

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24日 水曜日

Oracle APEX 21.2新機能(21) - 開発ツールの機能追加

Oracle APEX 21.2では、アプリケーション・ビルダーやSQLワークショップに含まれるツールに、以下の3つの機能追加があります。

1. 静的アプリケーション・ファイル、静的ワークスペース・ファイルのエディタ機能が追加されました。JavaScriptとCSSのMinifyおよびLessのコンパイルをファイルの保存時に自動で行います。
2. ブレッドクラムを編集する画面が追加されました。
3. SQLコマンドにて、32Kバイトの制限が無くなりました。

追加された機能を確認していきます。

静的ファイルのエディタ

Oracle APEX 21.2より、共有コンポーネントの静的アプリケーション・ファイルまたは静的ワークスペース・ファイルを開いた画面に、ファイルの作成が追加されています。



以前はエディタの機能がなかったため、**ファイルのアップロード**がありました。アップロードするファイルは手元のPCで作成し、それをアップロードする必要があります。ファイルを編集するには、一旦ファイルをダウンロードします。

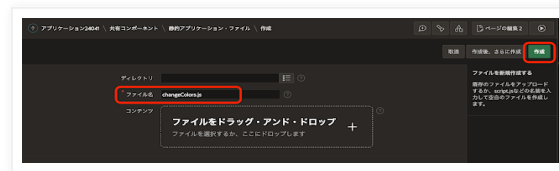


ファイルの作成をクリックすると、ディレクトリの指定(任意)、**ファイル名**の指定があります。すでに作成済みのファイルをアップロードすることも可能です。

以前に対話グリッドの表示に色を付ける実装を行ったことがあります。(記事：[対話グリッドの列の文字の色を条件によって変更する - JavaScriptによる実装](#)) この実装では、**ページ・プロパティ**のJavaScriptの**ページ・ロード時に実行**と**動的アクション**に、直接JavaScriptを記述しています。

このJavaScriptの記述をひとつの静的アプリケーション・ファイルにまとめてみます。

一覧画面で**ファイルの作成**をクリックします。**ファイル名**をchangeColors.jsとして**作成**をクリックします。



ファイルに記載する内容は以下です。**参照**のパスを**コピー**しておきます。色を設定する部分のコードは共通にしています。

```
// 部署ごとに色を設定する。
function setColor(elems, deptno) {
  if (elems.length == 1) {
    let elem = elems[0];
    switch(deptno) {
      case '10': // 会計は赤
        if (elem.classList.contains("u-success-text")) {
          elem.classList.remove("u-success-text");
        }
        if (elem.classList.contains("u-warning-text")) {
          elem.classList.remove("u-warning-text");
        }
        if (! elem.classList.contains("u-danger-text")) {
          elem.classList.add("u-danger-text");
        }
        break;
      case '20': // 研究開発は黄色
        if (elem.classList.contains("u-success-text")) {
          elem.classList.remove("u-success-text");
        }
        if (elem.classList.contains("u-danger-text")) {
          elem.classList.remove("u-danger-text");
        }
        if (! elem.classList.contains("u-warning-text")) {
          elem.classList.add("u-warning-text");
        }
        break;
      case '30': // セールスは緑
        if (elem.classList.contains("u-warning-text")) {
          elem.classList.remove("u-warning-text");
        }
        if (elem.classList.contains("u-danger-text")) {
          elem.classList.remove("u-danger-text");
        }
        if (! elem.classList.contains("u-success-text")) {
          elem.classList.add("u-success-text");
        }
        break;
    }
  }
};
```

```
// ページ・ロード時の処理
function changeColor(region) {
  let view = apex.region(region).widget().interactiveGrid("getViews", "grid"),
      menu$ = view.rowActionMenu$;
  // 行メニューの設定をかえる。
  menu$.menu("option").items[0] =
  {
```

```

type:"action",
label:"Alt Single Row View",
action: function(menu, element) {
    // 最初に単一行ビューに切り替える。
    apex.region(region).widget().interactiveGrid( "getActions" ).invoke( "single-row-view" );
    // 単一ビューに切り替えた後は、DOMの操作を行なう。
    let elem = document.getElementById(region);
    let enameElems = elem.getElementsByClassName("u-bold");
    let deptnoElems = elem.getElementsByClassName("this-deptno");
    let deptnoValElems = deptnoElems[0].getElementsByClassName("a-RV-fieldValue");
    let deptno = deptnoValElems[0].innerText;
    setColor(enameElems, deptno);
}
}
};

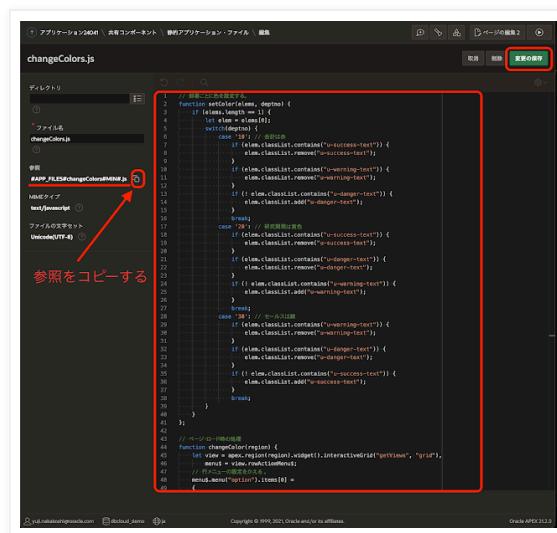
```

