TRANSACTIONAL NETWORK

Graphic Representation invoice.

Version 1.0.

Content

Preface

This document aims to detail specifications, functionalities and technical section of the development of the json component / page that will show the graphical representation of the receipts on the TransactionalNetwork site.

Introduction

The objective of the project will be to graphically represent the receipt with tipoCFE = 111 that corresponds to the type of document E-Factura.

Functional requirements..

1. Develop a component that shows the graphical representation of a receipt. + The graphic representation of the receipt should be a json page per document type. The graphic representation must be the same as the design sent in the pdf "Redesign of XSLs Documents.pdf" page 2.
Budget: 500 usd

2. Embed the component in the full site page of the final product TN, developer in REACT. You will be provided with a project in GIT to integrate into it.
Budget: 150 usd

3. Integrate the REST API to obtain the XML to be displayed on the screen according to design.
Budget: 150 usd

4. Bug Fixing.

Budget: 150 usd

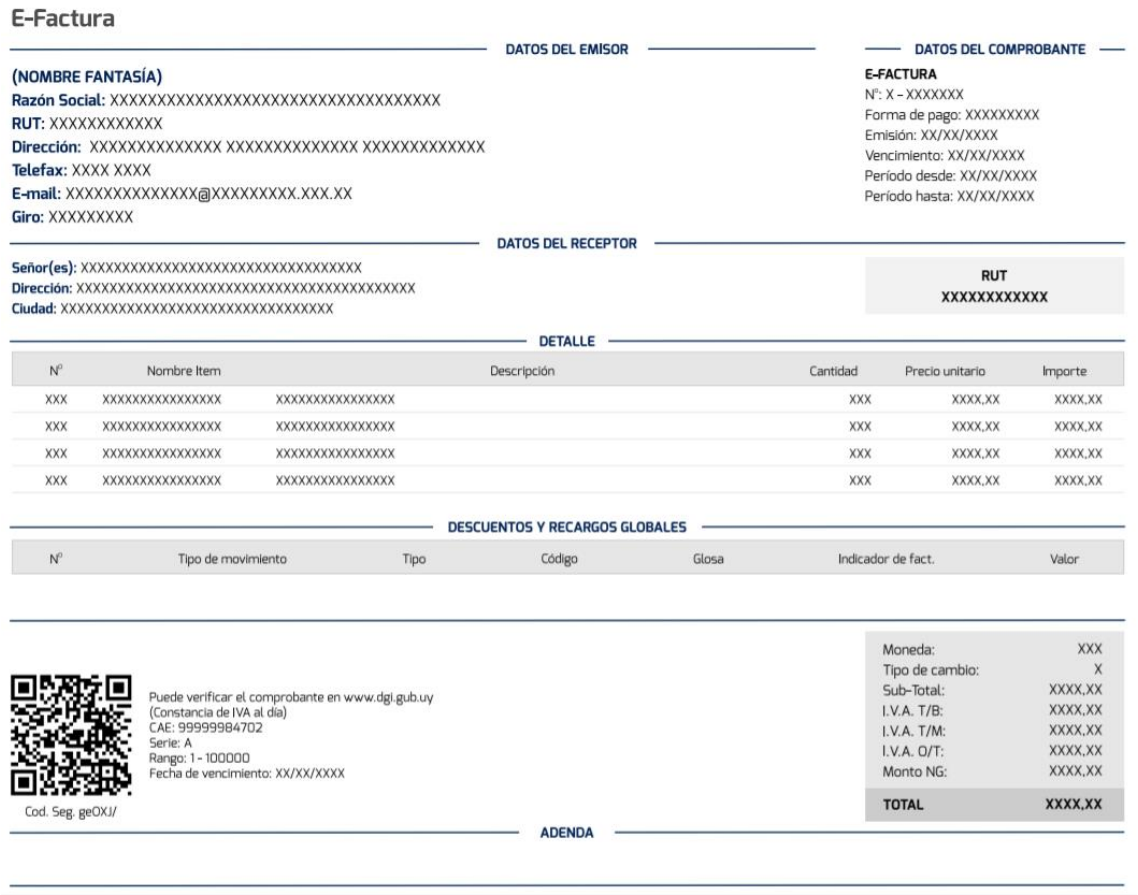
XML mapping.

