

# Dynamic Actions 2.0?

Verwaltung komplexer Formulare in APEX

# Jürgen Sieben

Geschäftsführender Inhaber  
ConDeS GmbH & Co. KG

[j.sieben@condes.de](mailto:j.sieben@condes.de)

[Github.com/j-sieben](https://github.com/j-sieben)



# Komplexe Formulare in APEX

- Was macht ein Formular komplex?
- Wohin mit der Geschäftslogik?
- Wie kontrolliere ich die Anwendungsseite?
- Wie reduziere ich Komplexität?

# Wohin mit der Geschäftslogik?

- Deklarative Validierungen und Berechnungen stoßen bei komplexen Formularen an ihre Grenzen
- Aufteilung in deklarative und innerhalb der DB implementierte Teile schafft ein Wartungschaos
- Deklarative Validierungen und Berechnungen sind nicht wiederverwendbar und schwer zu überblicken

# Wohin mit der Geschäftslogik?

- Lösung:
  - Ein Package pro Seite in der DB, konzentriert sich auf:
    - Entgegennahme der APEX-Daten
    - Aufbereitung der Schreibweise
    - Evtl. Surrogation fehlender Eingabewerte
  - Ein oder mehrere Packages zur Implementierung der Geschäftslogik
    - Validieren und Speichern die Eingaben
  - KEINE Logik auf der APEX-Seite

# Beispiel (App-Alias DWH)

- Auf der Anwendungsseite
  - Validation: `dwh_ui_pkg.validate_<page_alias>;`
  - Verarbeitung: `dwh_ui_pkg.handle_<page_alias>;`
- In der Datenbank (Package `dwh_ui_pkg`)
  - Hilfsmethode liest typischer alle Seitenelementwerte
  - Validate-Methode veranlasst Validierung und Füllung von Standardwerten
  - Handle-Methode veranlasst Speicherung in DB
- s. Skriptbeispiel

# Wie kontrolliere ich die Anwendungsseite?

- Dynamische Aktionen unterliegen dem gleichen Problem wie Triggerprogrammierung
  - Unklar, wann welche DA auslöst
  - Verteilung von Code über die Seite
  - Vielfältige Implementierungsmöglichkeit erschwert  
Wartung
- Aber: DA bzw. JavaScript sind alternativlos.

# Wie kontrolliere ich die Anwendungsseite?

- ODER?
- Welche Alternativen existieren?
  - Eine zentrale JavaScript-Datei pro Anwendungsseite
  - State Chart Toolkit-Plugin (SCT)
- Anwendungsbeispiel



# State Chart Toolkit (SCT)

- Umkehrung der Zuständigkeit: DB kontrolliert Anwendungsstatus
- Ein generisches Plugin auf der Seite genügt
- Zentralisierte Verwaltung der Darstellungslogik
- Minimaler Wartungsaufwand auf der Anwendungsseite

# Arbeitsweise SCT

- Kernidee: Zentralisierte Regeln als SQL-Abfrage gegen den Sessionstatus
- Prinzip der Regelabfrage:

```
with session_state as(<HOLE SESSION STATUS>),  
    data as (<REGEL-VIEW, Regeln in WHERE-Klausel>)  
select <Regel, die ausgeführt werden soll>
```
- Regeln werden mit Aktionen in PL/SQL oder JavaScript verknüpft

# Arbeitsweise SCT

- Durch Analyse der Regeln automatische Konfiguration des Plugins
- APEX-Anwendung erleichtert Verwaltung und Analyse der Regeln
- Umfangreiches Logging erleichtert Analyse des Ablaufs der Regelsteuerung

# Arbeitsweise SCT

- Ex- und Importmöglichkeit für Regeln erleichtert Deployment
- Zukünftiges Potenzial durch Integration in APEX-IDE wäre möglich und würde die Bedienung drastisch vereinfachen

# Was noch?

- Oops, I did it again ...  
Neuerscheinung  
Oracle APEX – das umfassende Handbuch
  - Erscheinungstermin Herbst 2017
- Expertenseminar  
Datenbankprogrammierung für PL/SQL- und APEX-Entwickler
  - Termin: 27./28.06.2017, DOAG, Berlin

# Bewertung

Link zur Bewertung:

<http://bit.ly/2commW1u>