

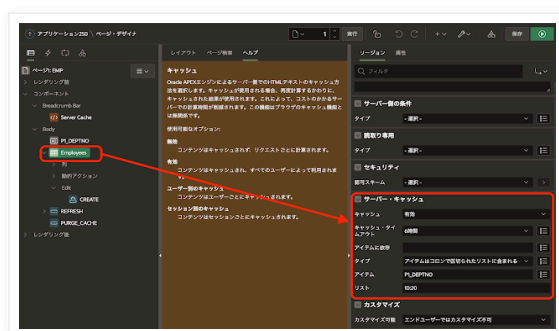
日々是Oracle APEX

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。

2023年8月18日 金曜日

リージョン属性のサーバー・キャッシュについて

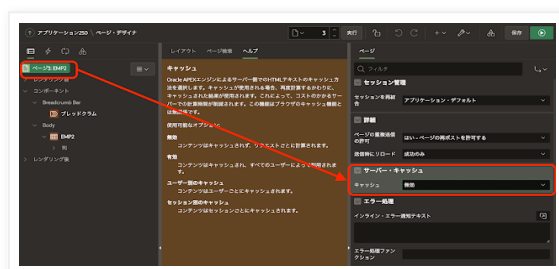
リージョン属性に**サーバー・キャッシュ**というのがあるけど、これを**有効**にしたら表示が速くなるの？と聞かれたので、調べてみました。この機能を使った経験はありません。



マニュアルの記載は以下になります。

アプリケーション・ビルダー・ユーザーズ・ガイド Release 22.2
12.4.3.4 リージョン・キャッシュの有効化

ページ属性にも同様に**サーバー・キャッシュ**があります。



ページ属性のサーバー・キャッシュについては、マニュアルに説明を見つけないことができませんでした。本来であれば、**7.4 ページ属性の管理**で説明されているはずの属性です。

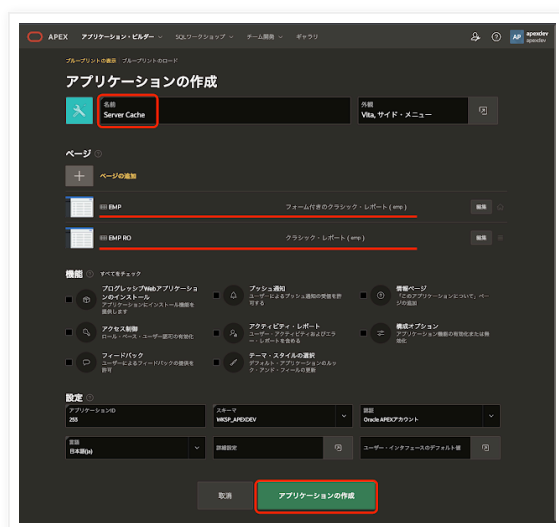
最近のWebページは、すべてのHTMLをサーバー側で生成するというのではなく、クライアント側のJavaScriptによる画面の書き換えも一般的です。このサーバー・キャッシュの機能は、JavaScriptの使用がまだ一般的ではなかった頃（そしてサーバー側のリソースが今より弱かった頃）は有効だった、古の機能と言えます。

これより、リージョン属性の**サーバー・キャッシュ**を中心に、その動作を確認してみます。

動作確認に使用するアプリケーションは、以下の手順で作成します。

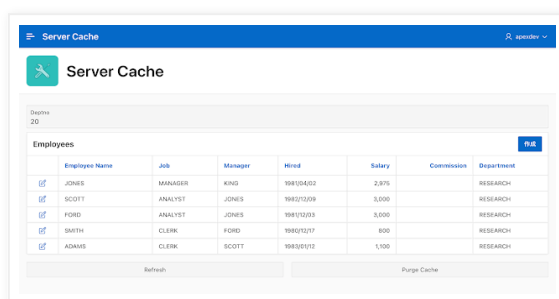
アプリケーション作成ウィザードを起動します。アプリケーションの名前は**Server Cache**とします。

