLX1VTRVJOQU1FIiwiYXVkIjoiVEVOQU5UX0NMSUVOVF9JRCIsIm5iZiI6MTYzNjQ0OTIzMiwic2NvcGUiOlsib3BlbmlkIl0sImlzcyI6Imh0dHBzOlwvXC9URU5BTlQuUE9SVEFMLkRPTUFJTiIsImV4cCI6MTYzNjQ0OTUzMiwiaWF0IjoxNjM2NDQ5MjMyLCJqdGkiOiI0NjkyNTUxMC1mNjY0LTQzNTktODIyYS1jMTdiNTlmNzNhOGUifQ.mmM6iEiGCLIURqaKaJV_LbddUP1i5wCJMJvuasM8i6Wu_Ynix0W_EeghvMcQ94QvLhNYq2KshGQlkl0N5186KCqpHpG6z2ZXbuP35oY4yRFNvhqWOt8drvyxw13aVfehk1_KPLLDgrKGmHTUgxNDvssQq1u6Xd7QxPz0_d0jnaosl78pIO_tV-auGMhYQo6SHHMbFHgJLYBlPUq81eBknqbu8W9Omr4FuDmzlr9VFI4grJ_guxlUuri8lx-C4mRtSbg6bfUYlH7PuAM8bDfaOZ_qhAQ9-KTYF-ZiShDnuJMlVz0u_97ky5kNm_IUOrH6XzWfGL8MboYLagxOHmzNMQ",
  "refresh_token" : "8FuXWpwMZI9oA8ASvCUrqap61N7RvPON6DjWFk-Saiv4dOR8y2tNf9eKf36woAaWYKwW99bpBAQVNWA7P8yM9jiBiGcix42ttYzvRoeMoEBoqYInBgnNMC8jTRTrKDEq",
```

```
  "scope" : "openid",
  "id_token" :
"eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNT
0NLX1VTRVJOQU1FIiwiYXVkIjoiVEVOQU5UX0NMSUVOVF9JRCIsImF6cCI6IlRFTkFOVF9DTElFTlRfSUQiLCJpc3MiOiJodHRwczpcL1wv
VEVOQU5ULlBPUlRBTC5ET01BSU4iLCJleHAiOjE2MzY0NTEwMzIsImlhdCI6MTYzNjQ0OTIzMiwianRpIjoiMGMzODNiZTktOGFiNy00Yz
EwLTg5NWQtMzYwNjgzODg3MmZiIn0.i4Zywl6O5KF7iiivV-8d4Yok7CZr_eQNI8mTS3BkaRCIKiMzXJ55-
T55XEonOViUE7s_Z4eMlyInm5-oLmk36NXrkO460LHEwxr8o5BlAnMhC4bd7xX3U3JrQISi6CpJxEn0UXXfJrtHnmR-
yxAGNLFkoijM_qV1KWe6Y_OxxKe4FPfM2PwjYACt-
XQgs4JsJOQk_UiSnHnvyvbpWTB8ZZrIlwwxrNErZxdr09HBWhsQQ5fjJNviSilNLKD5fYYMz0yhl-
YxDgMJ7s9tnfpDsNXyX25VpFtjdL4L13d1VAMPs2F5fTFBHX-p9LjoqF2slJFEBbapgOX5EO-E_v1IFQ",
  "token_type" : "Bearer",
  "expires_in" : 299
}
```


响应参数

参数	数据类型	描述
access_token	String	OAuth 2.0 Access Token (JWT)。
refresh_token	String	OAuth 2.0 Refresh Token。
scope	String	Access Token 的 Scope。
id_token	String	OIDC ID Token (JWT)。
token_type	String	Token 类型，目前取固定值 `Bearer` 。
expires_in	Number	Access Token 有效期，单位秒。

> **说明**
> CIAM 返回的是 JWT 格式的 ID Token，请参考 OIDC 官方文档 对 ID Token 进行解密验证。也可以直接使用相关的开发库完成解密验证。验证所需的公钥通过调用 获取 JWT 公钥 接口获得。

异常响应示例

- client_id 参数缺失或有误。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8
{
"error" : "invalid_request"
}
```

- client_id 与获取授权和获取 Token 时使用的不一致。

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8
{
"error" : "invalid_client"
}
```

- grant_type 参数有误。

```
HTTP/1.1 401 Unauthorized
```

- code 参数有误。

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8
```

```
{
"error" : "invalid_client"
}
```

