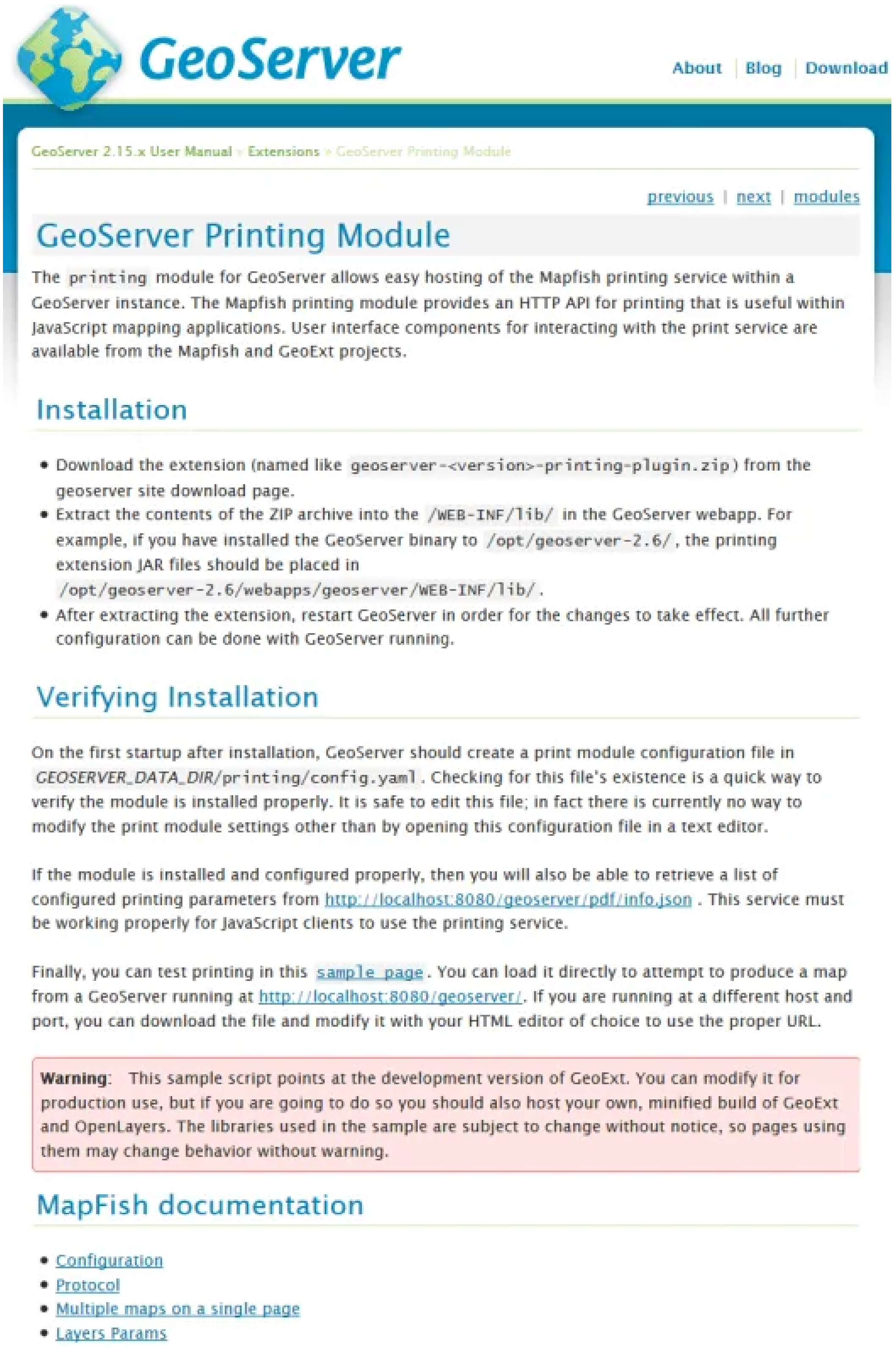
7 DECEMBER 2018 BY ATILIO FRANCOIS

Print web maps with geoserver: avoid the plugin!