デフォルトで作成されている**ホーム・ページ**を削除し、表**EMP**をソースとした**フォーム付きのクラシック・レポート**と、同様に表**EMP**をソースとした（フォームを付けない）**クラシック・レポート**のページを追加します。



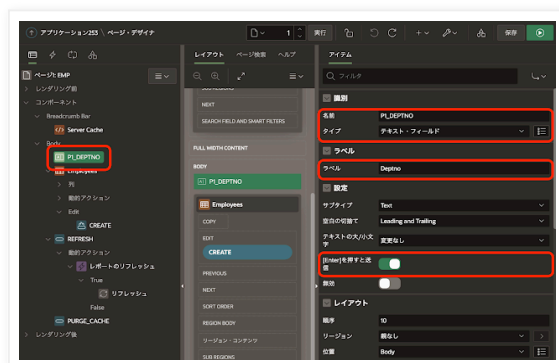
以上でアプリケーションを作成します。

作成したアプリケーションの**ホーム・ページ**が以下になるよう、**Deptno**の**ページ・アイテム**、**レポート**をリフレッシュするボタン、キャッシュされたリージョンのデータをパージするボタンを作成します。



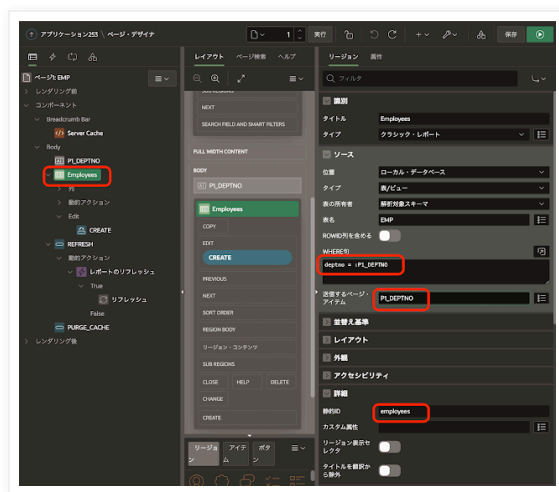
レポートの検索条件に使用する部門番号を保持する**ページ・アイテム**を作成します。

識別の名前は**P1_DEPTNO**、**タイプ**は**テキスト・フィールド**を選択します。**ラベル**は**Deptno**とし、設定の**[Enter]**を押すと送信を**オン**にします。



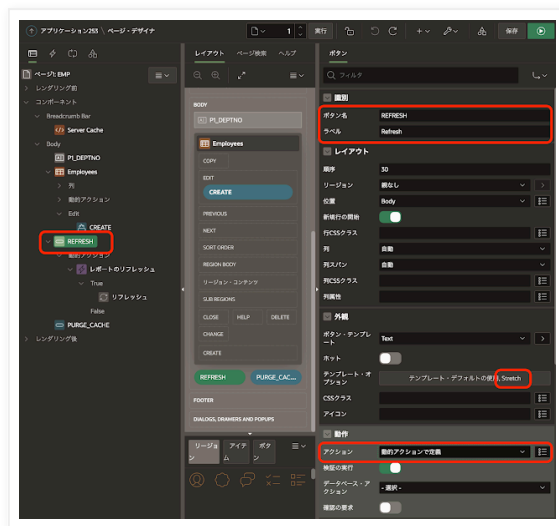
クラシック・レポートのリージョンのソースのWHERE句にdeptno = :P1_DEPTNOを記述します。**送信するページ・アイテム**としてP1_DEPTNOを設定します。これでクラシック・レポートは、指定された部門に所属する従業員のみを一覧表示します。

