

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11月26日 金曜日

Oracle APEX 21.2新機能(23) - その他

Oracle APEX 21.2の新機能およびその他の変更で、気に留まったものを紹介します。

テンプレート・ディレクティブの条件追加

Oracle APEX 21.2までのテンプレート・ディレクティブのifの条件はページ・アイテムまたは列の値が、**空文字列ではなく、かつ、F、N、0の文字列でもない**ときに真と評価されました。Oracle APEX 21.2より**?**および**=**のプレフィックス・オペレータが追加されています。

```
{if P2_COND/}  
  P2_CONDの条件では、&P2_COND.は空文字列ではなくNOFでもないので印刷される。<br>  
{endif/}  
{if =P2_COND/}  
  =P2_CONDの条件では、&P2_COND.はNOFではないので空文字列でも印刷される。<br>  
{endif/}  
{if ?P2_COND/}  
  ?P2_CONDの条件では、&P2_COND.は空文字列ではないので印刷される。<br>  
{endif/}
```

=オペレータはページ・アイテムまたは列の値が**N、0、F以外であれば、空文字列を含め真**となります。つまり**真偽値**だけを検証します。

?オペレータはページ・アイテムまたは列の値が**空文字列以外であれば真**となります。N、0、Fであっても条件としては真になります。つまり**空文字列かどうか**だけを検証します。

P2_CONDとしてAが渡されると、上記のスク립トは以下を印刷します。

P2_CONDの条件では、Aは空文字列ではなくNOFでもないので印刷される。
=P2_CONDの条件では、AはNOFではないので空文字列でも印刷される。
?P2_CONDの条件では、Aは空文字列ではないので印刷される。

F(または0、N)を渡した場合は以下になります。

?P2_CONDの条件では、Fは空文字列ではないので印刷される

空文字列の場合は以下になります。

=P2_CONDの条件では、はNOFではないので空文字列でも印刷される。

空文字列とN、0、Fといった真偽値の判定を異なる条件として扱えるようになり、条件設定が容易になっています。

POSTリクエストに問い合わせ引数p_contextを追加

ORDSにアクセス・ログを出力する設定を追加し、POSTリクエストのURLを確認してみます。アクセス・ログを出力する設定については、[こちらの記事](#)が参考になります。

フォームの送信でのログは以下のようにになっています。簡易URLはONです。

```
10.0.2.2 - - [26/Nov/2021:04:25:12 +0000] "POST /ords/xepdb1/wwv_flow.accept?p_context=employees/employee/8528313532441 HTTP/1.1" 200 101
```

Ajaxのリクエストの場合は以下です、

```
10.0.2.2 - - [26/Nov/2021:04:25:13 +0000] "POST /ords/xepdb1/wwv_flow.ajax?p_context=employees/employees-report/8528313532441 HTTP/1.1" 200 22038
```

簡易URLがOFFの場合の例です。

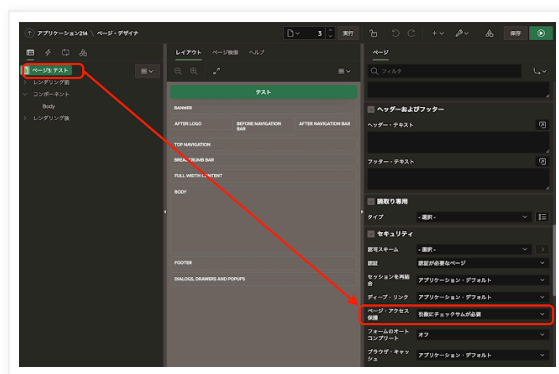
```
10.0.2.2 - - [26/Nov/2021:04:24:07 +0000] "POST /ords/xepdb1/wwv_flow.accept?p_context=4550:1:8705366998123 HTTP/1.1" 302 0
```

GETリクエストの場合はアプリケーション名(もしくはID)、ページ名(または番号)およびセッションIDがURLに含まれていました。POSTやAjaxはそうではなかったのですが、これからはURLから呼び出し元を確認できます。

アクセス・ログの解析が、今までより容易になりました。

新規ページ作成時のページ・アクセス保護のデフォルト値

新規にページを作成したときのページ・プロパティページ・アクセス保護の値が引数にチェックサムが必要になりました。今までは制限なしだったので、デフォルト値がより安全な設定になっています。



apex.util.escapeHTMLAttrの追加

これまでJavaScript APIとして[apex.util.escapeHTML](#)がありましたが、より多くの特殊文字をエスケープの対象とする[apex.util.escapeHTMLAttr](#)が追加されました。英数字とカンマ、ピリオド、ダッシュ、アンダースコア以外はエスケープされます。

例えば、`!#$%&`をそれぞれエスケープすると、結果は以下になります。

```
apex.util.escapeHTML("!#$%&");
```

```
'!#$%&amp;';
```

```
apex.util.escapeHTMLAttr("!#$%&");
```

```
'&#x21;&#x23;&#x24;&#x26;';
```

クロスサイト・スクリプティングに対応するために追加されたファンクションになります。HTMLの属性となる文字列は、`apex.util.escapeHTMLAttr`を呼び出してエスケープすることが推奨されています。

[リリース・ノート](#)などの記載から気に留まった変更点は以上になります。

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: 14:36

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.