- code_verifier 参数有误。

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8
{
"error" : "invalid_client"
}
```

普通授权码模式

最近更新时间：2023-08-14 15:50:22

接口描述

应用系统通过普通授权码模式获得认证门户返回的 `code` 之后，调用此接口获取 Access Token 和 ID Token，完成登录。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
POST
```

请求路径

```
/oauth2/token
```

请求 Content-Type

```
application/x-www-form-urlencoded
```

请求示例

```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&grant_type=authorization_code&code=MOCK_CODE&re
direct_uri=https%3A%2F%2Fexample.com%2Fcallback
```

请求参数

参数	可选	描述
client_id	false	应用的 `client_id` 。需要与获取授权时使用的一致。
client_secret	false	应用的 `client_secret` 。可参考 应用管理页面 > **选定指定应用** > 单击**应用配置** > 对应的“client_secret”。
grant_type	false	填固定值 `authorization_code` 。
code	false	获取授权时返回的授权码。
redirect_uri	false	授权成功后的重定向地址。需要与获取授权时指定的地址一致。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" :
"eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNT
0NLX1VTRVJOQU1FIiwiYXVkIjoiVEVOQU5UX0NMSUVOVF9JRCIsIm5iZiI6MTYzNjQ0OTIzMiwic2NvcGUiOlsib3BlbmlkIl0sImlzcyI6Im
h0dHBzOlwvXC9URU5BTlQuUE9SVEFMLkRPTUFJTiIsImV4cCI6MTYzNjQ0OTUzMiwiaWF0IjoxNjM2NDQ5MjMyLCJqdGkiOiJmYmM0N
WQ3NS1lYmRjLTQzNjUtOWU0MS01YWM1OTg5ZDdhZTIifQ.SnBaXHhZ0jz3sbo-FPcO91YJn2LqOAuBxWTFRdIzl-
```

```
PChdh4oXdaxmlWIcaLPU-
niBo9TqAAmeocnomkTgt6TS2hp5IOjX8FihshlgVdagQNYM6__Zr5NHw9DH2zEps0AGA_7pyGg9trbcRBkjb2xRLyJpQ5lPkpGiNKA18
SfCcsBBoy9E69wrZCZaKo3Y6iHO9v5qxlOTchajR5FI5VDZlxDLX9H3njf3C-
KG5NlB7VQZBa0O4TZJm6od_sh8eCmCa5TKF7s5Zhw7JU83JfEh76WKT3rD5uEb4SbFhBGP3hL7Xsj3SJMP7nA4LlQAlINWyFzntAluJ
1VryH3JmQCg",
  "refresh_token" : "Ugvo1lO7Se8vvIPrlOwn_eBe0hoi5-5ynR3H-aFYl0e1Gej-
SfUAaBDBXkWmojm_Q7utcpwz_swzLZwGGz_I5wHAkB6iaepCX3LvqZGvt5hmMXDCt3wZY3SnhridZC85",
  "scope" : "openid",
  "id_token" :
"eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNT
0NLX1VTRVJOQU1FIiwiYXVkIjoiVEVOQU5UX0NMSUVOVF9JRCIsImF6cCI6IlRFTkFOVF9DTElFTlRfSUQiLCJpc3MiOiJodHRwczpcL1wv
VEVOQU5ULlBPUlRBTC5ET01BSU4iLCJleHAiOjE2MzY0NTEwMzIsImlhdCI6MTYzNjQ0OTIzMiwianRpIjoiMDBlOWQxMjYtYmJkYi00Zjg
zLWI4NTYtNTAyYjI0OWJmNjVmIn0.enLzJ-pJlnKpFCBR0gQc7bJH2GA1q_PpqWNjxJuTE1rWuQzvH9Y0oNNi-
Wc1vUQ1u0hPSAC7in4E8vUpWDuDFyXrKWvwNxxj5uyLfFrAc2hnrju0ZUOT58lOqYd0Y48hA6THuxm9aA_EuOHvR3SxIS0cj_O5xJC
wvhSqGUldRXGVgB4yMXXvZWkxWm5Q3B_hJc8aZtLpmo76-
AKTz2OQouZ5bDE3WD89THuKN4mVxygyyIGTgpuleyP1x14aTgvNKp2N-HGRoUwCzks_fLMcxmH5vqkI5alMi-
XDfG8LWWhObM9j54oShSTpkok51B9VGTTOwbK4ZXeVRI48sCxWmg",
  "token_type" : "Bearer",
  "expires_in" : 299
}
```


> **说明**
> CIAM 返回的是 JWT 格式的 ID Token，请参考 OIDC 官方文档 对 ID Token 进行解密验证。也可以直接使用相关的开发库完成解密验证。 验证所需的公钥通过调用 获取 JWT 公钥 接口获得。

响应参数

参数	数据类型	描述
access_token	String	OAuth 2.0 Access Token (JWT)。
token_type	String	Token 类型，目前取固定值 `Bearer` 。
expires_in	Number	Access Token 有效期，单位秒。
scope	String	Access Token 的 Scope。
refresh_token	String	OAuth 2.0 Refresh Token。
id_token	String	OIDC ID Token (JWT)。

> **说明**
> CIAM 返回的是 JWT 格式的 ID Token，请参考 OIDC 官方文档 对 ID Token 进行解密验证。也可以直接使用相关的开发库完成解密验证。 验证所需的公钥通过调用 获取 JWT 公钥 接口获得。

异常响应示例

- client_id 参数缺失或有误。

