



JSON data +
RML template =

PDF report

Stefano Cotta Ramusino
<whitone@gmail.com>
2011.06.23



Target

Request:

create a PDF
starting from data
formatted with a “flexible” layout

Requirements:

limited disk space (no LaTeX)



What's “flexible”?

No hope:

no one knows when layout reaches the final version, if it ever reaches one

Error-free:

handle some missing input data

Dynamic:

graphics generated on the fly

Python PDF Report Engines



ReportLab

rst2pdf

pisa (xhtml2pdf)

Sphinx with rst2pdf and rst2latex

PyReport with LaTeX

matplotlib (PdfPages method)

pod with LibreOffice in server mode



ReportLab

Canvas
matryoshka mode

Platypus
(Page Layout and Typography Using Scripts)

RML
(Report Markup Language)



rml2pdf

ReportLab PLUS (commercial)

z3c.rml (Zope Community)



RML

Learning path:

RML For Idiots

RML Samples

RML User Guide

z3c.RML Reference



RML right now

odt2rml

based on sxw2rml
(Tiny/OpenERP)



pyjon.reports

Create PDF in Python
using Genshi and RML

by

Jonathan Schemoul



Genshi's power

template language

XInclude support



Ready to use

```
pip install pyjon.reports pil
```

installs also:

