

MS Office als Reports Designer

Roland Hörmann
SIB Visions GmbH
Wehlistraße 29 / Stiege 1 / 2.Stock, 1200 Wien

Schlüsselworte

Reports, effizient Business Berichte erstellen, Java, Open Source, JVx Framework

Einleitung

Im Kundenprojekt für einen österreichischen Schulungsanbieter war die Anforderung die Berichte für den Kunden flexible anpassbar zu gestalten. Jeder der schon ein paar Reporting Tools verwendet hat, weiß dass diese für Applikationsbenutzer wirklich nicht geeignet sind. Da entstand bei uns die Idee die Berichte durch die Benutzer selbst in Word und EXCEL anpassen zu lassen.

Ich zeige Ihnen wie Sie mit Word und EXCEL ihre Berichte designen können. Als Basis dafür verwende ich Kundenreports und ersetze die variablen Daten mit Platzhaltern. Diese werden bei der Erstellung des Reports automatisch befüllt. Ich arbeite mit beliebig geschachtelten LOOP Tags um die Daten von beliebigen Datenquellen (Tabellen, Views, ...) zu verwenden. Daten abhängige Formatierungen sind dabei auch umsetzbar. Neben Word und EXCEL, können die Berichte auch in PDF Format erstellt werden. Derselbe Mechanismus lässt sich auch für E-Mails im HTML Format anwenden.

Bei der Umsetzung haben wir verschiedene Open Source Frameworks kombiniert (Apache POI, iText, PDF Creator) um das Ergebnis zu erzielen. Ich werde Ihnen an Hand einiger Java Code Beispiele die Möglichkeiten zeigen.

Das Platzhalterkonzept

Wenn man für die öffentliche Hand die Schulungen der Arbeitslosen in Förderprojekten durchführt, ist natürlich einiges an Berichtswesen für die Sachbearbeiter der verschiedenen Stellen der Behörden umzusetzen. Natürlich sind die Berichte für jedes Projekt unterschiedlich, aber sie sind sogar für gleiche Projekttypen pro Bundesland und Sachbearbeiter unterschiedlich. Verständlicherweise ändern sich diese auch jährlich. Daher musste das Berichtswesen flexibel und durch die Benutzer anpassbar sein.

Nachdem wir die Idee geboren haben die Berichte durch die Benutzer in Word bzw. EXCEL anpassbar zu machen, tüftelten wir an einer Lösung dafür. Uns gefiel einfach die Mächtigkeit von Word und EXCEL und die Tatsache dass die Benutzer damit umgehen können. Doch wie bekommen wir die Daten aus unserer Datenbank in das RTF bzw. XLS/XLSX Format. Auf Grund der Einfachheit gefielen uns Platzhalter, welche mit den Daten befüllt werden. Daher wird das Layout in den Office Applikationen wie gewohnt erstellt und danach werden die Bereiche die mit Daten befüllt werden sollen, einfach durch Platzhalter in eckigen Klammern ersetzt.

Format: [SPALTENNAME] - Z.B.: [VORNAME]

Das hat auch den Vorteil dass wir keine VB Plugins für Office programmieren mussten, um die Datenbindung durch den Benutzer definieren zu können. Die Platzhalter zu verschieben, zu löschen oder zusätzliche aus der Liste der möglichen im Office zu platzieren ist für jeden Benutzer einfach

lösbar.

Da rein die Spalte aus einer Tabelle, View oder beliebigen Query nicht ausreichend ist, haben wir noch die Angabe der Tabelle/Query und eine Formatangabe hinzugefügt.

