

Workshop OpenOffice.org Java API

Wouter van Reeven

1 JDeveloper configureren voor OpenOffice.org

OpenOffice.org (OOo) kan op twee manieren aangestuurd worden, namelijk lokaal en via een netwerkverbinding. Merk op dat het ook mogelijk is een netwerkverbinding op te zetten met een lokale machine. Aangezien de class room pc's geen OOo installatie hebben, zullen we alle verbindingen via het netwerk doen.

Kopieer de project folder vanaf het netwerk naar een lokale plaats, bijvoorbeeld C:\workshop. Open JDeveloper 10.1.3 en open het project. In de package nl.amis.ooo.helper vind je de class OOoHelper welke algemene aan OOo gerelateerde methods bevat. De method om OOo lokaal te starten (en welke we nu niet gaan gebruiken) heet localBootstrap. De method om een verbinding te maken met een OOo instance die remote draait, gebruik je de remoteConnect method. De remoteConnect method vraagt als parameters de hostnaam en de port waarop OOo luistert.

Voordat we de OOoHelper en de andere classes kunnen gebruiken, moeten we JDeveloper vertellen dat er bepaalde jars zijn die gebruikt moeten worden. Click in JDeveloper op Tools -> Manage Libraries. Maak een nieuwe User Library aan en noem die (bijvoorbeeld) OpenOffice.org. En voeg de vijf jars in C:\workshop\OOoProject\client\lib toe aan het Class Path van deze library.

2 Een eerste connectie

Maak nu een nieuwe class aan in de package nl.amis.ooo.client en noem deze OOoWriterClient (of OOoCalcClient als je met hoofdstuk 7 aan de slag gaat). Voeg de volgende code toe aan deze class

```
public static void main(String[] args) {
    OOoHelper oooHelper = new OOoHelper();
    OOoWriterHelper oooWriterHelper = new OOoWriterHelper();
    OOoCalcHelper oooCalcHelper = new OOoCalcHelper(); // ivm hoofdstuk 7
    try {
        Object desktop = oooHelper.remoteConnect(<host>, <port>);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

waarbij <host> en <port> Strings zijn die respectievelijk de hostnaam en de port hebben waarop OOo luistert. Als je nu de klasse runt lijkt er niets te gebeuren. Maar dat is niet zo. Er wordt namelijk een verbinding gemaakt met OOo die op de opgegeven host draait. Aangezien er verder geen opdrachten aan OOo gegeven worden, zal OOo blijven wachten tot er iets gebeurt. De enige manier om de "applicatie" nu te laten stoppen is deze te terminaten door op het rode vierkantje te klikken. Deze situatie kan voorkomen worden door de regel

```
System.exit(0);
```

toe te voegen in de code. Later komt er een andere, betere manier om dit te doen.

3 Writer

Zoals de naam van de class al doet vermoeden, gaan we met Writer aan de slag. Om dit te kunnen doen, zijn twee stappen nodig:

1. We moeten een document openen of maken
2. We hebben de XTextDocument component hiervan nodig

Merk op dat als het document dat we openen niet een Writer document is, de poging om uit het document een XTextDocument op te vragen wel lukt, maar dat dan een NULL referentie teruggegeven wordt.

Het openen of maken van een document gaat in OOO als volgt. Er moet een URL opgegeven worden welke aangeeft welk document geopend wordt. De URL `private:factory/swriter` geeft aan dat een nieuw Writer document aangemaakt moet worden. In dit geval gaan we uit van een bestaand document. Voeg daarom in `OOoWriterClient` na de `remoteConnect` regel de volgende regels toe

```
XComponent xComponent = oooHelper.getXComponent(desktop,
"file:///C:/OOoPresentation/FieldsTemplate.odt", true, true);
XTextDocument xTextDocument = oooWriterHelper.getXTextDocument(xComponent);
```

De directory `C:\OOoPresentation` bestaat niet op je pc! Maar deze bestaat WEL op de server. Hieruit kun je dus opmaken dat er geen OOO handelingen op je pc uitgevoerd worden.

De signature van de `getXComponent` method in `OOoHelper` laat zien dat de twee booleans "readonly" en "hidden" heten. Er zijn nog veel meer eigenschappen van Writer documenten die gezet kunnen worden. Zie voor een lijst de eerste tabel in

http://api.openoffice.org/docs/DevelopersGuide/OfficeDev/OfficeDev.xhtml#1_1_5_Handling_Documents

Als je in je code "hidden" op false zet, zul je op de host je document kunnen bekijken. de readonly mode zorgt er alleen voor dat zichtbare documenten niet met een toetsenbord bewerkt kunnen worden. Met de Java API kan het dan nog wel.

Zelfs er een `System.exit(0);` in je code staat zal het document dat geopend wordt open blijven in OOO! Elk document moet actief gesloten worden. Om dit te doen kun je voor de `System.exit(0);` regel deze regel zetten