We will discuss here how to set up a solution to generate pdf for printing from an interactive mapping page, the OpenLayers 4 encoded page and the Geoserver 2.14 server. Technically, to read the Geoserver doc,everything should be very simple: Geoserver **Print** plugin installation , installing test, configuration and that's it


GeoServer
About | Blog | Download
GeoServer 2.15.x User Manual » Extensions » GeoServer Printing Module
previous | next | modules

GeoServer Printing Module

The `printing` module for GeoServer allows easy hosting of the Mapfish printing service within a GeoServer instance. The Mapfish printing module provides an HTTP API for printing that is useful within JavaScript mapping applications. User interface components for interacting with the print service are available from the Mapfish and GeoExt projects.

Installation

- Download the extension (named like `geoserver-<version>-printing-plugin.zip`) from the geoserver site download page.
- Extract the contents of the ZIP archive into the `/WEB-INF/lib/` in the GeoServer webapp. For example, if you have installed the GeoServer binary to `/opt/geoserver-2.6/`, the printing extension JAR files should be placed in `/opt/geoserver-2.6/webapps/geoserver/WEB-INF/lib/`.
- After extracting the extension, restart GeoServer in order for the changes to take effect. All further configuration can be done with GeoServer running.

Verifying Installation

On the first startup after installation, GeoServer should create a print module configuration file in *GEOSERVER_DATA_DIR*`/printing/config.yaml`. Checking for this file's existence is a quick way to verify the module is installed properly. It is safe to edit this file; in fact there is currently no way to modify the print module settings other than by opening this configuration file in a text editor.

If the module is installed and configured properly, then you will also be able to retrieve a list of configured printing parameters from http://localhost:8080/geoserver/pdf/info.json . This service must be working properly for JavaScript clients to use the printing service.

Finally, you can test printing in this `sample page`. You can load it directly to attempt to produce a map from a GeoServer running at http://localhost:8080/geoserver/. If you are running at a different host and port, you can download the file and modify it with your HTML editor of choice to use the proper URL.

Warning: This sample script points at the development version of GeoExt. You can modify it for production use, but if you are going to do so you should also host your own, minified build of GeoExt and OpenLayers. The libraries used in the sample are subject to change without notice, so pages using them may change behavior without warning.

MapFish documentation

- Configuration
- Protocol
- Multiple maps on a single page
- Layers Params

Although installation is quick and easy, the problems start when you want to use the proposed test page to make sure everything is working properly.Impossible to make it work and each time we find the source of the problem and solve it, a new error message appears and another problem arises.

The script paths are no longer valid, the contents of the libraries have changed, in short, if you want to waste your time, you can always try.

You can also look for solutions on the net and as me, find posts in forums with the same questions asked … and without any answer.

Let's start from the very beginning…

The print module must generate a pdf file containing a map and information. This map is produced by Geoserver and the request is generated by an html page written with OpenLayers. Even if you can do it all without another library, you often use the GeoExt library which simplifies the application coding.

As for the generation of the pdf, it relies on the java Mapfish Print
library.

If for one, Geoserver version does not affect the result, it is not the same for the other components.

The installation described in the Geoserver documentation refers to Mapfish version 2. It has been replaced by MapFish3 since 2015. It should be understood that the two versions are completely different in terms of syntax and workflow.

MapFish V2 is used with the GeoExt V2 library which relies on OpenLayers
2. That is to say old versions.

If you are working with
OpenLayers 4 or 5, the version of GeoExt to use is version 3, and the Mapfish version 3.

Therefore, the installation the Geoserver print plugin (MapFish V2) is not needed.

This is not a problem; MapFish3 can be installed quite easily. The problem or problems come when you are looking in the net for documentation,examples, information on error messages. You will have, anyhow, references of all the versions but without knowing which version it refers to …

