



ORACLE

# APEX 重磅新特性及实用开发实践

Oracle低代码平台

**Mile & Wangbo**

Oracle SE-hub

## 日程



Oracle APEX

- 工作审批流
- 智能数据搜索
- PWA
- 第三方认证集成
- 外部应用集成
- APEX计划自动化任务管理

# 1. workflow/审批流

APEX 22.1

## 工作流和自动化



任务定义

自动化

1

通过配置任务定义、任务详细信息和统一任务列表来管理审批。

- 审批组件处理Oracle APEX中人工审批的所有方面，允许**开发人员将审批功能**放入应用程序中。
- 建立任务主要信息：时间，备注，历史，优先级，审批人分配，撤回，取消，同意，拒绝等
- 人工建立任务流程和事件（**email模版快速搭建，变量传导，存储过程等**）
- **自动建立**任务列表，搜索，待审批列表
- **自动建立**详情页
- 系统中可设置完成后的任务多少天后被回收（1-30，默认7天）

操作					
Q 搜索: 所有文本列					
开始 操作 编辑					
<input type="checkbox"/>	≡	名称	结果	发生事件时	操作类型
<input checked="" type="checkbox"/>	≡	New exception received to Us...		创建	发送电子邮件
<input type="checkbox"/>	≡	On approval	已批准	完成	执行代码
<input type="checkbox"/>	≡	New exception Notify Admins		创建	发送电子邮件
<input type="checkbox"/>	≡	When approved	已批准	完成	发送电子邮件
<input type="checkbox"/>	≡	When Rejected	已拒绝	完成	发送电子邮件



## 2. 智能数据搜索

APEX快速建立搜索筛选报表解决方案

将新属性 Tokenize Row Search 添加到应用程序定义页面。以前，任何执行行搜索的组件都会执行完全匹配搜索。现在更普遍地应用搜索词。

**标记行搜索** - 将此属性设置为On分别处理搜索词的每个单词。当搜索中的单词一起或独立包含在任何列中时，记录作为匹配项返回。

多行搜索  
智能标记  
同名筛选

Sample Reporting

### 标记搜索

Q Search

Project

- Migrate from Legacy Server (10)
- Client Server Conversion (9)
- ACME Web Express Configuration (7)
- Migrate Desktop Application (7)
- Bug Tracker (5)

Show All

Assigned To

- Bob Nite (14)
- John Watson (14)
- Pam King (12)
- Myra Sutcliff (6)
- James Cassidy (5)

Show All

Status

- Closed (31)
- Pending (19)
- Open (17)
- On-Hold (6)

这是使用创建页面向导创建的分面搜索页面。最终用户可以使用左侧的标记设置过滤器。构面显示可能的值以及结果集中的出现次数。最终用户更改构面后，结果、相关构面和出现计数会立即刷新。

Total Row Count 73

Project	Task Name	Start Date	End Date	Status	Assigned To	Cost	Budget
ACME Web Express Configuration	Identify server requirements	02-SEP-22	03-SEP-22	Closed	John Watson	\$200.00	\$500.00
ACME Web Express Configuration	Determine Web listener configuration(s)	04-SEP-22	04-SEP-22	Closed	James Cassidy	\$600.00	\$500.00
ACME Web Express Configuration	Run installation	07-SEP-22	07-SEP-22	Closed	James Cassidy	\$200.00	\$200.00
ACME Web Express Configuration	Create pilot workspace	09-SEP-22	09-SEP-22	Closed	John Watson	\$100.00	\$100.00
ACME Web Express Configuration	Specify security authentication scheme(s)	14-SEP-22	14-SEP-22	Open	John Watson	\$200.00	\$300.00
ACME Web Express Configuration	Configure Workspace provisioning	15-SEP-22	15-SEP-22	Open	John Watson	\$200.00	\$100.00
ACME Web Express Configuration	Select servers for Development, Test, Production	18-SEP-22	20-SEP-22	Open	James Cassidy	\$200.00	\$600.00
Bug Tracker	Document quality assurance procedures	19-JUL-22	22-JUL-22	Closed	Myra Sutcliff	\$3,000.00	\$2,000.00
Bug Tracker	Review automated testing tools	23-JUL-22	25-JUL-22	Closed	Myra Sutcliff	\$750.00	\$1,500.00
Bug Tracker	Implement bug tracking software	07-AUG-22	07-AUG-22	Closed	Myra Sutcliff	\$100.00	\$100.00
Bug Tracker	Train developers on tracking bugs	14-AUG-22	19-AUG-22	On-Hold	Myra Sutcliff	\$1,000.00	\$2,000.00
Bug Tracker	Measure effectiveness of improved QA	26-AUG-22	26-AUG-22	Pending	Myra Sutcliff	\$0.00	\$500.00
Client Server Conversion	End-user Training	28-SEP-22	28-SEP-22	Pending	Myra Sutcliff	\$0.00	\$2,500.00
Client Server Conversion	Rollout migrated Client Server in ACME Web Express	29-SEP-22	29-SEP-22	Pending	Pam King	\$0.00	\$200.00
Client Server Conversion	User acceptance testing	22-SEP-22	24-SEP-22	Pending	Russ Saunders	\$0.00	\$2,500.00



# 3. PWA

Progressive Web App  
渐进式网页应用

什么是PWA：一种理念，使用多种技术来增强web app的功能，可以让网站的体验变得更好，能够模拟一些原生功能，比如通知推送。在移动端利用标准化框架，让网页应用呈现和原生应用相似的体验

**安装PWA** - Windows, Mac, 手机上生成APP，实现点击就可进入应用，方便用户近原生体验。

应用程序 50888 \ 渐进式 Web 应用程序

定义 安全性 全球化 用户界面 **渐进式 Web 应用程序**

应用程序 50888

全部显示 一般信息 外观 服务 Worker 配置

一般信息

启用渐进式 Web 应用程序

可安装

外观

显示 独立

屏幕方向 任何

主题颜色

背景色

iOS 状态栏样式 半透明

应用程序说明

QC\_Dashboard

Quota Analysis

Associate ITP01

Month/Period Period 1

Category	Used Quota (%)	Used Quota (\$)	Unused Quota (%)	Unused Quota (\$)
ABOVE GROUND POOLS	11.23%	\$29449196	88.77%	\$3723683
AIRBED,FURNITURE/PUMPS	9.96%	\$5228628	90.04%	\$578298
BOATS	11.36%	\$2146816	88.64%	\$275015
INTOYZ	9.21%	\$505966	90.79%	\$51351
SPA	9.31%	\$4648785	90.69%	\$477224
WETSET	12.15%	\$3386749	87.85%	\$1297659

