

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。

2023年3月8日水曜日

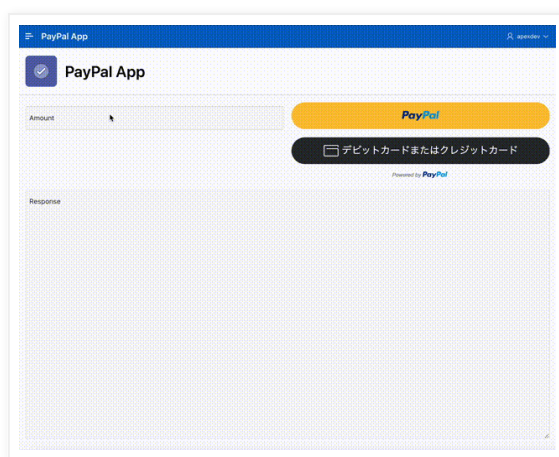
PayPal決済をアプリに組み込む

PayPalによる決済をAPEXアプリケーションに組み込んでみます。元ネタはJon Dixonさんのブログ記事[Receiving Payments from Oracle APEX with PayPal](#)です。

Jon Dixonさんが記事で作成されているAPEXアプリケーションは、デモとして一通りの機能が実装されています。本記事では**PayPal決済を組み込む実装に絞って解説**します。ショッピング・カートの実装は行いません。

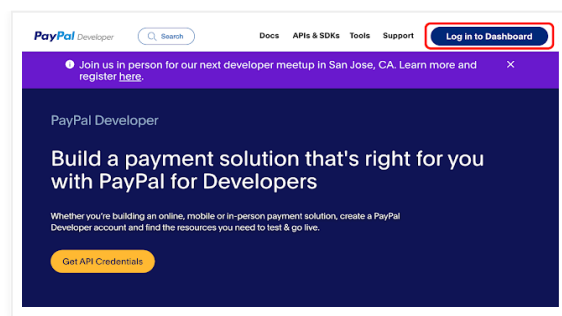
作成するアプリケーションは以下のように動作します。

ページ・アイテム**Amount**に決済金額を入力し、**PayPalボタン**を押して支払いを行います。支払いの完了メッセージはJSON形式のままページ・アイテム**Response**に印刷します。

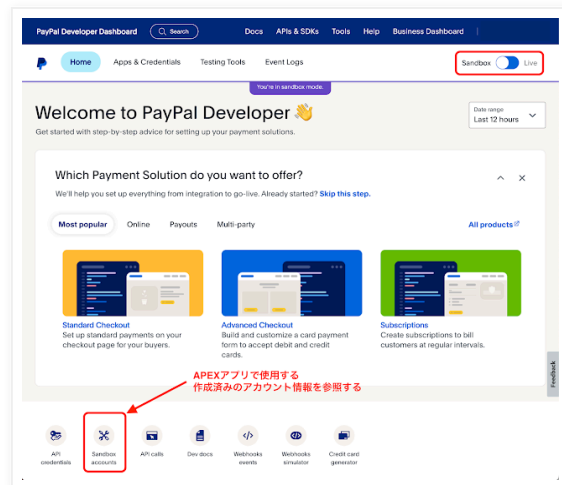


PayPal側の準備から始めます。[PayPal Developer](#)のサイトに接続し、開発者アカウントを作成します。画面右上の**Log in to Dashboard**をクリックして作業を進めます。

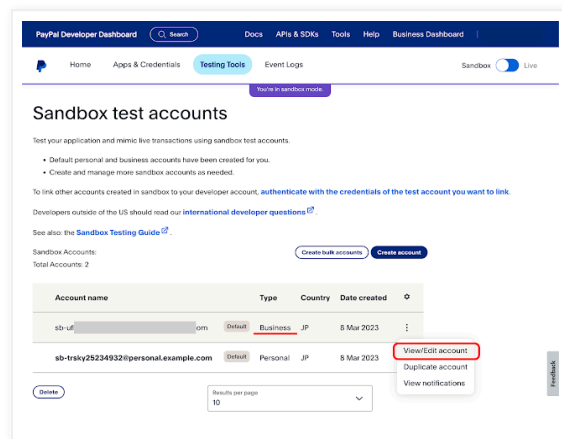
私は一般のPayPalのアカウント（PayPalによる決済ができるアカウント）が作成済みだったため、そのアカウントでPayPal Developerのサイトに接続できるようになりました。[こちら](#)を参照するとPayPalビジネスアカウントの作成といった手順が含まれていますが、これから紹介する作業では開発者アカウントに含まれている[Sandbox test accounts](#)を使用するため、開発者アカウント（その元となった一般のPayPalアカウント）以外のアカウントは登録しません。