```
oooHelper.closeDocument(xComponent);
```

De `System.exit(0);` is alleen onder Windows nodig. Ik weet niet waarom, maar onder *nix eindigt de applicatie op het moment dat het OOO document gesloten wordt.

4 Het opslaan, exporteren en printen van documenten

OOO ondersteunt naast de OOO documentformaten ook veel externe formaten, onder andere van Microsoft Office. Let op dat er zekere compatibiliteitsproblemen bestaan tussen OOO en MsO. Een document kan opgeslagen worden door de `XStorable` component uit het document te halen en deze de bestandsnaam en het bestandstype mee te geven. Het exporteren van een document naar PDF heeft nog een extra stap nodig, namelijk het aangeven van een `CompressMode`. Zonder deze `CompressMode` zal het exporteren naar PDF in een exception resulteren!

Voeg de volgende code toe voor het sluiten van het document om het bestand op te slaan als Word document

```
oooHelper.storeDocComponent(xComponent,
"file:///C:/ooo/FieldsTest.doc", "MS Word 97");
```

Pas de naam van je document naar iets unieks aan zodat jouw document niet door anderen overschreven wordt. Om te exporteren naar PDF gebruik je

```
oooHelper.exportPdfDocComponent(xComponent, "file:///C:/ooo/Certificate.pdf");
```

met dezelfde kanttekening voor de bestandsnaam.

Om te printen heb je de XPrintable component van het document nodig. Voor het printen van een document is het nodig twee property collecties te specificeren. De eerste collectie bevat de properties voor de printer, de tweede die voor de printtaak.

Door de property Name de waarde van de printernaam te geven en deze met setPrinter op de XPrintable component te zetten, kun je aangeven welke printer gebruikt moet worden. De naam moet een bekende naam in OOo zijn, anders werkt het niet maar wordt geen foutmelding gegeven. Met de properties van de printtaak kun je bijvoorbeeld aangeven welke pagina's geprint kunnen worden. Voor je code is het voldoende deze regel toe te voegen

```
oooHelper.printDocComponent(xComponent, printernaam);
```

Hierbij is printernaam de naam van de printer. Merk op dat in een Java String het "\" teken een escape teken is. Om een "\" te krijgen moet je "\\" opgeven.

5 Cursors

Tekstoperaties worden in OOo gedaan met behulp van cursors. OOo kent twee typen cursors, namelijk zichtbare en onzichtbare. er is EEN type zichtbare cursor en dat is de cursor waar je tekst kunt typen. Als een document wordt geopend als readonly, is deze cursor uitgeschakeld.

De typen onzichtbare cursors zijn XTextRange, XTextCursor, XWordCursor, XSentenceCursor en XParagraphCursor. Deze cursors inheriten van elkaar. Een XTextRange kan de cursor naar het begin of eind van een tekst bewogen worden, met een XTextCursor kan dit ook gebeuren en ook de tekst selecteren. Met de andere cursors kan dit ook per entiteit (word, sentence of paragraph). Het feit dat de cursors inheriten betekent dat met een XWordcursor ook losse karakters geselecteerd kunnen worden en met een XParagraphCursor ook losse woorden of zinnen.

Zodra met een cursor een stuk tekst geselecteerd is kunnen de properties van deze tekst aangepast worden. We willen nu het font van de tekst aanpassen. Dit kan gedaan worden door de tekst property CharFontName de juiste waarde te geven. Voeg deze code toe voor het opslaan van het document

```
XTextCursor xTextCursor = oooWriterHelper.getXTextCursor(xTextDocument);  
xTextCursor.gotoStart(false);  
xTextCursor.gotoEnd(true);  
XPropertySet xPropertySet = oooHelper.getXPropertySet(xTextCursor);  
xPropertySet.setPropertyValue("CharFontName", "Brush Script MT");
```

Merk op dat het lettertype "Brush Script MT" bekend moet zijn bij OOo.

Om de kleur van de karakters aan te passen kun je de volgende code gebruiken