\*网页展现

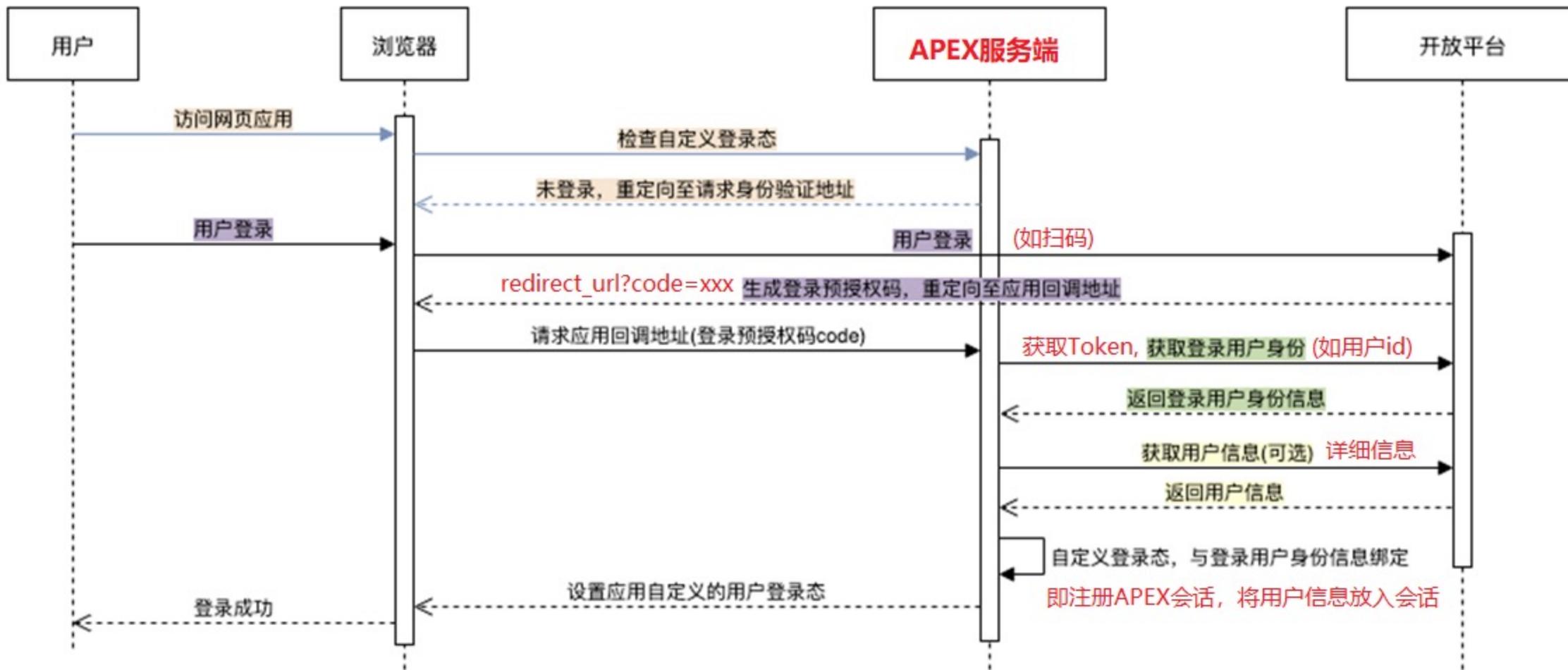
**APEX - 应用本身适配各种机型**



## 4. APEX认证集成

企业微信，飞书，钉钉，IDCS身份认证云

## APEX与开放平台认证集成 - OAuth2



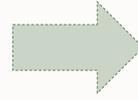
## 前置条件

1. 在企业微信开发者管理后台创建应用，记录下APP\_ID, APP\_SECRET等信息
2. 配置好应用的访问地址，接受authorization\_code的重定向URL，服务器IP/域名白名单等



扫码登录URL:

[https://open.work.weixin.qq.com/wwopen/sso/qrConnect?appid=ww85d34cdfcadcfbc4&agentid=1000003&redirect\\_uri=https%3A%2F%2Fpplxnxz650luzfig-omldb.adb.ap-seoul-1.oraclecloudapps.com%2Fords%2Fr%2Fhysunspace%2Fapexwecom-wechat%2Flogin&response\\_type=code&scope=snsapi\\_base&state=HYSUNSTATE#wechat\\_redirect](https://open.work.weixin.qq.com/wwopen/sso/qrConnect?appid=ww85d34cdfcadcfbc4&agentid=1000003&redirect_uri=https%3A%2F%2Fpplxnxz650luzfig-omldb.adb.ap-seoul-1.oraclecloudapps.com%2Fords%2Fr%2Fhysunspace%2Fapexwecom-wechat%2Flogin&response_type=code&scope=snsapi_base&state=HYSUNSTATE#wechat_redirect)



身份认证开始

获取Authorization code

获取Access Token

获取用户信息

注册APEX会话

完成身份认证

# Apex与企业微信身份认证集成REST API调用

STEP1:取得 authorization code (重定向URL中取)

```
begin
  l_code := :CODE;

  apex_web_service.g_request_headers(1).name := 'Content-Type';
  apex_web_service.g_request_headers(1).value := 'application/json; charset=utf-8';
```

```
l_clob := apex_web_service.make_rest_request(
  p_url => 'https://qyapi.weixin.qq.com/cgi-bin/gettoken?corpId=ww85d34cdfcadcfbc4&corpsecret=2rPF8t2'
  p_http_method => 'GET');
```