Format : [TABELLE! SPALTENNAME (TYP=FORMAT)]

Z. B. : [PRODUKT! PREIS (NUMBER=#, ##.00)]

Neben der Formatierung von Beträgen, haben wir uns noch für folgende Formate entschieden:

[TABELLE! SPALTENNAME (CHECKBOX)]

[TABELLE! SPALTENNAME (CHECKBOX=Yes , No)]

[TABELLE! SPALTENNAME (DATE=d. MMMM yyyy)]

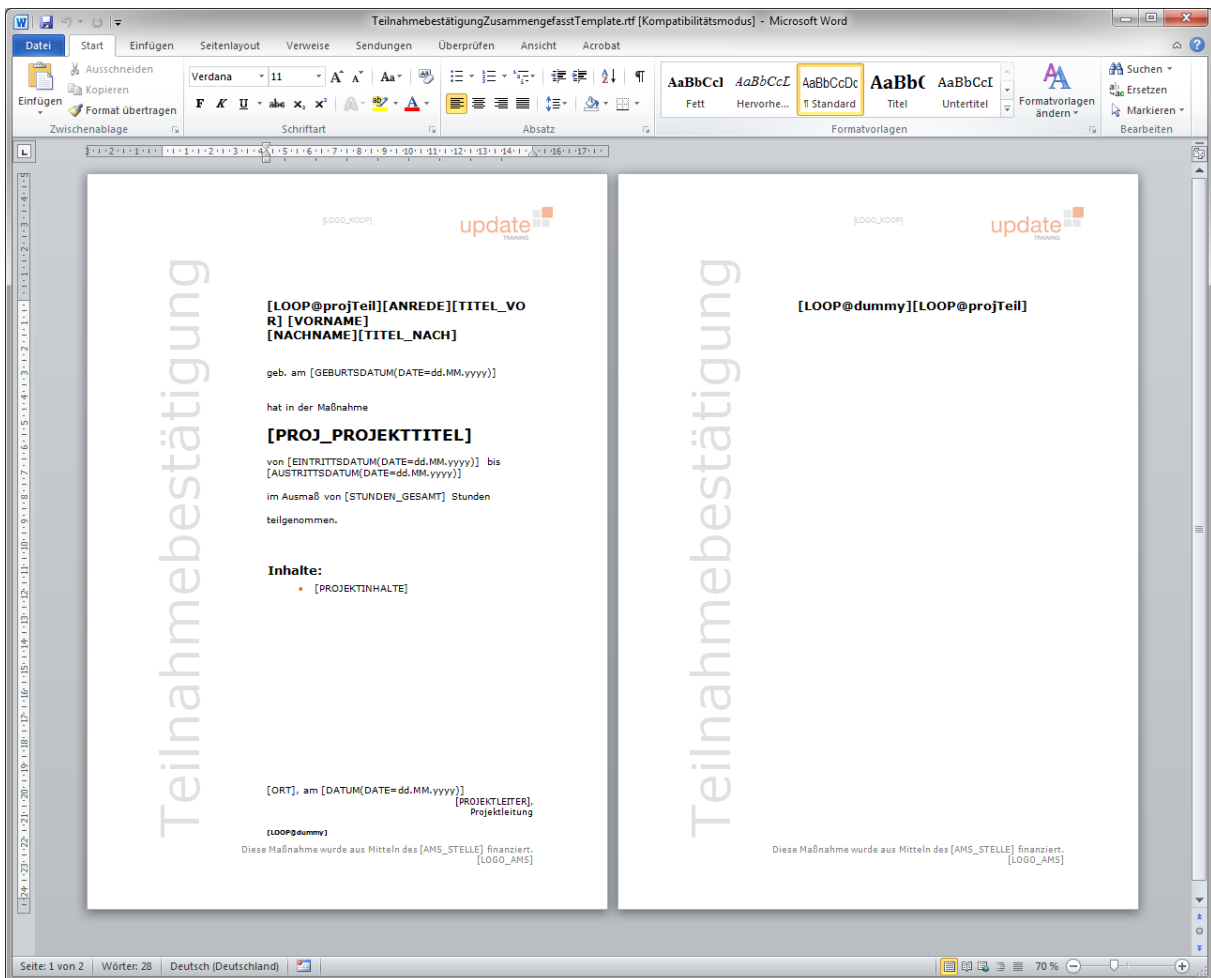


Abb. 1: Teilnehmerbestätigung Report-Template

In Berichten iteriert man typischerweise über mehrere Queries, die auch voneinander abhängig sind. Wie z.B.: Die Liste aller Kontakte und pro Kontakt die Liste seiner Adressen.