```
HTTP/1.1 401 Unauthorized
```

- client_id 与获取授权和获取 Token 时使用的不一致。

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_client"
}
```

- grant_type 参数有误。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "unsupported_grant_type",
"error_description" : "OAuth 2.0 Parameter: grant_type",
"error_uri" : "https://datatracker.ietf.org/doc/html/rfc6749#section-5.2"
}
```

- code 参数有误。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_grant"
}
```

客户端凭证模式

最近更新时间：2023-08-14 15:50:22

接口描述

使用 OAuth 客户端凭证模式 (client_credentials) 获取 Access Token。

支持的应用类型

Web 应用、M2M 应用。

请求方法

```
POST
```

请求路径

```
/oauth2/token
```

请求 Content-Type

```
application/x-www-form-urlencoded
```

请求示例

```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

grant_type=client_credentials&client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&scope=identity_proofin
g
```

请求参数

参数	可选	描述
grant_type	false	填固定值 client_credentials 。
client_id	false	应用的 client_id 。可参考 应用管理页面 > 选定指定应用 > 单击应用配置 > 对应的“Client Id”。
client_secret	false	应用的 client_secret 。可参考 应用管理页面 > 选定指定应用 > 单击应用配置 > 对应的“client_secret”。
scope	true	申请授权的 scope ，多个 scope 之间使用空格分隔。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
 "access_token" :
"eyJraWQiOiJmOTY5NGQ5My1kNTQxLTQ5ODUtODhkYy00MjIyOTg3MzAwOGUiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiO
iJURU5BTlRfQ0xJRU5UX0lEIiwiYXVkIjoiVEVOQU5UX0NMSUVOVF9JRCIsIm5iZiI6MTY0MDU4ODgyOCwic2NvcGUiOlsiaWRlbnRpdHlf
cHJvb2ZpbmciXSwiaXNzIjoiaHR0cHM6XC9cL3NhbXBsZS5wb3J0YWwudGVuY2VudGNpYW0uY29tIiwiZXhwIjoxNjQwNTg5MTI4LCJ
pYXQiOjE2NDA1ODg4MjgsImp0aSI6Ijk5MzliYzNmLTVlZGYtNDZjMy04ZjVjLTJiMWRjMWFjZmE5MCJ9.D8khbU3BkWTHD6sRTY9M_b
bRq7AF4MGina2S1ycFA2HOUo-k_P_f81V4fP1DLO7n-
1_5em_Dmxo768wsFcvUlYWRlSLh91kXPTyBV5mHt6IZRnT3eMpqTiMKC_ubzUc_7DSpE8-99-
```

```
CpfrQ19hcpMwkoLYh1dwYjUJB6xyZPzqlezrHy4unUHLTpyjGeecgNNLUScY9W0diGxVnbsMuTW7T00OsltNoJj11qtOllbh5kd1B4umv
A3UJpGucVMZQ2YTvltHyDBefWabP4ektzktAwDTALGIu1EsQfb5j9Rru3R0L0nAvjT8HiIqthLvvMdkjXAW983zj0yCPe3GqF3g",
  "scope" : "identity_proofing",
  "token_type" : "Bearer",
  "expires_in" : 299
}
```


响应参数

参数	数据类型	描述
access_token	String	Access Token (JWT)。
token_type	String	Token 类型，目前返回的是固定值 `Bearer` 。
expires_in	Number	Access Token 有效期，单位秒。
scope	String	Access Token 的 Scope。

> **说明**
> 客户端凭证模式的响应中不包含 Refresh Token。当 Access Token 过期时，应用再次调用此接口获取新的 Access Token。

异常响应示例

- 应用不存在、未开启或应用密钥校验失败。

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_client"
}
```

