Proxysign-printer

This service is composed by two components:

- Virtual printer (printer-node folder) is written in NodeJS 12.x
- workflow-printer is a java application with spring boot (https://spring.io/projects/spring-boot) his goal is to apply acrofield signed on a pdf document and send it with the workflow to proxysign-workflow service.

Get source and installation

Clone the project

```
git clone ssh://idel.fuschini@tordivalle.intra.linkverse.com/proxysign-printer
```

The virtual printer (directory printer-node) is a node project for use it install nodejs from (https://nodejs.org/it/download/) and install the package:

```
npm install
```

The root directory is a java project and for compile it's possible to use as IDE eclipse or intellij, remember to import the project as a maven project and run with 'package' option.

Configuration and run

Virtual Printer

the configuration of the virtual is in external file called config.json:

```
{
    "virtualPrinterName": "gdpr virtual printer",
    "virtualPrinterPort": 632,
    "virtualPrinterUri" : "/printer/gdpr",
    "virtualPrinterMaxListeners": 100000,
    "hostWorkflowPrinter": "http: //localhost:8080/"
}
```

important: the virtual printer using bonjour so virtual printer name must be unique in the network also host of workflow-printer (next paragraph), after configure it launch the command below:

```
node printer.js
```

workflow-printer

For launch the workflow-printer you need jdk 1.8.x or more (https://www.oracle.com/java/technologies/javase-downloads.html), for launch:

```
java -jar document-printer-0.0.1-SNAPSHOT.jar
```

The default options are:

```
server.port=8080
spring.servlet.multipart.max-file-size=200MB
spring.servlet.multipart.max-request-size=200MB
printerUpload.path=C:\\printer\\upload
printerUpload.imageTargetWidth=229
printerUpload.imageTargetHeight=56
printerUpload.imageTargetSize=38472
printerUpload.imageTargetType=jpg
printerUpload.hostGdprWorkflow=https://testdev.proxysign.
it//grapho/rest/request/startworkflow
printerUpload.graphoUsername=proxysign
printerUpload.graphoPassword=proxysign
printerUpload.recipient=station7
printerUpload.signatureType=graphometric_simple
printerUpload.targetDevice=tablet
```

for change some parameters the syntax is:

```
java -jar document-printer-0.0.1-SNAPSHOT.jar --<option>=
<value> --<option2>=<value2> .. --<option n>=<value n>
```

for example if you want to change the listen port of application and change the recipient:

```
java -jar document-printer-0.0.1-SNAPSHOT.jar --
server.port=8181 --printerUpload.recipient=station6
```

Options

Option	Description
server.port	Listen port
spring.servlet.multipart.max-file-size	specifies the maximum file size allowed for multipart/form-data
spring.servlet.multipart.max-request-size	specifies the maximum size allowed for multipart/form-data requests
printerUpload.path	the path where workflow-printer store the file, beforee shutdown the files are deleted
printerUpload.imageTargetWidth	specify width of image target
printerUpload.imageTargetHeight	specify height of image target
printerUpload.imageTargetSize	specify size of image target
printerUpload.imageTargetType	specify type of image target (png, jpg etc.)
printerUpload.hostGdprWorkflow	specify host where send the workflow gdpr service
printerUpload.graphoUsername	specify username for access to workflow gdpr service

printerUpload.graphoPassword	specify password for access to workflow gdpr service
printerUpload.recipient	specify recipient where to active the workflow, **if recipient is 'none' that means the recipient name arrived from virtual printer**
printerUpload.signatureType	specify type of signature (graphometric, graphometric_simple)
printerUpload.targetDevice	specify target device where activate (tablet,STU-530 etc...)