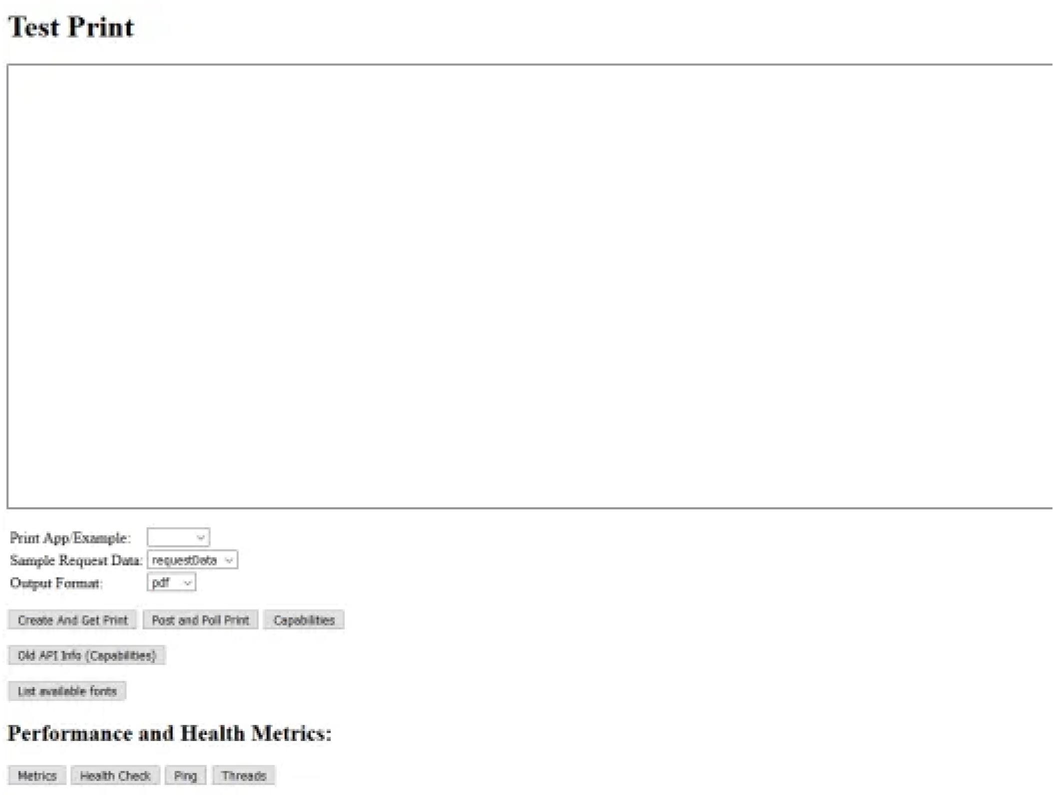
Installing Mapfish 3

The Mapfish Print WAR file is a Java archive that can be run on any Javaweb application server such as Tomcat or Jetty. To use Mapfish Print as a stand-alone Web application, you must:

- Install Java 7 or later

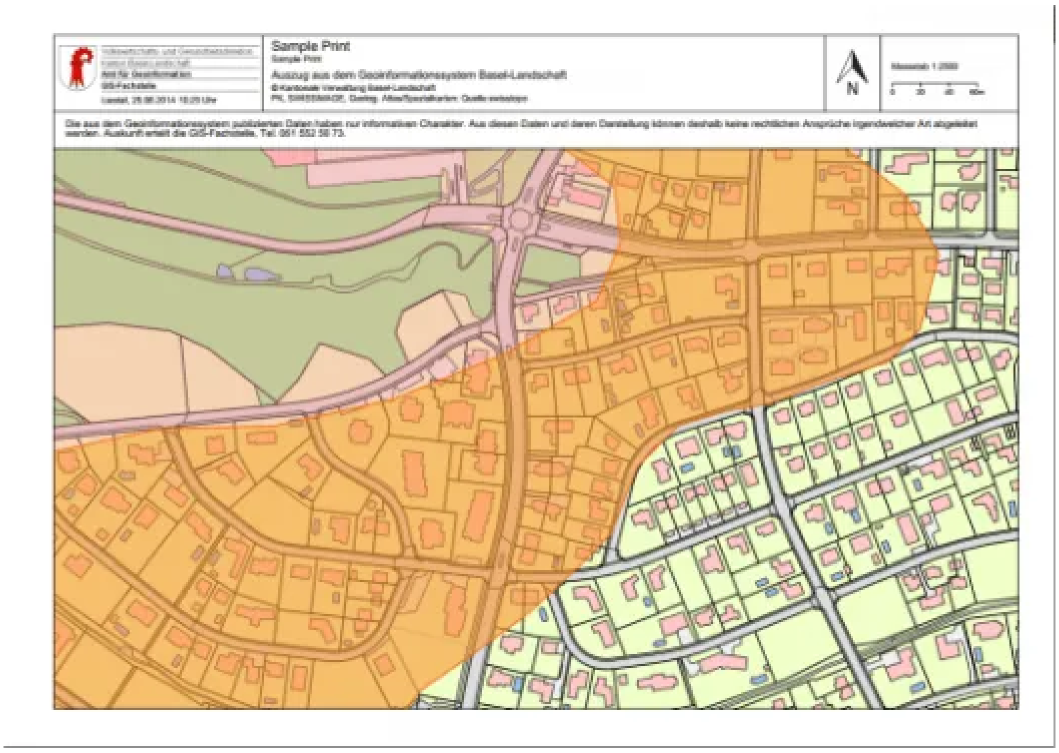
- Download the web application server: war file from http://mapfish.github.io/mapfish-print-doc/download.html
- Place the WAR print Mapfish file in the web apps directory of the web application server.
- Rename the Mapfish Print WAR file to a more practical name, such as print.war (the name will be part of the URL).
- Start the web application server(Tomcat or Jetty)