```
XWordCursor xWordCursor = oooWriterHelper.getXWordCursor(xTextDocument);  
xWordCursor.gotoStart(false);  
xWordCursor.gotoEndOfWord(true);  
XPropertySet = oooHelper.getXPropertySet(xWordCursor);  
XPropertySet.setPropertyValue("CharColor", 0xe6e64c);
```

Het hexadecimale getal voor de kleur bestaat uit drie hexadecimale paren welke de waarden voor Rood, Groen en Blauw aangeven. Zo is 0x000000 zwart en 0xfffffff wit. Fel blauw is 0x0000ff enzovoorts.

Ook karakters en paragrafen hebben flink wat eigenschappen die gezet kunnen worden. Zie

http://api.openoffice.org/docs/DevelopersGuide/Text/Text.xhtml#1_3_2_Formatting

6 Pagina's

Ook pagina's hebben veel eigenschappen die via de API gezet kunnen worden. Zo kan de achtergrondkleur ingesteld worden en of een pagina landscape of portrait is. Bijvoorbeeld als volgt

```
oooWriterHelper.setBackgroundColor(xTextDocument, 0x00dcff);  
oooWriterHelper.setLandscape(xTextDocument);
```

Mooi he :-) Zie voor (zeer uitgebreide info)

http://api.openoffice.org/docs/DevelopersGuide/Text/Text.xhtml#1_4_Overall_Document_Features

7 Calc

Zorg voor dat je aan dit onderdeel begint dat je de stappen in "JDeveloper configureren voor OOo" (hoofdstuk 1) en "Een eerste connectie" (hoofdstuk 2) hebt gevolgd.

Zoals de naam van de class al doet vermoeden, gaan we met Calc aan de slag. Om dit te kunnen doen, zijn weer twee stappen nodig:

3. We moeten een document openen of maken
4. We hebben de XSpreadsheetDocument component hiervan nodig

Merk op dat als het document dat we openen niet een Calc document is, de poging om uit het document een XSpreadsheetDocument op te vragen wel lukt, maar dat dan een NULL referentie teruggegeven wordt.

Het openen of maken van een document gaat in OOo als volgt. Er moet een URL opgegeven worden welke aangeeft welk document geopend wordt. De URL `private:factory/scalc` geeft aan dat een nieuw Calc document aangemaakt moet worden.

Een spreadsheet component in OOo bestaat (net als in Excel overigens) uit twee onderdelen. Ten eerste is er het spreadsheet document, ten tweede de losse sheets. De losse sheets kunnen in OOo uit het document gehaald worden met een tussenstap, namelijk de verzameling van spreadsheets. Gebruik de volgende code hiervoor

```
XComponent xComponent = oooHelper.getXComponent(desktop, oooHelper.CALC, false,  
false);  
XSpreadsheetDocument xSpreadsheetDocument =  
oooCalcHelper.getXSpreadsheetDocument(xComponent);  
XSpreadsheets xSpreadsheets =  
oooCalcHelper.getXSpreadsheets(xSpreadsheetDocument);  
String[] elementNames = xSpreadsheets.getElementNames();  
for (int i = 0; i < elementNames.length; i++) {  
    System.out.println("Found sheet with name " + elementNames[i]);  
}
```

```
XSpreadsheet xSpreadsheet =
oooCalcHelper.getXSpreadsheetByName(xSpreadsheetDocument, "Sheet1");
oooHelper.closeDocument(xComponent);
System.exit(0);
```

Hierbij is `OOoHelper.CALC` een String constante voor de URL voor een lege spreadsheet.

BELANGRIJK: Bij het opstellen van dit document ondervond ik dat als een Spreadsheet in readonly mode wordt aangemaakt, deze NIET met de Java API bewerkt kan worden! Zet daarom hier readonly op false! Hidden heeft geen invloed en mag dus true zijn.

Deze code print alle namen van de sheets in een nieuwe spreadsheet uit en vraagt de sheet met naam "Sheet1" op. Er geldt, net als in hoofdstuk 3, dat het document actief gesloten moet worden en dat de applicatie actief beëindigd moet worden.

Voeg de volgende code toe voor het sluiten van het document om het bestand op te slaan als Excel document

```
oooHelper.storeDocComponent(xComponent,
    "file:///C:/ooo/Spreadsheet.xls", "MS Excel 97");
```

Pas de naam van je document naar iets unieks aan zodat jouw document niet door anderen overschreven wordt. Om te exporteren naar PDF gebruik je

