

日々は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。

2023年3月3日 金曜日

日付範囲ピッカーを実装する

OracleのSteve Muenchさんが彼のブログに、APEX 22.2から使用できる新しい日付ピッカーをカスタマイズして、範囲を指定する方法を紹介されています。

Designer Collab for Date Ranges

<https://diveintoapex.com/2023/02/07/designer-collab-for-date-ranges/>

記事からダウンロードできるサンプル・アプリケーションに含まれる静的アプリケーション・ファイル、**dateRangePicker.js**、**dateRangePicker.css**を取り出し、自分のアプリケーションに取り込むことで簡単に再利用できるようになっています。

色々な技が使われていて再利用が容易になっているのですが、実装を理解するにはスキルが必要です。もう少し実装を分かりやすくするため、動的アクションによる実装に変更してみました。

dateRangePicker.cssはそのまま流用し、日付ピッカーのdayFormatterについてもほぼ同じ実装を流用します。

実装した日付範囲ピッカーは以下のように動作します。日付範囲ピッカーは**表示形式**を**Inline**にした**日付ピッカー**で、**開始日**と**終了日**はそれぞれ異なるページ・アイテムに設定しています。



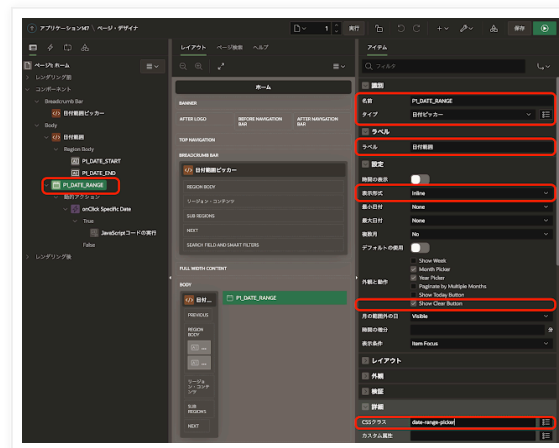
以下より実装について説明します。

ページ・アイテムとして**開始日**を保存する**P1_DATE_START**、**終了日**を保存する**P1_DATE_END**を作成します。上のGIF動画では、これらのページ・アイテムの値が画面に表示されていますが、一般的には非表示にしたいと思います。

日付範囲ピッカーですが、ページ・アイテム**P1_DATE_RANGE**として作成しています。

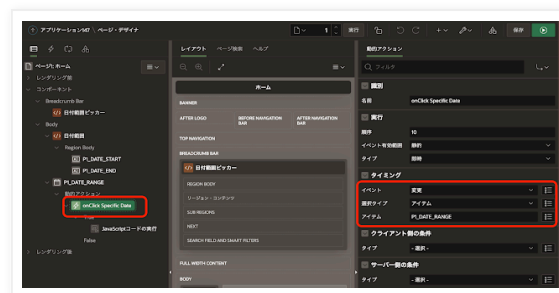
タイプは**日付ピッカー**です。**設定**の**表示形式**として**Inline**を選択します。**外観と動作**の**Show Clear Button**に**チェック**を入れます。表示形式が**Inline**の場合、ページ・アイテムの値をクリアするためにこのボタンが必要です。

詳細のCSSクラスに**date-range-picker**を設定します。これはdateRangePicker.cssで定義されているCSSクラスを適用するために必要です。



ページ・アイテムP1_DATE_RANGEに動的アクションを作成します。

タイミングのイベントは変更、選択タイプはアイテム、アイテムとしてP1_DATE_RANGEを指定します。日付範囲ピッカー上で日付を選択（変更）したときに、動的アクションが実行されます。