PayPal Developer DashboardからSandbox accountsを開きます。APEXのアプリケーションで使用するClient IDと、支払いのテストで使用するPersonalアカウントとパスワードを確認します。



BusinessとPersonalの2つのタイプのアカウントが作成されています。最初にBusinessのアカウントよりClient IDを確認します。View/Edit accountを実行します。



ダイアログが開きます。

API Credentialsのタブを選択し、REST AppsのDefault Applicationを開きます。

Account details

Profile API Credentials Funding Settings

NVP/SOAP Sandbox API Credentials

Username:
sb-ufb-...@sample.com

Password
[Redacted]

Signature:
[Redacted]

REST Apps

App name
[Redacted]

Default Application

Close

Default Applicationの設定が開きます。SANDBOX API CREDENTIALSのClient IDをコピーしておきます。この値はAPEXアプリケーションに作成する置換文字列G_CLIENT_IDの置換値として設定します。

App feature optionsのAccept paymentsにチェックが入っていることを確認します。

Businessアカウントでの確認は以上です。

Home Apps & Credentials Testing Tools Event Logs Sandbox Live

Default Application

App display name: APEX Demo

SANDBOX API CREDENTIALS

Sandbox Account
sb-ufb-...@sample.com

Client ID
sb-ufb-...@sample.com

Secret
[Redacted]

SANDBOX APP SETTINGS

Return URL: Users are redirected to this URL after live transactions. Allow up to three hours for the change to take effect. [Show](#)

App feature options

- ☒ **Accept payments** Enable one-time and subscription payments. [Advanced options](#)
- ☒ **Invoicing** Issue invoices for payments owed, manage partial balances due, and enable custom net payment terms.
- ☒ **Payments** Send batch payments to multiple PayPal accounts at once. You can vary the amount by recipient and select if you'd like it delivered by phone number or email.
- ☒ **Customer Disputes** Use the PayPal Disputes API to list disputes, provide evidence, accept claims, show dispute details, and appeal disputes.
- ☐ **PayPal Here** Process swiped/cash present card transactions.
- ☐ **Log in with PayPal** Identity service that enables your customers to login with their PayPal login.
- ☐ **Transaction Search** Access your PayPal transaction history.

[Save](#) [Cancel](#)

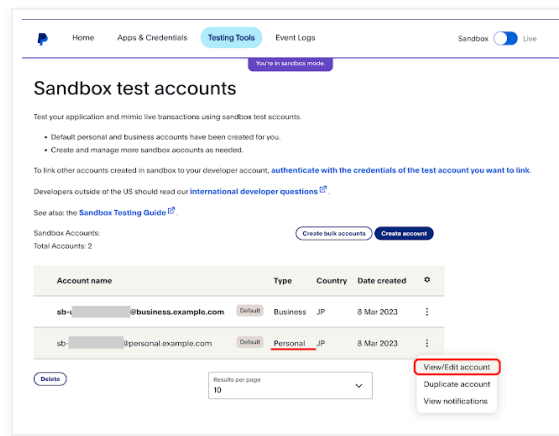
SANDBOX WEBHOOKS

Configure webhooks to notify your app when certain events occur. To configure a webhook, define your webhook listener URL and a list of events for which to listen. You can configure up to ten webhooks. Each webhook can subscribe to either specific events or all events. To learn more about webhooks, see [webhooks notifications](#).

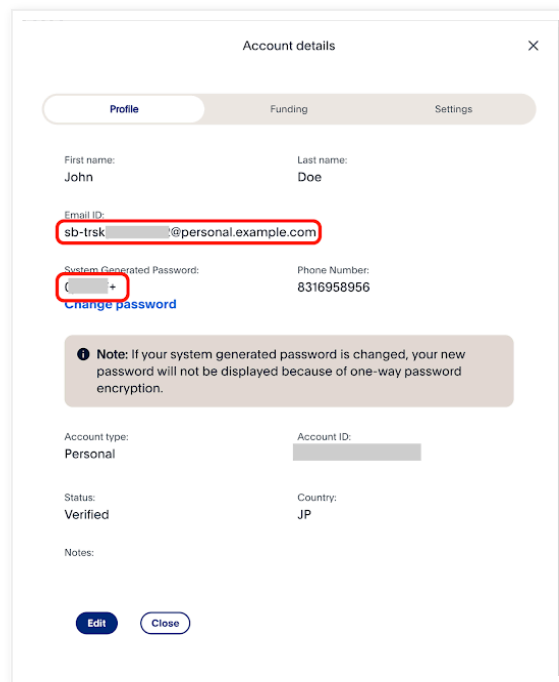
| Webhook | Webhook id | Event Tracked |
|---|------------|---------------|
| You do not have any events for the selected application | | |