STEP2 : 调用API获取Access Token

```
apex_json.parse(
  p_values => l_resp,
  p_source => l_clob);
```

```
l_token := apex_json.get_varchar2(
  p_path => 'access_token',
  p_values => l_resp);
```

```
l_clob := apex_web_service.make_rest_request(
  p_url => 'https://qyapi.weixin.qq.com/cgi-bin/user/getuserinfo?access_token=' || l_token || '&code='
  p_http_method => 'GET');
```

```
apex_json.parse(
  p_values => l_resp,
  p_source => l_clob);
```

```
l_errcode := apex_json.get_number(
  p_path => 'errcode',
  p_values => l_resp);
```

```
if l_errcode > 0 then
  return;
end if;
```

```
l_userid := apex_json.get_varchar2(
  p_path => 'UserId',
  p_values => l_resp);
```

```
l_clob := apex_web_service.make_rest_request(
  p_url => 'https://qyapi.weixin.qq.com/cgi-bin/user/get?access_token=' || l_token
  || '&userid=' || l_userid,
  p_http_method => 'GET');
```

STEP3 : 获取详细用户信息

```
apex_json.parse(
  p_values => l_resp,
  p_source => l_clob);
```

```
l_errcode := apex_json.get_number(
  p_path => 'errcode',
  p_values => l_resp);
```

```
if l_errcode > 0 then
  return;
end if;
```

```
l_username := apex_json.get_varchar2(
  p_path => 'name',
  p_values => l_resp);
```

```
if l_username is not null then
  APEX_CUSTOM_AUTH.POST_LOGIN(
    p_uname => l_username,
    p_session_id => v('APP_SESSION_ID'),
    p_app_page => :APP_ID || ':1'
  );
end if;
```

```
end;
```

STEP4 : 企业微信认证通过后, 调用 APEX API注册一个已认证的APEX会话, 并将认证后的用户放入会话

# 在Apex登录页面执行认证逻辑

The screenshot displays the Oracle APEX Page Designer interface for 'Page 9999: Login Page'. The left-hand 'Process' pane shows a 'Pre-Rendering' process named 'WeComAuthn' selected. A red box highlights this process, with a red-bordered text box containing the Chinese text: '在公共页面上预执行认证逻辑, 如登陆页面。' (Execute authentication logic in advance on public pages, such as the login page).

The central 'Layout' pane shows the page structure with a 'BODY HEADER' and a 'CONTENT BODY' containing an 'ApexWeCom(Wechat)' region. The 'Items' pane at the bottom shows a list of items including 'P9999\_USERNAME', 'P9999\_PASSWORD', 'P9999\_REMEMBER', 'CODE', and 'STATE'.

The right-hand 'Process' pane shows the configuration for the 'WeComAuthn' process:

- Identification:** Name: WeComAuthn, Type: Execute Code, Editable Region: - Select -
- Source:** Location: Local Database, Language: PL/SQL
- PL/SQL Code:**

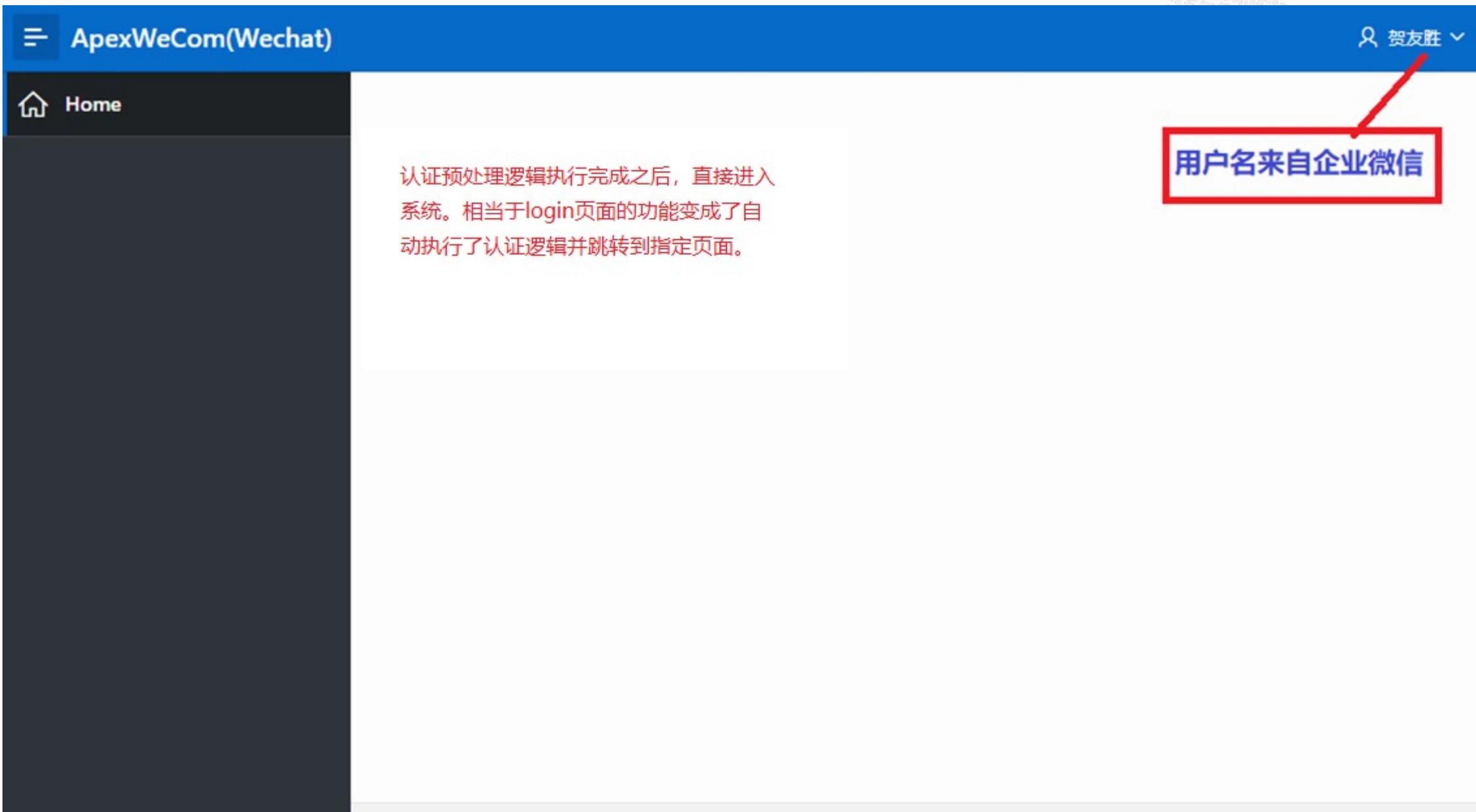
```
l_code := :CODE;

apex_web_service.g_request_headers(1).name := 'Content-Type';
apex_web_service.g_request_headers(1).value :=
'application/json; charset=utf-8';

l_clob := apex_web_service.make_rest_request(
  p_url => 'https://qyapi.weixin.qq.com/cgi-bin/gettoken?
  corpid=ww85d34cdfcadcfbc4&corpsecret=2rPF8t2fchAkhNBpGoAdnRm8c0fMI
  9Pp0uabk1Fd4c',
  p_http_method => 'GET');
```