TRUEアクションはJavaScriptコードの実行、設定のコードとして以下を記述します。クリックしたときの状態に応じて、選択された日付を開始日または終了日として設定しています。最後に日付範囲ピッカーをリフレッシュし、設定された開始日と終了日の間に色を付けます。

```
/*
 * 影響を受ける要素として、開始日と終了日のページ・アイテム
 * 開始日のページ・アイテムを先に設定する。
 */

let dateItemStart = this.affectedElements[0];
let dateItemEnd = this.affectedElements[1];
let dateItemClicked = this.triggeringElement;
```

```
/*
 * 値が設定されていないときは、クリアがクリックされている。
 * 開始日と終了日をクリアして、リフレッシュする。

```

```
*/  
if (!dateItemClicked.value) {  
    dateItemStart.value = "";  
    dateItemEnd.value = "";  
    dateItemClicked.refresh();  
}
```

```

    return;
}

if (!dateItemStart.value) {
    // 開始日が未設定なので、選択された値は開始日として扱う。
    dateItemStart.value = dateItemClicked.value;
    if (!dateItemEnd.value) {
        /*
         * 終了日も未設定なので、同じ日付を終了日に設定する。
         * 開始日と終了日が同じ日であることを禁止する場合は、この処理を変更する。
         */
        dateItemEnd.value = dateItemClicked.value;
    }
    // 開始日が未設定で終了日が設定されていることは無いはず。なので無視する。
}
else
{
    // 開始日はすでに設定されている。
    const dateClicked = apex.date.parse(dateItemClicked.value, DATE_FORMAT);
    const dateStart = apex.date.parse(dateItemStart.value, DATE_FORMAT);
    if (!dateItemEnd.value) {
        // 終了日が設定されていない。
        if (apex.date.isBefore(dateClicked, dateStart)) {
            // 選択した日付が開始日より前であれば、開始日と終了日を入れ替えて期間を設定する。
            dateItemEnd.value = dateItemStart.value;
            dateItemStart.value = dateItemClicked.value;
        }
        else
        {
            /*
             * 終了日を設定する。
             */
            dateItemEnd.value = dateItemClicked.value;
        }
    }
    else
    {
        // 開始日と終了日の両方が設定済み。
        const dateEnd = apex.date.parse(dateItemEnd.value, DATE_FORMAT);
        if (apex.date.isBefore(dateClicked, dateStart)) {
            // 開始日より前の日付を選択したときは開始日を変更する。
            dateItemStart.value = dateItemClicked.value;
        }
        else if (apex.date.isAfter(dateClicked, dateEnd)) {
            // 終了日より後の日付を選択したときは終了日を変更する。
            dateItemEnd.value = dateItemClicked.value;
        }
        else if (apex.date.isBetween(dateClicked, dateStart, dateEnd)) {
            // 開始日と終了日の間の日付を選択したときは、終了日を変更する。

```

```

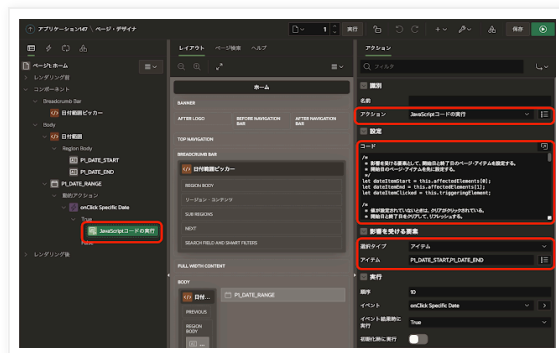
dateItemEnd.value = dateItemClicked.value;
// 開始日を後ろに移動するには、一旦クリアする。
}
}
/*
 * このページ・アイテムにはつねに開始日を設定したのちにリフレッシュを行う。
 * 日付の範囲を正しく表示するために必要。
 */
dateItemClicked.value = dateItemStart.value;
dateItemClicked.refresh();
}

```

dateRangePickerDA.js hosted with ❤ by GitHub

[view raw](#)

影響を受ける要素の選択タイプにアイテムを選び、アイテムとしてP1_DATE_START、P1_DATE_ENDの2つのページ・アイテムを設定します。開始日を保持するページ・アイテムを先頭に配置します。