[Add Webhook](#)

続いてPersonalアカウントのパスワードを確認します。View/Edit accountを実行します。



ProfileのタブよりEmail IDとSystem Generated Passwordを確認します。PayPal決済を試す際に、支払いを行うユーザー名とパスワードとして使用します。

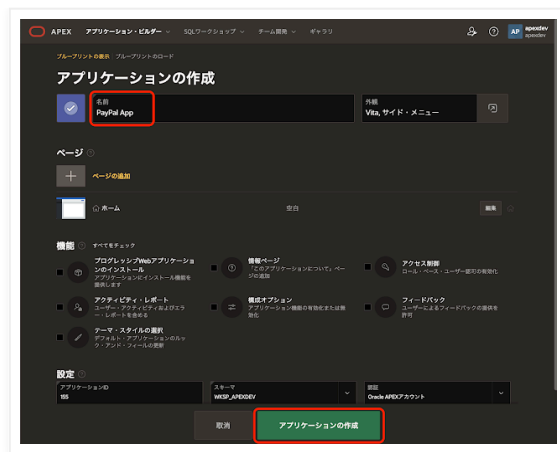


Fundingタブを開くと、テストに使用できるBank Account、Credit Card、PayPalのBalanceも確認できます。PayPalのBalanceはテストで支払いを実行すると、支払った分だけ少なくなります。Balanceには最初に5000円が振り込まれています。

PayPal側の準備は以上です。これからOracle APEXのアプリケーションを作成します。

アプリケーション作成ウィザードを起動し、空のアプリケーションを作成します。

名前はPayPal Appとします。PayPal決済はホーム・ページに実装します。



アプリケーションが作成されたら**アプリケーション定義**を開き、**置換文字列**を設定します。

置換文字列として**G_CLIENT_ID**、**置換値**として**BusinessアカウントのREST AppのClient ID**を設定します。



ページ・デザイナーで**ホーム・ページ**を開きます。

決済金額を入力する**ページ・アイテムP1_AMOUNT**を作成します。

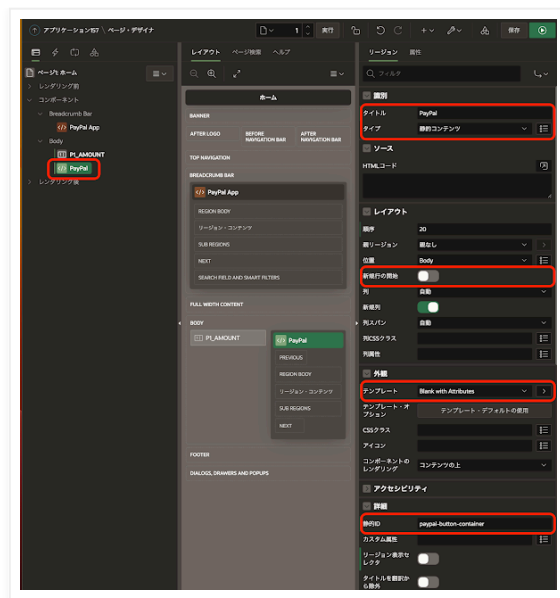
タイプは**数値フィールド**、**ラベル**は**Amount**とします。必ずしも必要ではありませんが、**設定の最小値**に**100**、**最大値**に**1000**を指定します。



PayPalの決済ボタンを表示する領域を**リージョン**として作成します。

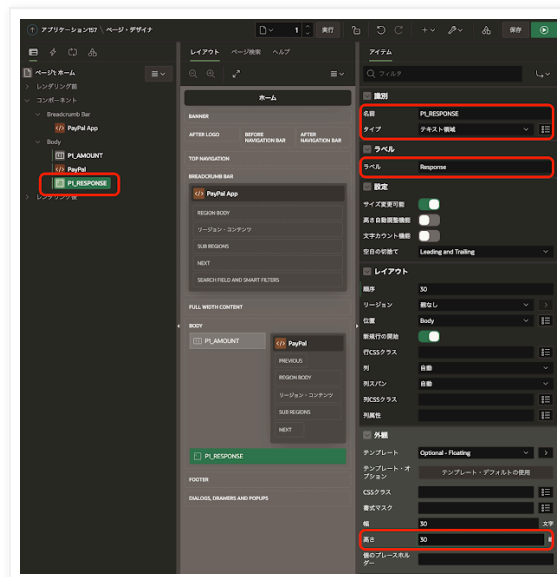
識別の**タイトル**は**PayPal**、**タイプ**として**静的コンテンツ**を選択します。**レイアウトの新規行の開始**を**OFF**として、**ページ・アイテムP1_AMOUNT**の右隣に配置します。不要な装飾を省くため**外観のテンプレート**として**Blank with Attributes**を選択します。

