

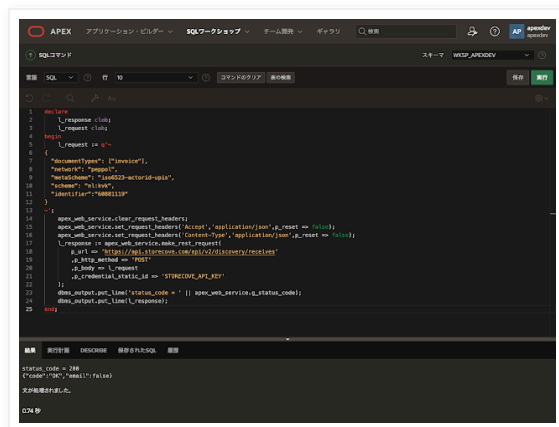
日々是Oracle APEX

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4月27日 木曜日

Storecove社のデジタルインボイスAPIを呼び出してみる

Storecove社が提供しているデジタルインボイスAPIを、Oracle APEXから呼び出してみます。



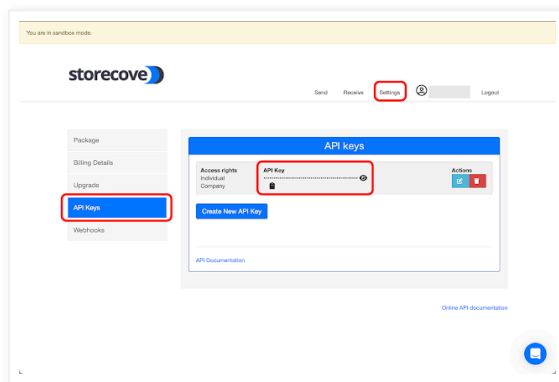
Storecove社のサイトは以下です。

<https://www.storecove.com/>

Storecove社は、[デジタル庁が推進しているPeppol](#)のネットワークにアクセスするサービスおよびRESTful APIを提供しています。

APIの呼び出しを試すには、サンドボックス環境を取得する必要があります。サンドボックス環境は無料で取得できます。詳しくは上記のStorecove社のサイトを参照してください。

開発者アカウントを作成しログインすると、**Settings**タブより**APIキー**の生成と取得ができます。



デジタルインボイスの送受信を行うには、受送信を行うLegal Entityを作成する必要があります。Legal Entityを作成する際にPeppol IDが必要であり、このIDは「法人番号」と「適格請求書発行事業者登録番号」のどちらか、「インボイス制度」における「適格請求書」を送るためには「適格請求書発行事業者登録番号」が必要になります。(Storecove社のFAQ: [Peppol IDとは？](#)を参照)

これらの情報を準備できなかったため、以下の作業のみを実施しました。

Storecove API Documentation

1.1.3 Make your first API call

Storecove APIを呼び出すにあたって、APIキーの保存の仕方とパッケージAPEX_WEB_SERVICEの使い方を以下より紹介します。

デジタルインボイスの作成（StorecoveのAPIではインボイスはJSONオブジェクトとして作成します）や、送受信のAPI呼び出しは含みません。

Web資格証明の作成

StorecoveのAPIは以下の形式のAuthorizationヘッダーにより、認証を行います。（1.1.3 Make your first API callのcurlのサンプルより）

Authorization: Bearer API_KEY_HERE

この形式の認証は、**認証タイプ**がHTTPヘッダーである**Web資格証明**として作成できます。

ワークスペース・ユーティリティの**Web資格証明**を開きます。



作成する**Web資格証明**に、以下の値を設定します。

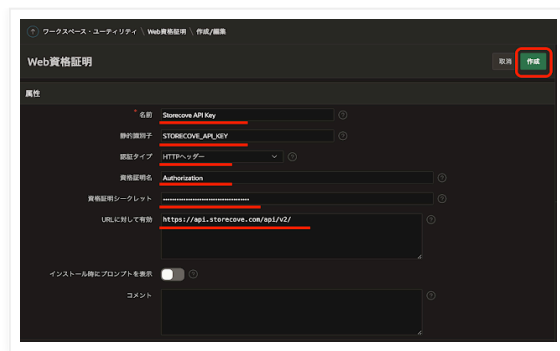
名前は任意です。今回は**Storecove API Key**としました。**静的識別子**はAPI呼び出しの際に指定する文字列になります。今回は**STORECOVE_API_KEY**としています。

認証タイプとして**HTTPヘッダー**を選択します。**資格証明名**はヘッダー名である**Authorization**を指定します。**資格証明シークレット**として、**Bearer**で始め、**空白**で区切り、Storecoveの**Settings**の**API Keys**の画面で生成した**APIキー**の値を続けて設定します。

Bearer API_KEY_HERE

URLに対して有効に、APIのベースURLである以下を設定します。

https://api.storecove.com/api/v2/



Web資格証明が作成できたので、**SQLコマンド**よりAPIの呼び出しを行います。

APIの呼び出し

最初のAPIとして、**Discover Network Participant**が呼び出されています。

https://app.storecove.com/en/docs#_receiving_documents

リクエストの本体であるJSONドキュメントは**DiscoverableParticipant**です。

以下のPL/SQLコードを実行します。

SQLワークショップの**SQLコマンド**より実行します。(記事の先頭のスクリーンショット)

```
declare
    l_response clob;
    l_request clob;
begin
    l_request := q'~
{
  "documentTypes": ["invoice"],
  "network": "peppol",
  "metaScheme": "iso6523-actorid-upis",
  "scheme": "nl:kvk",
  "identifier":"60881119"
}
~';

    apex_web_service.clear_request_headers;
    apex_web_service.set_request_headers('Accept','application/json',p_reset => false);
    apex_web_service.set_request_headers('Content-Type','application/json',p_reset => false);
    l_response := apex_web_service.make_rest_request(
        p_url => 'https://api.storecove.com/api/v2/discovery/receives'
        ,p_http_method => 'POST'
        ,p_body => l_request
        ,p_credential_static_id => 'STORECOVE_API_KEY'
    );
    dbms_output.put_line('status_code = ' || apex_web_service.g_status_code);
```

```
dbms_output.put_line(l_response);  
end;
```

storecove-make-your-first-api-call.sql hosted with ❤ by GitHub

[view raw](#)

応答として[DiscoveredParticipant](#)が返されます。

今回の実行では、以下の応答が返されます。

```
{"code": "OK", "email": false}
```

以上で、Oracle APEXからStorecove APIを呼び出せることが確認できました。

ドキュメントの送信 ([3.2.2 Sending a document](#))、インボイス ([5.2.32 Invoice](#))、送信証明の取得 ([3.3.5 Sending Evidence](#))、ドキュメントの受信 ([3.4.2. Received Document Webhook](#)) といったAPIも何かの機会があれば、試してみたいと思います。

Oracle APEXによる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: 15:20

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.