

日々是Oracle APEX

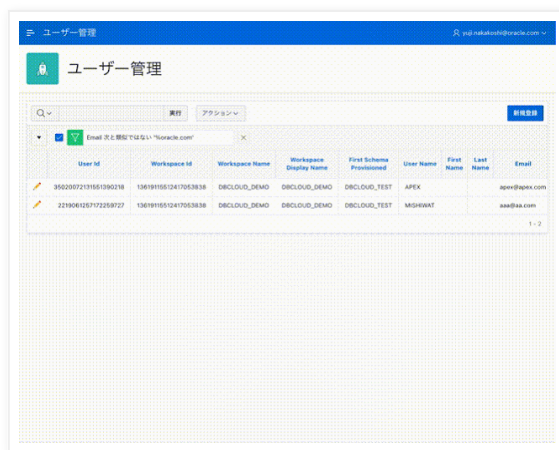
Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2021年2月20日 土曜日

ユーザー管理用のAPIを呼び出す

以前に[管理サービスの禁止](#)方法を記事にしました。管理サービスを禁止すると、色々と不便になりますが、最初に困るのはユーザー管理でしょうか。

Oracle APEXはユーザー管理を行うAPIを提供しているので、それを使うことで管理画面を自作することができます。試しに簡単なアプリを作ってみました。Oracle APEXが提供しているビューとAPIを使うことで比較的容易にアプリケーションは作れました。



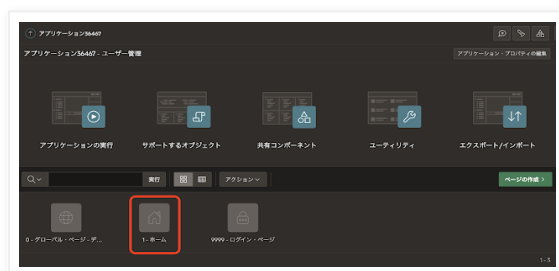
とはいえ、管理サービスを禁止するくらいセキュリティについて意識するのであれば、ユーザー管理はOAuth2などの認証プロトコルを使用して、外部の認証サービスを使うようにすべきでしょう。そこまでいかなくても、自前で表を作ってユーザー情報を保存し、カスタム認証を実装しても良いのではないかと思います。

アプリケーション自体は単純な作りですが、ページ・アイテムが多く、それを手作りする必要があるので、最初にアプリケーションのエクスポートを紹介します。

<https://github.com/ujnak/apexapps/blob/master/exports/sampleusermgmt.sql>

では、作り方の方を紹介します。

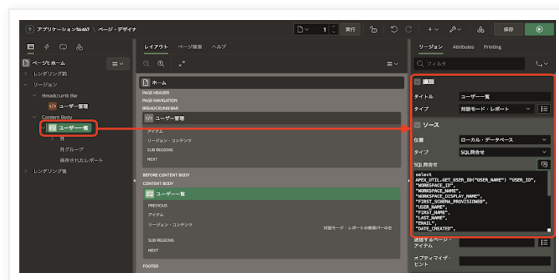
最初に空のアプリケーションを作成します。名前は**ユーザー管理**とします。



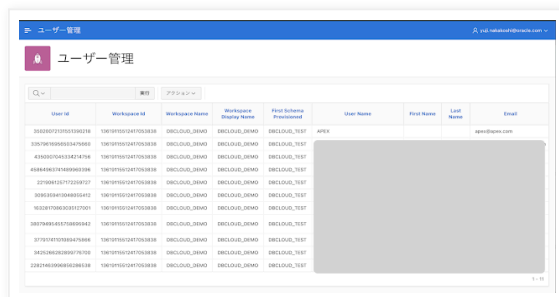
ホーム・ページに登録済みのユーザーを一覧する対話モード・レポートを作成します。一覧に使用するSQLは以下です。ビュー**APEX_WORKSPACE_APEX_USERS**の検索結果を表示しています。

```
select
APEX_UTIL.GET_USER_ID("USER_NAME") "USER_ID",
"WORKSPACE_ID",
"WORKSPACE_NAME",
"WORKSPACE_DISPLAY_NAME",
"FIRST_SCHEMA_PROVISIONED",
"USER_NAME",
"FIRST_NAME",
"LAST_NAME",
"EMAIL",
"DATE_CREATED",
"DATE_LAST_UPDATED",
"AVAILABLE_SCHEMAS",
"IS_ADMIN",
"IS_APPLICATION_DEVELOPER",
"ACCOUNT_LOCKED",
"DESCRIPTION",
"PASSWORD_VERSION",
"ACCOUNT_EXPIRY",
"FAILED_ACCESS_ATTEMPTS",
"PROFILE_IMAGE_NAME",
"PROFILE_MIMETYPE",
"PROFILE_FILENAME",
"PROFILE_CHARSET"
from "APEX_WORKSPACE_APEX_USERS"
```