詳細の**静的ID**として**paypal-button-container**を設定します。



PayPalの決済完了の応答を印刷するページ・アイテムP1_RESPONSEを作成します。

タイプはテキスト領域、ラベルはResponseとします。外観の高さに30を設定します。



コンポーネントの配置は以上です。

PayPal決済に必要なコーディングを行います。

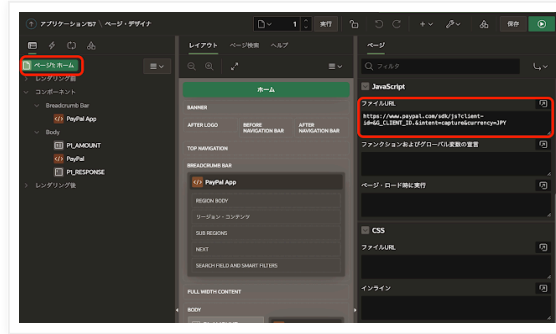
ページ・プロパティのJavaScriptのファイルURLに、JavaScript SDKを読み込みを指定します。読み込み時にスクリプトの構成オプションを指定します。

構成オプションについては、PayPalのドキュメントのJavaScript SDK script configurationに説明があります。

<https://developer.paypal.com/sdk/js/configuration/>

今回はclient_id、intentおよびcurrencyを設定します。

https://www.paypal.com/sdk/js?client-id=&G_CLIENT_ID.&intent=capture¤cy=JPY



ページ・ロード時に実行するJavaScriptのコードとして以下を記述します。

```

/*
 * PayPalの決済ボタンを構成する。
 *
 * style属性でボタンの表示方法を指定する。
 * createOrderでは決済をJSON形式で作成し、PayPalでの決済を依頼する。
 * onApproveはPayPalで決済が完了したときに呼び出され、決済の結果を受け取る。
 */
const paypalButtonsComponent = paypal.Buttons({
  // ボタンの表示形式を決める。
  style: {
    color: "gold",
    shape: "pill",
    layout: "vertical"
  },

  // P1_AMOUNTが決済金額。
  createOrder: (data, actions) => {
    // 決済の作成。
    const createOrderPayload = {
      purchase_units: [
        {
          amount: {
            value: apex.items.P1_AMOUNT.getValue()
          }
        }
      ]
    };
    // PayPalに決済を依頼する。
    return actions.order.create(createOrderPayload);
  },

  // PayPalで決済が完了したので、その結果をページ・アイテムP1_RESPONSEに保存する。
  onApprove: (data, actions) => {
    // PayPalによる決済の結果をP1_RESPONSEに保存する。
    const captureOrderHandler = (details) => {
      const payerName = details.payer.name.given_name;
    }
  }
});

```

```

apex.items.P1_RESPONSE.setValue(JSON.stringify(details, null, 2));
};
// 決済の結果をPayPalから取得する。
return actions.order.capture().then(captureOrderHandler);
},

// 回復不能なエラーに対応する。
onError: (err) => {
    // PayPalのエラー処理を行う。本来はもっといろいろな処理が必要。
    console.error('An error prevented the buyer from checking out with PayPal');
}
});

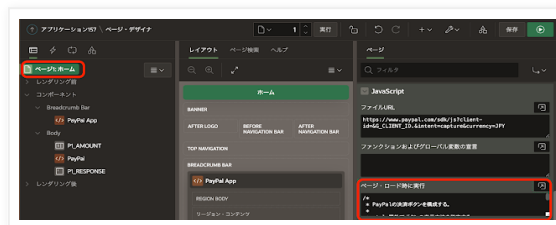
// PayPalのボタンを表示する。
paypalButtonsComponent
    .render("#paypal-button-container")
    .catch((err) => {
        // PayPalのボタンの表示がエラーになったときの処理。
        console.error('PayPal Buttons failed to render');
    });
});

```

paypal-apex.js hosted with ❤ by GitHub

[view raw](#)

PayPalのドキュメントの、[style](#)、[createOrder](#)、[onApprove](#)、[onError](#)を参照して、これらの記述内容をより詳細にできます。



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すると、記事の先頭のGIF動画のように動作します。支払いにはSandbox accountのPersonalアカウント使用します。

決済の履歴などはPayPal Developerのダッシュボードから確認できます。

今回作成したAPEX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/paypal-app.zip>

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: 22:54

共有

[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.
