

Validierung der Dateneingabe mit Oracle ADF

Jürgen Menge

Oracle Deutschland B.V. & Co. KG

München

Schlüsselworte

Oracle ADF, Oracle Forms, Validierung, ADF Faces, JavaScript

Einleitung

Die Validierung der vom Anwender eingegebenen Daten ist ein wesentlicher Funktionsbaustein von Unternehmens-Applikationen. Nachdem in einem ersten Schritt das Masken-Layout umgesetzt wurde, kann die Implementierung notwendiger Validierungen einen erheblichen Aufwand verursachen.

Java EE/ADF ist nicht Forms !

Bereits bei der Entwicklung mit Oracle Forms galt die Empfehlung, dass bestimmte Validierungen in mehreren Schichten der Applikation zu implementieren sind:

Schicht	Zweck	Mittel	Zeitpunkt d. Ausführung
Datenbank	Gewährleistung der Integrität und Konsistenz der Daten	DB Constraints, DB Trigger (PL/SQL)	Post/Commit
Forms-Anwendung	Frühzeitiges Abfangen fehlerhafter Eingaben; benutzerfreundliche Reaktion auf Fehler	Deklarative Validierungen, Forms Trigger (PL/SQL)	unterschiedliche Zeitpunkte gemäß Event-Modell von Forms

Es gibt zwar eine Übersicht der "ADF Equivalents of Common Oracle Forms Triggers" (<http://www.oracle.com/technetwork/developer-tools/jdev/index-092937.html>). Diese sollte aber nicht dazu verleiten, schematisch die vorhandenen Forms-Validierungen gemäß der Übersicht in der Java/ADF-Welt zu implementieren.

Für den Entwickler von Web-Applikationen besteht die Herausforderung darin, fachlich gleichartige Validierungen innerhalb des Mehrschichten-Modells und in Übereinstimmung mit dem Request-Response-Modell einer Web-Applikation so zu implementieren, dass sowohl die Integrität der Daten als auch die Ergonomie der Benutzerführung gewährleistet werden. Insbesondere Kreuzprüfungen (Cross Check Validation) sind teilweise kompliziert zu implementieren, da die Reihenfolge der Eingabe bzw. Korrektur der Daten durch den Anwender zumeist nicht vorhersagbar sind.

Die konkreten Möglichkeiten der Implementierung sind vom verwendeten Technologie-Stack abhängig. In diesem Vortrag werden speziell Web-Applikationen auf Basis des Oracle Fusion Stacks betrachtet:

Schicht	Zweck	Mittel	Zeitpunkt d. Ausführung
Datenbank	Gewährleistung der Integrität und Konsistenz der Daten	DB Constraints, DB Trigger (PL/SQL)	Post/Commit
ADF Business Components	Abfangen fehlerhafter Eingaben vor der Übermittlung an die Datenbank	Deklarative Validierungen (darunter Regular Expressions, Groovy Scripts) Programmatische Validierungen (Java)	Submit

Binding Layer	Abfangen fehlerhafter Eingaben vor der Übermittlung an den Business Service (z.B. Web Service)	Deklarative Validierung von Binding-Attributen im Pagedef-File	Submit
ADF Faces server-seitig	Abfangen fehlerhafter Eingaben vor der Übermittlung an den Business Service	Validierungs-Attribute, Default und Custom Validatoren (Java)	Submit
ADF Faces client-seitig	Frühestmögliche Reaktion auf fehlerhafte Eingaben	Validierungs-Eigenschaften, Default und Custom Validatoren (JavaScript); explizite Validierung mittels JavaScript	Client-seitige Events

ADF Business Components

Validierungen sollten soweit wie möglich in dieser Schicht implementiert werden. Das Framework bietet auch die meisten Möglichkeiten, erforderliche Validierungen umzusetzen. Die eingebauten Validierungen sind für alle Seiten wirksam, die auf diesen Komponenten basieren.

Neben deklarativen Validatoren (s. Tabelle) gibt es die Möglichkeit, komplexe Validierungen auch auf programmatischem Weg zu implementieren.

Validator	Zuordnung	Charakteristik
Collection Validator	Entity Object (EO)	Gruppenoperationen auf Attributen (Sum, Avg, etc.)
Unique Key Validator	Entity Object (EO)	Prüfung auf eindeutige Schlüssel
Compare Validator	EO, EO Attribut	Vergleich von Werten
Key Exists Validator	EO, EO Attribut	Prüfung auf existierenden Schlüssel
Length Validator	EO, EO Attribut	Prüfung der Länge
List Validator	EO, EO Attribut	Prüfung auf Existenz eines Wertes in einer Liste
Range Validator	EO, EO Attribut	Prüfung, ob Wert in einem Bereich liegt
Regular Expression Validator	EO, EO Attribute	Package java.util.regex
Script Validator	EO, EO Attribute	Verwendung von Groovy-Scripten

ADF Binding Layer (Model)

ADF bietet die Möglichkeit, bestimmte Validierungen für Attribute im Binding Layer zu definieren. Dies ist jedoch in einer Fusion Web-Applikation normalerweise nicht erforderlich, wenn die Validierungen bereits in der Business Service Schicht (ADFbc) implementiert wurden.