静的IDのemployeesは、キャッシュされたリージョンのデータをパージする際に必要な、リージョンIDを取得するために使用します。

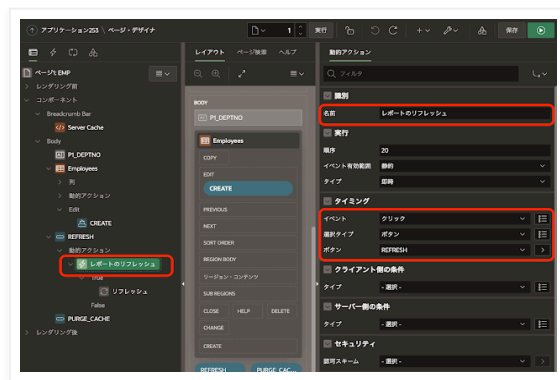


ボタンREFRESHを作成し、ボタンをクリックするとレポートをリフレッシュする動的アクションを作成します。

ラベルはRefresh、外観のテンプレート・オプションのWidthにStretchを指定しています。動作のアクションは動的アクションで定義を選択します。

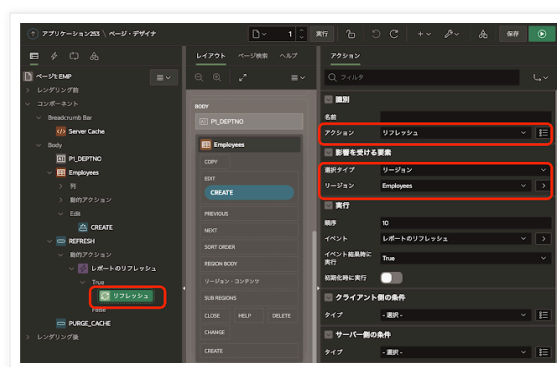


動的アクションの名前はレポートのリフレッシュ、タイミングのイベントはボタンのデフォルトであるクリックです。



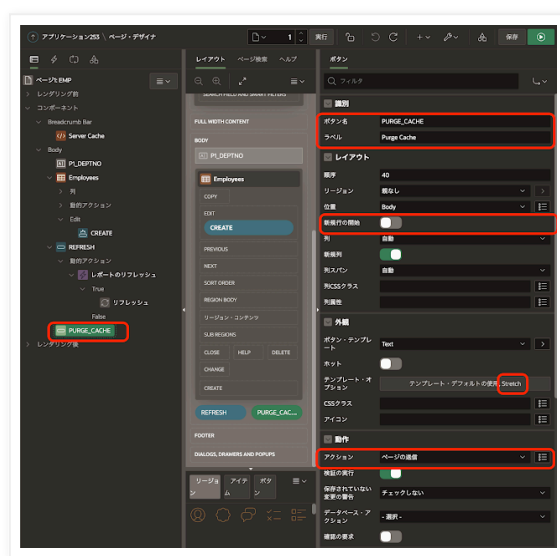
TRUEアクションとしてリフレッシュを選択します。影響を受ける要素の選択タイプとしてリージョンを選び、リージョンとしてクラシック・レポートであるEmployeesを選択します。

以上でボタンREFRESHを押すと、クラシック・レポートが更新されるようになりました。



キャッシュされたリージョンのデータをパージするボタンPURGE_CACHEを作成します。

ラベルはPurge Cache、レイアウトの新規行の開始をオフにして、ボタンREFRESHの右隣に配置します。外観のテンプレート・オプションのWidthを、ボタンREFRESHと同様にStretchに変更します。動作のアクションはデフォルトのページの送信から変更しません。



このボタンPURGE_CACHEをクリックしたときに実行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、プロセスを新規作成します。

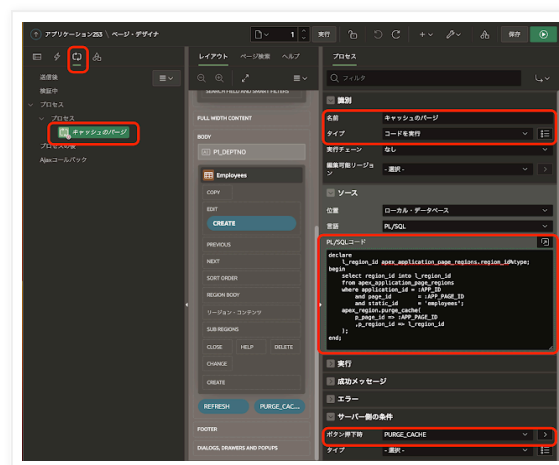
識別の名前はキャッシュのページ、タイプとしてコードを実行を選択します。ソースのPL/SQLコードとして以下を記述します。

```
declare
    l_region_id apex_application_page_regions.region_id%type;
begin
    select region_id into l_region_id
    from apex_application_page_regions
    where application_id = :APP_ID
        and page_id      = :APP_PAGE_ID
        and static_id     = 'employees';
    apex_region.purge_cache(
        p_page_id => :APP_PAGE_ID
        ,p_region_id => l_region_id
    );
end;
```