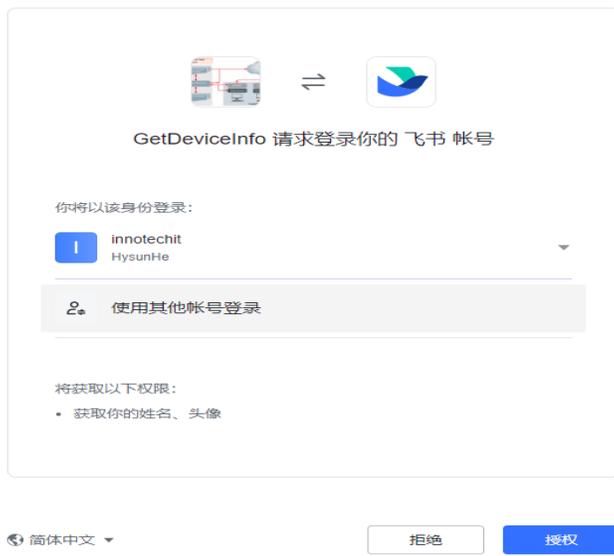


## 在浏览器通过扫码登录应用



## 前置条件

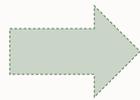
1. 在飞书开发者管理后台创建应用，记录下APP\_ID, APP\_SECRET等信息
2. 配置好应用的访问地址，接受authorization\_code的重定向URL，服务器IP/域名白名单等



扫码登录URL:

[https://open.feishu.cn/open-](https://open.feishu.cn/open-apis/authen/v1/index?app_id=cli_a2ceb2d6f0fa900d&redirect_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fhyspoc%2Fr%2Ffeishu%2Flogin&&state=authn)

[apis/authen/v1/index?app\\_id=cli\\_a2ceb2d6f0fa900d&redirect\\_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fhyspoc%2Fr%2Ffeishu%2Flogin&&state=authn](https://open.feishu.cn/open-apis/authen/v1/index?app_id=cli_a2ceb2d6f0fa900d&redirect_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fhyspoc%2Fr%2Ffeishu%2Flogin&&state=authn)



身份认证开始

获取Authorization code

获取Access Token

获取用户信息

注册APEX会话

完成身份认证

```

declare
  APP_ID varchar2(50) := 'cli_~' || '900d';
  APP_SECRET varchar2(50) := 'JUM2M ~' || 'UmG8N6xgf!' || ', , ZF';
  l_clob      clob;
  l_payload   clob;
  l_code      varchar2(50);
  l_resp      apex_json.t_values;
  l_login     varchar2(50);
begin
  l_code := :CODE;

  -- Get Access Token
  apex_web_service.g_request_headers(1).name := 'Content-Type';
  apex_web_service.g_request_headers(1).value := 'application/json; charset=utf-8';

  l_payload := '{"app_id": "" || APP_ID || "", "app_secret": "" || APP_SECRET
  || "", "code": "" || l_code || "", "grant_type": "authorization_code" || '}';

  l_clob := apex_web_service.make_rest_request(
    p_url => 'https://open.feishu.cn/connect/qrconnect/oauth2/access_token/',
    p_http_method => 'POST',
    p_body => l_payload);

```

STEP1:取得 authorization code (重定向URL中取)

STEP2 : 调用飞书API获取Access Token

```

apex_json.parse(
  p_values => l_resp,
  p_source => l_clob);

l_login := apex_json.get_varchar2(
  p_path => 'name',
  p_values => l_resp);

apex_debug.info('l_login: %s', l_login);

if l_login is not null then
  APEX_CUSTOM_AUTH.POST_LOGIN(
    p_username => l_login,
    p_session_id => v('APP_SESSION'),
    p_app_page => :APP_ID || ':1'
  );
end if;
end;

```

STEP3 : 获取用户信息

STEP4 : 飞书认证通过后, 调用 APEX API注册一个已认证的APEX会话, 并将认证后的用户放入会话

# 在Apex登录页面执行认证逻辑

The screenshot shows the Oracle APEX Page Designer interface for a login page. The left sidebar shows the page structure, with 'Pre-Rendering' selected. A red box highlights 'Pre-Rendering' with the text '在公共页面上执行认证逻辑 (如登陆页面)'. The main canvas shows the 'Login Page' layout with a 'FeiShu' region containing 'P9999\_USERNAME' and 'P9999\_PASSWORD' items. The right sidebar shows the 'Process' configuration for 'FeishuAuthn', set to 'Execute Code' and 'Local Database'. The 'PL/SQL Code' section contains the following code:

```
declare
  APP_ID varchar2(50) := 'cli_a2ceb2d6f0fa900d';
  APP_SECRET varchar2(50) := 'JUM2MIV8LXXHiomG8N6xgf1I7cp8GpZF';
  l_clob clob;
  l_payload clob;
  l_code varchar2(50);
  l_resp apex_json.t_values;
  l_login varchar2(50);
begin
  l_code := :CODE;
```

The text '认证逻辑' (Authentication Logic) is written in red next to the PL/SQL code.



The screenshot shows the FeiShu application interface. At the top, there is a blue header with the FeiShu logo on the left and navigation icons on the right, including 'Install App', a chat icon, a help icon, and a user profile icon labeled 'hysunhe'. Below the header, there is a 'Test' section with a red text message: '认证预处理逻辑执行完成之后, 直接进入系统。相当于login页面的功能变成了自动执行了认证逻辑并跳转到指定页面。'. To the right of this message, a red box highlights the text '用户信息来自飞书', with a red arrow pointing to the user profile icon in the header. Below the test section, there is a date separator '今天' and a time '11:16'. At the bottom, there is a notification box with the FeiShu logo and the following text: '登录操作通知', '登录应用: GetDeviceInfo', '登录时间: 2022-09-14 11:16:03', and 'CST'.



