

APEX Datenverwaltung –Wo sind die Daten gerade?

Dr. Gudrun Pabst
Trivadis GmbH
München

Schlüsselworte:

APEX, Sessionverwaltung, Dynamic Actions

Einleitung

Eine APEX-Anwendung wird erst durch zusätzliche Logik benutzerfreundlich. Dabei stößt der Programmierer aber bald auf das Problem, dass er sich nicht darüber klar ist, in welcher Schicht die benötigten Daten gerade zur Verfügung stehen.

Im folgenden wird gezeigt, zu welchen Zeiten die Daten auf der Client-Seite im Browser oder in der APEX-Session in der Datenbank gespeichert sind und mit welchen Methoden Daten vom Browser in die APEX-Session und umgekehrt übertragen werden können.

Voraussetzungen

Alle in diesem Dokument beschriebenen Funktionalitäten setzen mindestens APEX 4.0 mit jQuery 1.4.2 und jQuery UI 1.8 voraus.

Start einer Anwendung: Wo sind die Daten?

Bevor der Anwender die APEX-Anwendung startet, liegen die Daten entweder gespeichert in der Datenbank oder noch nicht verarbeitet in irgendeiner Form beim Anwender.

Startet der Benutzer die Anwendung und lädt damit eine Seite, sind die Daten in der Datenbank und teilweise in der HTML-Seite, aber nicht im Session-Cache (dazu später mehr).

Gibt der Benutzer nun Daten in die HTML-Seite ein, speichert aber noch nicht, sind diese Daten ebenfalls in der HTML-Seite, aber noch nicht in der Datenbank und nicht im Session-Cache.

Wie kommen die Daten von der Datenbank in die HTML-Seite?

Einer der Wege, wie Daten von der Datenbank in die HTML-Seite übertragen werden, ist der "Page Load" – Vorgang:

- Beim Laden der Seite werden die Reports auf der Seite ausgeführt.
- Formularfelder werden durch Seitenprozesse und Berechnungen gefüllt.
- Die ermittelten Daten werden in die Seite eingefügt.

Die meisten dieser Daten werden aber nicht in den Session-Cache geschrieben.

Das Laden einer Seite wird initiiert durch

- einen Submit: die aktuellen Daten in der abgeschickten Seite werden in die Datenbank und den Session-Cache transferiert, dann wird die neue Seite geladen.

- ein Redirect auf eine URL (z.B. durch den Cancel-Button): die aktuelle Seite und ihre Daten werden verworfen, die angeforderte Seite wird geladen.
- ein Reload der Seite: die aktuelle Seite wird mit den Daten aus der Datenbank neu geladen.

Session-Cache

Die mit APEX entwickelten Anwendungen sind Web-Anwendungen und haben daher keine permanente Verbindung zur Datenbank.

Damit eine APEX-Anwendung aber dennoch Informationen über die aktuelle Session zur Verfügung hat, verfügt APEX über den Session-Kontext:

- Jede Session hat eine eindeutige ID.
- Die Session-ID wird auf der Client-Seite in einem Cookie gespeichert.
- Bei jeder Anfrage an die Datenbank wird die Session-ID mitgegeben.
- Die Daten für jede Session werden mit dieser Session-ID versehen und in einer Tabelle in der Datenbank gespeichert: Dies ist der Session-Cache.

Der Session-Cache kann über den Button „Session“ angezeigt werden:



Wie kommen die Daten von der HTML-Seite in den Session Cache?

Um Daten von der HTML-Seite in den Session Cache zu übertragen, gibt es zunächst zwei Wege:

- Links: Parameter, die mittels Links übergeben werden, werden in den Session Cache geschrieben.
- Submit: Alle Werte für alle Formularfelder werden im Session Cache gespeichert.

Im oben gezeigten Beispiel befindet sich auf Page 2 ein Report, der Button links im Report öffnet ein Formular (Page 3) mit Einzeldarstellung. Der Link übergibt die gewünschte ID an die Page 3.

Vor dem Aufruf der Page 3 sieht der Session Cache für diese Seite so aus:

Page Items						
Application	Page	Item Name	Display ▲	Item Value	Status	Encrypted
130	3	P3_LOC_ID	Hidden			No
130	3	P3_LOC_CTR_ID	Select List			No
130	3	P3_LOC_STREET	Text Field			No
130	3	P3_LOC_CITY	Text Field			No
130	3	P3_LOC_POSTAL_CODE	Text Field			No
130	3	P3_LOC_STATE_PROVINCE	Text Field			No

1 - 6

Nach der Navigation auf die Page 3 zeigt das Formular die gewünschten Daten an:

Im Session Cache befindet sich nur der durch den Link gesetzte Wert für P3_LOC_ID, nicht die aus der Datenbank eingelesenen Werte:

Page Items						
Application	Page	Item Name	Display ▲	Item Value	Status	Encrypted
130	3	P3_LOC_ID	Hidden	1300	Inserted	No
130	3	P3_LOC_CTR_ID	Select List			No
130	3	P3_LOC_STREET	Text Field			No
130	3	P3_LOC_CITY	Text Field			No
130	3	P3_LOC_POSTAL_CODE	Text Field			No
130	3	P3_LOC_STATE_PROVINCE	Text Field			No

