relatorio Documentation

Release 0.8.1

Nicolas Évrard, CGaëtan de Menten, Cédric Krier

Dec 04, 2017

Contents

Contents:

CHAPTER 1

Quick Example

In this page we will show you how you can create OpenOffice documents using Relatorio.

1.1 Data

We need some data objects to work on, so let's first create a fake invoice object. Please create a file named `data.py` in your favorite text editor, and copy the following content:

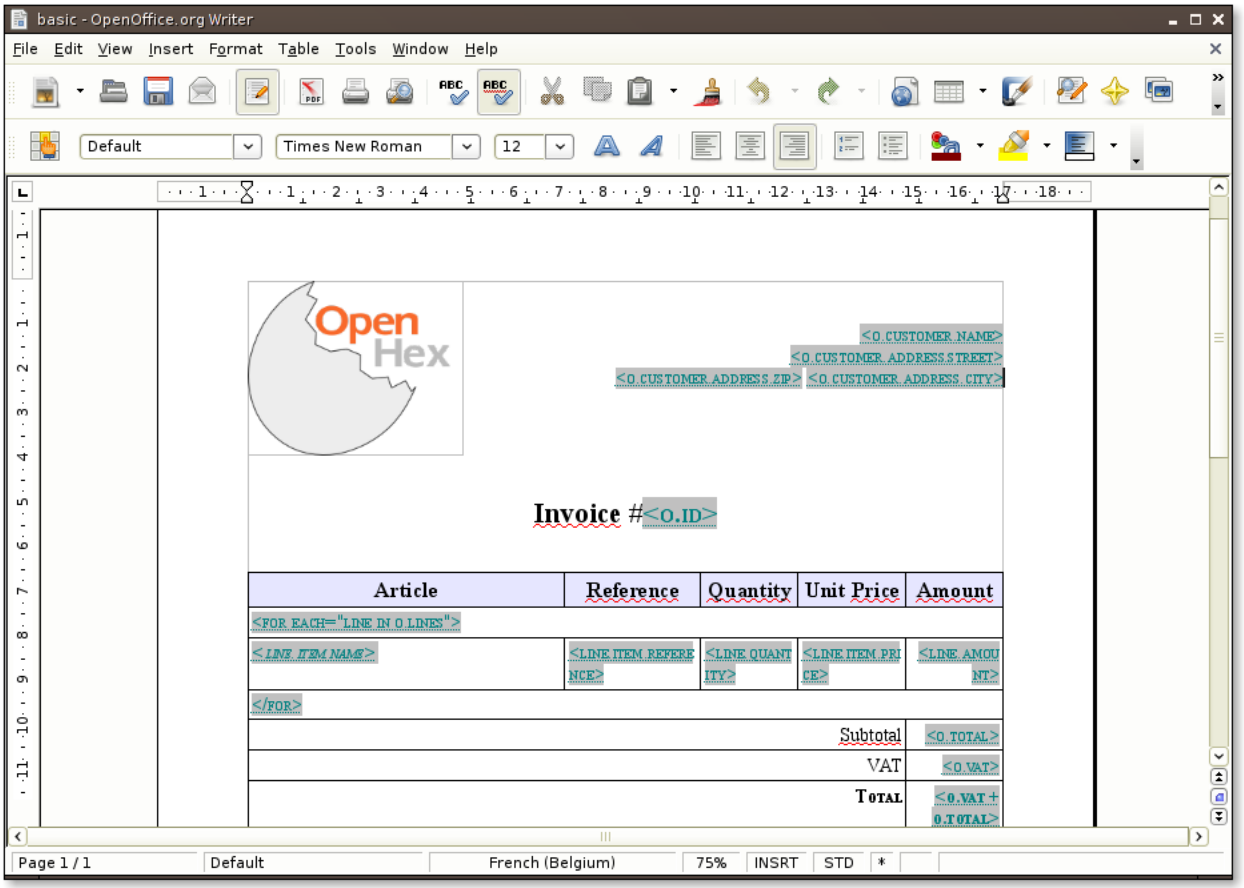
```
from os.path import join, dirname


class Invoice(dict):

    @property
    def total(self):
        return sum(l['amount'] for l in self['lines'])

    @property
    def vat(self):
        return self.total * 0.21


inv = Invoice(customer={'name': 'John Bonham',
                        'address': {'street': 'Smirnov street',
                                    'zip': 1000,
                                    'city': 'Montreux'}},
              lines=[{'item': {'name': 'Vodka 70cl',
                               'reference': 'VDKA-001',
                               'price': 10.34},
                      'quantity': 7,
                      'amount': 7 * 10.34},
                     {'item': {'name': 'Cognac 70cl',
                               'reference': 'CGNC-067',
                               'price': 13.46},
```

```
                    'quantity': 12,
                    'amount': 12 * 13.46},
                   {'item': {'name': 'Sparkling water 25cl',
                             'reference': 'WATR-007',
                             'price': 4},
                    'quantity': 1,
                    'amount': 4},
                   {'item': {'name': 'Good customer',
                             'reference': 'BONM-001',
                             'price': -20},
                    'quantity': 1,
                    'amount': -20},
                  ],
           id='MZY-20080703',
           status='late',
           bottle=(open(join(dirname(__file__), 'bouteille.png'), 'rb'),
               'image/png'))
```

So we created the data for an invoice for the famous Led Zeppelin's drummer and his favorite addiction.

1.2 Template

The next thing to do is to create a template for invoices. We will use the one displayed below. To create the Genshi directives, you need to create a text-type placeholder field, and fill it with the expression you want to use.