Um dies zu ermöglichen, haben wir einen Platzhalter (Loop Tag) zum Iterieren der Daten festgelegt.

[LOOP@TABELLE] . . . [LOOP@TABELLE]

Z. B. :

```
[LOOP@teilnehmer] [ANREDE] [TITEL] [VORNAME] [NACHNAME] ...  
[LOOP@teilnehmer]
```

Dieser wird am Anfang der ersten Zeile, und in der Folge Zeile die zu iterieren ist hinzugefügt. Der Inhalt zwischen beiden Loop Tags wird für jede Datenzeile des Queries kopiert und die Daten eingefüllt. Die Formatierung bleibt dabei wie in Word oder EXCEL festgelegt erhalten.

Nachdem wir unser Report Template erstellt haben, geht's jetzt an den Java Code um die Daten in das Template einzufüllen.

```
// Master Knoten  
ListNode lsnMaster = new ListNode();  
  
// Liste der Teilnehmer (getTeilnehmer()) für "LOOP@teilnehmer" Tag  
StorageNode snTeilnehmer = new StorageNode(getTeilnehmer());  
lsnMaster.setLoopDefinition("teilnehmer", snTeilnehmer);  
  
// Worker für .rtf Dateien holen, Template laden und Daten einfüllen  
AbstractWorker worker = AbstractWorker.getWorker(".rtf");  
worker.loadDocument(new FileInputStream(new File("/temp",  
                                             "/reports/Teilnehmer.rtf")));  
worker.fillInData(lsnMaster);  
  
// TeilnehmerListe.rtf zurückliefern  
RemoteFileHandle rfh = new RemoteFileHandle("TeilnehmerListe.rtf");  
worker.saveDocument(rfh.getOutputStream());  
  
return rfh;
```

Wir definieren unter dem Master Knoten eine Liste und befüllen Sie mit unseren Teilnehmern (getTeilnehmer()) und machen sie unter dem Namen „teilnehmer“ für unseren Loop Tag LOOP@teilnehmer verfügbar. Danach erstellen wir eine Worker Instance für das RTF Format (getWorker(„.RTF“), laden unser Template (loadDocument) und befüllen es mit den Teilnehmern(fillInData). Schlussendlich liefern wir unsere fertige TeilnehmerListe.rtf zurück. Mehr Code ist es nicht!

Word, EXCEL, PDF oder E-Mail

Das Konzept ist aber nicht auf das RTF Format begrenzt. Wir haben in diesem Projekt eine Implementierung für EXCEL (.XLS/.XLSX) umgesetzt. Dabei haben wir POI benutzt um den Mechanismus umzusetzen. In Projekten braucht es aber auch oft unveränderliche Berichte im PDF Format. Das ist bei unserem Konzept mit einem PDF Konverterservice möglich. Dazu kann man entweder einen der Open Source basierten wie Z.B.: PDF Creator oder einen der kommerziellen wie z.B. die BCL EasyPDF SDK verwenden. Diese funktionieren wie ein Druckertreiber, daher wird über MS Office das Dokument gedruckt und das Ergebnis wird über einen Postcript Druckertreiber in das PDF Format konvertiert und als Datei gespeichert. Darum haben wir im Projekt ein entsprechendes Java Webservice implementiert um es in unsere Applikation zu integrieren. Unser Konzept haben wir auch für Text bzw. HTML Format E-Mail verwendet. Das im Projekt für verschiedene Info E-Mails Verwendung fand.