## 前置条件

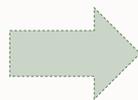
1. 在钉钉开发者管理后台创建应用，记录下APP\_ID, APP\_SECRET等信息
2. 配置好应用的访问地址，接受authorization\_code的重定向URL，服务器IP/域名白名单等



Scan QR Code with DingTalk [Refresh](#)

扫码登录URL:

[https://oapi.dingtalk.com/connect/qrconnect?appid=dingiuebm7adzt61ayks&response\\_type=code&scope=snsapi\\_login&state=RANDOMSTATE&redirect\\_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fr%2Fhyspoc%2Fapexding%2Flogin](https://oapi.dingtalk.com/connect/qrconnect?appid=dingiuebm7adzt61ayks&response_type=code&scope=snsapi_login&state=RANDOMSTATE&redirect_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fr%2Fhyspoc%2Fapexding%2Flogin)



身份认证开始

获取Authorization code

获取Access Token

获取用户信息

注册APEX会话

完成身份认证

# Apex与钉钉身份认证集成REST API调用

```
begin
  l_code := :CODE;
  l_timestamp := currentTimestampInMillis();
  l_sig := get_signature(l_timestamp, APP_SECRET);

  apex_web_service.g_request_headers(1).name := 'Content-Type';
  apex_web_service.g_request_headers(1).value := 'application/json; charset=utf-8';

  l_clob := apex_web_service.make_rest_request(
    p_url => 'https://oapi.dingtalk.com/gettoken?appkey=' || APP_KEY || '&apps',
    p_http_method => 'GET');

  apex_json.parse(
    p_values => l_resp,
    p_source => l_clob);

  l_token := apex_json.get_varchar2(
    p_path => 'access_token',
    p_values => l_resp);

  l_payload := '{"tmp_auth_code":"' || l_code || '"}';
  l_clob := apex_web_service.make_rest_request(
    p_url => 'https://oapi.dingtalk.com/sns/getuserinfo_bycode?timestamp=' || l_timestamp,
    p_http_method => 'POST',
    p_body => l_payload);

  apex_json.parse(
    p_values => l_resp,
    p_source => l_clob);

  l_unionid := apex_json.get_varchar2(
    p_path => 'user_info.unionid',
    p_values => l_resp);
```

STEP1:取得 authorization code (重定向URL中取)

STEP2 : 调用API获取Access Token

STEP3 : 获取详细用户信息

1. getuserinfo\_bycode
2. getbyunionid
3. user/get

```
l_payload := '{"unionid":"' || l_unionid || '"}';
l_clob := apex_web_service.make_rest_request(
  p_url => 'https://oapi.dingtalk.com/topapi/user/getbyunionid?access_token=' || l_token,
  p_http_method => 'POST',
  p_body => l_payload);
```

```
apex_json.parse(
  p_values => l_resp,
  p_source => l_clob);

l_userid := apex_json.get_varchar2(
  p_path => 'result.userid',
  p_values => l_resp);
```

```
l_payload := '{"language": "zh_CN", "userid": "' || l_userid || '"}';
l_clob := apex_web_service.make_rest_request(
  p_url => 'https://oapi.dingtalk.com/topapi/v2/user/get?access_token=' || l_token,
  p_http_method => 'POST',
  p_body => l_payload);
```

```
apex_json.parse(
  p_values => l_resp,
  p_source => l_clob);

l_username := apex_json.get_varchar2(
  p_path => 'result.name',
  p_values => l_resp);
```

```
:P9999_USERDING := l_username;
:P9999_NICK := l_nick;
```

```
if l_nick is not null then
  APEX_CUSTOM_AUTH.POST_LOGIN(
    p_uname => l_nick,
    p_session_id => v('APP_SESSION'),
    p_app_page => :APP_ID || ':1'
  );
end if;
```

end;

STEP4 : 钉钉认证通过后, 调用 APEX API注册一个已认证的APEX会话, 并将认证后的用户放入会话

# 在Apex登录页面执行认证逻辑

The screenshot displays the Oracle APEX Page Designer interface for configuring a login page. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', 'Team Development', and 'Gallery'. The user 'Hysun He hyspoc' is logged in. The page is identified as 'Page 9999: Login Page'.

**Left Panel (Page Designer):**

- Page 9999: Login Page** (highlighted with a red box)
- Pre-Rendering** (highlighted with a red box)
- Before Header
- Processes
  - DingTalkAuthn\_MoreDetailUserInfo (highlighted with a green box)
  - DingTalkAuthn\_BasicUserInfo
  - Get Username Cookie
- After Header
- Before Regions
- Components
  - Content Body
    - ApexDing
      - Region Body
        - P9999\_USERNAME
        - P9999\_PASSWORD
        - P9999\_REMEMBER
        - CODE

**Center Panel (Page Editor):**

- Layout** (selected)
- Page Search
- Help
- Search icons
- Page Title: Login Page
- Regions: BACKGROUND IMAGE, BODY HEADER, CONTENT BODY
- Item: ApexDing (selected)
- Buttons: Regions, Items (selected), Buttons
- Item Properties: C..., G..., Pi..., Pi..., Di..., O..., File Br..., G... A..., Hi..., List M..., M... E..., N... Fi..., P..., P... G...