名前はユーザー一覧、ソースのタイプはSQL問合せです。



作成したページを実行すると、登録されているユーザーの一覧が表示されます。

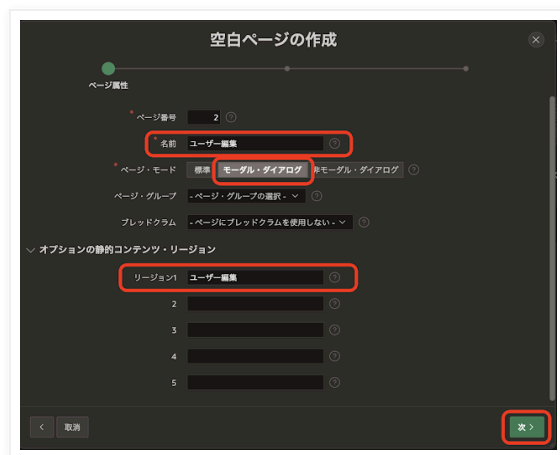


次にフォームとなるページを作成します。静的コンテンツを使ってページを作ります。

ページ作成ウィザードを起動し、コンポーネントの空白ページを選びます。



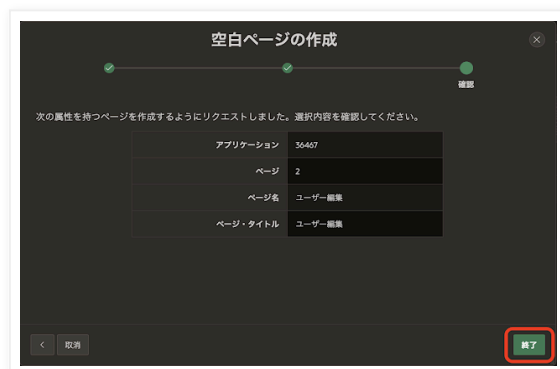
名前をユーザー編集、ページ・モードをモーダル・ダイアログとします。オプションの静的コンテンツ・リージョンとして、リージョン1にユーザー編集を指定します。次に進みます。



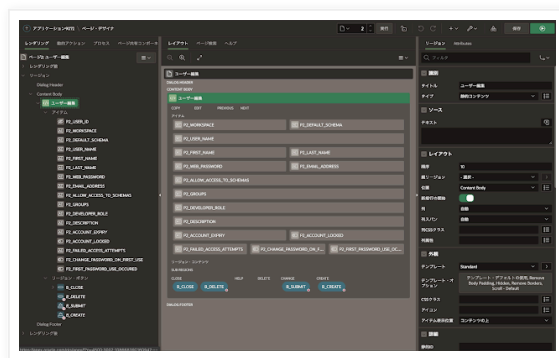
ダイアログのページなので、ナビゲーションのプリファレンスとしては、このページとナビゲーション・メニューを関連付けないを選択します。次に進みます。