An XML will be provided that must be represented in a .json page with the design specified in page 2 of the annex "Redesign of XSLs Documents.pdf".

The XML is made up of a set of TAGS that will give body to the graphic representation and will make up the voucher.

Shown below:

1. the final design that the XML should adopt.



E-Factura

DATOS DEL EMISOR

(NOMBRE FANTASÍA)
Razón Social: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
RUT: XXXXXXXXXXXX
Dirección: XXXXXXXXXXXXXX XXXXXXXXXXXXXX XXXXXXXXXXXXX
Telefax: XXXX XXXX
E-mail: XXXXXXXXXXXXXX@XXXXXXXXX.XXX.XX
Giro: XXXXXXXXX

DATOS DEL COMPROBANTE

E-FACTURA
Nº: X – XXXXXXX
Forma de pago: XXXXXXXXX
Emisión: XX/XX/XXXX
Vencimiento: XX/XX/XXXX
Período desde: XX/XX/XXXX
Período hasta: XX/XX/XXXX

DATOS DEL RECEPTOR

Señor(es): XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Dirección: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Ciudad: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

RUT
XXXXXXXXXXXX

DETALLE

Nº	Nombre Item	Descripción	Cantidad	Precio unitario	Importe
XXX	XXXXXXXXXXXXXXX	XXXXXXXXXXXXXXX	XXX	XXXX,XX	XXXX,XX
XXX	XXXXXXXXXXXXXXX	XXXXXXXXXXXXXXX	XXX	XXXX,XX	XXXX,XX
XXX	XXXXXXXXXXXXXXX	XXXXXXXXXXXXXXX	XXX	XXXX,XX	XXXX,XX
XXX	XXXXXXXXXXXXXXX	XXXXXXXXXXXXXXX	XXX	XXXX,XX	XXXX,XX

DESCUENTOS Y RECARGOS GLOBALES

Nº	Tipo de movimiento	Tipo	Código	Glosa	Indicador de fact.	Valor

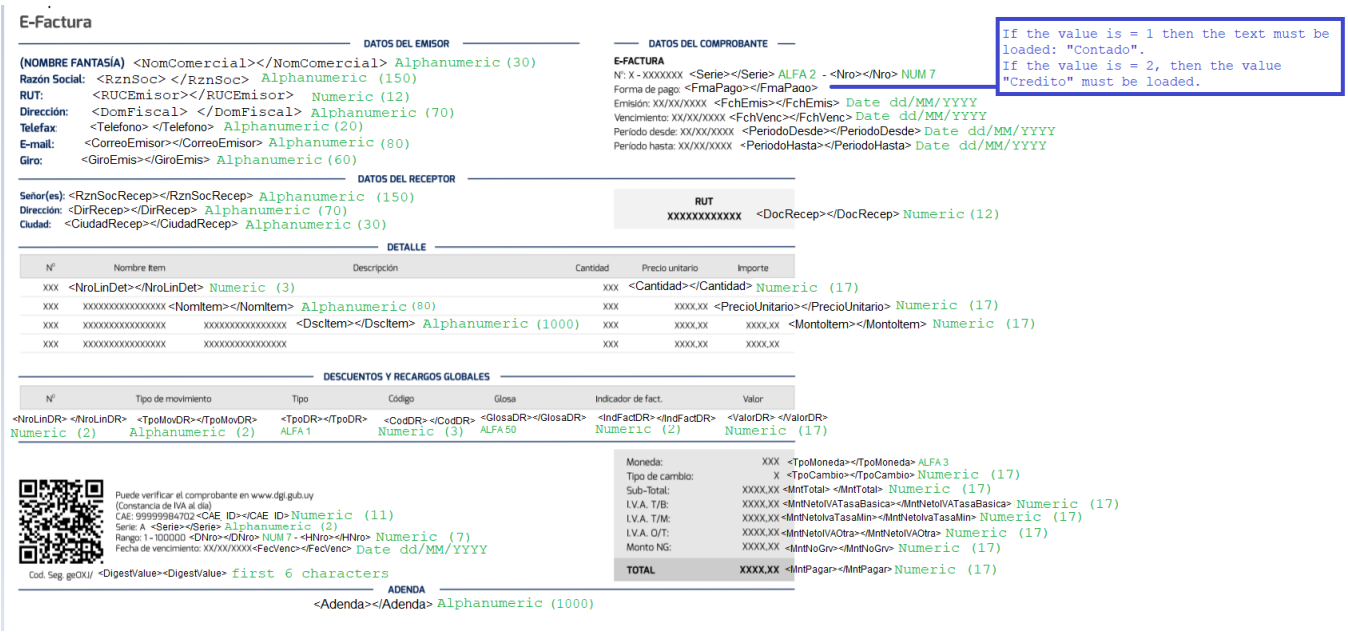
Puede verificar el comprobante en www.dgi.gub.uy
(Constancia de IVA al día)
CAE: 99999984702
Serie: A
Rango: 1 - 100000
Fecha de vencimiento: XX/XX/XXXX

