

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。

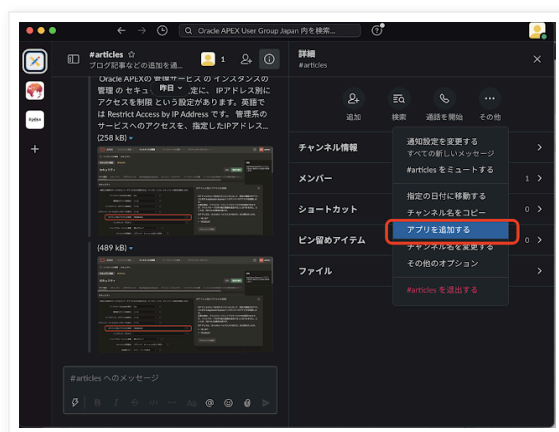
2021年4月15日 木曜日

APEXのアプリケーションからSlackに通知を行う

Oracle APEXのアプリケーションからSlackに通知を行う方法について紹介します。通知には、SlackのアプリIncoming Webhookを使用します。

SlackのチャンネルにアプリケーションIncoming Webhookを追加します。デスクトップ・アプリケーションから設定を行います。(多分Webからでも可能だとは思いますが)

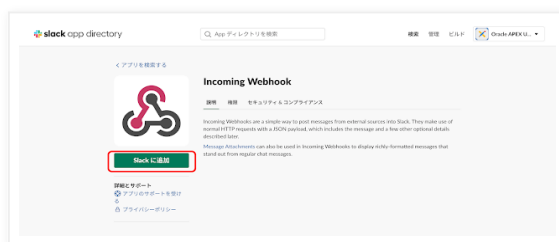
通知を送信するチャンネルを選択し、**その他からアプリを追加する**を実行します。



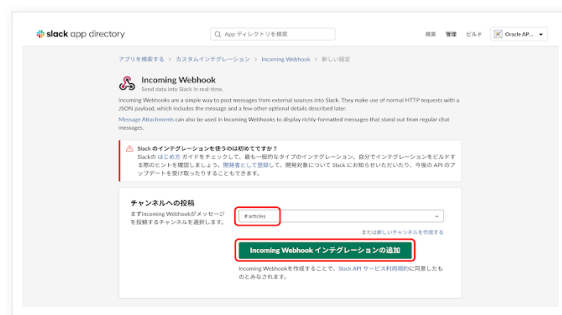
Incoming Webhookを検索し、見つけたアプリケーションをインストールします。



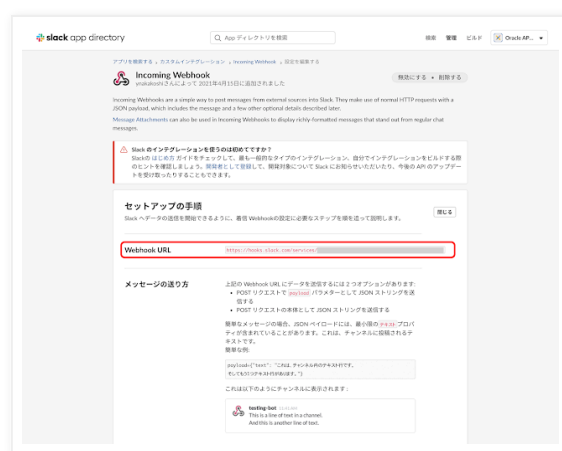
ブラウザが開き、Incoming Webhookの概要が表示されます。**Slackに追加**を実行します。



Incoming Webhookでは通知を投稿と呼んでいます。**チャンネルへの投稿**として、投稿するチャンネルを選択します。そして、**Incoming Webhookインテグレーションの追加**を実行します。



Incoming Webhookが追加され、投稿に必要な情報がブラウザに表示されます。**Webhook URL**として返されているURLに、投稿する内容をJSON形式でPOSTすると、Slackのチャンネルに表示されます。