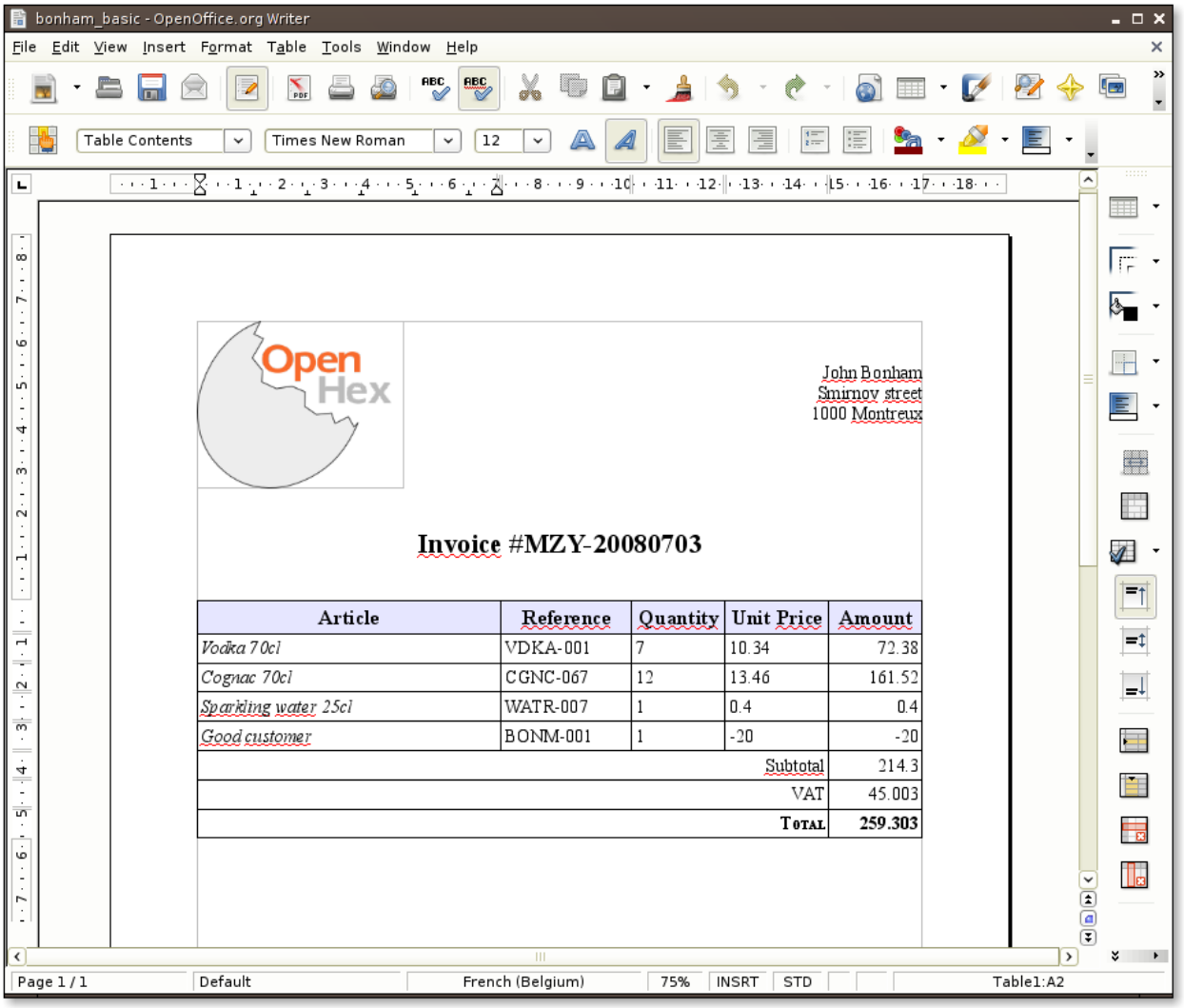


1.3 Generate the final documents

Now that we have both a template and some data, we can now start to use Relatorio to create John Bonham's particular invoice. So fire up your favorite python interpreter (we suggest using IPython) and type (or copy-paste) the following commands:

```
from relatorio.templates.opendocument import Template
from data import bonham_invoice
basic = Template(source='', filepath='basic.odt')
basic_generated = basic.generate(o=bonham_invoice).render()
file('bonham_basic.odt', 'wb').write(basic_generated.getvalue())
```

On the first line we import the opendocument Template engine. This class has the same signature as the one from Genshi but uses only the filepath argument. On the fourth line, we generate the final document from the template and the data. Note how we pass *o=bonham_invoice* as argument to generate. This is the same "o" variable as was used in the O dt template we just created. render() returns us a StringIO object, which is then used to pipe the result to a file.



And so here is our invoice with all the fields completed according to the Invoice object we created earlier. Notice how the style we set in the template are also applied in the resulting invoice.

In this example, we only used the *py:for* directive, but Relatorio also supports other Genshi directives: *py:if*, *py:choose*

/ *py:when* / *py:otherwise* and *py:with*.

CHAPTER 2

Indepth Example

In this page I will detail the way I created the reports that can be found in the examples directory.