purge_application_cache.sql hosted with ❤️ by GitHub

[view raw](#)

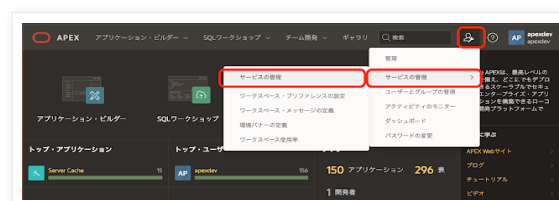
サーバー側の条件のボタン押下時にPURGE_CACHEを設定します。



以上でサーバー・キャッシュの動作を確認するアプリケーションは完成です。

キャッシュされているデータの確認は、ワークスペースの管理から行います。

ワークスペースの管理からサービスの管理を開きます。



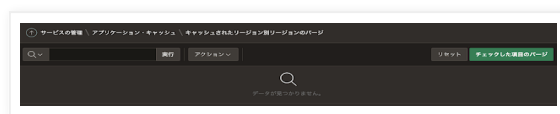
画面右のアプリケーション・キャッシュを開きます。



リージョン単位でキャッシュされたデータを確認するには、**キャッシュされたリージョン別リージョンのページ**を開きます。

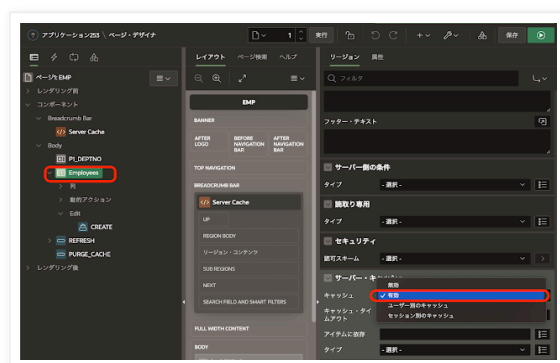


今の所何もキャッシュされていないので、何も表示されません。



リージョン**Employees**の**サーバー・キャッシュ**の**キャッシュ**を**有効**にします。**有効**の場合、すべてのアクセスで同じキャッシュを使用します。**ユーザー別**または**セッション別**にキャッシュを持つことも可能です。

それ以外は特に設定せず、効果を確認してみます。



アプリケーションを実行します。

最初は部門番号が設定されていないため、以下のように一行も従業員が表示されません。

しかし、リージョンのキャッシュは有効なので、この時点でレポートのHTML出力はキャッシュされています。

アプリケーション・キャッシュの画面から確認できます。

Deptnoに10を入力し、Enterを入力します。P1_DEPTNOとして10が送信され、レポートが再表示されますが、**データが見つかりません**と表示されます。

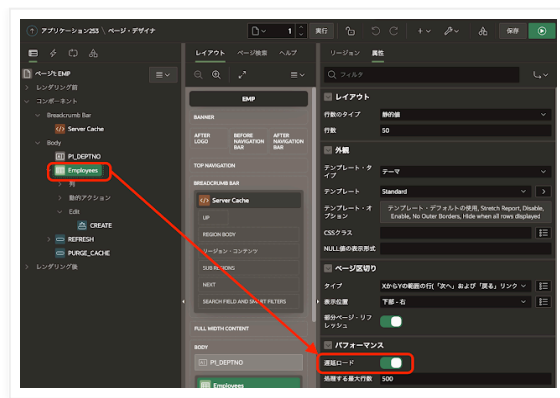
これはキャッシュにヒットしているため、この時点ではDeptnoに何を入力しても、結果は**データが見つかりません**となります。