Cod. Seg. geOXJ/

Moneda:	XXX
Tipo de cambio:	X
Sub-Total:	XXXX,XX
I.V.A. T/B:	XXXX,XX
I.V.A. T/M:	XXXX,XX
I.V.A. O/T:	XXXX,XX
Monto NG:	XXXX,XX
TOTAL	**XXXX,XX**

ADENDA

2. The mapping of tags to document fields.



Generation of the QR code.

QR Code =

“https://www.efactura.dgi.gub.uy/consultaQR/cfe?”

+

<RUCEmisor></RUCEmisor> + ","

+

<TipoCFE></TipoCFE> + ","

+

<Serie></Serie> + ","

+

<Nro></Nro> + ","

+

<MntPagar></MntPagar> with decimal separator using point "." + ","

+

IF <TmstFirma></TmstFirma> IS NOT NULL

+ <TmstFirma></TmstFirma> in format dd/MM/YYYY + ","

IF <TmstFirma></TmstFirma> IS NULL

+ ","

+ <DigestValue></DigestValue>

Example:

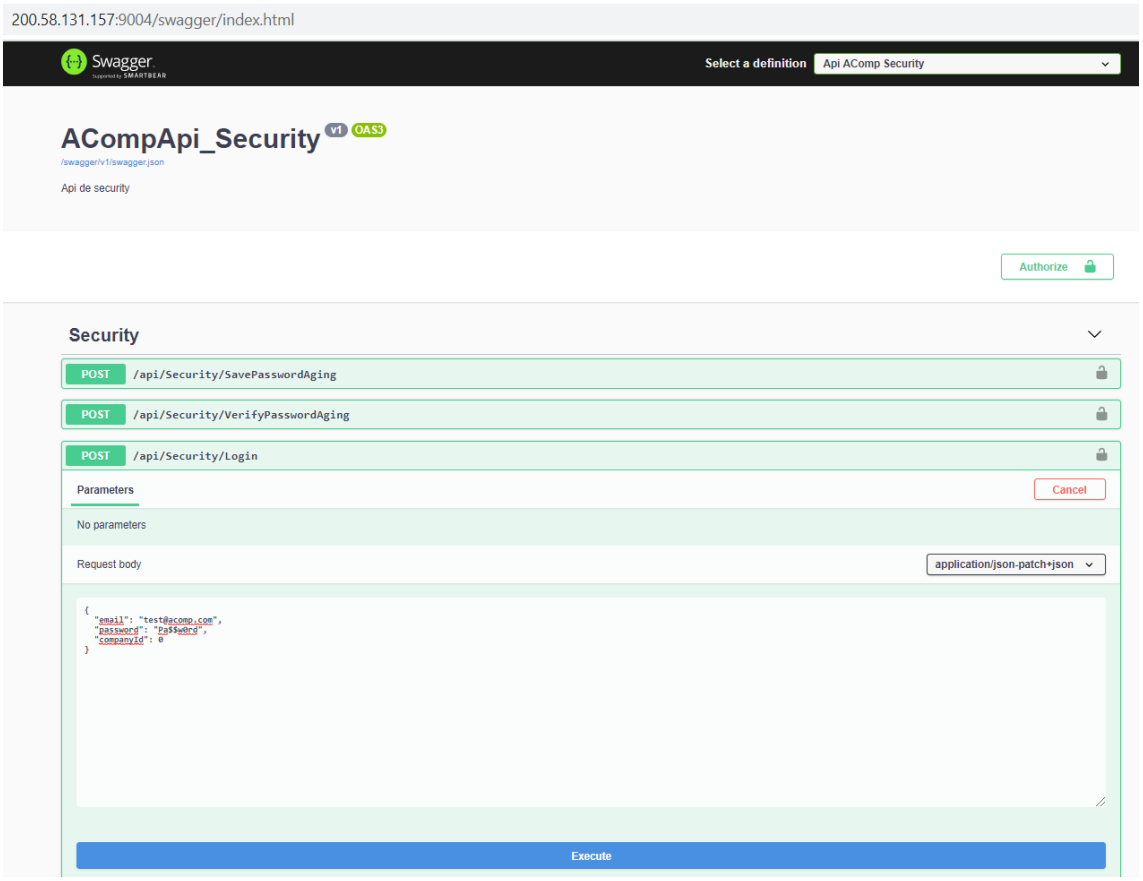
https://www.efactura.dgi.gub.uy/consultaQR/cfe?210094030014,101,A,0056232,302.00,18/10/2012,pBzXv%2F6NuZDM0Ctx%2BlnC3nrxQ2E%3D

Integrate the REST API to obtain the XML to be displayed

Api_Security:

http://200.58.131.157:9004/swagger/index.html

Api_Security In this api we must login (POST / api / Security / Login) and obtain the token and then authenticate ourselves in the other apis. The token is stored in a portal session variable.



With the obtained token you must authenticate to consume the Api Documents. The authentication is of type Bearer Token.

The token is obtained in the response of the Api Security.