開始日から終了日の間に色をつける処理は、日付範囲ピッカーに実装するdayFormatterが行います。

ページ・プロパティのJavaScriptのファンクションおよびグローバル変数の宣言に以下を記述します。

```
const DATE_FORMAT = "yyyy/mm/dd";
```

ページ・ロード時に実行でdayFormatterを定義します。開始日にCSSクラスdateRangeStart、終了日にdateRangeEnd、開始日と終了日の間の日付にはdateRangeMiddleを設定しています。開始日と終了日が同じ場合はCSSクラスとしてdateRangeSingleDayを設定しています。

```

apex.item("P1_DATE_RANGE").dayFormatter = (iso8860DateString) => {
  // Parse the day being formatted using ISO8860
  const currentDate = apex.date.parse(iso8860DateString, "YYYY-MM-DD");
  const startDateValue = apex.items.P1_DATE_START.value;
  // Parse start date using start page item format
  const startDate = startDateValue ?
    apex.date.parse(startDateValue, DATE_FORMAT) : null;
  const endDateValue = apex.items.P1_DATE_END.value;
  // Parse end date using end page item format

```

```

const endDate = endDateValue ?
    apex.date.parse(endDateValue,DATE_FORMAT) : null;

let dateRangeClass = "";
// 開始日の設定は必須。
if (startDateValue) {
    // 開始日が処理対象である。
    if (apex.date.isSame(currentDate,startDate)) {
        // 開始日と終了日が同じであれば、スタイルdateRangeSingleDayを適用する。
        if (endDateValue && apex.date.isSame(currentDate,endDate)) {
            dateRangeClass = "dateRangeSingleDay";
        }
        else {
            dateRangeClass = "dateRangeStart";
        }
    }
    // 処理対象が開始日以外。
    else if (endDateValue) {
        // 処理対象が終了日であれば、スタイルdateRangeEndを適用する。
        if (apex.date.isSame(currentDate,endDate)) {
            dateRangeClass = "dateRangeEnd"
        } else if (apex.date.isBetween(currentDate,startDate,endDate)) {
            // 処理対象が開始日と終了日の間であればスタイルdateRangeMiddleを適用する。
            dateRangeClass = "dateRangeMiddle";
        }
    }
    // それ以外はスタイルを適用しない。
};
return {
    disabled: false,
    class: dateRangeClass
};
};

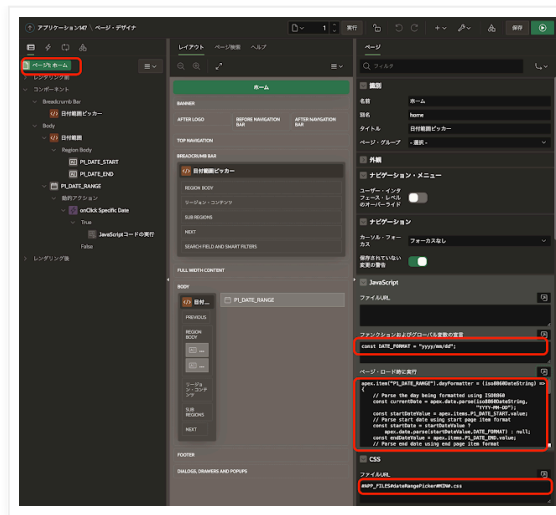
```

dateRangePickerFormatter.js hosted with ❤ by GitHub

[view raw](#)

CSSのファイルURLに以下を記述します。CSSファイルは[こちら](#)から、または英語の元記事よりサンプル・アプリケーションをダウンロードし、dateRangePicker.cssを取り出して静的アプリケーション・ファイルとして保存しておきます。

#APP_FILES#dateRangePicker#MIN#.css



以上で実装は完了です。

今回作成したAPEXアプリケーションのエクスポートを以下に置きました。
<https://github.com/ujnak/apexapps/blob/master/exports/date-range-picker.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: 14:46

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.