設定内容を確認して、終了をクリックします。



以上で、ダイアログとなるページが作成されました。作成された静的リージョンにページ・アイテムとボタンを配置します。



作成するページ・アイテムは以下です。

P2_USER_ID (非表示)、P2_WORKSPACE、P2_DEFAULT_SCHEMA、P2_USER_NAME、P2_FIRST_NAME、P2_LAST_NAME、P2_WEB_PASSWORD、P2_EMAIL_ADDRESS、P2_ALLOW_ACCESS_TO_SCHEMAS、P2_GROUPS、P2_DEVELOPER_ROLE、P2_DESCRIPTION、P2_ACCOUNT_EXPIRY、P2_ACCOUNT_LOCKED(切替え)、P2_FAILED_ACCESS_ATTEMPTS、P2_CHANGE_PASSWORD_ON_FIRST_USE(切替え)、P2_FIRST_PASSWORD_USE_OCCURED(切替え)

見やすくなるようにレイアウトを変更しましょう。

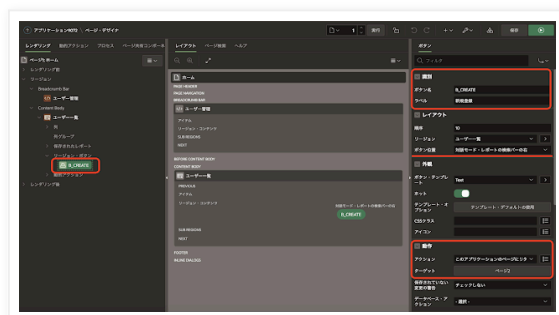
次にボタンですが、

B_CREATE、B_SUBMIT、B_DELETE、B_CLOSE

の4つのボタンを作成します。それぞれ、**作成**、**変更の適用**、**削除**と**閉じる**、というラベルを割り与えます。**B_CREATE**は主キーである**P2_USER_ID**が**NULL**の場合に表示し、**B_SUBMIT**、**B_DELETE**は反対に**P2_USER_ID**が**NULL**でない場合に表示するようサーバー側の条件を構成します。標準で、Create、Change、Delete、Closeとして推奨されているボタン位置があります。ただ、Deleteだけは作成と変更の適用から離すために、Closeの場所に配置しました。

次にレポートのページに戻り、ユーザーの新規登録のボタンを作成します。**ボタン名**を**B_CREATE**とし、**ラベル**を**新規登録**、**ボタン位置**は**対話モード・レポートの検索バーの右**に配置します。

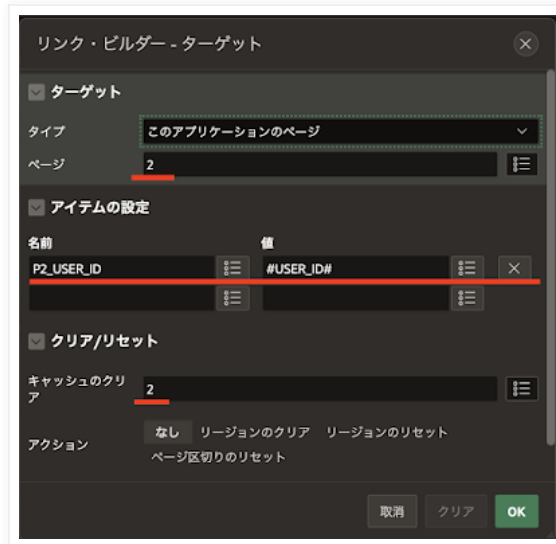
動作のアクションはこの**アプリケーションのページ**に**リダイレクト**を選択し、**ターゲット**は**ページ 2**として、**フォーム**を作成したページを指定します。



選択した行の編集フォームを開くよう、対話モード・レポートの**Attributes**に含まれる**リンク**を設定します。**リンク列**は**カスタム・ターゲット**への**リンク**を選択します。