// 動的アクションの処理

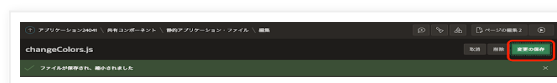
```

function changeColorDA(region, o) {
    let view =
        apex.region(region).widget().interactiveGrid("getCurrentView");
    if (view.singleRowMode) {
        let model = o.data.model;
        let record = o.data.selectedRecords[0];
        let deptno = model.getValue(record, "DEPTNO");
        let elem = o.triggeringElement;
        let enameElems = elem.getElementsByClassName("u-bold");
        setColor(enameElems, deptno);
    }
}
};

```



変更の保存をクリックするとファイルが保存され、縮小されましたと表示されます。Minifyされたファイルが生成されています。



取消をクリックし、ファイルの一覧画面に戻ります。Minifyされたファイルも一覧に表示されていることが確認できます。

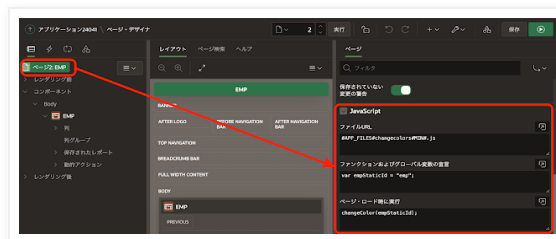


静的アプリケーション・ファイルに記載したJavaScriptファンクションを呼び出すように、ページ・プロパティの設定を変更します。

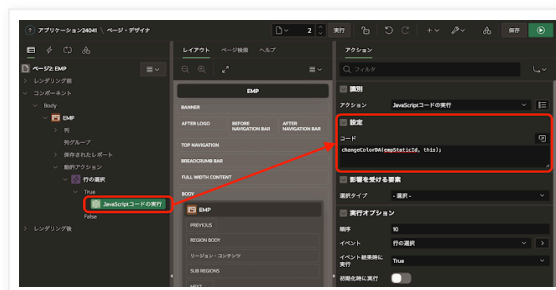
ファイルURLは以下です。
#APP_FILES#changeColors#MIN#.js

ファンクションおよびグローバル変数の宣言は以下です。
var empStaticId = "emp";

ページ・ロード時に実行は以下です。
changeColor(empStaticId);



動的アクションのコードも以下に変更します。
changeColorDA(empStaticId, this);



以上でアプリケーションの変更は完了です。動作自体はまったく変わりませんが、コードのメンテナンスに関しては遥かに容易になっています。

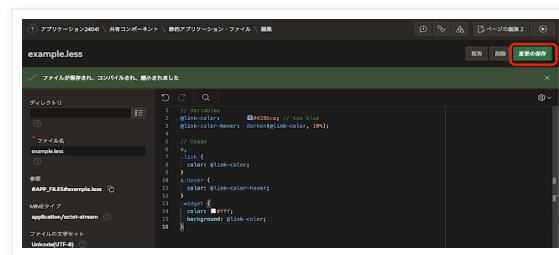
続いてLessの記載について確認してみます。[Lesscss.org](http://lesscss.org)にある例を静的アプリケーション・ファイルとして作成します。ファイル名はexample.lessとします。

```
// Variables
@link-color:    #428bca; // sea blue
@link-color-hover: darken(@link-color, 10%);

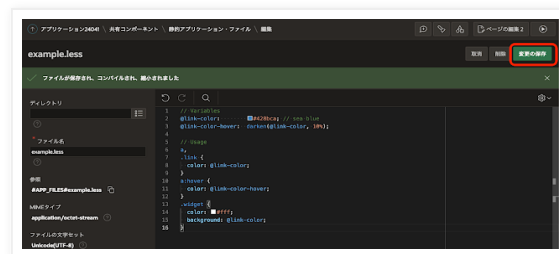
// Usage
a,
.link {
  color: @link-color;
}
a:hover {
  color: @link-color-hover;
}
```

```
.widget {
  color: #fff;
  background: @link-color;
}
```

変更の保存をクリックすると、**ファイルが保存され、コンパイルされ、縮小されました**と表示されます。



一覧画面に戻ると、example.lessに加えてexample.css、example.min.cssが作成されていることが確認できます。



CSSについてもページ・プロパティのCSSにインラインで記載するより、静的アプリケーション・ファイル（または静的ワークスペース・ファイル）にまとめて記載の方が生産性が上がるでしょう。