**Right Panel (Process Editor):**

- Process** (selected)
- Filter
- Source
  - Location: Local Database
  - Language: PL/SQL
  - PL/SQL Code


```
l_errmsg      varchar2(4000);
e_error       exception;
begin
  l_code := :CODE;
  l_timestamp := currentTimestampIn();
  l_sig := get_signature(l_timestamp, APP_SECRET);

  apex_web_service.g_request_headers(1).name := 'Content-Type';
  apex_web_service.g_request_headers(1).value := 'application/json; charset=utf-8';
```

- Execution Options
- Sequence: 10
- Point: Before Header

在公共页面上执行认证逻辑 (如这里的Login页面)

认证逻辑





用户信息来自钉钉

认证预处理逻辑执行完成之后，直接进入系统。相当于login页面的功能变成了自动执行了认证逻辑并跳转到指定页面。

## 共同点及差异

名称	临时Token	Unionid	AccessToken	UserId	UserName	Apex登录认证
企微	Url	无	Appld, AppSecret	AccessToken, 临时Token	AccessToken, UserId	用户名
钉钉	Url	Appld, AppSecret, 临时Token	Appld, AppSecret	AccessToken, Unionid	AccessToken, UserId	用户名
飞书	Url	无	Appld, AppSecret, 临时Token			用户名

Rest调用



Oracle 身份云服务(IDCS)提供多租户云平台账号群组以及访问管理等功能，针对于企业的身份认证，它可以提供不同的登陆认证方式，不论是单点登录，还是社交媒体登录都可以通过IDCS实现。

## IDCS主要特点包括：

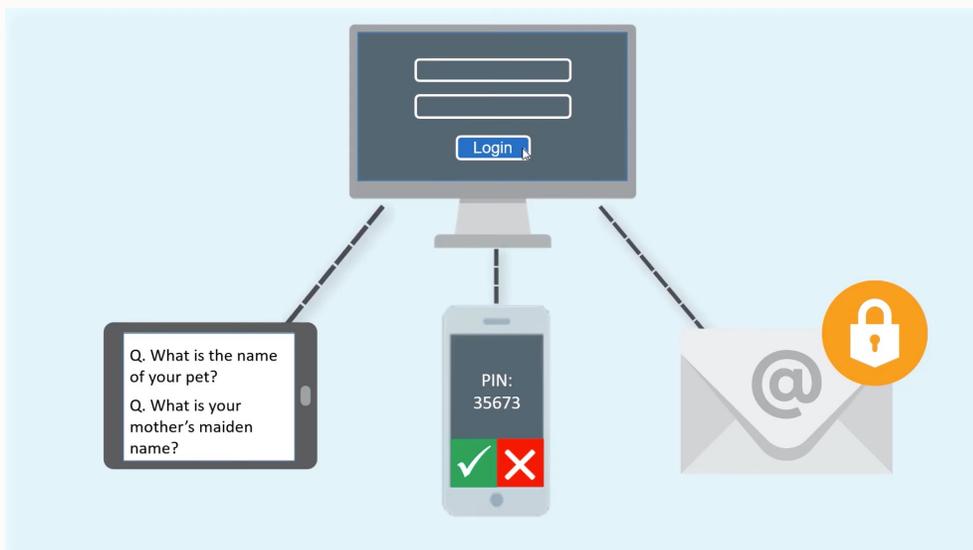
**用户和组群管理**- 账号管理，组群管理机制，Email SMS MFA等多重审核

**单点登录** – 用户不需要每一个应用都重新登录，通过单点登录（SSO）实现只需要登录一次就可以访问所有相互信任的应用系统

**同步** – 桥接代理标准协议，可同步外部AD用户信息到IDCS之中。

**分析** – 用户登录行为分析

Oracle 身份云服务的名称存在两个名称，如下：  
新：Oracle Domain  
旧：IDCS



# 5.外部应用集成

## 案例分享 – Apex与Onfido集成

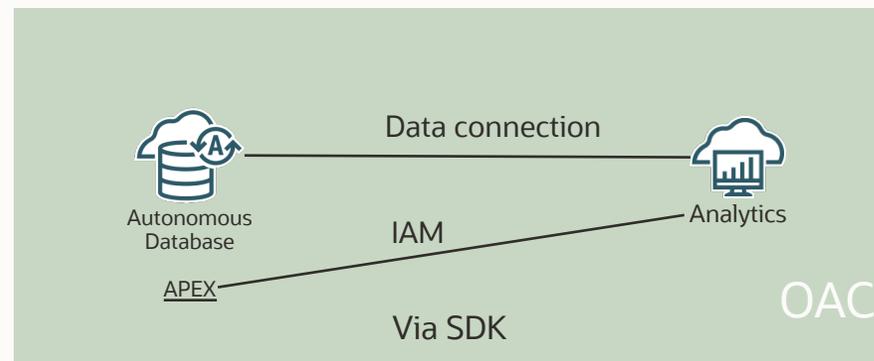
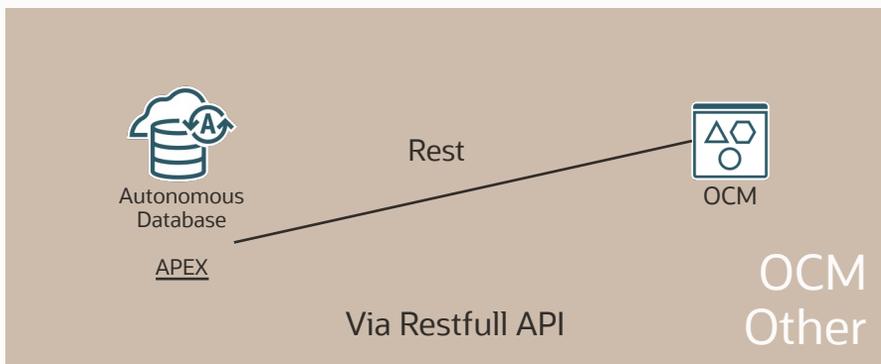
备注：Onfido是一个国外的一种身份验证平台

API 参考：<https://developers.onfido.com/guide/web-sdk-quick-start#start-here>

□ APEX与其他服务间的集成，主要集成方式有两种方式

- ✓ 方式1：SDK
- ✓ 方式2：RESTful API

只有通过IAM认证获取token做安全认证，才可调用接口或引入SDK。



备注：接下来的案例分享包含集成的两种集成方式。



# Apex与Onfido集成 流程图

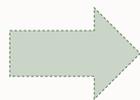
**前置条件**  
在Onfido管理后台创建token，扫码  
记录Api Token信息



扫码登录URL:

[https://open.feishu.cn/open-apis/authen/v1/index?app\\_id=cli\\_a2ceb2d6f0fa900d&redirect\\_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fhyspoc%2Fr%2Ffeishu%2Flogin&&state=authn](https://open.feishu.cn/open-apis/authen/v1/index?app_id=cli_a2ceb2d6f0fa900d&redirect_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fhyspoc%2Fr%2Ffeishu%2Flogin&&state=authn)