しかし、ボタンRefreshをクリックすると、Deptno = 10（ACCOUNTING）に所属する従業員が一覧されます。これはクライアント側でレポートの描画を行なっているためです。

Employee Name	Job	Manager	Hired	Salary	Commission	Department
KING	PRESIDENT		1981/01/17	5,000		ACCOUNTING
CLARK	MANAGER	KING	1981/06/09	2,450		ACCOUNTING
MILLER	CLERK	CLARK	1982/01/23	1,300	200	ACCOUNTING

クライアント側で描画されたレポートはサーバー側でキャッシュされません。Deptnoに20を入力しEnterを押すと、キャッシュにヒットするため**データが見つかりません**と表示されます。

レポートの遅延ロードがオンの場合も、動的アクションによるリフレッシュと同様にクライアント側で描画されるため、サーバー・キャッシュの影響を受けません。



Deptnoに30を入力しEnterを押します。遅延ロードがオンの場合は、部門がSALESの従業員が一覧されます。

Deptno: 30

Employee Name	Job	Manager	Hired	Salary	Commission	Department
BLAKE	MANAGER	KING	19810501	2,850		SALES
ALLEN	SALESMAN	BLAKE	19810202	1,600	300	SALES
WARD	ANALYST	BLAKE	19810202	1,200	1000	SALES
MARTIN	SALESMAN	BLAKE	19810208	1,200	1400	SALES
TURNER	ANALYST	BLAKE	19810908	1,500	0	SALES
JAMES	CLERK	BLAKE	19810303	900		SALES

Refresh Purge Cache

遅延ロードをオフに戻し、データの更新を行ってみます。

Deptnoに20を入力し、Purge Cacheをクリックします。

今までキャッシュされていたリージョンのデータがパージされます。その後にDEPTNO = 20の条件でレポートが表示されます。このレポートは新たにキャッシュされます。(そのため、Deptnoに10や30を入れてEnterを押しても、部門RESEARCHの一覧が表示されます。)

Deptno: 20

Employee Name	Job	Manager	Hired	Salary	Commission	Department
JONES	MANAGER	KING	19810402	2,975		RESEARCH
SCOTT	ANALYST	JONES	19820309	3,000		RESEARCH
FORD	ANALYST	JONES	19810303	3,000		RESEARCH
SMITH	CLERK	FORD	19800107	800		RESEARCH
ADAMS	CLERK	SCOTT	19830102	1,100		RESEARCH

Refresh Purge Cache

従業員ADAMSの編集フォームを開き、Salaryを1100から2200に変更します。

変更の適用をクリックします。

Server Cache

Deptno: 20

Employees

Employee Name	Job
JONES	MANAGER
SCOTT	ANALYST
FORD	ANALYST
SMITH	CLERK
ADAMS	CLERK

Refresh

Employee

Employee Name: ADAMS

Job: CLERK

Manager: SCOTT

Hired: 19830102

Salary: 2200

Commission:

Department: RESEARCH

保存の適用

フォームが閉じると、ADAMSのSalaryが2200に更新されています。

Deptno
20

Employees

Employee Name	Job	Manager	Hired	Salary	Commission	Department
JONES	MANAGER	KING	19810402	2,975		RESEARCH
SCOTT	ANALYST	JONES	19821209	3,000		RESEARCH
FORD	ANALYST	JONES	19811203	3,000		RESEARCH
SMITH	CLERK	FORD	19801217	800		RESEARCH
ADAMS	CLERK	SCOTT	19831012	2,200		RESEARCH

Refresh Purge Cache

しかし、DeptnoでEnterを押してレポートを更新すると、ADAMSの給与は1100に戻ります。フォームを閉じたときにリージョンがリフレッシュ（つまり、クライアント側で描画）されるためです。データベースで保持されているSalaryは2200に更新されています。