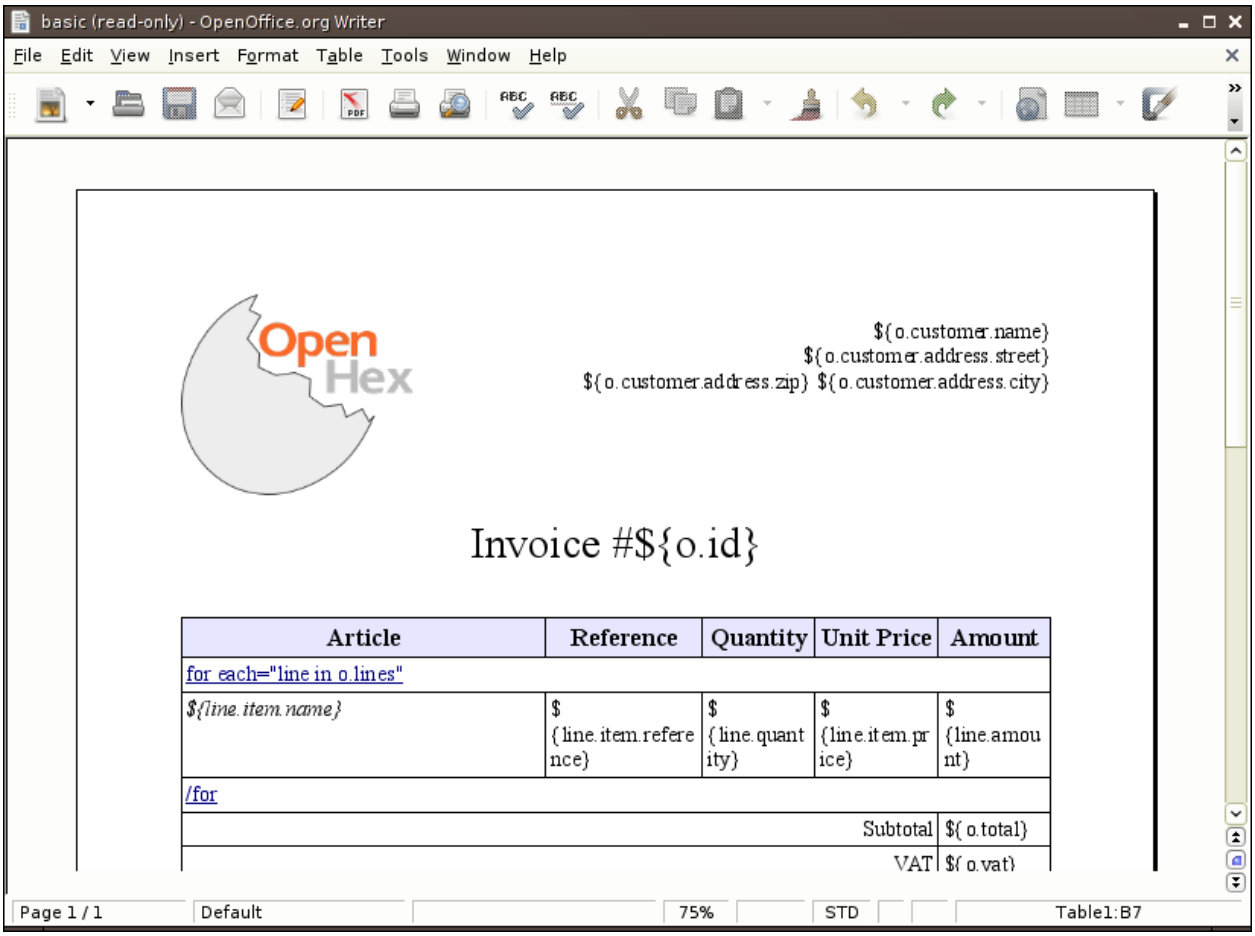
Let's start with the content of `common.py`, this file stores the definition of an invoice that will be used to create the different reports. The invoice is a simple python dictionary with some methods added for the sake of simplicity:

```
from os.path import join, dirname


class Invoice(dict):

    @property
    def total(self):
        return sum(l['amount'] for l in self['lines'])

    @property
    def vat(self):
        return self.total * 0.21


inv = Invoice(customer={'name': 'John Bonham',
                        'address': {'street': 'Smirnov street',
                                    'zip': 1000,
                                    'city': 'Montreux'}},
              lines=[{'item': {'name': 'Vodka 70cl',
                               'reference': 'VDKA-001',
                               'price': 10.34},
                      'quantity': 7,
                      'amount': 7 * 10.34},
                     {'item': {'name': 'Cognac 70cl',
                               'reference': 'CGNC-067',
                               'price': 13.46},
                      'quantity': 12,
                      'amount': 12 * 13.46},
                     {'item': {'name': 'Sparkling water 25cl',
                               'reference': 'WATR-007',
```

```
                        'price': 4},
                    'quantity': 1,
                    'amount': 4},
                   {'item': {'name': 'Good customer',
                             'reference': 'BONM-001',
                             'price': -20},
                    'quantity': 1,
                    'amount': -20},
                  ],
        id='MZY-20080703',
        status='late',
        bottle=(open(join(dirname(__file__), 'bouteille.png'), 'rb'),
            'image/png'))
```

2.1 Create a simple OpenOffice Writer template

Let's start with the simple template defined in `basic.odt`.

