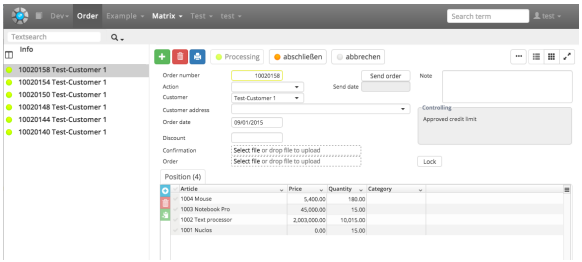
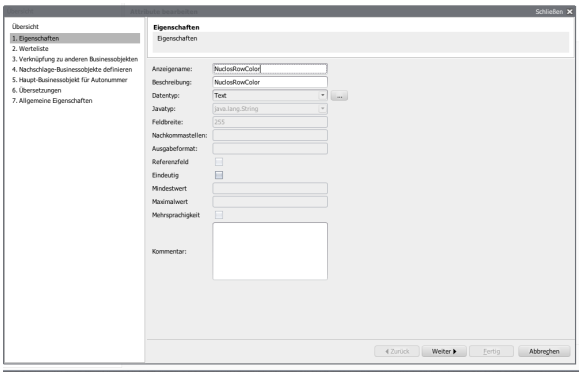
Farbige Zeilen

Ausgangssituation

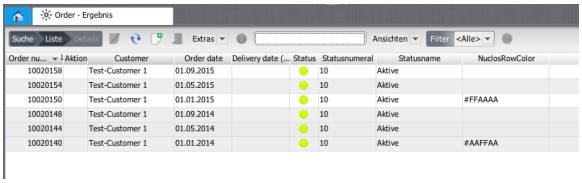
Man möchte die Zeilen in der Listenansicht der Bo-Klasse "Order" farblich hinterlegen:



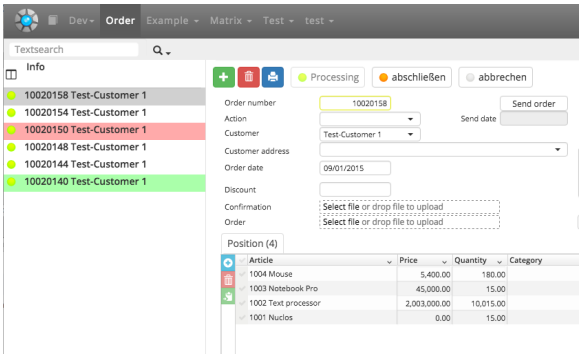
Zu diesem Zweck fügt man ein Attribut (NuclosRowColor) im Bo-Wizard zu der Bo-Klasse "Order" hinzu. Groß- und Kleinschreibung ist an dieser Stelle egal.



Man kann, wie in diesem Fall, ein Text-Objekt verwenden, oder auch ein berechnetest Attribut. Die Zeilenfarbe kann nun für jeden Datensatz in der Form RGB-Hexadezimal (#RRGGBB) definiert werden:



Entsprechend werden die Zeilen nun farblich hinterlegt:



Umsetzungsempfehlung

Groovyregeln zur Farbhinterlegung von Zeilen (zu hinterlegen im BO-Editor) werden im Webclient (wie alle Groovyregeln) nicht mehr funktionieren.

Daher besteht folgende Empfehlung zur Umsetzung gewünschter Farbehinterlegungen, die in beiden Clients funktionieren soll:

a) Anlegen eines neuen Attributes “NuclosRowColor” (String) in der gewünschten Entität.
b) Dieses als berechnetes Attribut mit einer DB-Funktion wie gewünscht befüttern. Gefüttert werden muss es mit einem HTML-Farb-Hexcode (z.B. #FFFFFF), so wie heute schon in den Groovyregeln. Die eigentliche Farbermittlungslogik muss also in einer DB-Funktion stecken, nicht in einer Groovy-Funktion. Das ist auch wesentlich sinnvoller, weil man nur in einer DB-Funktion auch Zugriff auf andere Entitäten hat (z.B. um eine Auftragsposition in Abhängigkeit vom aktuellen Lagerbestand farblich zu hinterlegen).

Der Webclient wird ein solches Attribut zukünftig nicht in der Spaltenliste anzeigen, sondern grundsätzlich als Farbe für die Zeile interpretieren.

Damit es auch im Rich Client weiter funktioniert, muss die Groovy-Regel also nur noch wie folgt aussehen, der Webclient ignoriert sie dann einfach:
“return <NUCLET>.<BO>.NuclosRowColor;”

Die Groovy-Regel darf also nur noch den Rückgabewert der DB-Funktion (an den Rich Client) weiterreichen.

Hier ein MS SQL Server Beispiel für eine solche DB-Funktion:

```
CREATE FUNCTION R0X7_CA_AP_ROWCOLOR(@id NUMERIC) RETURNS VARCHAR(255) AS
BEGIN
DECLARE @val VARCHAR(255);
DECLARE @free NUMERIC(9,2);
DECLARE @used NUMERIC(9,2);

select @free = min(bestand-reserviert) from R0X7_V_BESTANDSPLANUNG where intid_strauftragsposition =
@id;
select @used = dblmenge from R0X7_MESSEAUFTRAGSPOSITION where intid = @id;

if (@free > (@used + 10))
select @val = '#01DF01';
else
if (@free > @used)
select @val = '#F7FE2E';
else
select @val = '#FF0000';

return @val;
END;
```