- 应用无权限使用 `client_credentials` 模式获取 Token。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "unauthorized_client"
}
```

- scope 参数有误或超出应用权限。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_scope"
}
```

获取用户信息

最近更新时间：2022-03-28 11:03:41

接口描述

获取已登录用户的用户信息。调用此接口时，需携带登录成功时得到的具备 `openid scope` 的 Access Token。

支持的应用类型

Web 应用、单页应用、移动 App、M2M 应用、小程序应用。

请求方法

```
GET
```

请求路径

```
/userinfo
```

请求示例

```
GET /userinfo HTTP/1.1
Authorization: Bearer ACCESS_TOKEN_WITH_OPENID_SCOPE
Host: sample.portal.tencentciam.com
```

请求头

名称	描述
Authorization	OAuth 2.0 Bearer Token，格式为 `Bearer <Token>`，其中 `Bearer` 为固定字符串，`<Token>` 为用户登录成功时得到的具备 `openid scope` 的 Access Token，`Bearer` 和 `<Token>` 之间用一个空格隔开。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "sub" : "MOCK_USER_ID",
  "email" : "MOCK_USERNAME@example.com",
  "name" : "MOCK_NAME",
  "nickname" : "MOCK_NICKNAME",
  "zoneinfo" : "Asia/Shanghai",
  "locale" : "zh-CN"
}
```

响应参数

参数	数据类型	描述
sub	String	用户标识，在用户池内唯一。

> **说明**
> 除 `sub` 字段一定返回外，其余返回哪些字段由应用参数配置中的 `Claims` 决定。

异常响应示例

- 未携带 Access Token。

```
HTTP/1.1 400 Bad Request
WWW-Authenticate: Bearer error="invalid_request", error_description="Bearer token not found in the request",
error_uri="https://tools.ietf.org/html/rfc6750#section-3.1"
```

- Access Token 无效（例如 Token 格式有误、已过期或已注销）。

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token", error_description="Error decoding JWT",
error_uri="https://tools.ietf.org/html/rfc6750#section-3.1"
```

- Access Token 不具备 `openid scope` 。

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_scope", error_description="The request requires higher privileges than
provided by the access token.", error_uri="https://tools.ietf.org/html/rfc6750#section-3.1"
```

- 找不到 Access Token 对应的用户。

```
HTTP/1.1 404 Not Found
Content-Type: application/json;charset=UTF-8

{
"error" : "user_not_found"
}
```

更新用户信息

最近更新时间：2022-03-28 11:04:30

接口描述

更新已登录用户的个人信息。调用此接口时，需携带登录成功时得到的具备 openid scope 的 Access Token。如果更新手机号或邮箱，则需先调用 发送 OTP 验证码 接口向用户发送验证码。

支持的应用类型

Web 应用、单页应用、移动 App、M2M 应用、小程序应用。

请求方法

PATCH

请求路径

/userinfo

请求 Content-Type

application/json

请求示例

```
PATCH /userinfo HTTP/1.1
Content-Type: application/json
Authorization: Bearer ACCESS_TOKEN_WITH_OPENID_SCOPE
Host: sample.portal.tencentciam.com

{
 "nickname" : "MOCK_NICKNAME"
}
```

请求头

名称	描述
Authorization	OAuth 2.0 Bearer Token，格式为 `Bearer <Token>`，其中 `Bearer` 为固定字符串，`<Token>` 为用户登录成功时得到的具备 `openid scope` 的 Access Token，`Bearer` 和 `<Token>` 之间用一个空格隔开。

请求体 JSON 参数

JSON 路径	数据类型	描述
phone_number	String	用户的手机号，限国内三大运营商11位手机号。传递此参数时，须同时传递 `phone_number_otp_token` 和 `phone_number_otp` 两个参数。
phone_number_otp_token	String	发送短信验证码成功后服务端返回的 `otp_token`。
phone_number_otp	String	用户手机收到的 OTP 验证码。
email	String	用户的邮箱地址。传递此参数时，须同时传递 `email_otp_token` 和 `email_otp` 两个参数。

email_otp_token	String	发送邮箱验证码成功后服务端返回的 otp_token 。
email_otp	String	用户邮箱收到的 OTP 验证码。
name	String	用户姓名。
nickname	String	用户昵称。
zoneinfo	String	用户时区，如 Asia/Shanghai 或 Europe/Paris 。
locale	String	用户 locale 信息，如 zh-CN 或 en-US 。

说明
其他参数的取值为用户属性标识。属性标识可以在 属性自定义页面 的属性详情界面查看。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "sub" : "MOCK_USER_ID",
  "email" : "MOCK_USERNAME@example.com",
  "name" : "MOCK_NAME",
  "nickname" : "MOCK_NICKNAME",
  "zoneinfo" : "Asia/Shanghai",
  "locale" : "zh-CN"
}
```