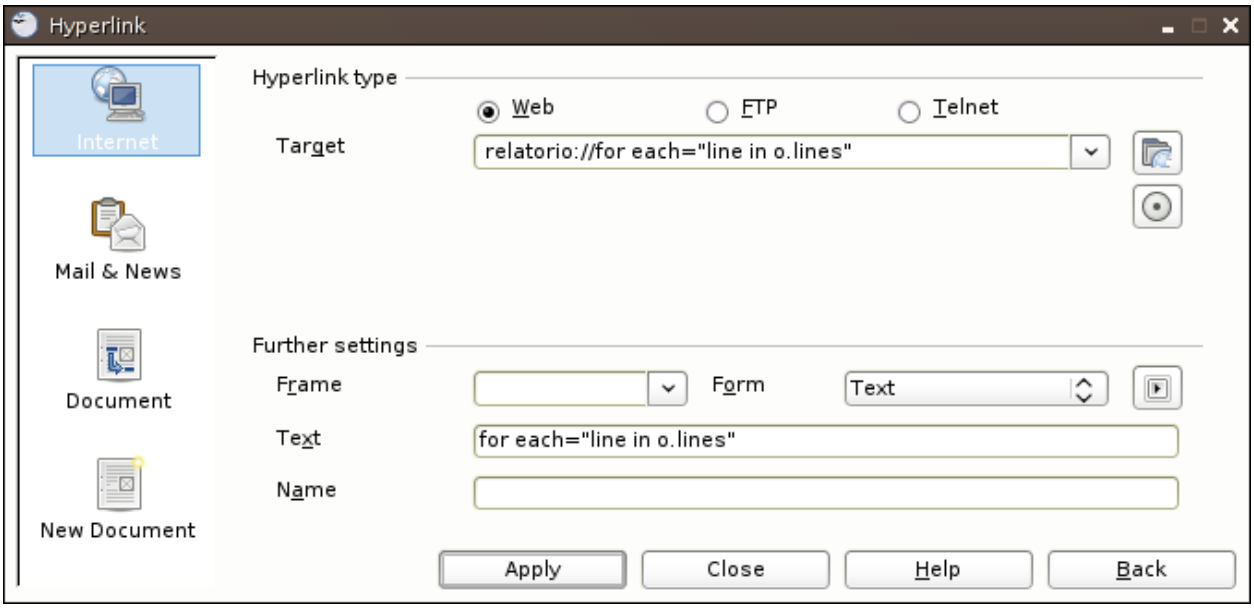


This report will be created and rendered with the following three line of code:

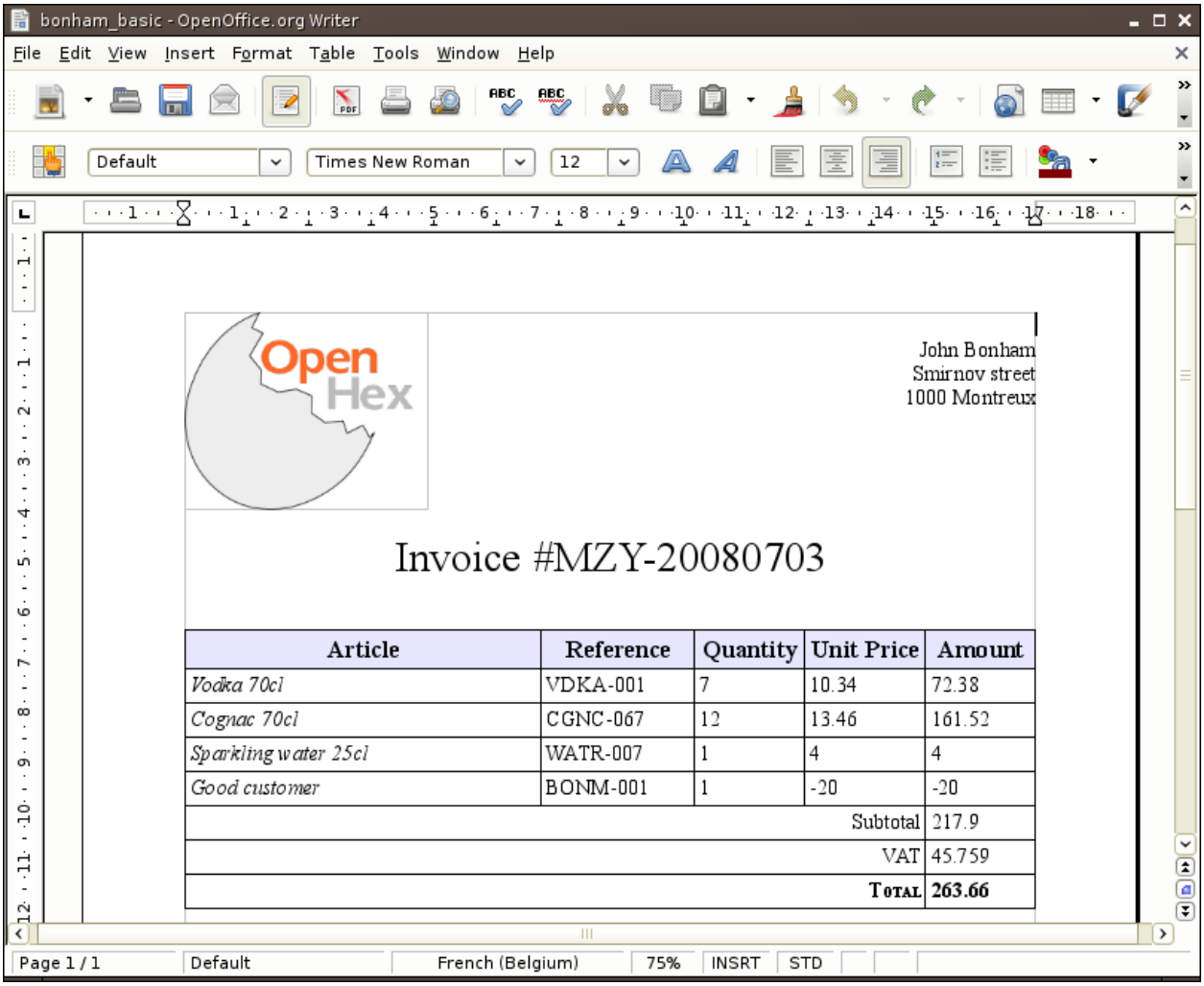
```
from relatorio.templates.opendocument import Template
basic = Template(source='', filepath='basic.odt')
file('bonham_basic.odt', 'wb').write(basic.generate(o=inv).render().getvalue())
```

Notice that the dictionary passed to *generate* is used to bind names to make them accessible to the report. So you can access the data of the invoice with a Text Placeholder containing *o.customer.name*. This is where you can see our Genshi heritage. In fact, all reports using relatorio are subclasses of Genshi's Template. Thus you can use most of the goodies provided by Genshi.

To iterate over a list you must use an hyperlink (created through 'Insert > Hyperlink') and encode as the target the Genshi expression to use. The URL-scheme used *must* be *relatorio*. You can use whatever text you want as the link text, but we find it way more explicit to display the Genshi code used. Here is the example of the for loop.