認証などで保護はされていないため、Webhook URLは投稿を行うアプリケーションの開発者に限定して伝えることが望ましいでしょう。

Incoming Webhookの設定として、**説明ラベル**、**名前**、**アイコン**を設定していると、投稿がどこから行われたか簡単に判断できるようになります。



Slack側の設定は以上で完了です。Oracle APEX側からはWebhook URLに投稿したい内容をJSON形式にして送信します。

マークダウン形式のテキストを投稿するプロシージャは以下になります。

```

create or replace procedure slk_invoke_webhook
(
  p_message in varchar2,
  p_webhook_url in varchar2
)
is
  l_body clob;
  l_ret clob;
  j_message json_object_t;
  j_blocks json_array_t;
  j_text json_object_t;
  j_ptext json_object_t;
begin
  j_blocks := new json_array_t;
  j_ptext := new json_object_t;
  j_ptext.put('type','mrkdwn');
  j_ptext.put('text', p_message);
  j_text := new json_object_t;
  j_text.put('type','section');
  j_text.put('text', j_ptext);
  j_blocks.append(j_text);
  --
  j_message := new json_object_t;
  j_message.put('blocks', j_blocks);
  l_body := j_message.to_clob;
  -- apex_debug.info(l_body);
  apex_web_service.g_request_headers.delete();
  apex_web_service.g_request_headers(1).name := 'Content-Type';
  apex_web_service.g_request_headers(1).value := 'application/json';
  l_ret := apex_web_service.make_rest_request
  (
    p_url => p_webhook_url,
    p_http_method => 'POST',
    p_body => l_body
  );
end slk_invoke_webhook;

```

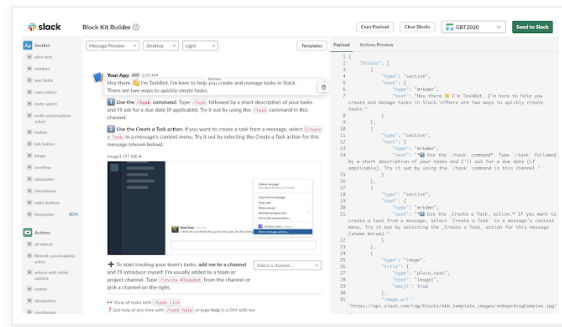
投稿するマークダウンを、Incoming Webhookによって投稿可能なJSON形式にするために、最低限必要な変換を行っています。

```

{
  "blocks": [
    {
      "type": "section",
      "text": {
        "type": "mrkdwn",
        "text": "投稿するメッセージ"
      }
    }
  ]
}

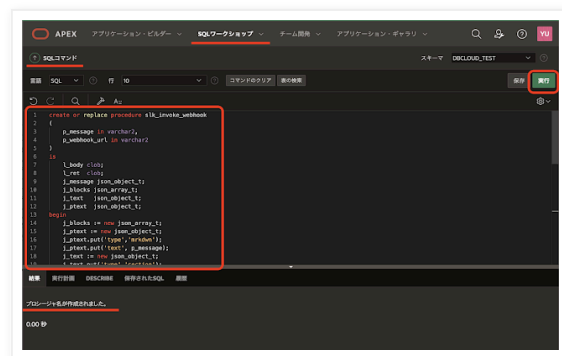
```

投稿するメッセージはSlackの[Block Kit](#)および[Block Kit Builder](#)を使うことで、容易に形式を決めることができます。Block Kitについては、Slackからの説明を参照してください。

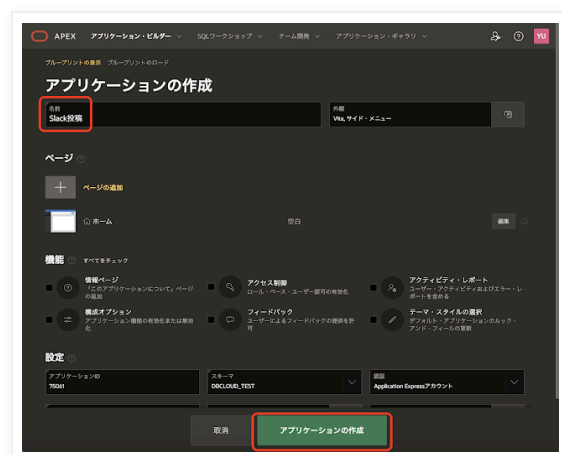


では、Incoming Webhookを検証する簡単なアプリケーションを作成してみます。

最初にSQLワークショップからSQLコマンドを開き、上記のプロシージャSLK_INVOKE_WEBHOOKを作成します。プロシージャのコードを貼り付け、**実行**をクリックします。