说明
除 sub 字段一定返回外，其余返回哪些字段由应用参数配置中的 Claims 决定。

异常响应示例

- 入参包含未知属性。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Unknown attribute(s) found."
}
```

- 入参包含不支持修改的属性。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_request",
"error_description" : "Unsupported user attribute(s) found."
}
```

- 电话号码格式不合法。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_phone_number"
}
```

- 电话号码已存在。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_phone_number"
}
```

- `phone_number_otp_token` 错误或已过期，或注册时使用的参数与发送验证码时不一致（例如：手机号不同）。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_phone_number_otp_token"
}
```

- `phone_number_otp` 错误或已过期。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_phone_number_otp"
}
```

- 邮箱格式不合法。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "malformed_email"
}
```

- 邮箱已存在。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_email"
}
```

- `email_otp_token` 错误或已过期，或注册时使用的参数与发送验证码时不一致（例如：邮箱不同）。

```
HTTP/1.1 400 Bad Request
```

```
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp_token"
}
```

- `email_otp` **错误或已过期。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp"
}
```

- **入参取值不合法（例如：不符合用户属性正则表达式）。**

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "illegal_parameter_value"
}
```

- `bearer_token` **缺失。**

```
HTTP/1.1 400 Bad Request
WWW-Authenticate: Bearer error="invalid_request", error_description="Bearer token not found in the request",
error_uri="https://tools.ietf.org/html/rfc6750#section-3.1"
```

- `bearer_token` **错误。**

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token", error_description="Error decoding JWT",
error_uri="https://tools.ietf.org/html/rfc6750#section-3.1"
```

- `bearer_token` **无效。**

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_scope", error_description="The request requires higher privil
```

修改用户密码

最近更新时间：2022-03-28 11:04:41

接口描述

修改已登录用户的密码。调用此接口时，需携带登录成功时得到的具备 `openid scope 的 Access Token` 。新密码需符合当前应用关联的账号密码认证源的密码策略，且不能与策略中指定的前 N 次历史密码相同。

支持的应用类型

Web 应用、单页应用、移动 App、M2M 应用、小程序应用。

请求方法

```
POST
```

请求路径

```
/change_user_password
```

请求 Content-Type

```
application/json
```

请求示例

```
POST /change_user_password HTTP/1.1
Content-Type: application/json
Authorization: Bearer ACCESS_TOKEN_WITH_OPENID_SCOPE
Host: sample.portal.tencentciam.com

{
 "old_password" : "MOCK_PASSWORD",
 "new_password" : "MOCK_NEW_PASSWORD"
}
```

请求头

名称	描述
Authorization	OAuth 2.0 Bearer Token，格式为 `Bearer <Token>` ，其中 `Bearer` 为固定字符串， `<Token>` 为用户登录成功时得到的具备 `openid scope 的 Access Token` ， `Bearer` 和 `<Token>` 之间用一个空格隔开。

请求体 JSON 参数

JSON 路径	数据类型	描述
old_password	String	旧密码。
new_password	String	新密码。

正常响应示例

```
HTTP/1.1 200 OK
```

异常响应示例

- 旧密码错误。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "wrong_old_password"
}
```

- 新密码与旧密码相同。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "duplicate_password"
}
```

- 新密码与历史密码相同。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "recurrent_password"
}
```

- 新密码不满足密码策略。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_new_password"
}
```

- 用户未找到。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "user_not_found"
}
```

- 用户被冻结。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "abnormal_user_status",
"error_description" : "User is frozen."
}
```

- 用户被锁定。

```
HTTP/1.1 400 Bad Request
```

```
Content-Type: application/json;charset=UTF-8

{
"error" : "abnormal_user_status",
"error_description" : "User is locked."
}
```

- `bearer_token` **缺失。**

```
HTTP/1.1 400 Bad Request
WWW-Authenticate: Bearer error="invalid_request", error_description="Bearer token not found in the request",
error_uri="https://tools.ietf.org/html/rfc6750#section-3.1"
```