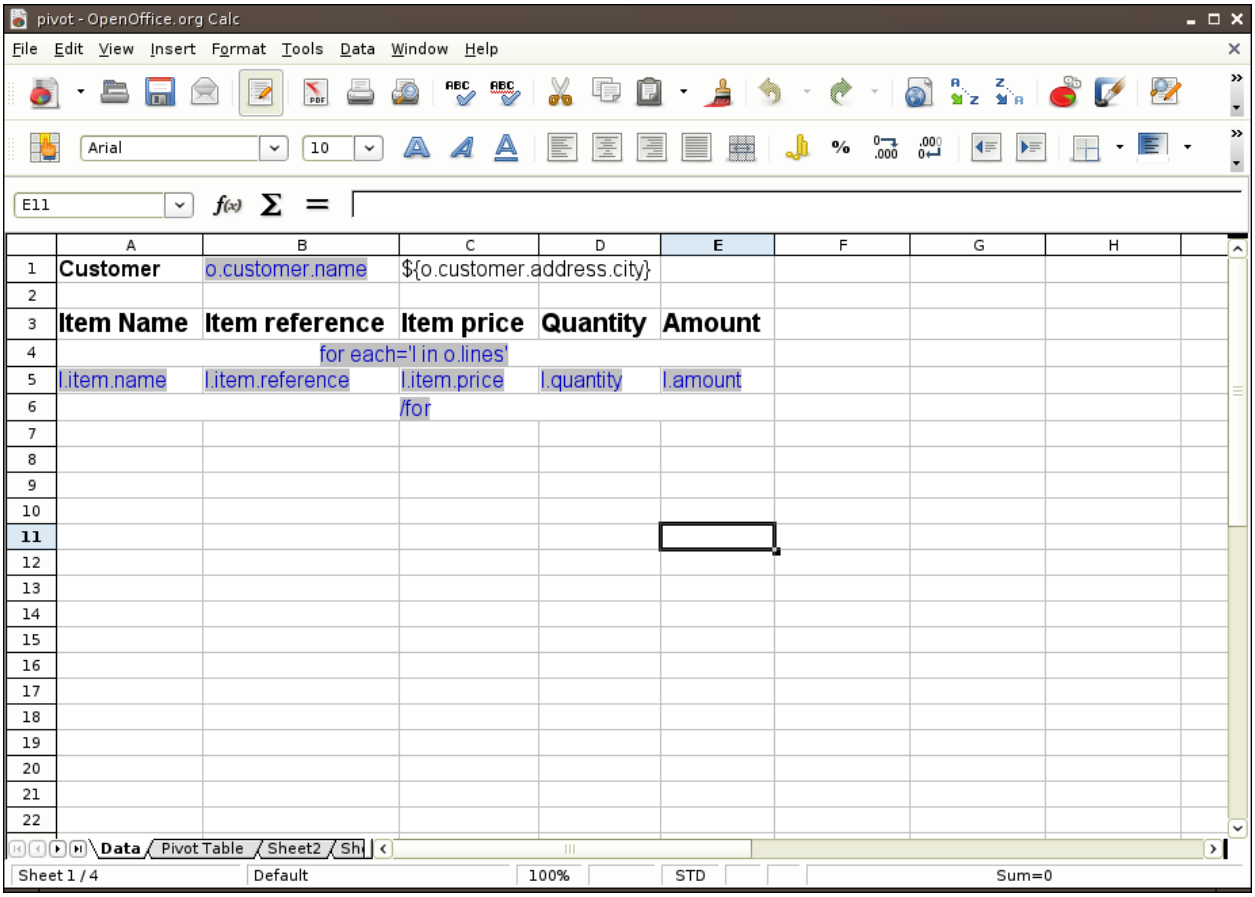


And thus here is our invoice, generated through relatorio:



2.2 One step further: OpenOffice Calc and OpenOffice Impress templates

Just like we defined a Writer template it is just as easy to define a Calc/Impress template. Let's take a look at `pivot.ods`.

