

# 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。

2022年9月28日 水曜日

## BLOBに格納したJSON配列をマルチバリュー・ファセットとして使用する

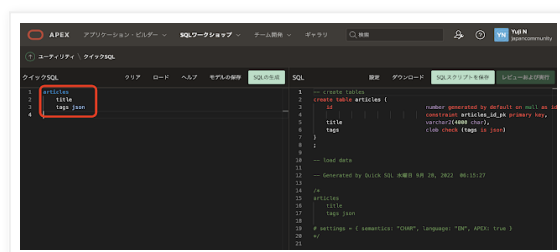
ファセット検索のマルチバリュー・ファセットのタイプとしてJSON配列を選択できます。しかし、JSONをBLOBとして格納すると文字列ではないため、そのままでは対話モード・レポートやフォームで扱うことができません。

BLOBに格納したJSON配列をマルチバリュー・ファセットとして使用するために作成した、いくつかの処理について紹介します。



実装サンプルとなるアプリケーションを作成するために、**クイックSQL**の以下のモデルより、表**ARTICLES**を作成します。

```
articles
  title
  tags json
```



レビューおよび実行までを行うと、**SQLスクリプトの編集画面**が開きます。クイックSQLで列の型としてJSONを指定した際のデフォルトはCLOBなので、これを**BLOBに変更**します。

その後、**実行**をクリックして表**ARTICLES**を作成します。

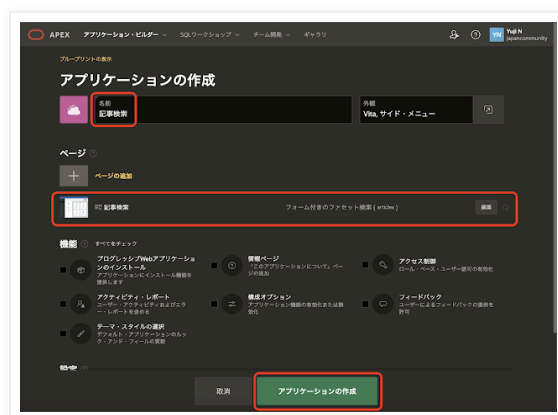


表ARTICLESが作成されたら、**アプリケーションの作成**を実行します。



**アプリケーション作成ウィザード**が起動します。デフォルトで作成されているページを削除し、ファセット検索のページを追加します。

**アプリケーションの名前とページ名の双方を記事検索**としています。



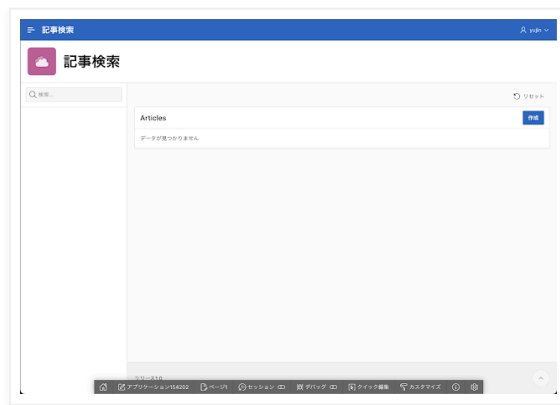
追加するファセット検索のページの**フォームを含める**にチェックを入れます。

表示形式は**レポート**、表として**ARTICLES**を選択します。

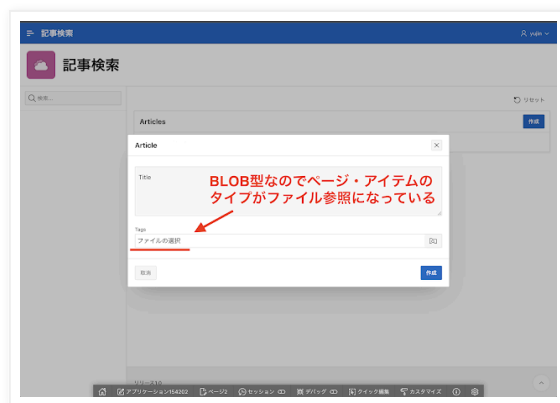


以上で**アプリケーションの作成**を実行します。

作成されたアプリケーションを実行すると、以下の画面が開きます。



作成をクリックし、データを作成します。フォームが開くと**Tags**の指定として**ファイル**を選択するようになっています。列の型がBLOBなので、ページ・アイテムのタイプが**ファイル参照**になっています。