Api_Document:

http://200.58.131.157:9000/swagger/index.html

In this api we are going to obtain the XML of the voucher to represent. The operation to invoke is: GET / api / Document / {documentId} / {downloadInPdf} The parameters for this operation should be:

200.58.131.157:9000/swagger/index.html

GET /api/Document/{documentId}/{downloadInPdf}

Parameters

Name	Description
documentId * required integer (path)	1
downloadInPdf * required boolean (path)	false

Execute

Responses

Curl

```
curl -X GET "http://200.58.131.157:9000/api/Document/1/false" -H "accept: application/json" -H "Authorization: bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0ZXN0QGFjb21wLmNvbSIsImp0aSI6IjI0MGVjZTkzLThlNTUtNDU2NC1hY2E2LTAyODQ1MDRj
L25hbWVpZGVudGlmaWVyIjoiMDkwODBlZDAtMzY5Yi00M2JlLWI2Y2ItZjlkNTkzNTc0OTQzIiwiZW1wcmVzYUNsYWltIjoiMSIsImh0dHA6Ly9zY2hlbWFzL
XMiOiJbe1wiSXNzdWVyXCI6XCJMT0NBTCB8VVRIT1JJVFlcIixcIk9yaWdpbmFsSXNzdWVyXCI6XCJMT0NBTCBVVRIT1JJVFlcIixcIlByb3BlcnRpZXNcIj
8PU1wiLFwiVmFsdWVcIjpcIlRPVEFMXCIsXCJWYWx1ZVR5cGVcIjpcImh0dHA6Ly93d3cudzMub3JnLzIwMDEvWE1MU2NoZW1hI3N0cmluZ1wifSx7XCJjc3N
UWVwiLFwiUHJvcGVydGllc1wiOnt9LFwiU3ViamVjdFwiOm51bGwsXCJUeXBlXCI6XCJDT01QUk9CQUSURVNcIixcIlZhbHVlXCI6XCJBU0lHTkFSXCIsXCJW
c3N1ZXJcIjpcIkxPQ0FMIEFVVEhPUklUWVwiLFwiT3JpZ2luYWxJc3N1ZXJcIjpcIkxPQ0FMIEFVVEhPUklUWVwiLFwiUHJvcGVydGllc1wiOnt9LFwiU3Via
```