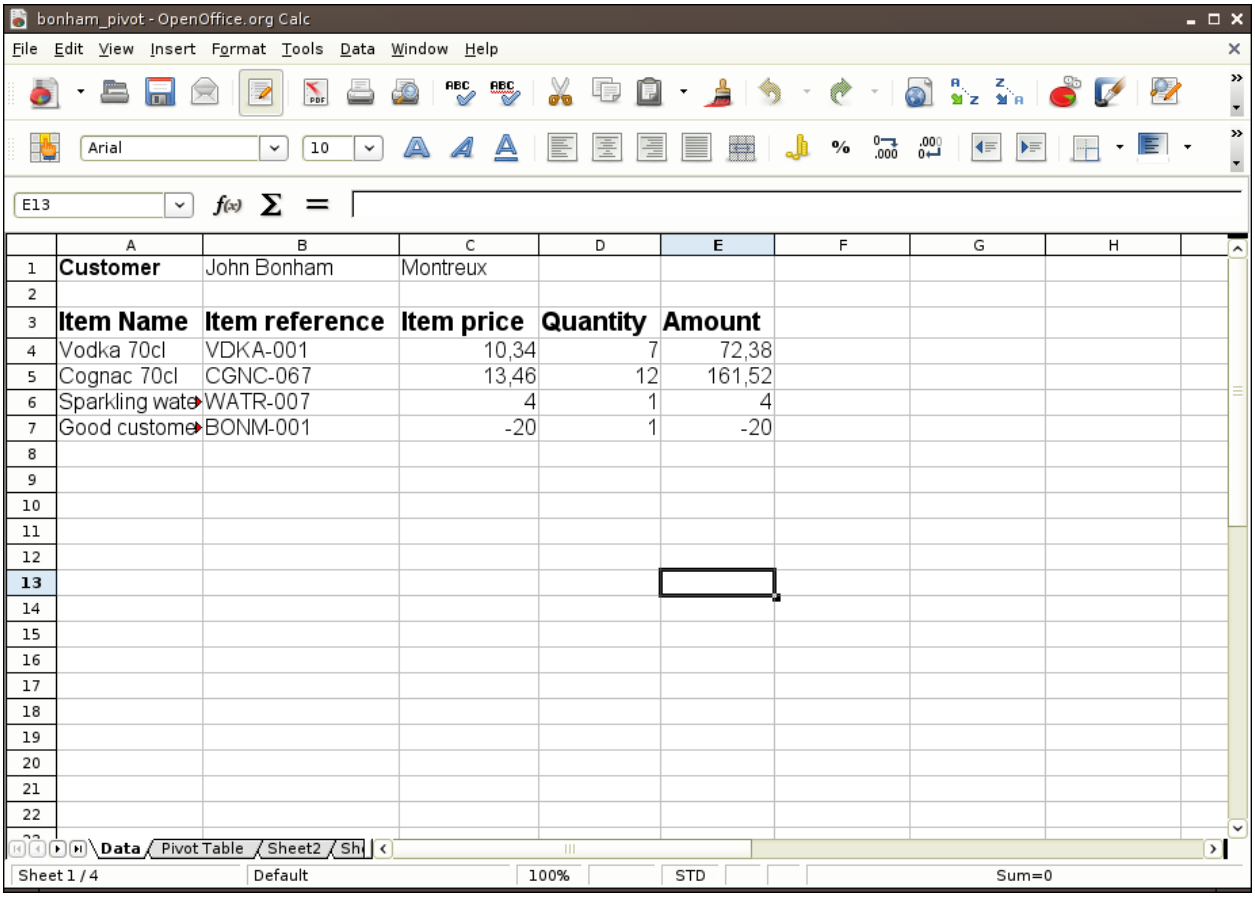


As usual you can see here the different way to make a reference to the content of the invoice object:

- through the Text Placeholder interpolation of Genshi
- or through the hyperlink specification I explained earlier.

Note that there is another tab in this Calc file used to make some data aggregation thanks to the data pilot possibilities of OpenOffice.

And so here is our rendered template:



Note that the type of data is correctly set even though we did not have anything to do.

2.3 Everybody loves charts

Now we would like to make our basic report a bit more colorful, so let's add a little chart. We are using PyCha to generate them from our `pie_chart` template:

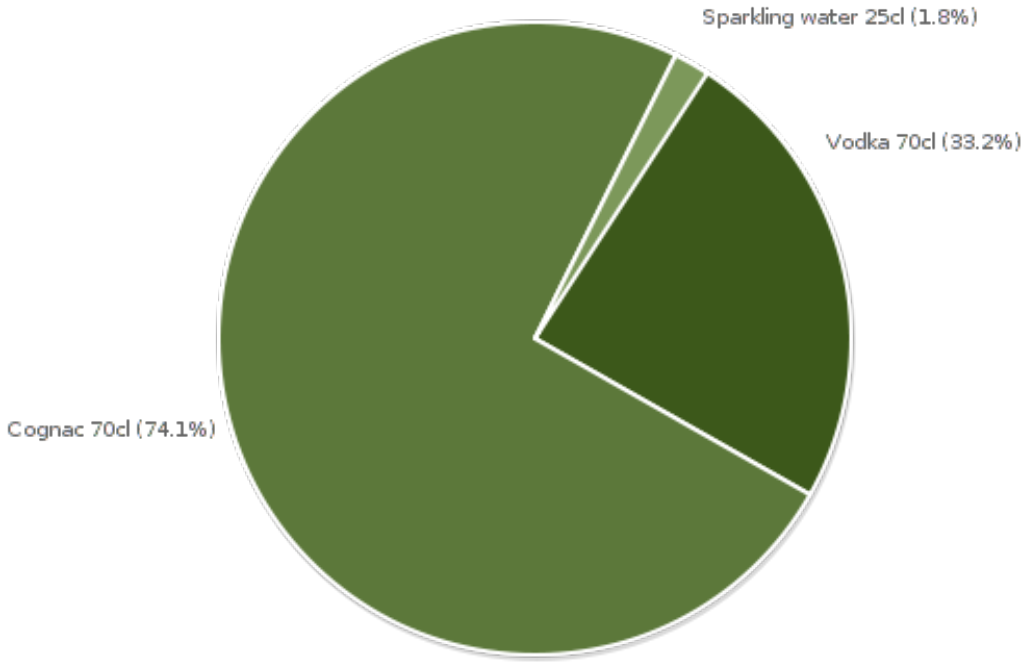
```
options:
    width: 600
    height: 400
    background: {hide: true}
    legend: {hide: true}
    axis: {labelFontSize: 14}
    padding: {bottom: 10, left: 10, right: 10, top: 10}
chart:
    type: pie
    output_type: png
    dataset:
    {% for line in o.lines %}
      - - ${line.item.name}
        - - [0, $line.amount]
    {% end %}
```

Once again we are using the same syntax as Genshi but this time this is a TextTemplate. This file follow the YAML format thus we can render it into a data structure that will be sent to PyCha:

- the options dictionary will be sent to PyCha as-is
- the dataset in the chart dictionary is sent to PyCha through its *.addDataset* method.