```
oooHelper.exportPdfDocComponent(xComponent, "file:///C:/ooo/Spreadsheet.pdf");
```

met dezelfde kanttkening voor de bestandsnaam.

8 Cellen

Een cell in Calc kan verschillende typen content hebben. Het default type is Number (int, double etc) maar het kan met wat extra handelingen ook tekst, een datum of een formule zijn. Verder is het mogelijk de formattering van de cellen aan te passen. In dit voorbeeld zullen we de formattering van de data in de cellen niet aanpassen, maar wel de kleuren van tekst en cellen en ook de borders van de cellen.

Eerst maar eens wat inhoud toevoegen. Cellen kunnen gevonden worden door te tellen van linksboven naar rechts en onder. De eerste cell bevindt zich op (0,0), de cell ernaast op (0,1) en de cell daaronder op (1,1) enzovoorts. De volgende code voegt wat teksten toe aan cellen om de inhoud van de data te verduidelijken:

```
oooCalcHelper.setTextValueOfXCellAtPosition(xSpreadsheet, 0, 0, "Number");
oooCalcHelper.setTextValueOfXCellAtPosition(xSpreadsheet, 0, 1, "One");
oooCalcHelper.setTextValueOfXCellAtPosition(xSpreadsheet, 0, 2, "Two");
oooCalcHelper.setTextValueOfXCellAtPosition(xSpreadsheet, 0, 3, "Three");
oooCalcHelper.setTextValueOfXCellAtPosition(xSpreadsheet, 0, 4, "Four");
oooCalcHelper.setTextValueOfXCellAtPosition(xSpreadsheet, 0, 5, "Total");
oooCalcHelper.setTextValueOfXCell(oooCalcHelper.getXCellByPosition(xSpreadsheet,
1, 0), "Int");
```

In de cellen in de tweede rij zetten we wat getallen neer

```
for (int i = 0; i < 5; i++) {
oooCalcHelper.setNumValueOfXCellAtPosition(xSpreadsheet, 1, i+1, new
Integer(i).doubleValue());
}
```

In de cell rechtsonder zetten we een formule neer

```
oooCalcHelper.setFormulaOfXCellAtPosition(xSpreadsheet, 1, 5, "=SUM(B1:B4)");
```

Door in de for-loop het getal dat ingevuld wordt te wijzigen, bijvoorbeeld in `Integer(i+19)`, kun je zien dat de optelsom goed werkt.

Een andere manier om cellen te benaderen is door middel van `XCellRange`. Dit is een blok cellen waarvan tegelijk de inhoud of opmaak aangepast kan worden. De volgende code, bijvoorbeeld, past de kleur en weight van het font van de bovenste twee cellen aan

```
XCellRange xCellRange = oooCalcHelper.getXCellRangeByPositions(xSpreadsheet, 0,
0, 1, 0);
XPropertySet xPropertySet = oooHelper.getXPropertySet(xCellRange);
xPropertySet.setPropertyValue("CharWeight", new
Float(com.sun.star.awt.FontWeight.BOLD));
xPropertySet.setPropertyValue("CharColor", 0x993300);
```

en deze code die van de onderste twee cellen

```
xCellRange = oooCalcHelper.getXCellRangeByPositions(xSpreadsheet, 0, 5, 1, 5);
xPropertySet = oooHelper.getXPropertySet(xCellRange);
xPropertySet.setPropertyValue("CharWeight", new
Float(com.sun.star.awt.FontWeight.BOLD));
xPropertySet.setPropertyValue("CellBackColor", 0x55ff55);
```

Zie verder

http://api.openoffice.org/docs/DevelopersGuide/Spreadsheet/Spreadsheet.xhtml#1_3_2_Formatting

9 Borders

OOo kent de klasse `TableBorder`. Deze klasse kan gebruikt worden om de border van cellen en tables aan te passen. Bekijk de code voor `OOoCalcHelper.createBorder` over een voorbeeld hiervoor. Om de border van de bovenste twee rijen in de tabel aan te passen, kun je de volgende code toevoegen NA de code die het font in de bovenste twee rijen aanpast en VOOR de code die het font van de onderste rijen aanpast

```
xPropertySet.setPropertyValue("TableBorder",
oooCalcHelper.createBorder(0x000000, false, false, true, false));
```

Deze code doet iets vergelijkbaars met de onderste rijen en moet dus NA de code voor de onderste rijen worden toegevoegd

```
xPropertySet.setPropertyValue("TableBorder",
oooCalcHelper.createBorder(0xaaaaaa, true, false, false, false));
```