EXCEL Formulare importieren

Nachdem das neue Reporting für unser Kundenprojekt sehr gut funktioniert hat, haben wir uns an etwas ganz verrücktes gewagt. Wir sind in der Vergangenheit bei Projekten schon öfter gefragt worden, ob wir Daten auch Offline in EXCEL Formularen erfassen können und diese dann in die

Applikation importieren können. Als Entwickler fragt man sich da gleich, warum verwendet ihr nicht die von uns entwickelte Maske zum Erfassen der Daten. Ja, sicher gibt es den Use Case Dinge offline zu erfassen, doch darf nicht vergessen werden, dass man oft noch Daten von anderen Unternehmen oder Abteilungen bekommt. Doch dafür gibt es nicht immer Schnittstellen, oder wie bei SAP sind diese nur mit sehr hohem Aufwand umzusetzen und stehen meist erst nach Monaten zur Verfügung. Warum nicht einfach die Daten aus EXCEL in die Applikation importieren.

Während der Entwicklung des Reportings über Platzhalter ist einen Kollegen die zündende Idee in den Kopf geschossen. Die Platzhalter sind eigentlich eine perfekte Beschreibung, wo die Daten ausgegeben werden. Aber sie sind genauso eine allgemeine Beschreibung wo sich die Daten für einen Import befinden. Wir haben einfach basierend auf unseren Reporting einen generischen EXCEL Importer geschrieben. Dies war zwar um einiges komplexer in der Implementierung, aber manche Dinge muss man einfach ausprobieren ob sie funktionieren.

Neben den bekannten Platzhaltern haben wir einen neuen sehr hilfreichen eingeführt. Format :

[TABELLE!SPALTENNAME (LIST=A;B;...)] oder
 [TABELLE!SPALTENNAME (LIST==DataArea)]
 Z. B. : [contacts!SALU_SALUTATION(LIST=Frau;Herr)]
 [contacts!SALU_SALUTATION(LIST==Anreden)]

Nämlich die Definition von Auswahllisten für EXCEL um die Fehlertoleranz zu reduzieren.

