

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3月24日 水曜日

リモート・デバッグの方法の紹介

カナダのInsumというパートナーさんが、APEX Instant Tipsというタイトルで、ちょっとしたOracle APEXの使い方を紹介しています。

その12回目にオラクルのJoel Kallmanさんが招待され、リモート・デバッグについて紹介していました。

録画はこちらより視聴できます。

Episode 12: How to remote debug a user's session

<https://www.insum.ca/apex-remote-debug-user-session/>

そこで説明されていた内容と、少々デバッグに関する情報を紹介します。

例えば利用者がOracle APEXで作られたアプリケーションにアクセスしていて、何か問題が発生したとします。電話やチャットで開発者にサポートが求められている状況を想定します。

開発者はアプリケーション・ビルダーにサインインすることができます。

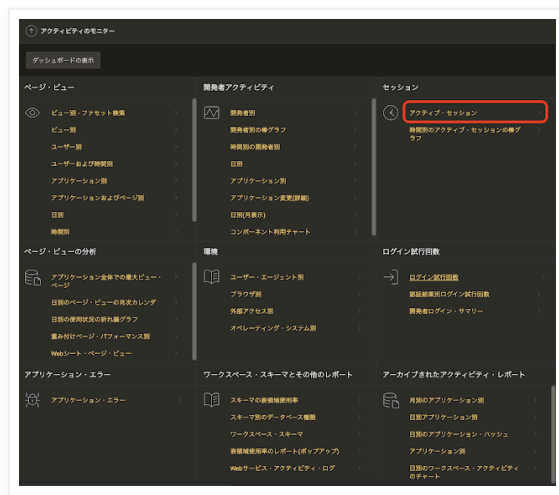
開発者は、連絡を受けて最初に利用者の名前を確認するでしょう。

名前、正確にはユーザー名の確認ができれば、そのユーザーのセッションを見つけることができます。

アプリケーション・ビルダーの右上の**管理**（スパナと人のアイコン）アイコンからメニューを開いて、**アクティビティのモニター**を実行します。



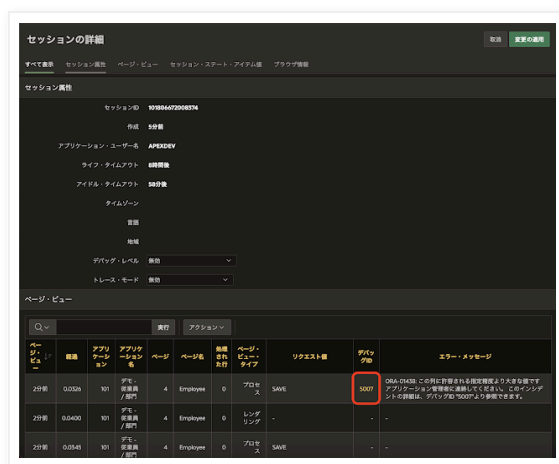
アクティビティのモニターに**アクティブ・セッション**があります。これを開きます。



アクティブ・セッションを開くと、その時点で有効であるセッションの一覧がユーザー名（レポート上は所有者）とともに表示されます。ユーザー名が分かっている場合は、セッションが作成（開始）されたタイミングや最新のビューから、どのセッションなのかは大体見当がつけられるでしょう。そのアクティブ・セッションをクリックして開きます。

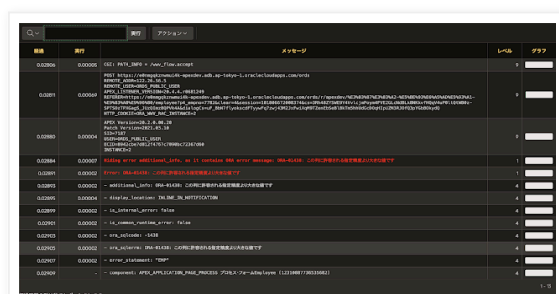


そのセッションの詳細が表示されます。そのユーザーがOracle APEXのアプリケーションに対して、どのような操作を行ってきたか、ページ・ビューとして一覧されます。



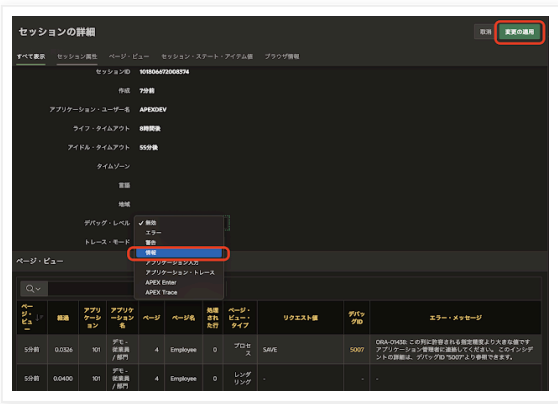
発生しているエラーや、それがどのページの何の処理で発生したのかを確認することができます。利用者にエラーメッセージなどを確認する必要はありません。

エラーが発生している場合は、デバッグIDをクリックして実行ログを確認します。



この時点で十分な情報が出力されていればよいですが、解析するには情報が足りない場合もあるでしょう。その場合はアクティブ・セッションに戻って**デバッグ・レベル**を変更します。