1 - 6

Jetzt wird ein neuer Datensatz eingetragen:

Nach dem Submit sind die eingetragenen Werte auch im Session Cache gespeichert:

Page Items						
Application	Page	Item Name	Display ▲	Item Value	Status	Encrypted
130	3	P3_LOC_ID	Hidden	3204	Updated	No
130	3	P3_LOC_CTR_ID	Select List	8	Inserted	No
130	3	P3_LOC_STREET	Text Field	Lehrer-Wirth-Straße 4	Inserted	No
130	3	P3_LOC_CITY	Text Field	München	Inserted	No
130	3	P3_LOC_POSTAL_CODE	Text Field	81829	Inserted	No
130	3	P3_LOC_STATE_PROVINCE	Text Field		Inserted	No
						1 - 6

Refresh

Neben dem Laden einer Seite können Daten auch über einen teilweisen Refresh der Seite von der Datenbank in die HTML-Seite gelangen. Hierzu wird eine Dynamic Action definiert mit folgenden Einstellungen:

- Type : „Advanced“
- Action : „Refresh“
- Selection Type : Typ der Objekte, die neu geladen werden sollen
- Selection : Liste der Objekte, die neu geladen werden sollen

Soll ein Report mit einem teilweisen Refresh neu geladen werden, muss für diesen Report die Einstellung „Enable Partial Page Refresh“ auf „Yes“ gesetzt sein.

Diese Funktionalität kann z.B. genutzt werden, um ein Suchfeld zu definieren und den Report dann mit den eingegebenen Einschränkungen neu auszuführen, ohne die gesamte Seite nochmals zu laden.

In diesem Fall stellt man dann allerdings fest, dass die im Suchfeld eingegebenen Daten zunächst nur in der HTML-Seite zur Verfügung stehen und vor dem Ausführen der Abfrage in der Datenbank erst an den Session Cache übertragen werden müssen.

Da in diesem Szenario vermieden werden soll, die komplette Seite neu zu laden, kann hierfür der Submit der Page nicht genutzt werden, sondern es muss eine Dynamic Action definiert werden mit diesen Eigenschaften:

- Type : „Advanced“
- Action : „Execute PL/SQL Code“
- Page Items to Submit : Liste der Page Items, deren Werte in den Session Cache geschrieben werden sollen

Settings

```

begin
  :P5_TEST := :P5_LAST_NAME;
end;

```

* PL/SQL Code

Page Items to Submit

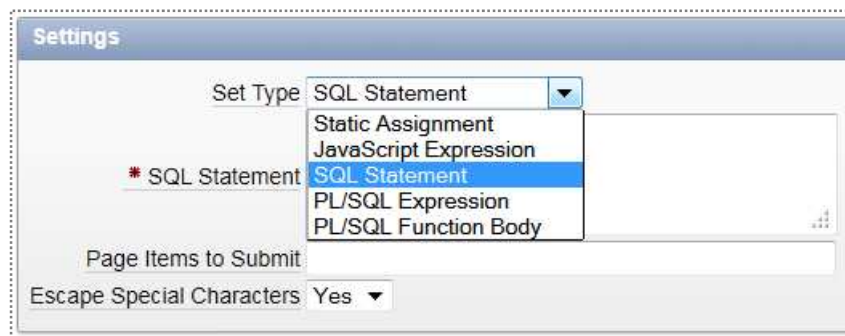
Nach Eingabe von Suchkriterien (z.B. W%) in das Feld P5_LAST_NAME und Auslösen der Suche zeigt der Session Cache den Wert des Suchfeldes:

130	5	P5_LAST_NAME	Text Field	W%	Inserted	No
-----	---	--------------	------------	----	----------	----

Setzen von Werten

Wenn in der Datenbank Werte vorhanden sind, die in der HTML-Seite in Javascript benötigt werden, aber beim Seitenaufbau nicht geladen werden konnten, können die Werte über eine Dynamic Action in die HTML-Seite übertragen werden. Diese Dynamic Action muss folgende Eigenschaften haben:

- Type : „Advanced“
- Action : „Set Value“
- Set Type : je nach gewünschter Funktionalität zum Ermitteln des Wertes



- Selection Type : Item(s)
- Item(s) : Liste der Items, die mit dem aus der Datenbank ermittelten Wert gefüllt werden sollen

Damit werden die gewünschten Items in der HTML-Seite auf den ermittelten Wert gesetzt; im Session Cache ist dieser Wert nicht sichtbar.

Fazit

Es ist wichtig, sich darüber im klaren zu sein, ob die zu programmierende Funktionalität in der Datenbank oder an der Oberfläche abläuft und ob die benötigten Daten an diesem Ort zur Verfügung stehen. Ist dies nicht der Fall, können die Daten mit den gezeigten Mittel an die Stelle transportiert werden, an der sie benötigt werden.

Kontaktadresse:

Dr. Gudrun Pabst
Trivadis GmbH
Lehrer-Wirth-Straße 4
D-81829 München

Telefon: +49 (0) 89-99 27 59 30
Fax: +49 (0) 89-99 27 59 59
E-Mail: gudrun.pabst@trivadis.com
Internet: www.trivadis.com