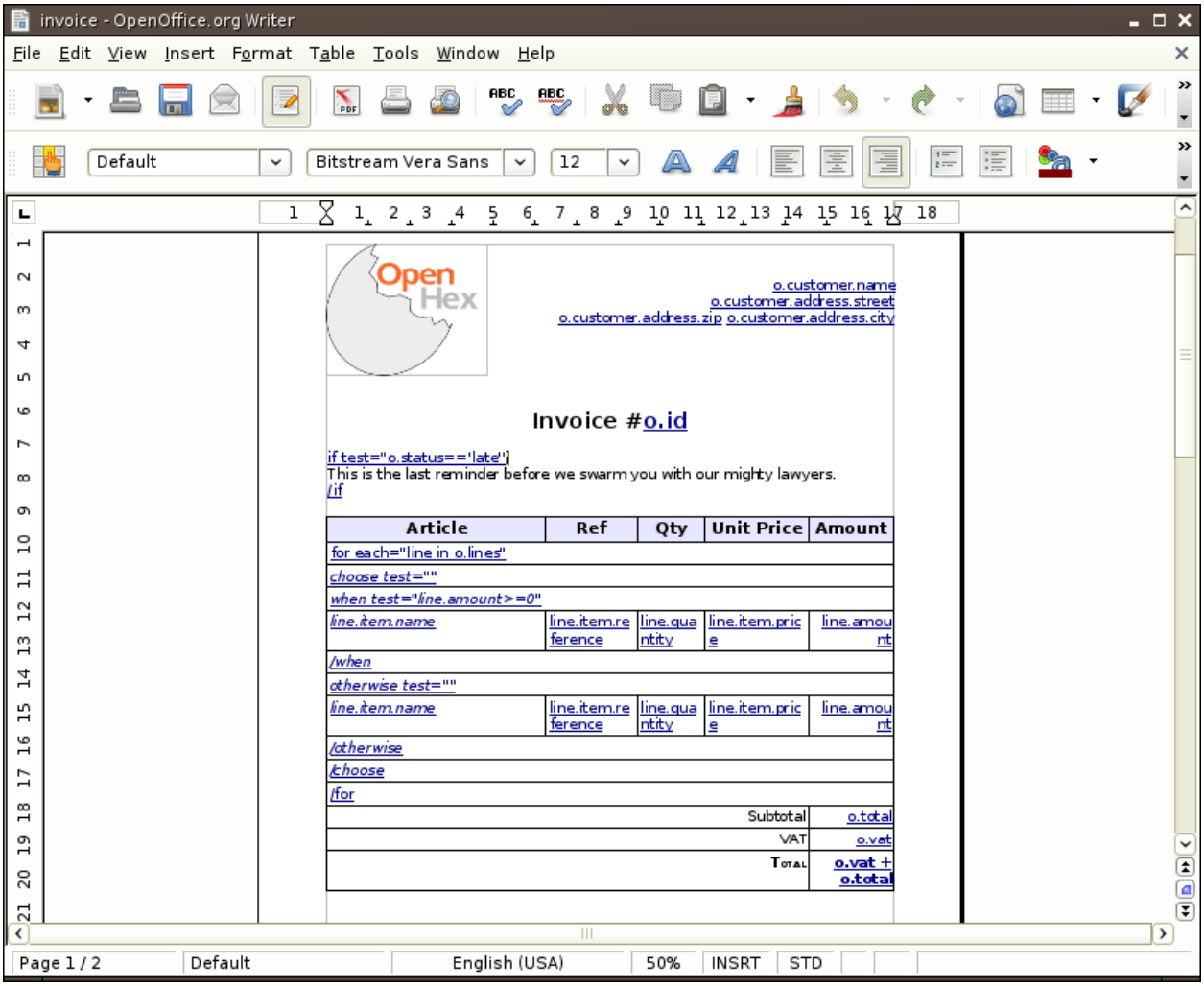
If you need more information about those go to the pycha website.

And here is the result:



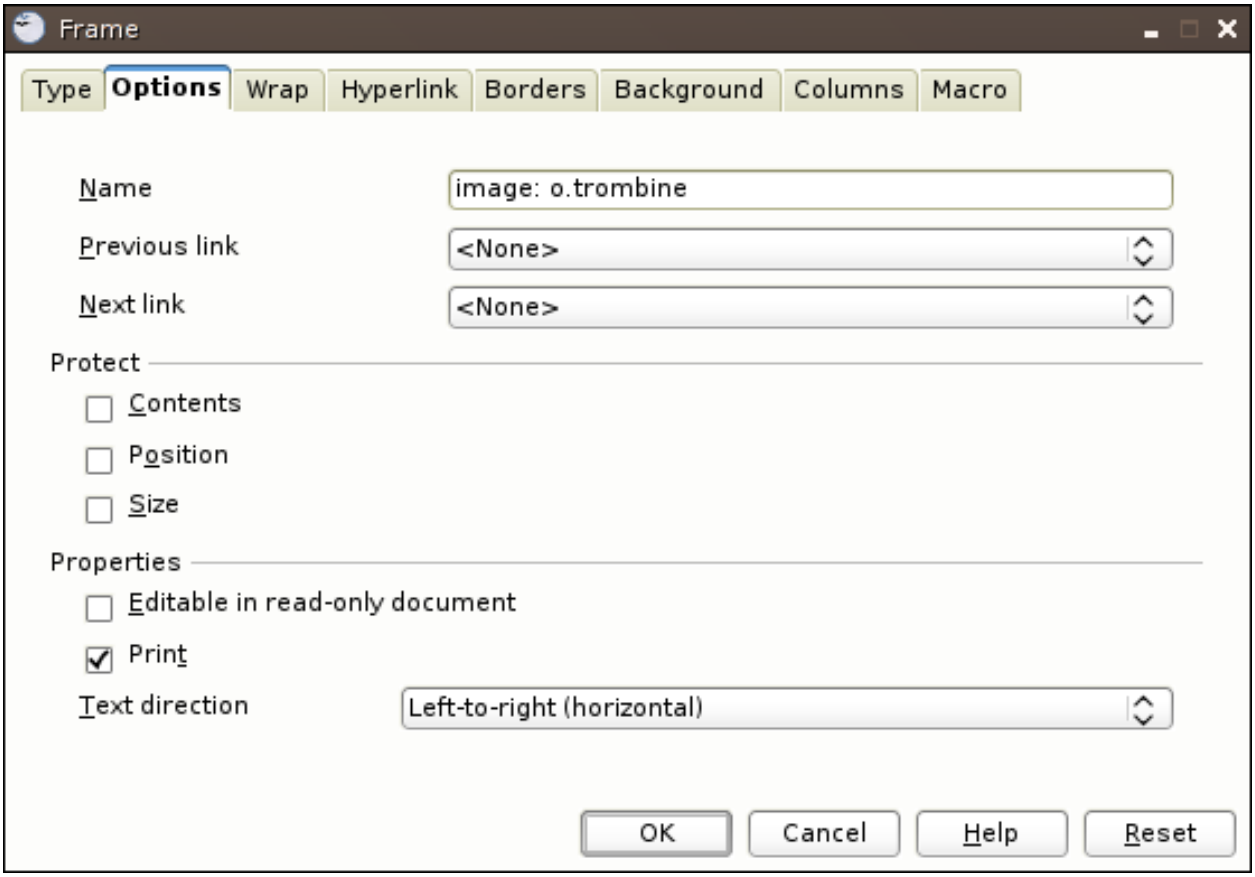
2.4 A (not-so) real example

Now that we have everything to start working on our complicated template `invoice.odt`, we will go through it one step at a time.