[https://open.feishu.cn/open-apis/authen/v1/index?app\\_id=cli\\_a2ceb2d6f0fa900d&redirect\\_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fhyspoc%2Fr%2Ffeishu%2Flogin&&state=authn](https://open.feishu.cn/open-apis/authen/v1/index?app_id=cli_a2ceb2d6f0fa900d&redirect_uri=https%3A%2F%2Fapex.oracle.com%2Fpls%2Fapex%2Fhyspoc%2Fr%2Ffeishu%2Flogin&&state=authn)



## 1. 下载Onfido SDK JS库并导入。'共享组件'->'静态应用程序文件'

名称	Mime 类型	大小	引用	文件	更新时间	更新者
style.css	text/css	273KB	#APP_FILES#style.css	下载	-	-
onfido.min.js	application/javascript	1MB	#APP_FILES#onfido.min.js	下载	-	-

## 2. 画面中引入Js、Css库。

## 3. 添加HTML标签。

4. 左侧标签引入成功后，需在Apex js部分引入初始化SDK的代码

```
Onfido.init({
  token: '<YOUR_WEB_SDK_TOKEN>',
  containerId: 'onfido-mount',
  onComplete: function(data) {
    console.log("A completion message")
  },
  steps: [
    'welcome',
    'document',
    'face',
    'complete'
  ]
});
```



# Apex与Onfido SDK集成REST API调用



```
1 DECLARE
2   l_clob          CLOB;
3   lv_par         VARCHAR2(4000);
4   ln_cnt         NUMBER:=0;
5 BEGIN
6   -- Onfido get api token
7   :P1_API_TOKEN := 'Token token=api_sandbox.cpv9oYOHreiW.Wk0aW0d50xbrcoEf_0ogTQqih3q7H00v1';
8   -- get user info
9   lv_par := '{"first_name": "test", "last_name": "test"}';
10  --
11  -- get token and applicant id
12  IF :APP_TOKEN IS NULL OR :APP_APPLICANT_ID IS NULL THEN
13    -- create applicant
14    apex_web_service.g_request_headers(1).name := 'Authorization';
15    apex_web_service.g_request_headers(1).value := :P1_API_TOKEN;
16    apex_web_service.g_request_headers(2).name := 'Content-Type';
17    apex_web_service.g_request_headers(2).value := 'application/json';
18  --
19  -- Get applicant id
20  l_clob := apex_web_service.make_rest_request(
21    p_url => 'https://api.onfido.com/v3.3/applicants',
22    p_http_method => 'POST',
23    p_body => lv_par);
24  apex_json.parse(l_clob);
25  :APP_APPLICANT_ID := apex_json.get_varchar2('id');
26  --
27  -- Generate an SDK token
28  lv_par := '{
29    "applicant_id": "' || :APP_APPLICANT_ID || '"';
30  k_clob := apex_web_service.make_rest_request(
31    p_url => 'https://api.eu.onfido.com/v3.3/sdk_token',
32    p_http_method => 'POST',
33    p_body => lv_par);
34  apex_json.parse(k_clob);
35  :APP_TOKEN := apex_json.get_varchar2('token');
36  END IF;
37 END;
```

STEP1:取得API Token

STEP2:创建申请人信息

STEP3:取得SDK Token

STEP4:在Apex中收集的证件信息和人脸信息(拍照)



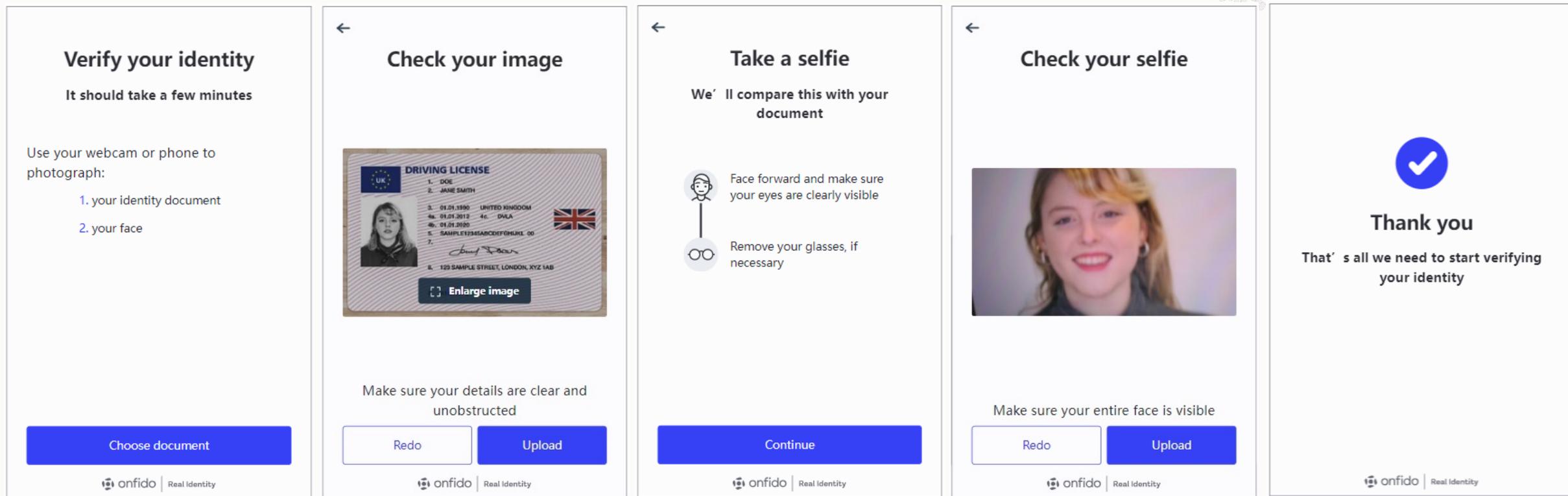
```
1 DECLARE
2   lv_clob        CLOB;
3   lv_par         VARCHAR2(4000);
4 BEGIN
5   apex_web_service.g_request_headers.delete();
6   apex_web_service.g_request_headers(1).name := 'Authorization';
7   apex_web_service.g_request_headers(1).value := :P1_API_TOKEN;
8   --
9   -- Get applicant id
10  --
11  lv_par := '{
12    "applicant_id": "' || :P1_ID || '"',
13    "report_names": ["document", "facial_similarity_photo"]';
14  --
15  lv_clob := apex_web_service.make_rest_request(
16    p_url => 'https://api.eu.onfido.com/v3.3/checks',
17    p_http_method => 'POST',
18    p_body => lv_par);
19  --
20  apex_json.parse(lv_clob);
21  apex_json.open_object;
22  apex_json.write('id', apex_json.get_varchar2('id'));
23  apex_json.close_object;
24  --
25  END;
```