  Verify that the application is running correctly by visiting http: // localhost: 8080 / print / index.html.



For the time being, do not do anything with this page. The fact that it is displayed is sufficient to confirm that the installation of MapFish has been successful.

How Mapfish 3 works The lack of online documentation is difficult to grasp because there is no clear introduction to the general architecture


Amt für Geoinformation
GIS-Fachstelle
Liestal, 25.08.2014 10:25 Uhr
Sample Print
Sample Print
Auszug aus dem Geoinformationssystem Basel-Landschaft
© Kantonale Verwaltung Basel-Landschaft
N
Massstab 1:2500
Die aus dem Geoinformationssystem publizierten Daten haben nur informativen Charakter. Aus diesen Daten und deren Darstellung können deshalb keine rechtlichen Ansprüche irgendwelcher Art abgeleitet werden. Auskunft erteilt die GIS-Fachstelle, Tel. 061 552 58 73.

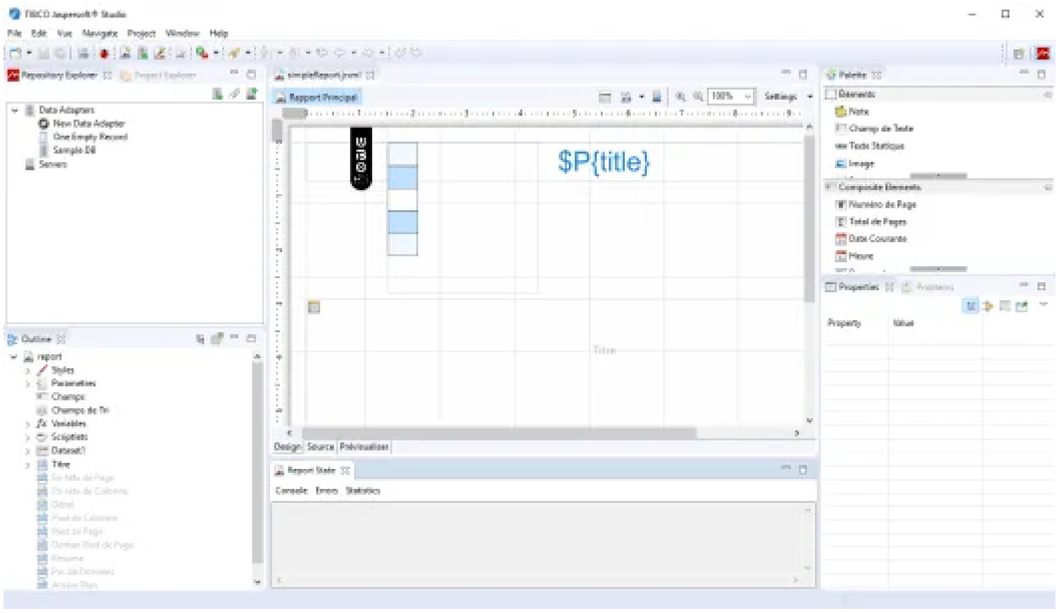
In order for the user to obtain a document of this type by clicking on a button «   To print   » located on a page with an interactive map(html page with code OpenLayers) there are three elements to consider:

- A form template created with Jasper Studio
- a config.yaml file configured on the print server
- an element "specs" to be added to the POST or GET order of the html page