Sinnvolle Anwendungsfälle für Validierungen im Binding Layer sind:

- wenn der Business Service keine deklarative Validierung unterstützt
- wenn die Validierung unnötige Network Roundtrips verhindert (z.B. bei einem Web Service)
- wenn die Parameter einer Parameter-Form vor der Ausführung validiert werden sollen.

ADF Faces

Validierungen spielen in der Oberflächen-Schicht eine wichtige Rolle, da sie die Schnittstelle zum Benutzer darstellt, in der auch Hinweise (Hints) und Fehlermeldungen im Zusammenhang mit den Validierungen angezeigt werden.

Einfache Standard-Validierungen werden durch **Validierungseigenschaften** (z.B. *Required*) der jeweiligen Komponente (z.B. *af:inputText*) abgebildet. Zusätzlich können in ADF Faces bestimmte **Default Validatoren** verwendet werden, die gegenüber den Validatoren des JavaEE Frameworks Java Server Faces (JSF) funktional erweitert wurden. Während JSF die Validierung nur server-seitig, d.h. nach einem Submit der Page, durchführt, fügt ADF Faces implizit eine client-seitige Validierung mittels JavaScript hinzu. Damit bekommen die Anwender Fehlermeldungen zum frühestmöglichen Zeitpunkt noch bevor die Daten an den Application Server geschickt werden.

Validatoren sind in ADF Faces eigenständige Komponenten (Tags), die der zu validierenden Komponente hinzugefügt werden können. Aus der Komponenten-Auswahl können sowohl JSF- als auch ADF Faces-Validatoren ausgewählt werden. Die ADF Faces-Validatoren (s. Tabelle) besitzen zusätzliche Eigenschaften, wie z.B. Texte für Hinweise (Hints) und Fehlermeldungen. Einer Komponente können mehrere Validatoren zugeordnet werden, die in der Reihenfolge der Zuordnung abgearbeitet werden.

Validator	Charakteristik
af:validateByteLength	Prüfung auf max. zulässige Länge in Bytes
af:validateDateRestriction	Prüfung auf unzulässige Tage und Monate
af:validateDateTimeRange	Prüfung auf zulässiges Zeitintervall
af:validateDoubleRange	Prüfung, ob Zahl vom Typ Double innerhalb eines Bereiches liegt
af:validateLength	Prüfung, ob Länge eines Strings innerhalb eines Bereiches liegt
af:validateLongRange	Prüfung, ob Zahl vom Typ Long innerhalb eines Bereiches liegt
af:validateRegExp	Prüfung auf Regular Expression (<code>java.util.regex</code>)

Reichen die Default Validatoren nicht aus, können eigene **Custom Validatoren** geschrieben und verwendet werden.

Wird ein Custom Validator nur innerhalb einer bestimmten Seite benötigt, kann die Validierungsmethode innerhalb einer Backing Bean definiert werden. Ist die Validierung allgemeingültig, sollte eine eigene JSF Validator-Klasse geschrieben werden, die das Interface `javax.faces.validator` implementiert. Diese Klasse muss in der `faces-config.xml` registriert werden.

Wie bereits erwähnt, findet in ADF Faces sowohl eine server- als auch client-seitige Validierung mit einem jeweils eigenen Request Life Cycle statt. Die client-seitige Validierung wird mittels JavaScript auf dem Arbeitsplatz des Anwenders ausgeführt. Tritt bei dieser Validierung eine Fehlersituation auf, wird die betreffende Seite nicht submitted, d.h. es werden keine Daten zum Application Server gesendet.

Wird dieses Verhalten auch für einen Custom Validator benötigt, muss eine eigene client-seitige Implementierung in JavaScript geschrieben werden. Hierfür gelten die Vorgehensweise und Empfehlungen der Verwendung von JavaScript in ADF Faces, d.h. die Funktionalität sollte gegen das JavaScript API von ADF Faces programmiert werden.

Verwandte Themen

Dazu zählen die Fehlerbehandlung (Exception Handling) und die Anzeige der Fehlermeldungen (u.a. Mehrsprachigkeit). Neben den Validatoren gibt es in JSF bzw. ADF Faces auch Converter, die Eingaben in bestimmte Formate konvertieren, die ähnlich behandelt werden.

Diese Themen werden jedoch im aktuellen Vortrag nicht berücksichtigt.

Weiterführende Informationen

Fusion Developer's Guide for Oracle Application Development Framework - 11.1.2.2.0

- ADF Business Components Kapitel 7 und 8
- ADF Model Kapitel 16

Web User Interface Developer's Guide for Oracle Application Development Framework - 11.1.2.2.0

- Converter und Validatoren Kapitel 7

Kontaktadresse:

Dr. Jürgen Menge
Oracle B.V. & Co. KG
Riesstr. 25
D-80992 München

Telefon: +49 (0) 89-1430 2239
Fax: +49 (0) 89-1430 2150
E-Mail juergen.menge@oracle.com
Internet: www.oracle.com