ターゲットは、ページを2、アイテムの設定として名前をP2_USER_ID、値は#USER_ID#を設定します。キャッシュのクリアも2を設定します。



以上で画面に必要なコンポーネントは配置できました。

ここからが今回の本題です。フォームのページに戻ります。

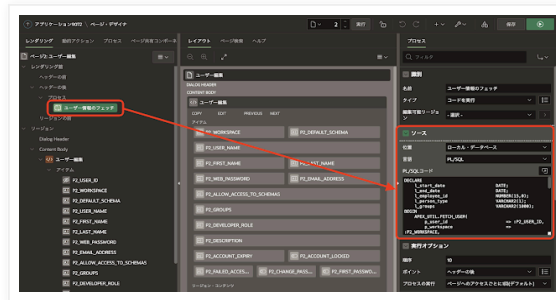
ユーザー情報をフェッチし、ページ・アイテムに設定するプロセスを登録します。APIとしては、[APEX_UTIL.FETCH_USER](#)を呼び出します。ヘッダーの後にプロセスを作成し、以下のコードをソースとして設定します。名前をユーザー情報のフェッチとし、タイプはコードを実行を選びます。

```
DECLARE
    l_start_date      DATE;
    l_end_date        DATE;
    l_employee_id     NUMBER(15,0);
    l_person_type     VARCHAR2(1);
    l_groups           VARCHAR2(1000);
BEGIN
    APEX_UTIL.FETCH_USER(
        p_user_id      => :P2_USER_ID,
        p_workspace    => :P2_WORKSPACE,
        p_user_name     => :P2_USER_NAME,
        p_first_name    => :P2_FIRST_NAME,
        p_last_name     => :P2_LAST_NAME,
        p_web_password  => :P2_WEB_PASSWORD,
        p_email_address => :P2_EMAIL_ADDRESS,
        p_start_date    => l_start_date,
        p_end_date      => l_end_date,
        p_employee_id   => l_employee_id,
        p_allow_access_to_schemas => :P2_ALLOW_ACCESS_TO_SCHEMAS,
        p_person_type   => l_person_type,
        p_default_schema => :P2_DEFAULT_SCHEMA,
        p_groups        => l_groups,
        p_developer_role => :P2_DEVELOPER_ROLE,
        p_description   => :P2_DESCRIPTION,
```

```

p_account_expiry          => :P2_ACCOUNT_EXPIRY,
p_account_locked          => :P2_ACCOUNT_LOCKED,
p_failed_access_attempts  => :P2_FAILED_ACCESS_ATTEMPTS,
p_change_password_on_first_use => :P2_CHANGE_PASSWORD_ON_FIRST_USE,
p_first_password_use_occurred => :P2_FIRST_PASSWORD_USE_OCCURED);
select listagg(apex_util.get_group_name(column_value),':') into :P2_GROUPS
from apex_string.split(l_groups, ':');
END;

```



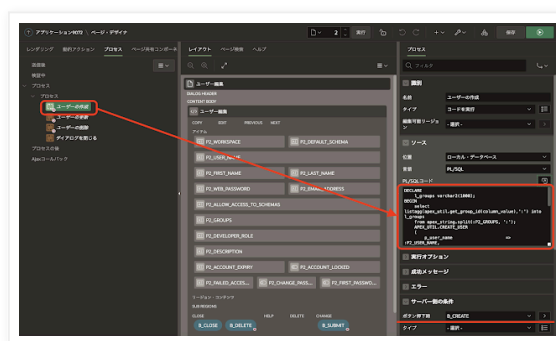
左ペインをプロセス・ビューに切り替え、データ操作に関するプロセスを3つ追加します。

最初にユーザーを作成するプロセスを追加します。APEX_UTIL.CREATE_USERを呼び出します。名前をユーザーの作成とし、ボタン押下時はB_CREATEとして、作成ボタンがクリックされたときに実行します。実行するコードは以下になります。

```

DECLARE
  l_groups varchar2(1000);
BEGIN
  select listagg(apex_util.get_group_id(column_value),':') into l_groups
  from apex_string.split(:P2_GROUPS, ':');
  APEX_UTIL.CREATE_USER
  (
    p_user_name          => :P2_USER_NAME,
    p_first_name         => :P2_FIRST_NAME,
    p_last_name          => :P2_LAST_NAME,
    p_description        => :P2_DESCRIPTION,
    p_email_address      => :P2_EMAIL_ADDRESS,
    p_web_password       => :P2_WEB_PASSWORD,
    p_group_ids          => l_groups,
    p_developer_privs    => :P2_DEVELOPER_ROLE,
    p_default_schema     => :P2_DEFAULT_SCHEMA,
    p_allow_access_to_schemas => :P2_ALLOW_ACCESS_TO_SCHEMAS,
    p_change_password_on_first_use => :P2_CHANGE_PASSWORD_ON_FIRST_USE
  );
END;

