统一身份认证第三方应用接入文档

目录

[统一身份认证第三方应用接入文档 1](#_Toc82623250)

[规范 2](#_Toc82623251)

[认证流程图 2](#_Toc82623252)

[核心接口 3](#_Toc82623253)

[oidc认证请求 3](#_Toc82623254)

[oidc token请求 4](#_Toc82623255)

[范例代码 5](#_Toc82623256)

[配置步骤 5](#_Toc82623257)

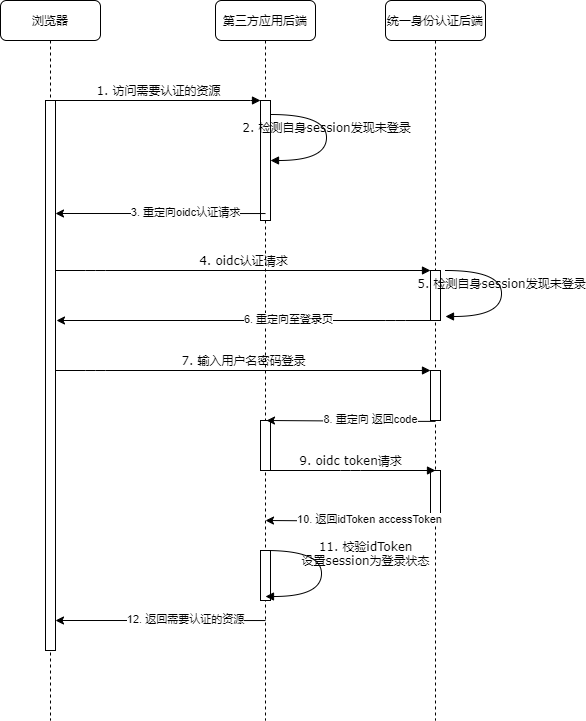
## 规范

统一身份认证服务遵循OpenID Connect协议。协议的官方文件地址如下

<https://openid.net/specs/openid-connect-core-1_0.html>

## 认证流程图

下面使用时序图将认证流程精确地表达出来。图中的OIDC是OpenID Connect的缩写。



1. 用户通过浏览器访问应用中需要认证的资源。
2. 第三方应用后端检测自身的session，如果该用户没有登录，转至步骤3，否则跳至步骤12。
3. 重定向至oidc认证请求。
4. 浏览器请求oidc认证请求接口。接口详情见下文核心接口 oidc认证请求。
5. 统一身份认证后端检测自身session，如果该用户没有登录，跳转至步骤6，否则跳至步骤8。
6. 重定向至登录页。
7. 用户输入用户名密码后点击登录按钮。
8. 统一身份认证服务校验账号密码通过后，重定向至步骤4认证请求中redirect\_uri参数，并添加code和state参数（步骤4传入的参数）。
9. 第三方应用后端发起token请求，携带步骤8中的code参数。
10. 统一身份认证服务向应用后端发送ID token和access token响应。
11. 校验idToken，设置第三方应用session为登录状态。
12. 返回需要认证的资源。

## 核心接口

### oidc认证请求

请求方式 GET, POST {统一认证服务器地址}/oap-svr/oap /authorize

参数说明

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| scope | query | 固定填openid |
| response\_type | query | 固定填code |
| client\_id | query | 指定本次请求的应用ID。需要预先在统一认证服务中配置第三方应用，具体步骤见《配置步骤》小节 |
| redirect\_uri | query | 认证成功后的重定向地址，由第三方应用自行提供，但需要预先在统一认证服务中配置。具体步骤见《配置步骤》小节 |
| state | query | 由第三方应用生成 |

下面是一个请求范例

GET /oap-svr/oap /authorize?

scope=openid

& response\_type=code

&client\_id=third-party-demo

&state=st

&redirect\_uri=https%3A%2F%2Fclient.example.org%2Fcb

响应

认证成功后，统一身份认证服务返回302重定向响应。响应的Location是请求参数中的redirect\_uri，加上code和state两个参数。下面是一个响应范例。