- `bearer_token` **错误。**

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Bearer error="invalid_token", error_description="Error decoding JWT",
error_uri="https://tools.ietf.org/html/rfc6750#section-3.1"
```

- `bearer_token` **无效。**

```
HTTP/1.1 403 Forbidden
WWW-Authenticate: Bearer error="insufficient_scope", error_description="The request requires higher priv
```

重置用户密码

最近更新时间：2023-08-14 15:50:22

接口描述

重置用户的密码。调用此接口前，需要先通过 发送_OTP_验证码 接口向用户发送验证码。

> **注意**
> 新密码需符合当前应用关联的账号密码认证源的密码策略，且不能与策略中指定的前 N 次历史密码相同。

支持的应用类型

Web 应用、单页应用、移动 App。

请求方法

```
POST
```

请求路径

```
/reset_user_password
```

请求 Content-Type

```
application/json
```

请求示例

```
POST /reset_user_password HTTP/1.1
Content-Type: application/json
Authorization: Basic VEVOQU5UX0NMSUVOVF9JRDpURU5BTlRfQ0xJRU5UX1NFQ1JFVA==
Host: sample.portal.tencentciam.com

{
   "password" : "MOCK_PASSWORD",
   "email" : "MOCK_EMAIL@163.com",
   "email_otp" : "MOCK_EMAIL_OTP",
   "email_otp_token" : "MOCK_EMAIL_OTP_TOKEN"
}
```

请求头

名称	描述
Authorization	HTTP Basic 认证请求头，格式为 Basic <credentials>，其中 Basic 为固定字符串，<credentials>的计算方式为 base64(url_encode(client_id) + ":" + url_encode(client_secret))，Basic 和 <credentials> 之间用一个空格隔开。

请求体 JSON 参数

JSON 路径	数据类型	描述
client_id	String	应用的 client_id。需要与发送验证码时使用的一致。
client_secret	String	应用的 client_secret。Web 应用须传递此参数。单页应用和移动 App 不传递此参数。
password	String	新密码。

email	String	用户的邮箱地址。发送邮箱 OTP 验证码时传递此参数。
email_otp_token	String	发送邮箱验证码成功后服务端返回的 otp_token。
email_otp	String	用户邮箱收到的 OTP 验证码。
phone_number	String	用户的手机号。发送短信 OTP 验证码时传递此参数。
phone_number_otp_token	String	发送短信验证码成功后服务端返回的 otp_token。
phone_number_otp	String	用户手机收到的 OTP 验证码。

正常响应示例

重置密码成功。

```
HTTP/1.1 200 OK
```

异常响应示例

- 新密码与历史密码相同。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "recurrent_password"
}
```

- 新密码不满足密码策略。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "invalid_new_password"
}
```

- 未找到与 email 对应的用户。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "user_not_found"
}
```

- 用户处于冻结状态，无法重置密码。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "abnormal_user_status",
"error_description" : "User is frozen."
}
```

- email_otp_token 错误或已过期，或重置密码时使用的参数与发送验证码时不一致（例如：邮箱不同）。

```
HTTP/1.1 400 Bad Request
```

```
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp_token"
}
```

- email_otp 错误或已过期。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
"error" : "bad_email_otp"
}
```