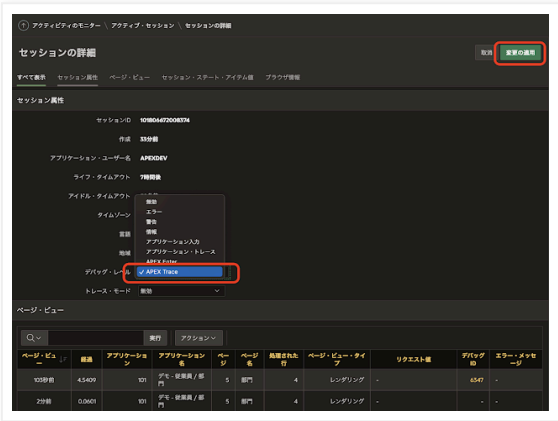
例えば**デバッグ・レベル**を**情報(LEVEL4)**に変更します。**変更の適用**をクリックします。利用者に問題が発生したオペレーションを再実行してもらえば、より詳細な実行ログを取得できます。



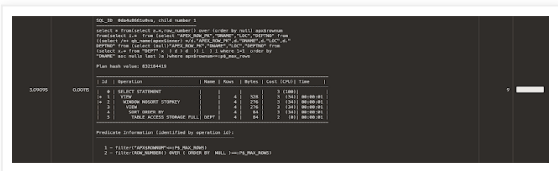
デバッグ・レベルを情報まであげると、ページ・アイテムに設定された値などもログに含まれるようになります。(ちなみにトレース・モードはSQLトレースをファイルに出力するためのオプションで、alter session set sql_trace = trueと同じです。データベース管理者の協力がなければ、ほぼ使うことのない指定です。)



パフォーマンスの問題などで実行計画を確認したい場合は、**デバッグ・レベル**を**APEX Trace(LEVEL9)**まで上げます。



デバッグ・レベルをAPEX Traceまで上げると大量の実行ログが出力されます。その中に実行計画の出力も含まれます。



速度は極端に遅くなるので、必要な情報を取得した後はデバッグ・レベルは下げた方がよいでしょう。

続いて注意点です。アプリケーション定義でデバッグがONになっていると、アプリケーションの利用者によってデバッグを有効にすることが可能になります。

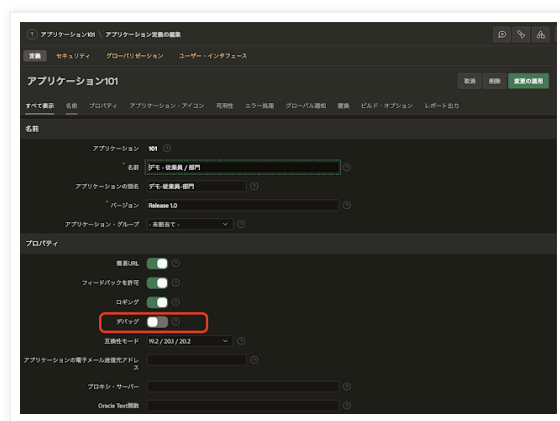
例えばf?p形式のURLの場合は

f?p=アプリケーションID:ページID:セッションID:リクエスト:**デバッグ**:::

簡易URLの場合は

debug=**デバッグ**

です。アプリケーションのURLでデバッグが指定されていると、リモートからの設定より優先されます。本番アプリケーションでは、ブラウザからデバッグを有効にできるようになっていると問題があるので、**デバッグはOFFにします。**

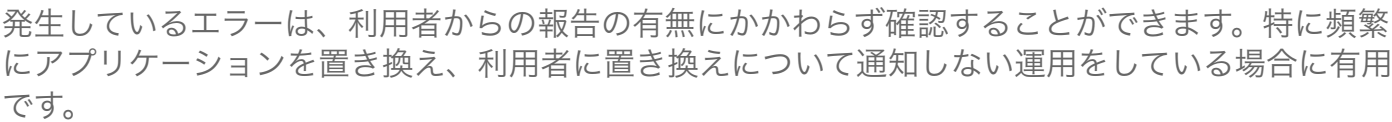


デバッグに関するマニュアルの記載は[こちら](#)です。

最後に開発したアプリケーション固有のデバッグ・ログを出力するために、[APEX_DEBUG](#)パッケージを使用できます。ERROR、WARN、INFO、TRACEといったデバッグ・ログを出力するプロシージャが含まれています。

プロセスなどに比較的長いコードを記述する場合に、APEX_DEBUG.INFOプロシージャを呼び出して、デバッグを容易にするメッセージを出力することをお勧めします。

今回紹介した[アクティビティのモニター](#)には、**アプリケーション・エラー**という項目もあります。



セッションやデバッグ・ログを確認するためにアプリケーション・ビルダーにサインインするのは手間であったり、または、サポート要員が確保できていれば、アプリケーションをデバッグするためのアプリケーションを開発することも可能です。

セッションの情報はビューAPEX_WORKSPACE_SESSIONS、ページのアクセスは、ビューAPEX_WORKSPACE_ACTIVITY_LOG、デバッグ・ログはビューAPEX_DEBUG_MESSAGESから取得できます。

アプリケーションが自分自身のセッションのデバッグを有効/無効にするために APEX_DEBUG.ENABLE/DISABLEが使えます。異なるセッションの場合は [APEX_SESSION.SET_DEBUG](#) を呼び出します。

Oracle APEXでは、フレームワーク側でセッションの情報、アクセス履歴やエラー・メッセージなどを記録してくれますし、保存されている情報はSQLで取り出すこともできます。ですので、デバッグ用のコードの埋め込みを少なくすることができます。

PL/SQLのコーディングなどを学ぶと、パッケージを使う、例外処理をきちんと書く、ということが推奨されています。確かにそうなのですが、Oracle APEXに埋め込むコードについていえば、例外をトラップしていなければ、アプリケーション・エラーとしてメッセージが記録されます。

APEXにはアプリケーションのデバッグを容易にするための機能が備わっているので、それを活用すると良いのではないかと思います。そうすることで、短期間でとりあえず使えるアプリケーションを作ることができます。

[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.