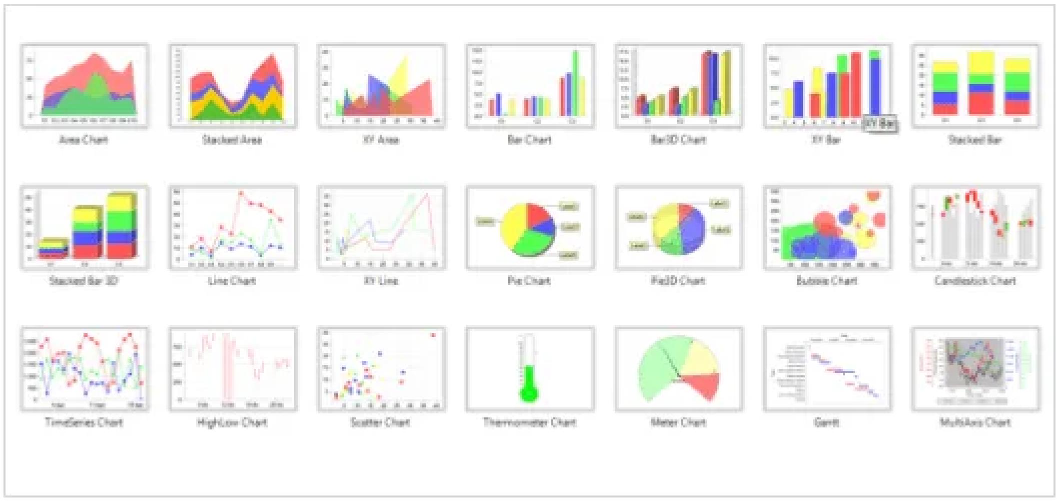
Let's discuss in more detail.

Form template

To create the form we use ***Jaspersoft Studio*** . You can download it from this page .



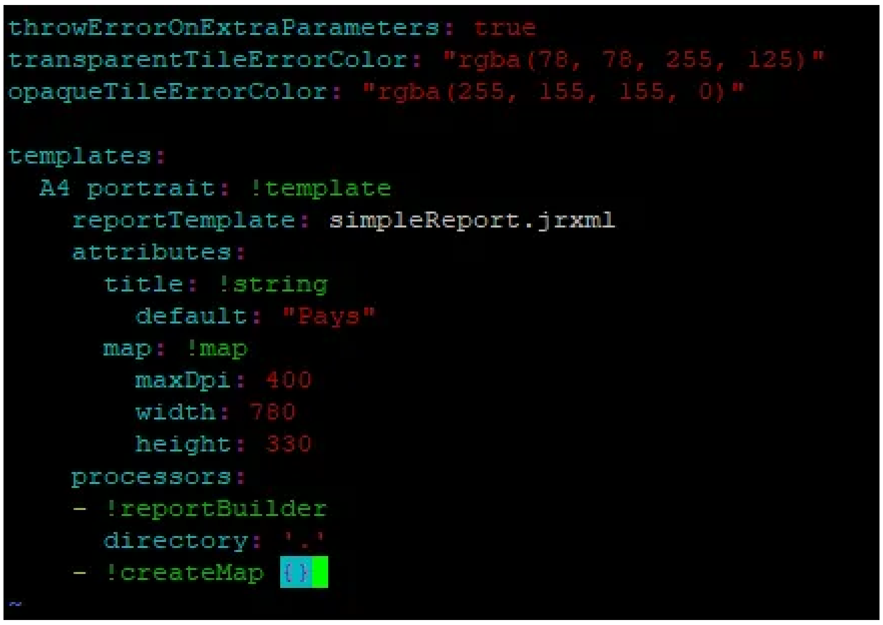
JasperStudio allows you to create a form template where you can place blocks of text, images, maps and captions but also other blocks such as graphics



The purpose of this work is to create a .jrxml file with the page template to create. The very content of the page will be defined elsewhere: the layers to be represented, the scale, the projection,…

Configuration file .yaml

Compared to Mapfish Print version 2, there is always a configuration file in yaml format but it is now much more simple, and focuses on the definition of models: the Jasper files to use (the previous.jrxml), and the parameters to send to the Jasper engine to fill the blocks of the model. Here is an example of a very simple configuration file


```
throwErrorOnExtraParameters: true
transparentTileErrorColor: "rgba(78, 78, 255, 125)"
opaqueTileErrorColor: "rgba(255, 155, 155, 0)"

templates:
  A4 portrait: !template
    reportTemplate: simpleReport.jrxml
    attributes:
      title: !string
        default: "Pays"
      map: !map
        maxDpi: 400
        width: 780
        height: 330
    processors:
    - !reportBuilder
      directory: '.'
    - !createMap {}
~
```

What has to be understood in this file:

- the **A4 portrait** line **:! template** defines the model name. When we want to use this configuration we will do it by referring to the layout ***A4portrait***
- the **report Template is** used to indicate the name of our .jrxml file created with JaspersoftStudio.
- the ***title*** and ***map*** **attributes** contain the information to be passed to the Jasperengine. This means that we have created a ***title*** box and a ***map*** box in the form template. Here we inform the content of these two boxes.