```

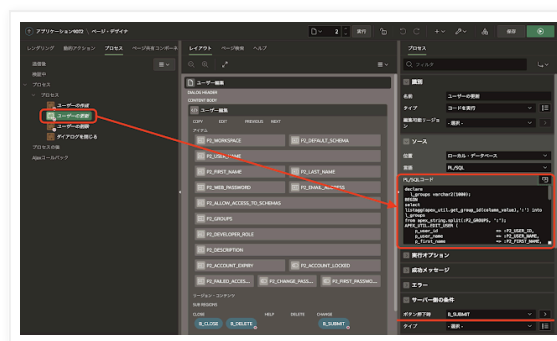


ユーザーの更新では、APEX_UTIL.EDIT_USERを呼び出します。ボタンB_SUBMITが押された時に実行します。

```

declare
l_groups varchar2(1000);
BEGIN
select listagg(apex_util.get_group_id(column_value),':') into l_groups
from apex_string.split(:P2_GROUPS, ':');
APEX_UTIL.EDIT_USER (
    p_user_id          => :P2_USER_ID,
    p_user_name        => :P2_USER_NAME,
    p_first_name       => :P2_FIRST_NAME,
    p_last_name        => :P2_LAST_NAME,
    p_web_password     => :P2_WEB_PASSWORD,
    p_new_password     => :P2_WEB_PASSWORD,
    p_email_address    => :P2_EMAIL_ADDRESS,
-- p_start_date       => l_start_date,
-- p_end_date         => l_end_date,
-- p_employee_id      => l_employee_id,
    p_allow_access_to_schemas => :P2_ALLOW_ACCESS_TO_SCHEMAS,
-- p_person_type      => l_person_type,
    p_default_schema   => :P2_DEFAULT_SCHEMA,
    p_group_ids        => l_groups,
    p_developer_roles  => :P2_DEVELOPER_ROLE,
    p_description      => :P2_DESCRIPTION,
    p_account_expiry   => :P2_ACCOUNT_EXPIRY,
    p_account_locked   => :P2_ACCOUNT_LOCKED,
    p_failed_access_attempts => :P2_FAILED_ACCESS_ATTEMPTS,
    p_change_password_on_first_use => :P2_CHANGE_PASSWORD_ON_FIRST_USE,
    p_first_password_use_occurred => :P2_FIRST_PASSWORD_USE_OCCURED);
END;

```

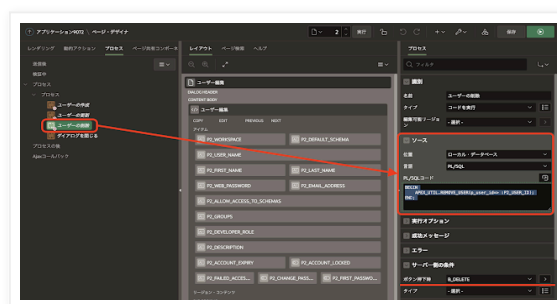


ユーザーの削除では、[APEX_UTIL.REMOVE_USER](#)を呼び出します。ボタンB_DELETEがクリックされたときに実行します。

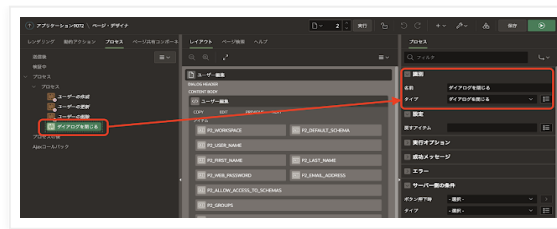
```

BEGIN
APEX_UTIL.REMOVE_USER(p_user_id=> :P2_USER_ID);
END;