302 Found

Location: https://client.example.org/cb?code=SplxlOBeZQQYbYS6WxSbIA&state=st

### oidc token请求

请求方式POST {统一认证服务器地址} /oap-svr/oap/token

（注：也可以使用示例代码中的OidcCore.tokenRequest方法，传入相应参数即可）

参数说明

|  |  |  |
| --- | --- | --- |
| 参数名 | 参数类型 | 说明 |
| Authorization | header | 该参数由client\_id和client\_secret计算而来。具体的算法见OAuth 2.0协议的2.3.1节（<https://datatracker.ietf.org/doc/html/rfc6749#section-2.3.1>）。用于校验第三方应用本身的身份。 |
| grant\_type | application/x-www-form-urlencoded | 指定使用Authorization Code流程。固定填authorization\_code。 |
| code | application/x-www-form-urlencoded | 指定授权码。授权码的值由前一步认证请求的响应中得来。 |
| redirect\_uri | application/x-www-form-urlencoded | 认证成功后的重定向地址，由第三方应用自行提供，但需要预先在统一认证服务中注册，具体步骤见《配置步骤》小节。 |

响应

Token端点以json格式返回ID token、access token、refresh token。其中的ID token是JWT格式。下面是一个响应范例。

{

"access\_token": "SlAV32hkKG",

"token\_type": "Bearer",

"refresh\_token": "8xLOxBtZp8",

"expires\_in": 3600,

"id\_token": "eyJhbGciOiJSUzI1NiIsImtpZCI6IjFlOWdkazcifQ.ewogImlzc

yI6ICJodHRwOi8vc2VydmVyLmV4YW1wbGUuY29tIiwKICJzdWIiOiAiMjQ4Mjg5

NzYxMDAxIiwKICJhdWQiOiAiczZCaGRSa3F0MyIsCiAibm9uY2UiOiAibi0wUzZ

fV3pBMk1qIiwKICJleHAiOiAxMzExMjgxOTcwLAogImlhdCI6IDEzMTEyODA5Nz

AKfQ.ggW8hZ1EuVLuxNuuIJKX\_V8a\_OMXzR0EHR9R6jgdqrOOF4daGU96Sr\_P6q

Jp6IcmD3HP99Obi1PRs-cwh3LO-p146waJ8IhehcwL7F09JdijmBqkvPeB2T9CJ

NqeGpe-gccMg4vfKjkM8FcGvnzZUN4\_KSP0aAp1tOJ1zZwgjxqGByKHiOtX7Tpd

QyHE5lcMiKPXfEIQILVq0pc\_E2DzL7emopWoaoZTF\_m0\_N0YzFC6g6EJbOEoRoS

K5hoDalrcvRYLSrQAZZKflyuVCyixEoV9GfNQC3\_osjzw2PAithfubEEBLuVVk4

XUVrWOLrLl0nx7RkKU8NXNHq-rvKMzqg"

}

## 范例代码

代码地址： <https://github.com/rebue/scx/tree/1.2.4/orp/third-party-demo>

代码说明见此目录下的README.md文件

## 配置步骤

1. 在rac数据库RAC\_APP添加一条记录

INSERT INTO `rac`.`RAC\_APP`(`ID`, `NAME`, `REALM\_ID`, `URL`, `MENU`, `REMARK`, `IS\_ENABLED`, `IMG\_URL`, `SEQ\_NO`, `IS\_CERTIFIED`) VALUES ('third-party-demo', '第三方示例应用', 'default', 'http://122.3.11.12:30010', NULL, NULL, 1, NULL, 0, 0);

1. 在oap数据库OAP\_APP添加一条记录，CLIENT\_ID用于oidc认证请求，SECRET用于oidc token请求

INSERT INTO `oap`.`OAP\_APP`(`ID`, `APP\_ID`, `IS\_ENABLED`, `CLIENT\_ID`, `SECRET`, `OBJ\_ID`, `CREATE\_TIMESTAMP`, `UPDATE\_TIMESTAMP`) VALUES (1, 'unified-auth', 1, 'unified-auth', '123456', NULL, 1, 1);

1. 在oap数据库OAP\_REDIRECT\_URI添加一条记录，其中APP\_ID是1. 2步骤中的ID字段的值，REDIRECT\_URI是认证成功后的重定向地址

INSERT INTO `oap`.`OAP\_REDIRECT\_URI`(`ID`, `APP\_ID`, `REDIRECT\_URI`, `CREATE\_TIMESTAMP`, `UPDATE\_TIMESTAMP`) VALUES (3, 1, 'http://auth.maiyuesoft.com\*', 1, 1);