

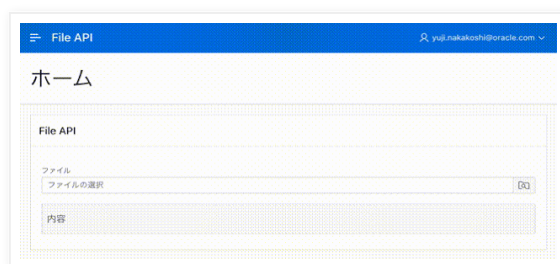
日々是Oracle APEX

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。

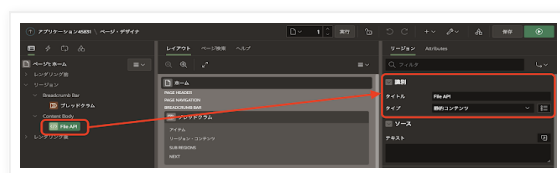
2021年2月6日土曜日

ファイルの内容をページ・アイテムに設定する

タイプがファイル参照...のページ・アイテムで選択したファイルの内容を、File APIを使って読み出し異なるページ・アイテムに設定します。



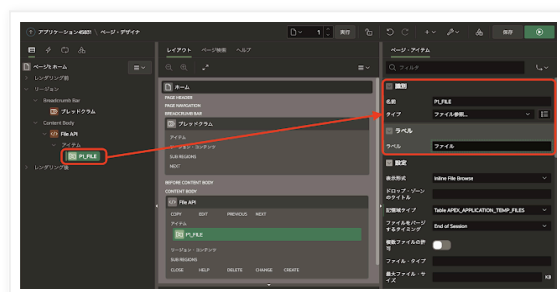
実装例を紹介するために使用するページ・アイテムを配置するリージョンを新規に作成します。名前は**File API**、タイプは**静的コンテンツ**とします。



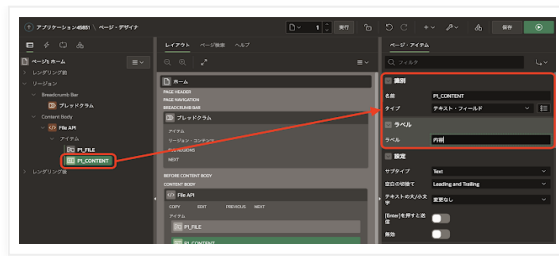
ページ・アイテムを配置するだけなので、リージョンであればタイプは実際のところ何でもかまいません。

作成したリージョンにページ・アイテムを作成します。

最初にファイルを選択するページ・アイテムを作成します。名前を**P1_FILE**、タイプを**ファイル参照...**にします。ラベルも、**ファイル**、としておきます。

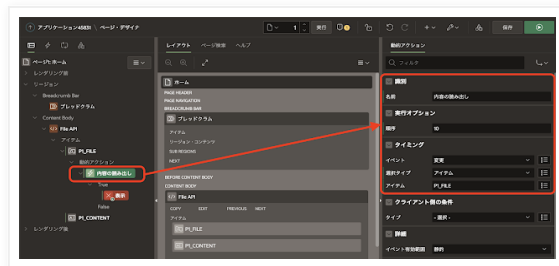


続けて、ファイルの内容を保持するページ・アイテムを作成します。名前を**P1_CONTENT**、タイプを**テキスト・フィールド**にします。ラベルも、**内容**、としておきます。



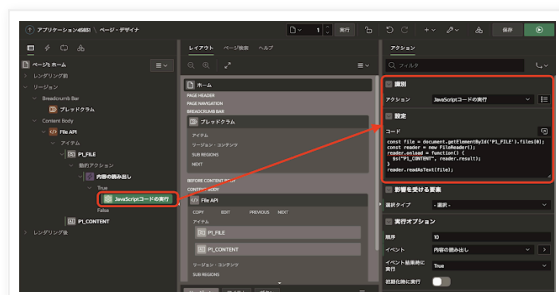
P1_FILEにファイルが選択されたときに、その内容を読み出す処理は動的アクションにて実装します。

P1_FILEに動的アクションを作成します。名前を内容の読み出し、とします。タイミングはデフォルトでアイテムのP1_FILEが変更されたときに発火するようになっています。



TrueアクションとしてJavaScriptコードの実行を選択します。設定のコードには以下を記述します。

```
const file = document.getElementById('P1_FILE').files[0];
const reader = new FileReader();
reader.onload = function() {
  $s("P1_CONTENT", reader.result);
}
reader.readAsText(file);
```



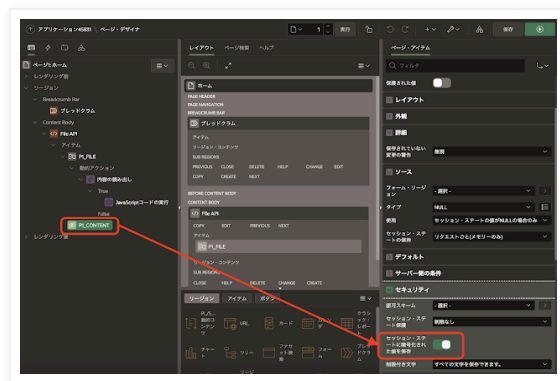
ファイルの内容が秘匿性が高い場合は、ページ・アイテムのプロパティを調整します。利用者に情報そのものを見せる必要はないので、タイプは非表示にします。タイプを非表示にしても、動的アクションによって値が設定できるよう、設定の保護された値をOFFにします。詳細の保存されていない変更の警告は無視します。変更を保存することはないためです。

ソースのセッション・ステートの保持はリクエストごと(メモリーのみ)を選択します。

ページの送信時（フォームのページで送信を行ったとき）にページ・アイテムの値はブラウザからサーバーに送信されます。セッションごと(ディスク)、ユーザーごと(ディスク)を選択している場合は、受信したページ・アイテムの値をディスクに保存します。結果としてサーバー側で内容を参照できるようになります。リクエストごと(メモリーのみ)の場合は、ディスクには保存しません。通常は異なるページに配置されているページ・アイテムの値を参照するコードは書かない（そのよ

うな場合はアプリケーション・アイテムを使用する) ので、リクエストごと(メモリーのみ)でディスクに保存する必要はありません。

Oracle APEXではページ・アイテムをセッション・ステートに保存する際に暗号化するオプションも提供しています。**セキュリティのセッション・ステートに暗号化された値を保存**です。



セッション・ステートをディスクに保存する場合は、こちらの設定をONにするのも有効な対応でしょう。

ファイルの内容をページ・アイテムに設定する方法の紹介は以上になります。

Oracle APEXによるアプリケーション作成の一助になれば幸いです。

追記

当初、想定していた要件についてはlocalStorageを使うのが正解でした。

完

Yuji N. 時刻: 12:57

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.