Deptno
20

Employees

Employee Name	Job	Manager	Hired	Salary	Commission	Department
JONES	MANAGER	KING	19810402	2,975		RESEARCH
SCOTT	ANALYST	JONES	19821209	3,000		RESEARCH
FORD	ANALYST	JONES	19811203	3,000		RESEARCH
SMITH	CLERK	FORD	19801217	800		RESEARCH
ADAMS	CLERK	SCOTT	19831012	1,100		RESEARCH

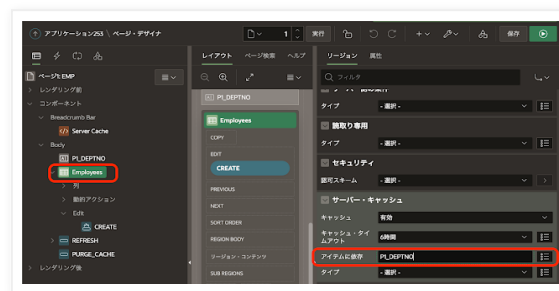
Refresh Purge Cache

まとめると、リフレッシュにより表示を更新できるリージョン・タイプでは、サーバー・キャッシュは役に立たないので設定すべきでない、といえます。

その点を考慮すると、クラシック・レポート（遅延ロードはオフ、リフレッシュが発生する処理は含めない）および動的コンテンツ（同様に遅延ロードはオフ、リフレッシュが発生する処理は含めない）であれば、サーバー・キャッシュを活用できそうです。

今回のようなケースではサーバー・キャッシュのアイテムに依存にP1_DEPTNOを設定します。

リージョンがキャッシュされるにあたり、指定したページ・アイテムの値が紐付けされます。



アイテムに依存にP1_DEPTNOを設定すると、Deptnoに10、20、30と値を入れてEnterを押したとき、指定した部門の従業員が一覧されるようになります。

Deptno
20

Employees

Employee Name	Job	Manager	Hired	Salary	Commission	Department
BLAKE	MANAGER	KING	19810501	2,850		SALES
ALLEN	SALESMAN	BLAKE	19810220	1,600	300	SALES
WARD	ANALYST	BLAKE	19810222	1,300	1000	SALES
MARTIN	SALESMAN	BLAKE	19810526	1,250	1400	SALES
TURNER	ANALYST	BLAKE	19810508	1,500	0	SALES
JAMES	CLERK	BLAKE	19811203	950		SALES

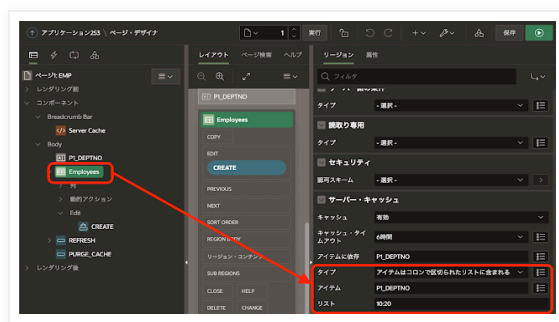
Refresh Purge Cache

レポートの表示を行う際に、P1_DEPTNOが前回の表示と同じ値であればキャッシュされたHTMLを表示します。SELECT文は実行されません。P1_DEPTNOの値が異なる場合は、キャッシュを破棄し、SELECT文を実行してHTMLを生成します。その後、次の要求に備えて生成したHTMLをキャッシュします。

検索条件を変えずにレポートの再表示が行われるケースが多ければ、サーバー・キャッシュを設定することによりサーバーの負荷を軽減できる可能性があります。

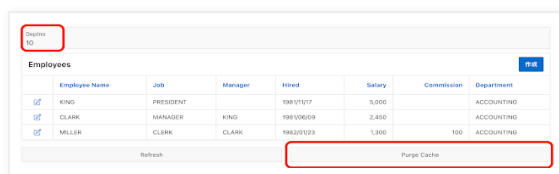
キャッシュが使用される条件を設定することもできます。

タイプとして**アイテム**は**コロンで区切られたリストに含まれる**を選択し、**アイテム**に**P1_DEPTNO**、**リスト**として**10:20**を指定します。

