



data . cloud . mobile

apex.server.process  
vs  
htmldb\_get

Oliver Lemm  
APEX NRW Meetup  
16.02.2016

# Facts & Figures

Technologie-orientiert  
Branchen-unabhängig

Hauptsitz  
Ratingen

240  
Beschäftigte



Inhabergeführt

24 Mio. Euro  
Umsatz

Gründung  
1994

Zertifizierter  
Partner von  
Oracle,  
Microsoft  
und SAP

Ausbildungs-  
betrieb



Niederlassung  
Frankfurt am Main

# über mich

---

- Oliver Lemm
- Seit 02.2007 bei der MT AG in Ratingen
  - Junior Berater > Berater > Senior Berater > Competence Center Leiter APEX
- Diplom Angewandte Informatik an der Universität Duisburg-Essen
- Projektleiter, IT-Architekt und Entwickler
- beschäftigt sich seit 2007 mit Oracle Application Express und Oracle Datenbanken
  
- Blog <http://oliverlemm.blogspot.de>
- Twitter <https://twitter.com/OliverLemm>



# Agenda

---

1. AJAX
2. `htmldb_get`
3. `apex.server.process`
4. Livedemo

„AJAX = Asynchronous JavaScript and XML“

„asynchrone Kommunikation zwischen Browser & Server“

„Nachladen von Daten auf einer HTML Seite,  
ohne das die Seite neu geladen wird.“

# AJAX

## *Einsatzzwecke*

- Dynamisches Nachladen basierend auf Datenbankinhalten
  - LOV's in Tabular Forms
  - Komplexe HTML Strukturen
- Eigene Locking Mechanik
- Clientseitige Validierung basierend mit Logik im PLSQL
- Wiederverwendbare JavaScriptlogik inklusive PLSQL Abhängigkeiten

# AJAX in APEX

---

- Dynamic Action mit PLSQL
- JavaScript mit `htmldb_get` / `apex.server.process` und OnDemand Application Process

# htmldb\_get

## *Parameter*

- undokumentiert & nicht supported!
- standardmäßig **synchron** (blocking)
- function htmldb\_Get(obj , flow, req, page, instance, proc, queryString)
  - obj - kann null gesetzt werden
  - flow - Anwendungs-ID - \$v("pFlowId") oder &APP\_ID.
  - req - REQUEST (case sensitive!) - APPLICATION\_PROCESS=my\_plsql
  - instance - Seiten-ID - \$v("pFlowStepId") oder &APP\_PAGE\_ID.
  - proc - in apex nicht benötigt
  - queryString - in apex nicht benötigt



# htmlDb\_get

## *Funktionen*

- add(name, value)
  - Parameter welche als PageItem oder ApplicationItem existiert
- addParam(name,value)
  - Parameter welcher als Globaler Parameter existiert. GI
  - apex\_application.g\_x01 – apex\_application.g\_x10
- get(mode, startTag, endTag)
  - Kann komplett ohne Parameter aufgerufen werden -> dann String
  - Ansonsten mit erstem Parameter XML und die beiden nächsten optional -> XML

# htmldb\_get

---

// Aufruf Application Process

```
var ajaxRequest = new htmldb_Get(null
                                , $v("pFlowId")
                                , "APPLICATION_PROCESS=my_plsql"
                                , $v("pFlowStepId"));
```

// Hinzufügen der Parameter

```
ajaxRequest.addParam('x01', $v('P1_TEXT'));
```

// Abfrage durchführen

```
var ajaxReturn = ajaxRequest.get();
```

// Ergebnis zuweisen

```
$s('P1_RESULT', ajaxReturn);
```

# apex.server.process

---

- Einführung in APEX 4.2 zur Ablösung von `htmldb_get`
- In APEX 5 wurde `htmldb_get` als deprecated gekennzeichnet  
[https://docs.oracle.com/cd/E59726\\_01/doc.50/e39143/toc.htm#HTMRN315](https://docs.oracle.com/cd/E59726_01/doc.50/e39143/toc.htm#HTMRN315)
- Funktionalität standardmäßig **asynchron**
- In APEX API Kapitel 23 - JavaScript APIs dokumentiert  
[https://docs.oracle.com/cd/E37097\\_01/doc.42/e35127/javascript\\_api.htm#AEAPI29523](https://docs.oracle.com/cd/E37097_01/doc.42/e35127/javascript_api.htm#AEAPI29523)

# apex.server.process

---

- apex.server.process( pAjaxIdentifizier, pData, pOptions )
  - pAjaxIdentifizier - Bezeichner des Application Process
  - pData - Eigenschaften welche als Parameter mit übergeben werden
  - pOptions - verschiedene Eigenschaften, wie mit der Rückgabe umgegangen wird und ein Ladeindikator der angezeigt wird
- refreshObject, refreshObjectData, clear, apexbeforerefresh, loadingIndicator, loadingIndicator, loadingIndicatorPosition, before, after, prepend, append

weitere Attribute unter jQuery.ajax einsehbar

```
apex.server.process(  
    'my_plsql',                // Name des APEX Anwendungsprozesses  
    {x01: $v("P1_TEXT")}      // Parameterübergabe an global Variable "x01"  
    pageltems: "#P1_DEPTNO,#P1_EMPNO"  
},  
{  
    success: function (pData) { // JavaScript-Code für den Erfolgsfall  
        $s('P1_RESULT', pData); // Ergebnis anzeigen  
    },  
    dataType: "text"          // Datentyp Server-Antwort (hier: Plain Text, ansonsten JSON)  
}  
);
```

# Quellen

---

- Ajax (Programmierung)  
[https://de.wikipedia.org/wiki/Ajax\\_\(Programmierung\)](https://de.wikipedia.org/wiki/Ajax_(Programmierung))
- AJAX in APEX  
<https://sumanruet.wordpress.com/2014/02/03/ajax-in-apex/>
- AJAX Programmierung in APEX-Anwendungen: Wer nutzt noch htmldb\_Get?  
<https://apex.oracle.com/pls/apex/germancommunities/apexcommunity/tipp/3341/index.html>
- Ajax in Apex: introduction and technique  
<http://tpetrus.blogspot.de/2013/03/ajax-in-apex-introduction-and-techniques.html>

Vielen Dank!



[@OliverLemm](https://twitter.com/OliverLemm)



<http://oliverlemm.blogspot.de/>



<http://de.linkedin.com/in/OliverLemm>



[https://www.xing.com/profile/Oliver\\_Lemm](https://www.xing.com/profile/Oliver_Lemm)