Of course, there are other options available for configuration files. But for now, let's keep it simple.

We have a .jrxml form template and a .yaml configuration file. All that remains is to send the pdf creation request.

Print request

From the html page we will make a POST to the server with a **spec** parameter that will contain, in json format, the parameters of our print request.

Here is an example of a **spec** :

{

"layout": "A4
portrait",

```
“attributes”:
{“map”: {

“center”: [

5

45

]

“rotation”: 0,

“longitudeFirst”: true,

“layers”: [{

“geoJson”: “file:
//countries.geojson”,

“style”: {

“*”:
{“symbolizers”: [{

“fillColor”: “#
5E7F99″,

“strokeWidth”: 1,

“fillOpacity”: 1,

“type”:
“polygon”,

“strokeColor”: “#
CC1D18″,

“strokeOpacity”: 1

}]},
```

“version”:
“2”

}

“type”:
“geojson”

}],

“scale”: 100000000,

“projection”:
“EPSG: 4326”,

“dpi”: 72

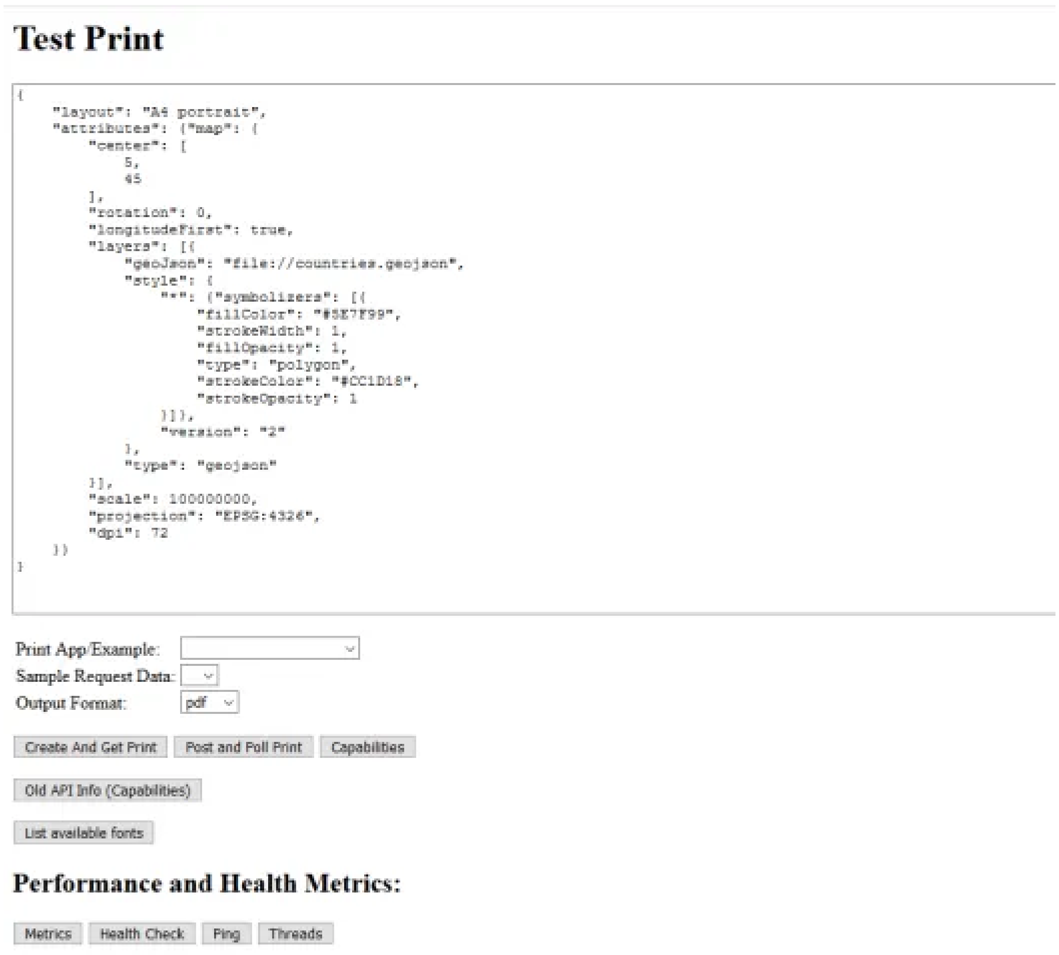
}}

}

The line **“ layout »:« A4 portrait** is used to indicate the link with the config.yaml file with its line **A4 portrait:! Template**