続いて、アプリケーション・ビルダーより空のアプリケーションを作成します。名前は**Slack投稿**とします。**アプリケーションの作成**を実行します。



アプリケーションが作成されたのち、Slackに投稿するフォームのページを作成します。**ページの作成**をクリックします。



フォームを選択します。



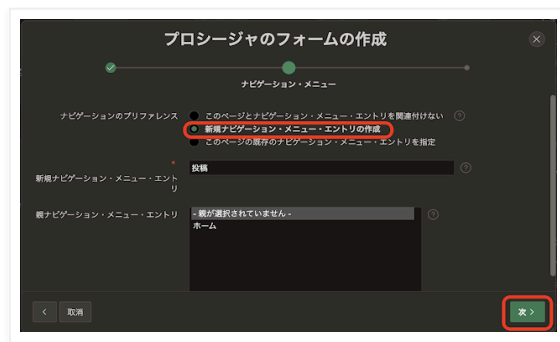
ローカル・プロシージャのフォームを選択します。



ページ名は**投稿**とします。ページ・モードは**標準**、送信時にここに**ブランチ**および**取り消してページに移動**には、作成される**ページ番号**と同じ数値（今回の例では**3**）を指定します。次に進みます。



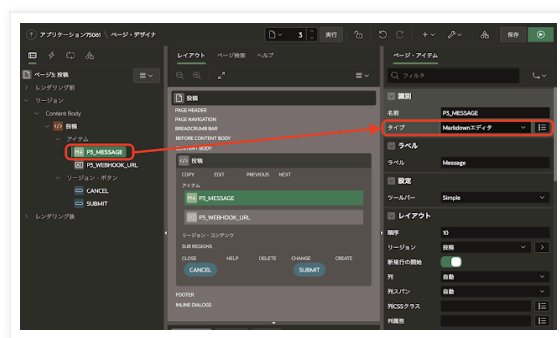
ナビゲーションのプリファレンスとして**新規ナビゲーション・メニュー・エントリ**の作成を選択します。次に進みます。



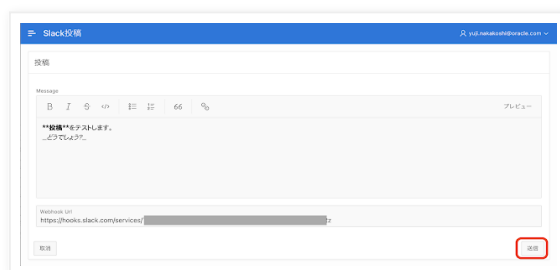
ストアド・プロシージャ名に先程作成したプロシージャSLK_INVOKE_WEBHOOKを選択します。作成をクリックします。



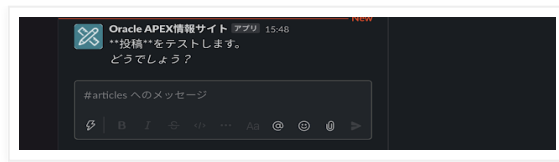
ページが作成されたら、ページ・アイテムP3_MESSAGEのタイプをMarkdownエディタに変更します。



作成したページを実行します。Webhook URLを入力し、マークダウンでメッセージを記述したのち、送信をクリックします。



Slackのチャンネルを確認します。



通知が確認できれば完了です。

今回作成したアプリケーションのエクスポートを以下に置きました。

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

Oracle APEXのアプリケーション作成の参考になれば幸いです。

完

Yuji N. 時刻: 15:59

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.