```

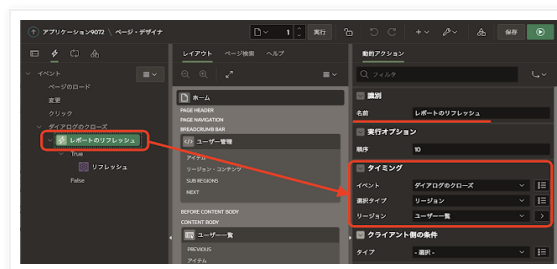


最後にダイアログを閉じるプロセスを追加します。



レポートのページにダイアログがクローズしたときに、レポートをリフレッシュする動的アクションを定義します。動的アクション・ビューを開き、ダイアログのクローズで動的アクションを作成します。

名前はレポートのリフレッシュとし、タイミングのイベントはダイアログのクローズ、選択タイプはリージョン、リージョンはユーザー一覧とします。



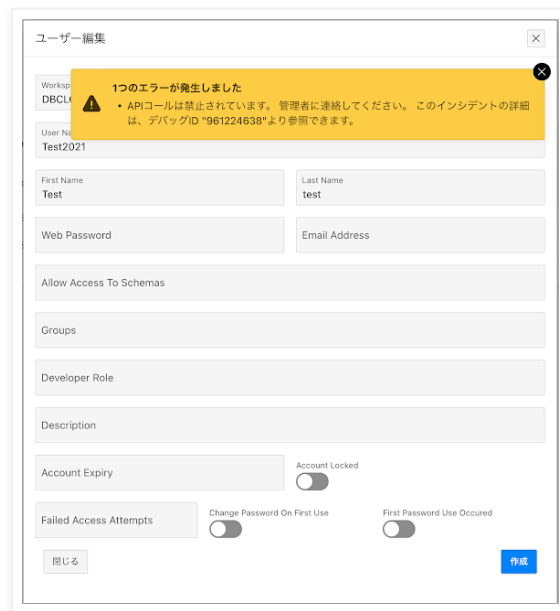
Trueアクションを作成します。アクションはリフレッシュ、影響を受ける要素は、選択タイプがリージョン、リージョンはユーザー一覧になります。



以上でアプリケーションは完成です。

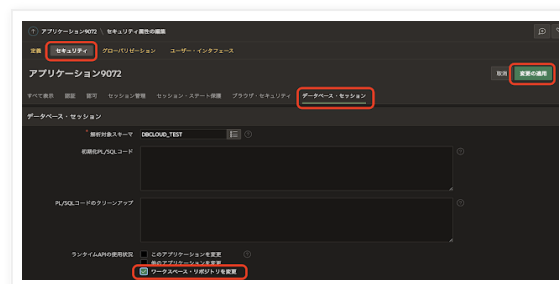
アプリケーションを実行し、ユーザーを登録しようとする、以下のエラーが発生します。

APIコールは禁止されています。管理者に連絡してください。



Oracle APEXで作成したアプリケーションからは、**管理系のAPIの呼び出しが禁止**されています。こちらの制限を解除する必要があります。

アプリケーション定義のセキュリティを開き、データベース・セッションのセクションに含まれる、ランタイムAPIの使用状況のワークスペース・リポジトリを変更にチェックを入れます。



再度、アプリケーションを実行すると、表題にあるGIF動画のような動作になります。

APIの確認が目的なので、アプリケーションとしては改良の余地は多々あります。たとえば、パスワードを隠す、など。また、本アプリを本番環境にインストールするのは全くお勧めしません。あくまでサンプルです。

本記事は以上になります。Oracle APEXのアプリケーション開発の一助になれば幸いです。

完

Yuji N. 時刻: 13:48

共有

<

ホーム

>

ウェブ バージョンを表示

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。
こちらの記事につきましては、免責事項の参照をお願いいたします。

[詳細プロフィールを表示](#)

Powered by Blogger.