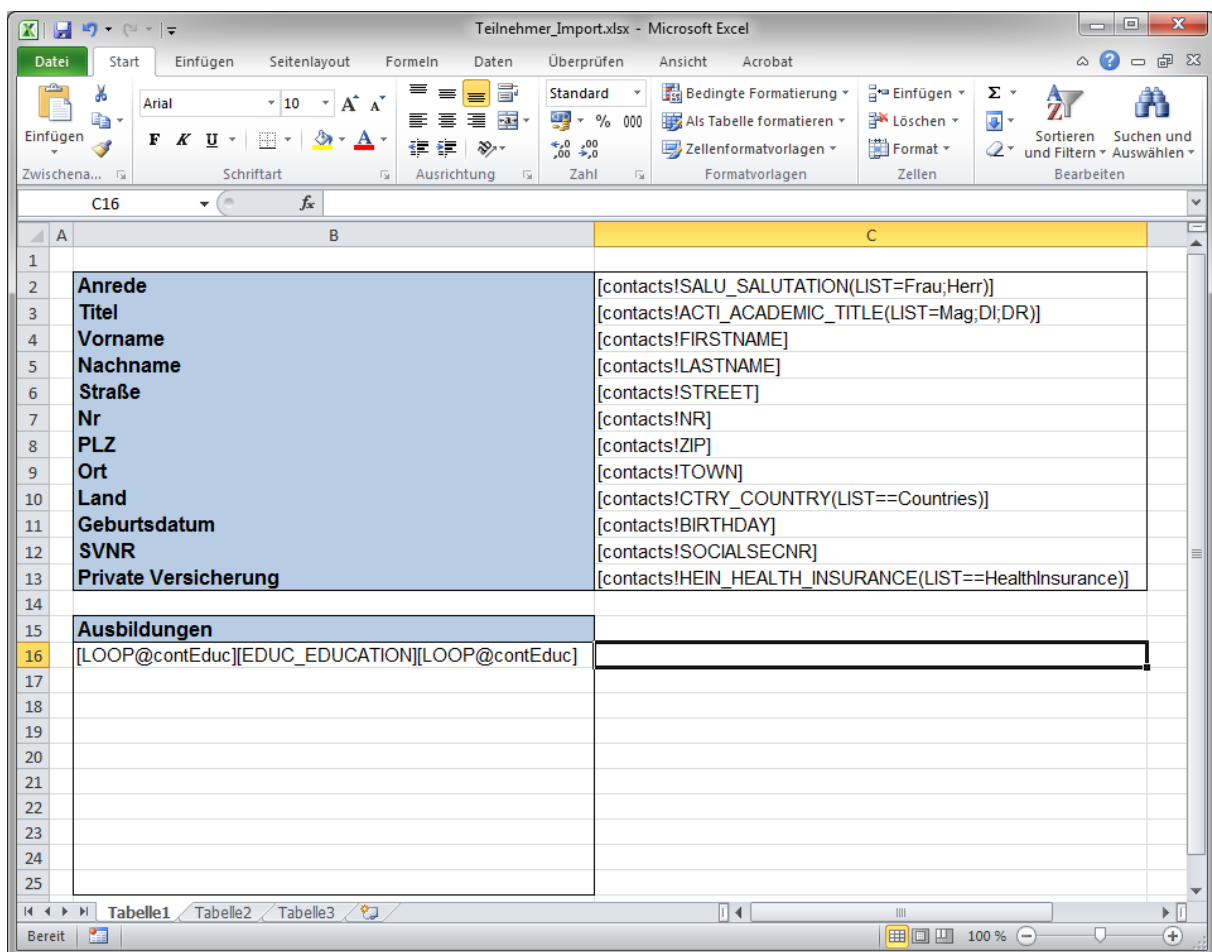


Abb. 2: Teilnehmerimport – Report/Import-Template

Diese Auswahllisten können direkt mit einer Werteliste, oder mit Datenbereichen definiert werden, die aus der Datenbank gefüllt werden können.

Den Importer konnten wir im Projekt für den flexiblen Import der Teilnehmer gut gebrauchen. In unseren Use Case werden die Teilnehmer über Prüfungen und Qualifikationsworkshop den geeigneten Kursen zugeordnet. Diese Qualifizierung wird von entsprechenden externen Instituten durchgeführt und die Teilnehmerlisten in unterschiedlichen EXCEL Listen geliefert. Ähnlich ist es bei den Teilnehmern die direkt vom AMS (Behörde für Österreichs Arbeitslose) ins Projekt zugeteilt werden. Damit konnten wir für die unterschiedlichen Formate, welche immer die gleichen Inputdaten für uns lieferten, einfach in unterschiedlichen Templates definieren.

Hübsches für die Entscheider in EXCEL

Für die Entscheider beim Kunden haben wir für dieses Projekt noch ein paar schöne grafische Management Reports benötigt. Dazu haben wir in unseren Reporttemplates einfach ein paar EXCEL Diagramme hinzugefügt, welche auf die Grunddaten im EXCEL verweisen. Damit kamen wir zu dem Schluss, dass sich jeder erdenkliche Report mit diesem Konzept umsetzen lässt und die Benutzer auch wirklich selbstständig diese anpassen können. Als Entwickler kann man mit diesem System auch das mühsame Layouten der Berichte und das wiederholte nachbesser und ändern auf die Benutzer auslagern. Damit bleibt einfach mehr Zeit für die interessanten Aufgaben im Projekt.

Kontaktadresse:

Roland Hörmann
SIB Visions GmbH
Wehlistr. 29 / Stiege 1 / 2.Stock
A-1200 Wien
Österreich

Telefon:	+43 1 934 6009 616
Fax:	+43 1 934 6009 999
E-Mail	roland.hoermann@sibvisions.com
Internet:	www.sibvisions.com
Fax:	+49 (0) 12-345 6788
E-Mail	Ihre@adresse.de
Internet:	www.adresse.de