In the list of sent attributes, in this simple example, we only have the map to be displayed (” **map** «) with its formatting parameters (centring, rotation, scale, projection, definition) and, above all,the definition of the layers to be mapped with the definition of their data sources (here a local geoJson file).

In response to this post the page receives the download url of the pdf document produced. To verify this, you can now copy and paste the json code corresponding to the spec in the Test Print window:

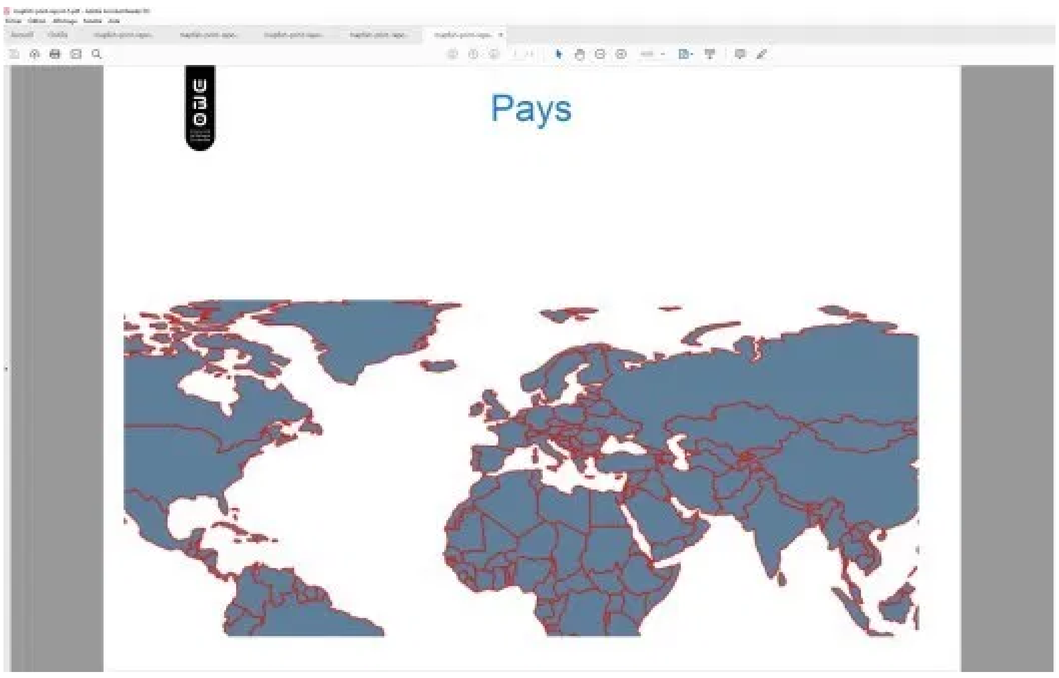

Test Print

```
{
    "layout": "A4 portrait",
    "attributes": {"map": {
        "center": [
            5,
            45
        ],
        "rotation": 0,
        "longitudeFirst": true,
        "layers": [{
            "geoJson": "file://countries.geojson",
            "style": {
                "*": {"symbolizers": [{
                    "fillColor": "#5E7F99",
                    "strokeWidth": 1,
                    "fillOpacity": 1,
                    "type": "polygon",
                    "strokeColor": "#CC1D18",
                    "strokeOpacity": 1
                }]},
                "version": "2"
            },
            "type": "geojson"
        }],
        "scale": 100000000,
        "projection": "EPSG:4326",
        "dpi": 72
    }}
}
```