ページ・デザイナーでフォームのページを開き、画面上は**カンマ区切りの文字列**として複数のタグを入力できるように変更します。

フォームを初期する際にBLOBに保存されているタグのJSON配列を、カンマ区切りの文字列に変更するプロセスを作成します。

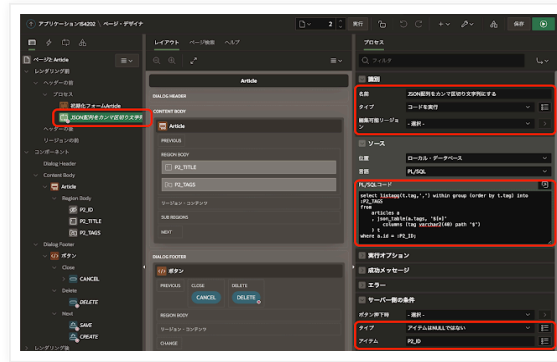
レンダリング前のヘッダーの前、初期化フォームArticleの直下に新規にプロセスを作成します。

識別の名前は**JSON配列をカンマ区切り文字列にする**とします。タイプとして**コードを実行**を選びます。編集可能リージョンは- 選択 -のまま変更しません。これは**未選択**を意味します。タグを保存するページ・アイテムP2\_TAGSとして現在は表ARTICLESの列TAGSが割り当てられていますが、この後に割り当てを変更しP2\_TAGSにプロセスの処理結果を保存します。そのため、割り当てるべき編集可能リージョンはありません。

ソースの**PL/SQLコード**として以下を記述します。変数宣言が不要な場合はbegin/endで囲む必要はありません。

```
select listagg(t.tag,',') within group (order by t.tag) into :P2_TAGS
from
  articles a
  , json_table(a.tags, '$[*]'
    columns (tag varchar2(40) path '$')
  ) t
where a.id = :P2_ID;
```

サーバー側の条件として、タイプに**アイテムはNULLではない**を選択し、アイテムに**P2\_ID**を指定します。



以上で、データの読み出し時にJSON配列からカンマ区切りの文字列へ変更するプロセスが作成されました。

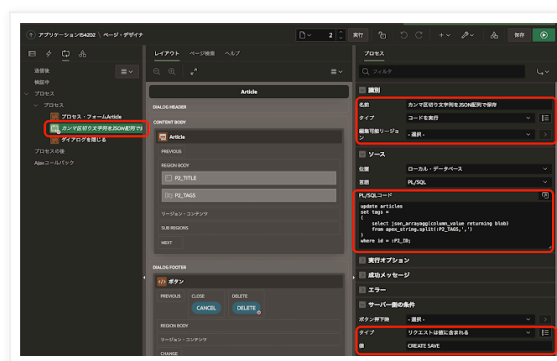
次にカンマ区切りの文字列をJSON配列に変換して、表ARTICLESに保存するプロセスを作成します。

左ペインで**プロセス・ビュー**を開き、**プロセス・フォームArticle**の直下に新規にプロセスを作成します。

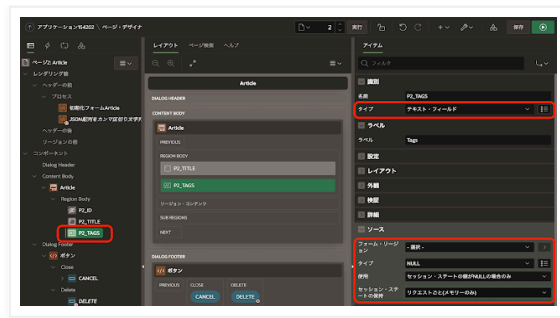
識別の名前は**カンマ区切り文字列をJSON配列で保存**とします。ソースのPL/SQLコードとして以下を記述します。

```
update articles
set tags =
(
  select json_arrayagg(column_value returning blob)
  from apex_string.split(:P2_TAGS,',')
)
where id = :P2_ID;
```

**サーバー側の条件**として、**タイプ**に**リクエストは値に含まれる**、**値**として**CREATE SAVE**を指定します。**作成**または**変更の適用**のボタンを押した時のみ、処理が実行されます。



ページ・アイテムP2\_TAGSの識別の**タイプ**を**テキスト・フィールド**に変更し、ソースの**フォーム・リージョン**を**- 選択 -**に変更（つまり未選択）します。**タイプ**は**NULL**、**セッション・ステートの保持**は**リクエストごと(メモリーのみ)**を指定します。