ReportLab
z3c.rml
Genshi
pyPdf



Data input format

JSON

included in Python 2.6+

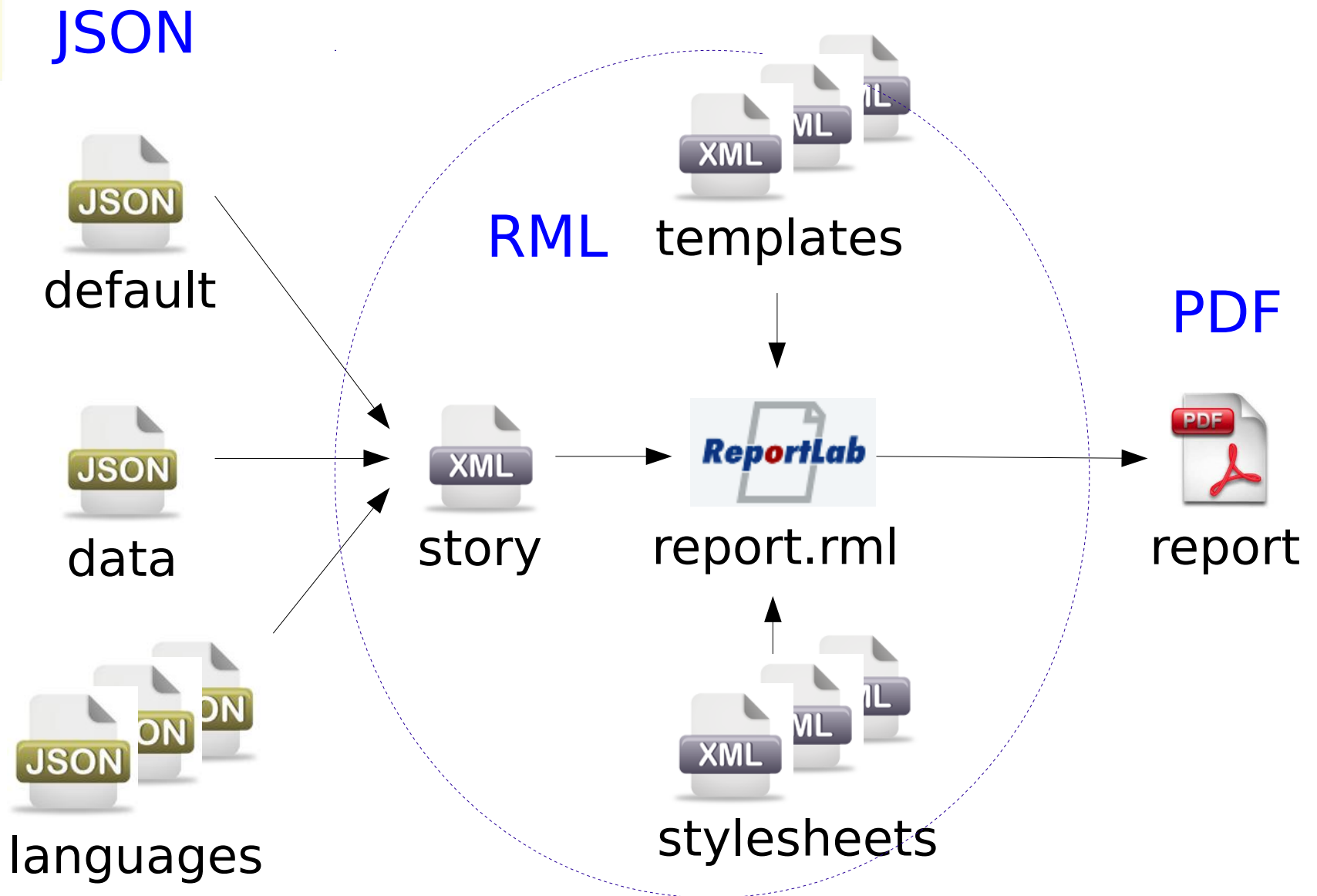
YAML

need PyYAML or PySick

Python itself



Working model





Main code

```
# args:
# -t template (default: report.rml)
# -o outfile (default: report.pdf)
# infile (default: data.json)

args = parser.parse_args()

with open("lib/data_file.py", "w") as datafile:
    print >>datafile, 'data_file = "' + args.infile + '"'

from pyjon.reports import ReportFactory

factory = ReportFactory()
factory.render_template(template_file = args.template)
factory.render_document(args.output)
factory.cleanup()
```



XInclude

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<?python from lib.data import *?>
<!DOCTYPE document SYSTEM "rml.dtd">
<document
  xmlns:py="http://genshi.edgewall.org/"
  xmlns:xi="http://www.w3.org/2001/XInclude"
  filename="report.pdf"
  compression="true">
  <xi:include href="rml/${filename}.xml"
    py:for="filename in ('docinit', 'stylesheet',
                        'template', 'story')">
    <xi:fallback>
      <py:choose test="filename">
        <docinit py:when="'docinit'"></docinit>
        <stylesheet py:when="'stylesheet'"></stylesheet>
        <template py:when="'template'"></template>
        <story py:otherwise=""></story>
      </py:choose>
    </xi:fallback>
  </xi:include>
</document>
```



JSON loader

```
try:
    from data_file import *
except:
    data_file = "data.json"

from json import load

# load defaults
with open("defs/default.json") as json:
    locals().update(load(json))

# load default lang and data
for jsonfile in "lang/" + lang + ".json", data_file:
    with open(jsonfile) as json:
        locals().update(load(json))

# load data lang
with open("lang/" + lang + ".json") as json:
    locals().update(load(json))
```




JSON example

```
{  
  "lang": "eng",  
  
  "date": "13.03.2008/15:49",  
  "name": "John",  
  "surname": "Doe",  
  "id": "1",  
  "birth": "25.12.1948",  
  "age": "59",  
  "sex": "M",  
  
  "speed": ["4.1", "8.5", "7.6"]  
}
```



Real RML content

```
<story xmlns:py="http://genshi.edgewall.org/">
  <title style="title">${title}</title>
  <para style="subtitle">${subtitle}</para>

  <spacer length="13"/>

  <blockTable style="table-user-data">
    <tr>
      <td>${date_field}:</td>
      <td>${date}</td>
    </tr>
    <tr>
      <td>${name_field}:</td>
      <td>${name}</td>
    </tr>
    <tr>
      <td>${surname_field}:</td>
      <td>${surname}</td>
    </tr>
```



Dynamic RML content

```
<illustration height="193" width="482">  
  <?python  
  
    # generate plot/report.png with matplotlib  
  
  ?>  
  <image x="0" y="0" file="plot/report.png"  
    height="193" width="482"  
    py:if="exists('plot/report.png')"/>
```



Questions and answers

identi.ca/whitone

www.whitone.tk

whitone@gmail.com