Tipsとして、MacであればCMD+S、WindowsであればCtrl+Sでエディタ上の変更が保存されます。変更の保存をクリックし、ページの送信を行う必要がありません。

以前に紹介したことのある[APEX Builder Extension by FOS](#)は、APEX 21.2とは互換性がないようで同時に使用できません。そもそも、この機能はStephan DobreさんがオラクルのAPEX開発チームにジョインして作ったもので、同じ人が開発しています。そのため、ほとんど機能的には同じなのですが、プラグインのBoilerplate Codeは含まれていないようです。

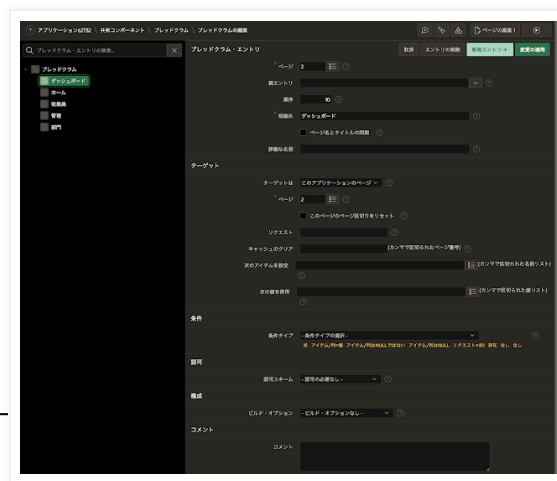
開発者のStephan Dobreさんが、Office Hourで静的ファイル・エディタの解説をしています。

Oracle APEX 21.2 New Features



ブレッどクラムの編集画面

ブレッどクラムの編集画面が刷新されました。以下の画面になっています。

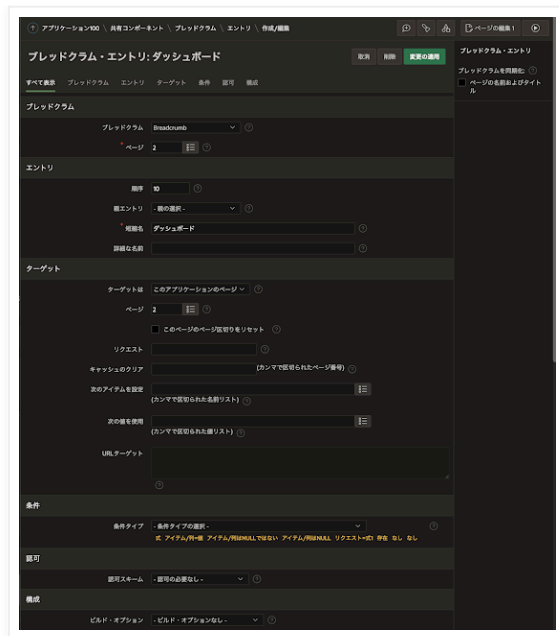


以前のバージョンでは1画面ではなく、ブレッどクラムの一覧画面と詳細画面に分かれていました。

一覧画面は以下でした。

名前	ページ	ページ ID	値
ダッシュボード	10	2	(null)
ホーム	10	1	(null)
従業員	10	3	(null)
管理	10	10000	(null)
部門	10	5	(null)

編集画面は以下でした。



設定できる内容に違いはありません。**ターゲットにURLを選択したときに限りURLターゲットの設定ができるようになったり、コメントが入れられるようになっています。**

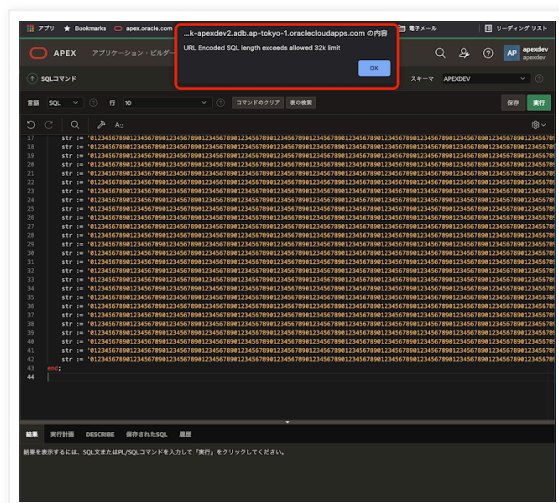
ブレッダクラムの設定が直感的にできるようになりました。

32Kバイトの制限の撤廃

Oracle APEX 21.2以前では、SQLコマンドで32Kバイト以上のコードを実行しようとすると、ダイアログが表示され実行できませんでした。

以下がメッセージです。

URL Encoded SQL length exceeds allowed 32k limit



21.2よりこの制限は無くなっています。

Oracle APEX 21.2で開発ツールに追加された機能の紹介は以上です。

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: 17:24

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.