Print App/Example:
Sample Request Data:
Output Format: pdf

Performance and Health Metrics:

Click **Create and Get Print** or **Post and Poll Print** , in return you will get your pdf file:



The same yaml configuration file and the same jrxml template can be called with different data. Here is another version of the code json spec which requests the dat the countries served in WMS by Geoserver:

{

“layout”: “A4
portrait”,

“outputFormat”:
“pdf”,

“attributes”: {

“map”: {

“projection”:
“EPSG: 3857”,

“dpi”: 72,

“rotation”: 0,

“center”:
[-8233518.5005945, 4980320.4059228],

“scale”: 130000000,

“layers”: [

{

“baseURL”:
“http://carto-dei-brest.fr/geoserver/postgres/wms”,

“opacity”: 1,

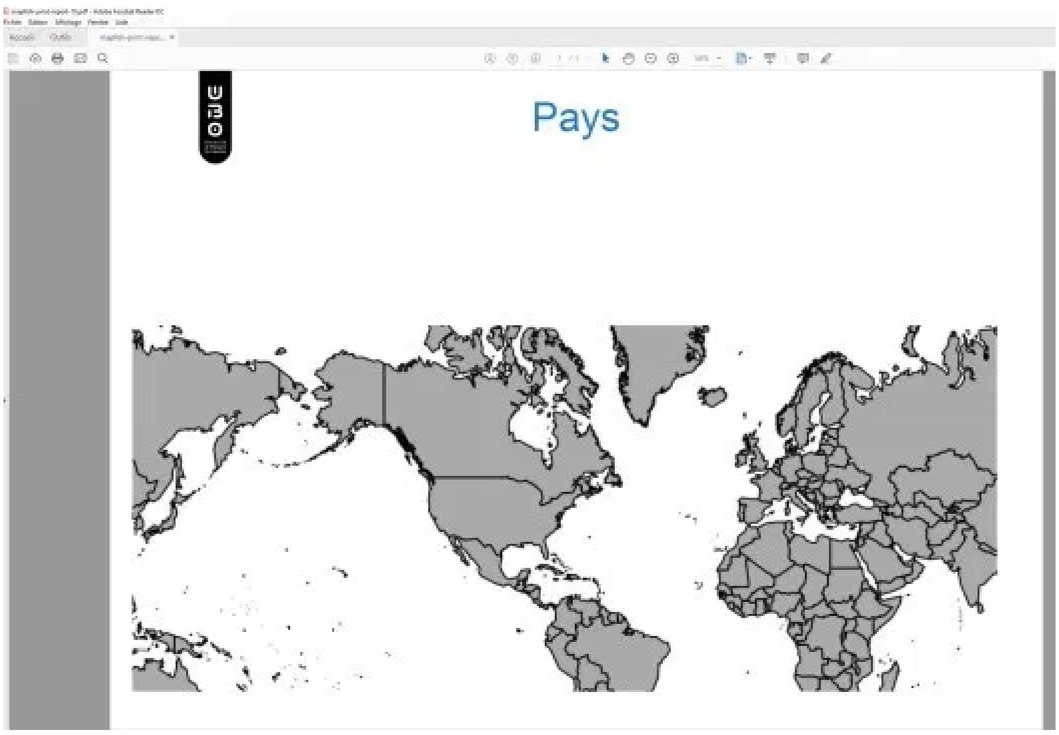
“type”:
“WMS”,

“layers”:
[“postgres: country”],

“imageFormat”:
“image / png”,

```
“styles”: [“polygon”],

“customParams”: {

“TRANSPARENT”:
“true”

}

}

]

}

}

}
```

In the print test page we replace the text json and the result is the following:



To go further, you have a presentation of CampToCamp and Mapfish documentation ,

And if you use the Geoserver print plugin…

If you still decide to use the plugin provided by Geoserver, you should know that:

- there is no way to use Jaspersoft to create a template. Everything you do with Jaspersoft and MapFish 3 you will have to do manually in the config.yaml file. And you will not have the same possibilities as,for example, to include graphics in your pdf output.
- the syntax of the yaml file is totally different. You will find its description on this page .
- If you want to use the GeoExtlibrary examples as a basis for your developments, you will need to install and use the GeoExt2 version. GeoExt 3 is not compatible withthe Geoserver Print module.

- 0
- 0
- 0
- 0
-

No related posts.

NON CLASSÉ