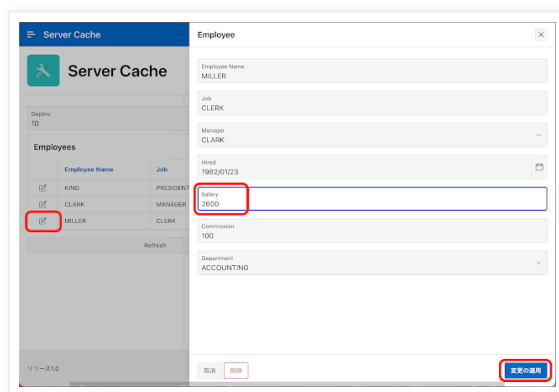


サーバー・キャッシュが有効なのはDeptnoが10と20のときに限る、という設定になります。

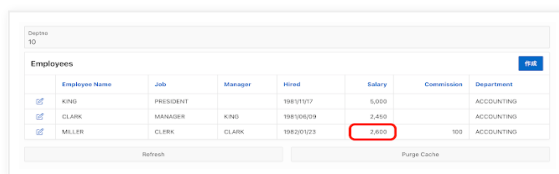
Deptnoに**10**を入力しボタン**Purge Cache**をクリックして、Deptnoが10のレポートがキャッシュされた状態にします。



従業員**MILLER**の編集フォームを開き、**Salary**を**2600**に変更します。



MILLERのSalaryはレポート上も2600に更新されますが、これはリージョンがリフレッシュされたため（クライアント側で再描画が行われたため）です。



Deptnoは10のまま変更せずEnterを入力すると、キャッシュされたデータで置き換えられるため、MILLERのSalaryは1300へ戻ります。

Employee Name	Job	Manager	Hired	Salary	Commission	Department
KING	PRESIDENT		1981/11/17	5,000		ACCOUNTING
CLARK	MANAGER	KING	1981/09/08	2,450		ACCOUNTING
MILLER	CLERK	CLARK	1982/01/23	1,300	100	ACCOUNTING

Deptnoの10および20はキャッシュの対象ですが、30と40は対象から外しています。Deptnoを30にして、同様の操作を行ってみます。

Deptnoに30を入力し、ボタンPurge Cacheをクリックします。

Employee Name	Job	Manager	Hired	Salary	Commission	Department
BLAKE	MANAGER	KING	1981/05/01	2,850		SALES
ALLEN	SALESMAN	BLAKE	1981/02/20	1,600	300	SALES
WARD	ANALYST	BLAKE	1981/02/22	1,300	1000	SALES
MARTIN	SALESMAN	BLAKE	1981/02/05	1,250	1400	SALES
TURNER	ANALYST	BLAKE	1981/09/08	1,500	0	SALES
JAMES	CLERK	BLAKE	1981/12/03	960		SALES

JAMESのSalaryを2000に変更します。

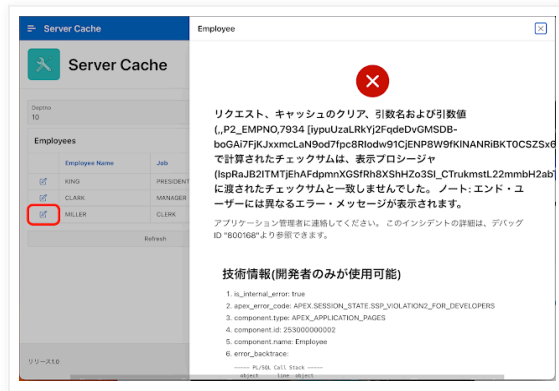
Employee Name	Job
BLAKE	MANAGER
ALLEN	SALESMAN
WARD	ANALYST
MARTIN	SALESMAN
TURNER	ANALYST
JAMES	CLERK

フォームが閉じるとJAMESのSalaryは2000になります。また、Deptnoの30はキャッシュの対象ではないため、Deptnoを30としてEnterを入力してもSalaryはデータベースに保存されている2000を表示します。