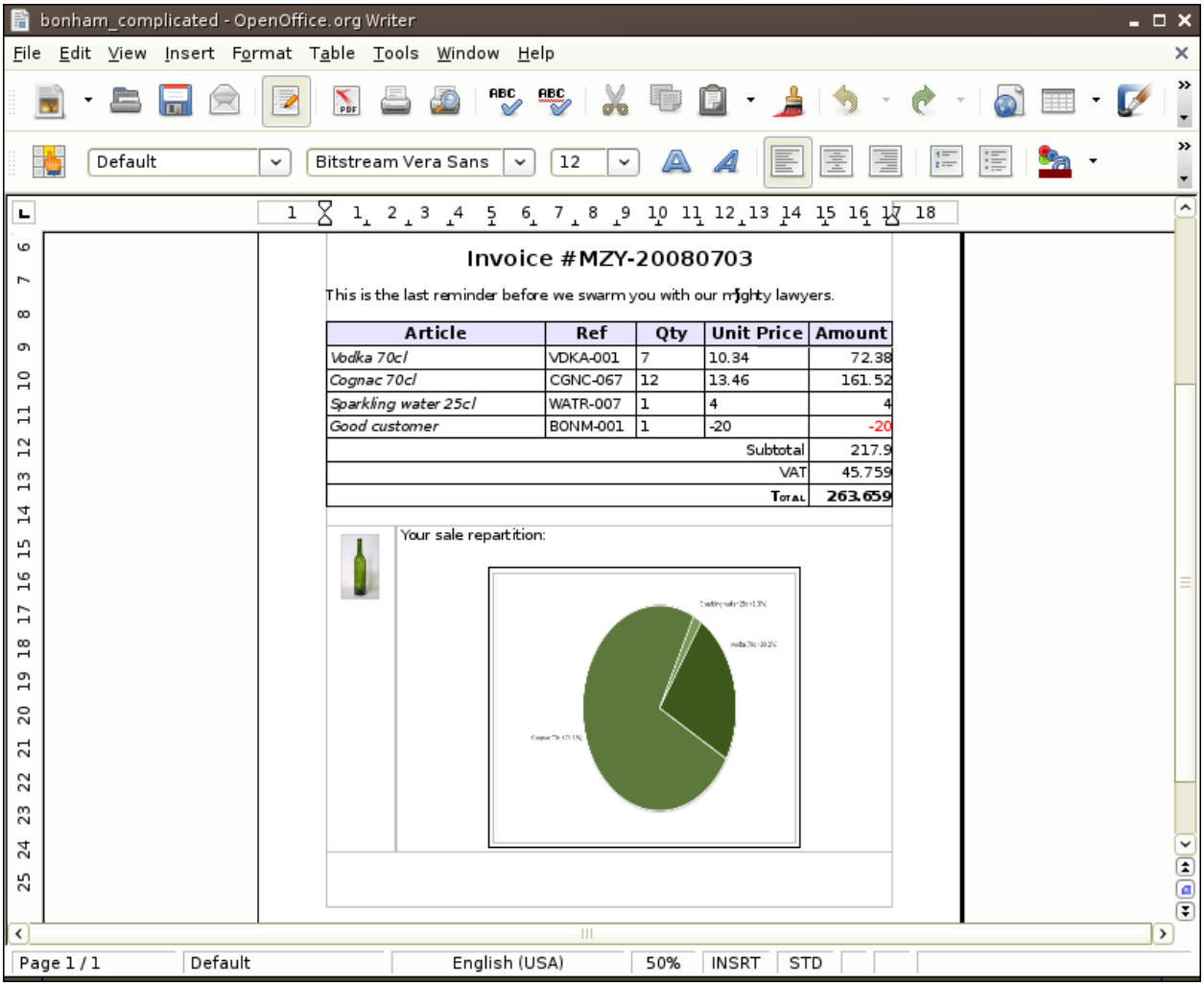
In this example, you can see that not only the openoffice plugin supports the *for directive*, it also supports the *if directive* and the *choose directive* that way you can choose to render or not some elements.

The next step is to add images programmatically, all you need to do is to create frame ('Insert > Frame') and name it *image: expression* just like in the following example:



The expression when evaluated must return a couple whose first element is a file object containing the image and second element is its mimetype. Note that if the first element of the couple is an instance of relatorio report then this report is rendered (using the same arguments as the originating template) and used as a the source for the file definition.

This kind of setup gives us a nice report like that:


bonham_complicated - OpenOffice.org Writer
File Edit View Insert Format Table Tools Window Help
Default
Bitstream Vera Sans
12
Invoice #MZY-20080703
This is the last reminder before we swarm you with our mighty lawyers.
Article
Ref
Qty
Unit Price
Amount
Vodka 70cl
VDKA-001
7
10.34
72.38
Cognac 70cl
CGNC-067
12
13.46
161.52
Sparkling water 25cl
WATR-007
1
4
4
Good customer
BONM-001
1
-20
-20
Subtotal
217.9
VAT
45.759
Total
263.659
Your sale repartition:
Page 1 / 1
Default
English (USA)
50%
INSRT
STD