STEP5:身份信息收集完成后,使用Rest Apt进行身份验证

API参考: [https://docs.oracle.com/en/database/oracle/apex/22.1/aeapi/MAKE\\_REST\\_REQUEST-Function.html#GUID-C77BB45A-8968-470C-8243-BADB63743DE9](https://docs.oracle.com/en/database/oracle/apex/22.1/aeapi/MAKE_REST_REQUEST-Function.html#GUID-C77BB45A-8968-470C-8243-BADB63743DE9)



STEP4：在Apex中收集的证件信息和人脸信息，如下图所示。



备注：

- Apex中开发的应用也支持手机原生功能，如摄像头，推送，指纹，人脸识别，位置等



# 身份认证Check信息

## 检查摘要

清除

### 检测到的问题



未检测到任何问题



人脸对比



[查看全尺寸媒体](#)

检查状态	最近更新时间	申请人编号	9834b420-794f-4f28-9c0f-c261e54fcee3	
完全的	2022年7月4日 10:26 (UTC)	检查身份证	e13b452f-e546-492f-8626-abddb550f9b	



## 面部相似度报告

清除

[在此处](#)查找有关面部相似性报告结果的更多详细信息

[标记问题](#)

### 视觉真实性

#### 欺骗检测

得分: 0.9512

### 图像完整性

#### 来源完整性

检测到人脸

### 人脸对比

#### 人脸匹配

得分: 0.6512

文档 uuid: 0756a18c-bf16-49dd-85d9-e1dafa2989d3

## 文件报告

清除

[在此处](#)查找有关文档报告结果的更多详细信息

[标记问题](#)

### 年龄验证

#### 最低接受年龄

### 警方记录

文件未被记录为丢失、被盗或损坏

### 图像完整性

#### 彩色图片

支持的文件

确凿的文件质量

画面质量

### 数据比较

出生日期

名

文件编号

文件类型

姓

到期日期

性别

发行国

### 数据一致性

国籍

姓

名

存在多个数据源

性别

出生日期

文件类型

发行国

文件编号

到期日期

### 受损文件

在我们的数据库中找不到文档

## 6. APEX计划自动化任务

案例分享 -某公司使用场景：每当一笔财务数据录入后，在后台不选择实时触发结算财务金额。为了节省资源，采用定时任务结算财务金额。

备注：在某些场景下，遇到紧急结算时..也支持实时调用。

# 自动化是什么？

自动化是一组连续的操作，由[监控数据的查询结果](#)触发，然后执行适当的操作。

## 自动化的主要特点包括：

**定义自动化操作** - 定义多个操作并让它们按顺序为每个自动化查询行执行。典型的自动化操作是 PL/SQL 代码，也可以进行邮件发送。通过[监控数据的查询结果](#)或者 PL/SQL 函数返回 TRUE 时，可以对查询结果的每一行执行自动化操作。可以通过条件控制自动化是否执行。查询结果行中的列值可用作绑定变量。

**编程方式触发** - 使用 PL/SQL 接口 APEX\_AUTOMATION 软件包触发自动化。

**查看自动化日志** - [执行日志](#)中可以查看自动化执行的详细信息。[自动化历史记录](#)中也可以查看自动化修改的历史。

## 创建Apex自动化：‘共享组件’->‘工作流和自动化’->‘自动化’

The screenshot shows the Oracle APEX Automation interface. At the top, there is a breadcrumb trail: '应用程序 129049 \ 共享组件 \ 自动化'. Below this, there are tabs for '自动化', '执行日志', and '历史记录'. A search bar and a '开始' button are visible. The main area contains a table with the following columns: '名称', '类型', '以下情况下执行操作:', '调度状态', '上次运行', '下次运行', and '运行'. The table has one row: '财务结算自动化', '已调度', '返回的行数', '已禁用', and empty cells for '上次运行' and '下次运行'. A '运行' button is present in the '运行' column. On the right side, there is a sidebar with the title '自动化' and a description: '自动化功能根据调度启动一组连续操作。它们用于监视数据，然后执行适当的操作（例如，自动批准特定请求和发送电子邮件预警）。' Below the description is a '了解详细信息...' link. At the bottom right of the sidebar, there are '复制' and '创建' buttons.

名称	类型	以下情况下执行操作:	调度状态	上次运行	下次运行	运行
财务结算自动化	已调度	返回的行数	已禁用			运行



# 自动化创建

### 创建自动化

\* 名称

类型

操作启动日期

执行调度

频率

间隔

执行时间   
午夜, 早晨 6 点, 中午, 下午 6 点

**设定自动化默认执行时间**

### 创建自动化

数据源

源类型

\* 输入 SQL SELECT 语句

```
1 SELECT SUM(AMOUNT) SUM_AMOUNT
2     , BUSINESS_UNIT
3     , ACCOUNT
4 FROM   GL_INTERFACE_TEMP
5 WHERE  COMPLETE_FLAG = 'N'
6 GROUP BY BUSINESS_UNIT
7     , ACCOUNT
```

以下情况下执行操作:

监控数据的查询结果

自动化创建完成后，可以在下图所示的操作中添加自动化的执行代码或者进行邮件推送设置。

## 自动化：财务结算自动化

优化程序提示

操作

搜索: 所有文本列

<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	名称	执行序列	操作类型	位置
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	新建操作	10	执行代码	本地数据库

已选择 1 行 总计 1



主页

财务金额Report

设置

## 设置

清除automation测试数据

Q

开始

操作

创建

	任务名称	定时任务执行时间
	财务结算定时任务	立即执行

1-1



ORACLE