Employee Name	Job	Manager	Hired	Salary	Commission	Department
BLAKE	MANAGER	KING	1981/05/01	2,850		SALES
ALLEN	SALESMAN	BLAKE	1981/02/20	1,600	300	SALES
WARD	ANALYST	BLAKE	1981/02/22	1,300	1000	SALES
MARTIN	SALESMAN	BLAKE	1981/02/05	1,250	1400	SALES
TURNER	ANALYST	BLAKE	1981/09/08	1,500	0	SALES
JAMES	CLERK	BLAKE	1981/12/03	2,000		SALES

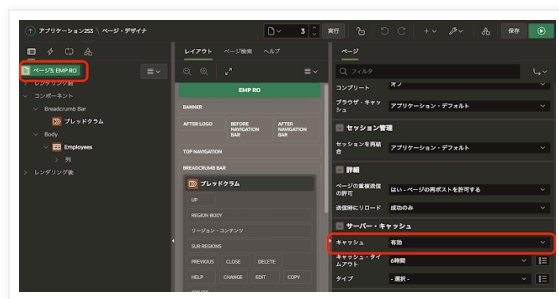
今までの手順でクラシック・レポートより更新フォームを開いていますが、この操作ができるのは同じセッションでキャッシュされたHTMLに限定されます。異なるセッションよりキャッシュされたHTMLの場合、編集フォームを開こうとするとチェックサムに関するエラーが発生します。

編集リンクに含まれているcs=で与えられているチェックサムの値は、リージョンをキャッシュしたセッションで計算されています。検証に使うチェックサムはキャッシュを参照しているセッションで計算されるので、セッションが異なるためチェックサムが一致しません。



以上のように、サーバー・キャッシュが活用できる場面はかなり限定されています。とはいえ、活用できる条件に合致さえすれば、間違いなくサーバーの負荷を減らすことができるでしょう。

ページのサーバー・キャッシュは、さらに活用できる場面は限定されています。



ページ上で対話的なオペレーションはなく、データベースのデータを参照してページを生成しているが、内容の更新が発生しない場合は、ページのサーバー・キャッシュを活用できそうです。

Employee Name	Job	Manager	Hire Date	Salary	Commission	Department
KING	PRESIDENT		1981/01/01	5,000		ACCOUNTING
BLAKE	MANAGER	KING	1981/03/01	2,850		SALES
CLARK	MANAGER	KING	1981/06/09	2,450		ACCOUNTING
JONES	MANAGER	KING	1981/09/02	2,975		RESEARCH
SCOTT	ANALYST	JONES	1982/03/09	3,000		RESEARCH
FORD	ANALYST	JONES	1981/01/03	3,000		RESEARCH
SMITH	CLERK	FORD	1980/01/01	800		RESEARCH
ALLEN	SALESMAN	BLAKE	1981/02/20	1,600	300	SALES
WARD	ANALYST	BLAKE	1981/02/22	1,300	1000	SALES
MARTIN	SALESMAN	BLAKE	1981/05/28	1,250	1600	SALES
TURNER	ANALYST	BLAKE	1981/09/08	1,500	0	SALES
ADAMS	CLERK	SCOTT	1980/01/12	2,200		RESEARCH
JAMES	CLERK	BLAKE	1981/01/03	2,000		SALES
MILLER	CLERK	CLARK	1982/01/23	2,600	100	ACCOUNTING

上記のように表EMPの一覧を表示しているような状態で、表EMPに変更があったときにAPEX_PAGE.PURGE_CACHEを呼び出してキャッシュを削除する、といった運用をすると、このページのHTMLを生成する負荷を削減できます。

キャッシュされたページについては、アプリケーション・キャッシュのキャッシュされたページ別ページのページより確認できます。



キャッシュ・モード = 有効を条件として、ページを一覧します。



サーバー・キャッシュについての説明は以上になります。

今回の説明に使用した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/server-cache.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: 17:29

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.