以上でタグの入出力については、対応ができました。

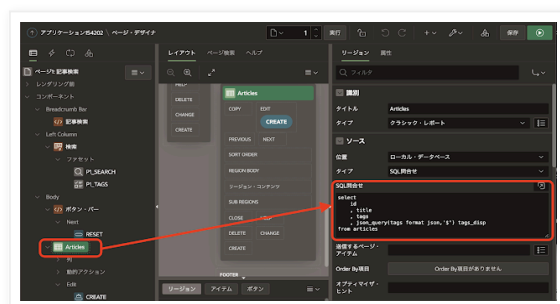
アプリケーションを実行し、データを入力してみます。

データを保存すると、レポートに**ダウンロード**のリンクが表示されます。列の型がBLOBの場合のデフォルトになります。

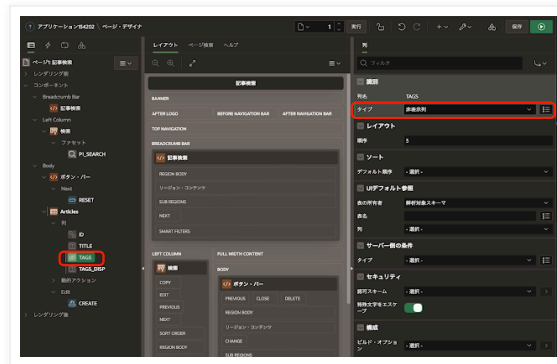
ダウンロードのリンクの代わりに文字列を表示させ、また、マルチバリュー・ファセットも作成します。

リージョン**Articles**の**ソース**の**SQL問合せ**を以下に変更します。手抜きですが、JSON配列をそのまま文字列として表示します。ファセットのソースとしてJSON配列を使うため、列TAGSはそのまま残します。文字列として表示する列はTAGS\_DISPとして新たに追加します。

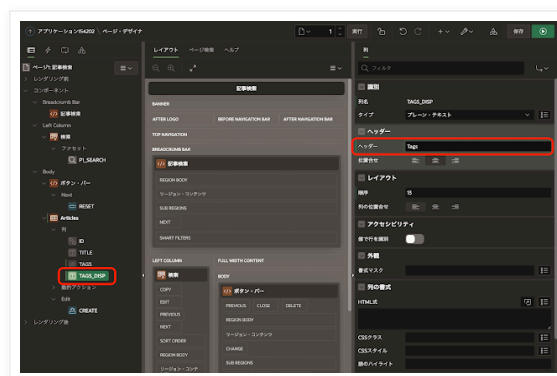
```
select
  id
  , title
  , tags
  , json_query(tags format json,'$') tags_disp
from articles
```



列TAGSは**非表示列**に変更します。



列TAGS\_DISPのヘッダーをTags DispからTagsへ変更します。



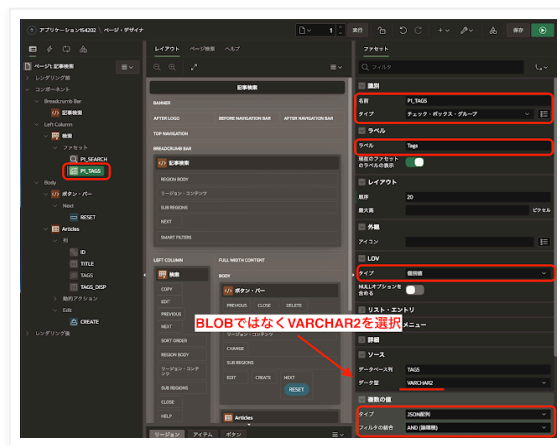
この時点でレポートを表示させると、ダウンロード・リンクの代わりにJSON配列がそのままの形式で表示されていることが確認できます。



最後にファセットを作成します。

識別の名前をP1\_TAGS、タイプとしてチェック・ボックス・グループを選択します。ラベルはTagsとします。LOVのタイプとして個別値、ソースのデータベース列はTAGS、データ型としてVARCHAR2を選択します。実際の型はBLOBなのですが、なぜかVARCHAR2にしないと正常に動作しません。

複数の値のタイプとしてJSON配列を選択し、フィルタの結合にはAND(論理積)を選択しました。



編集フォームが閉じたときにファセットも更新されるよう、動的アクションを追加します。

動的アクション・ビューを開き、動的アクションのレポートの編集 - ダイアログのクローズにTRUEアクションを作成します。

識別のアクションとしてリフレッシュを選択します。影響を受ける要素の選択タイプとしてリージョン、リージョンに検索（ファセットのリージョン）を選択します。



以上でアプリケーションは完成です。

実行すると記事の最初にあるGIF動画のように動作します。

今回作成したアプリケーションを以下に置きました。

<https://github.com/ujnak/apexapps/blob/master/exports/multivalue-facet-json-blob.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: 18:05

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](#)