获取 JWT 公钥

最近更新时间：2021-12-30 15:03:23

接口描述

JWT 公钥用于对 JWT 格式的 ID Token 和 Access Token 进行验证。

> **说明**
> JWT 密钥在创建用户目录时自动生成，不同用户目录的密钥不同。

支持的应用类型

Web 应用、单页应用、移动 App、M2M 应用、小程序应用。

请求方法

```
GET
```

请求路径

```
/oauth2/jwks
```

请求示例

```
GET /oauth2/jwks HTTP/1.1
Host: sample.portal.tencentciam.com
```

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "keys" : [ {
    "kty" : "RSA",
    "e" : "AQAB",
    "kid" : "f9694d93-d541-4985-88dc-42229873008e",
    "n" : "wYmf-
IL7_pXqEjtfHme7KqS06hRQ0BzhTzORjgwnsJD_CPexMHQAd82vZfOQioW9oaMXTiSAtkXslxNIxKVjiYMVzTLHQ9nqCARHOAONiftvc
OyMiDGwI_ZV2u2ltHCbQ1w8sMpREMxLiW46TYHANSQwgzg9gLojhzPEUmAS0ksTx3UURmQGLnFBEh6Ydbj8tPNnNxfZHRLtqTD0F
wLpPrn3wJvQRxNk_fcrJexM5v96XdQ1SLhhcIAMyqU-_-3IkyWIcS-Z-EvZCiKNJPCfVIEpWdz0oQdGXdADSRU8cV_sl-
YmUWSE355PSzK1UmbvNJWLMsO67ZsgZyetcvkaIw"
  } ]
}
```

响应参数

参数	数据类型	描述
keys	Array	包含公钥的数组。
keys[].kty	String	密钥类型，如 RSA。
keys[].kid	String	密钥标识。
keys[].e	String	RSA 公钥。
keys[].n	String	RSA 公钥。

刷新 Token

最近更新时间：2024-04-11 14:31:11

接口描述

使用 refresh_token 获取新的 access_token。

支持的应用类型

Web 应用、单页应用、移动 App、M2M 应用、小程序应用。

请求方法

POST

请求路径

/oauth2/token

请求 Content-Type

application/x-www-form-urlencoded

请求示例

```
POST /oauth2/token HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&grant_type=refresh_token&refresh_token=MOCK_REFR
ESH_TOKEN
```

请求参数

参数	可选	描述
client_id	false	应用的 client_id 。需要与获取授权时使用的一致。
client_secret	false	应用的 client_secret 。可通过租户管理平台的应用基本信息页面查看。
grant_type	false	填固定值 refresh_token 。
refresh_token	false	获取 Token 时返回的 refresh_token 。

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8

{
  "access_token" :
"eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNT
0NLX1VTRVJOQU1FIiwiYXVkIjoiVEVOQU5UX0NMSUVOVF9JRCIsIm5iZiI6MTYzNjQ0OTIzMSwic2NvcGUiOlsib3BlbmlkIl0sImlzcyI6Im
h0dHBzOlwvXC9URU5BTlQuUE9SVEFMLkRPTUFJTiIsImV4cCI6MTYzNjQ0OTUzMSwiaWF0IjoxNjM2NDQ5MjMxLCJqdGkiOiI1YTE4Zj
E0MS1lMzQxLTQzMzgtODVkNi1iYjRlNDFhODY3NjUifQ.cP9azxqZW8ELTvr70T2u8S0uxq9fnEKKVh55_6umF0RzBXMtRSeN8bW1uD
HzQbYVDfQZQp-vzee35XsfZkS5dYOFQmiOr_uXwd8wdgPvy_8ZHyamY4c4ytZvysx-
RDrr6oeHK63lPBrkoC1n_NI6AyZtMZxtxrodggzW593FfqG6vWE7OdPBoeQ8Qk-SEntuReNb60l9f6KYs-
H8DJt9f1tYbCe_4ECrxpAbezEyhjVxj_8B72oNCybs3toc1NfZ01ouhGyRIyaDqzKLFFX8QYp-
UgZV3N_muUyPNXHhvF4xsiFp_Sf0gSfQ_CYQUoXVs7R_ejSPexqqqYUyXHyzJA",
  "refresh_token" : "MOCK_REFRESH_TOKEN",
  "scope" : "openid",
  "id_token" :
"eyJraWQiOiJkNDliYzUwNS01NTcyLTRlZDYtOWU0OC0zODhjM2Q0NGJiNDYiLCJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdWIiOiJNT
0NLX1VTRVJOQU1FIiwiYXVkIjoiVEVOQU5UX0NMSUVOVF9JRCIsImF6cCI6IlRFTkFOVF9DTElFTlRfSUQiLCJpc3MiOiJodHRwczpcL1wv
```

VEVOQU5ULlBPUlRBTC5ET01BSU4iLCJleHAiOjE2MzY0NTEwMzEsImlhdCI6MTYzNjQ0OTIzMSwianRpIjoiMDEzYmFiMzktZjkxOC00NDhmLWEzOTYtY2U3N2VlZmZkYjM3In0.M_k7_RnA3E9ptcunBhSSZkxxqqHZhH7PIMn0QQoJGhJ0nja1ZxdMwhEMBmQDZ-cDNaWNUGFT3XZkAsJNBF06g69kAu_gMzRKeGlgCUrQUO_Z1DNoec6Q6oPg3bV5JzYtNjjySEeYrlKxG0PpY3L0uRjpQFXXL6vpm9n4LXz81gsmZiC0JAfJ-n7oDH6QJ8AmfWWJZMs2qobFXsKaFqwBehXFki5qTy8MLxkHMuW-kc4IK0gadlrvic3MRsy3qgfPnwHUfCLb7Ky77EPGXGwFbNzAYGtlimCe-ZN498RatwejTXptiqkBqUhaS7QqdEcI0etAd_4J2DmJYwcXJCFwSg",
  "token_type" : "Bearer",
  "expires_in" : 299
}