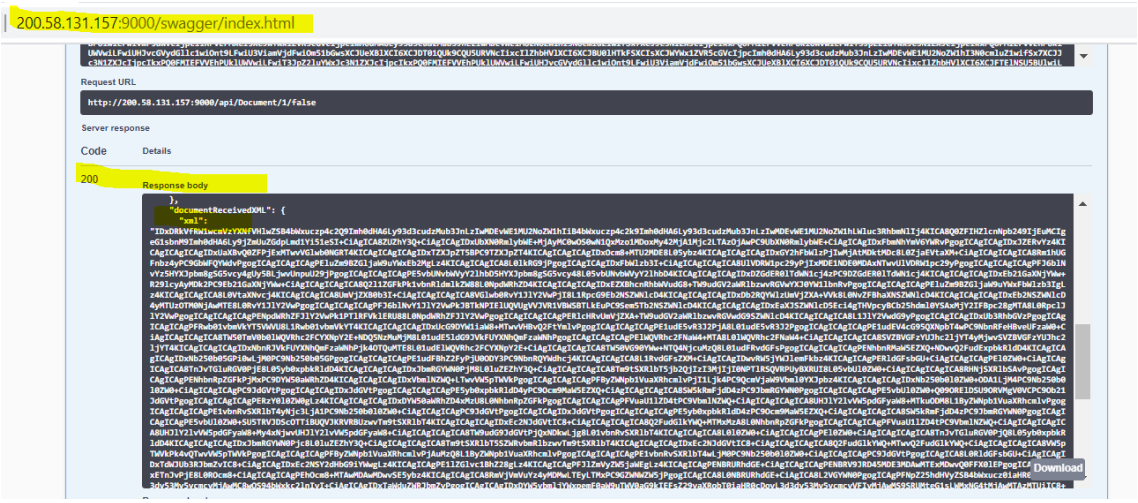
Request URL

```
http://200.58.131.157:9000/api/Document/1/false
```

Server response

Code	Details
200	Response body


```
{
  "error": {
    "message": null,
    "exceptionMessage": null,
    "code": 0
  },
  "data": {
    "documentId": 1,
    "serie": "123",
    "nro": 123,
    "fchEmis": null,
    "paymentMethod": null,
    "expireDate": "2020-09-22T10:09:19.8287894",
    "emitingRUC": "12345678901",
    "legalName": "11060418",
    "commercialNameEmisor": null,
    "emailEmisor": null,
    "typeDocRecep": null,
    "docRecep": "210154140015",
    "legalNameRecep": null,
    "additionalInfo": null,
    "purchaseIdentificatioNumber": null,
    "currencyTypeTrx": "USD",
```



In this operation, the response returns an "XML" parameter, here the xml that I must show will be in base 64. This file must be DECODE base 64 returned.

Non-functional requirements

• The product should be designed in a scalable way, with the possibility of incorporating new functionalities in the future. For example: new types of documents and the editing of lines of the document as shown in the video sent.

• The source code will remain the property of the company, in order to carry out future maintenance tasks.

Technology's and implementation

• Page/Component developed in REACT.