响应参数

参数	数据类型	描述
access_token	String	刷新后的 OAuth 2.0 Access Token (JWT)。
refresh_token	String	与请求参数中相同的 OAuth 2.0 Refresh Token。
scope	String	Access Token 的 Scope。
id_token	String	刷新后的 OIDC ID Token (JWT)。
token_type	String	Token 类型，目前取固定值 `Bearer` 。
expires_in	Number	Access Token 有效期，单位秒。

异常响应示例

refresh_token 参数有误。

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=UTF-8

{
  "error" : "invalid_grant"
}
```

注销 Token

最近更新时间：2021-12-30 15:03:36

接口描述

注销 OAuth 2.0 Token 。如果传入的是 access_token ，则仅注销该 access_token；如果传入的是 refresh_token，则该 refresh_token 以及与它相关联的 access_token 都将被注销。

支持的应用类型

Web 应用、单页应用、移动 App、M2M 应用、小程序应用。

请求方法

```
POST
```

请求路径

```
/oauth2/revoke
```

请求 Content-Type

```
application/x-www-form-urlencoded
```

请求示例

```
POST /oauth2/revoke HTTP/1.1
Host: sample.portal.tencentciam.com
Content-Type: application/x-www-form-urlencoded

client_id=TENANT_CLIENT_ID&client_secret=TENANT_CLIENT_SECRET&token=MOCK_ACCESS_TOKEN
```

请求参数

参数	可选	描述
client_id	false	应用的 client_id 。需要与获取授权和获取 Token 时使用的一致。
client_secret	false	应用的 client_secret 。可通过租户管理平台的应用基本信息页面查看。
token	false	access_token 或 refresh_token 的值。

正常响应示例

```
HTTP/1.1 200 OK
```

异常响应示例

client_id 与发起登录和获取 Token 时使用的不一致。

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json;charset=UTF-8

{
  "error" : "invalid_client"
}
```

获取 OpenID Provider 配置信息

最近更新时间：2021-12-30 15:03:41

接口描述

应用可以通过此接口获取 OIDC 授权服务器的配置信息，从而简化本地配置。具体的配置信息及其含义请参考 OpenID Connect Discovery 标准 。

支持的应用类型

Web 应用、单页应用、移动 App、M2M 应用、小程序应用。

请求方法

```
GET
```

请求路径

```
/.well-known/openid-configuration
```

请求示例

```
GET /.well-known/openid-configuration HTTP/1.1
Host: sample.portal.tencentciam.com
```

正常响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "issuer" : "https://sample.portal.tencentciam.com",
  "authorization_endpoint" : "https://sample.portal.tencentciam.com/oauth2/authorize",
  "token_endpoint" : "https://sample.portal.tencentciam.com/oauth2/token",
  "token_endpoint_auth_methods_supported" : [ "client_secret_basic", "client_secret_post" ],
  "jwks_uri" : "https://sample.portal.tencentciam.com/oauth2/jwks",
  "response_types_supported" : [ "code" ],
  "grant_types_supported" : [ "authorization_code", "client_credentials", "refresh_token" ],
  "subject_types_supported" : [ "public" ],
  "id_token_signing_alg_values_supported" : [ "RS256" ],
  "scopes_supported" : [ "openid" ]
}
```

响应参数

参数	数据类型	描述
issuer	String	OIDC Issuer。
authorization_endpoint	String	OAuth 2.0 获取授权的服务地址。
token_endpoint	String	OAuth 2.0 获取 Token 的服务地址。
jwks_uri	String	获取 JWT 公钥的服务地址。
grant_types_supported	Array	支持的 OAuth 2.0 Grant Type。
response_types_supported	Array	OAuth 2.0 获取授权服务支持的 Response Type。
token_endpoint_auth_methods_supported	Array	OAuth 2.0 获取 Token 服务支持的认证方式。

subject_types_supported	Array	支持的 OIDC Subject Identifier Type。
id_token_signing_alg_values_supported	Array	支持的 JWS 签名算法。
scopes_supported	Array	支持的 OAuth 2.0 Scope。