



Contenu [Afficher](#)

## Objective

- Combine the use of Office Word and Oracle ORDS to create a document generator controlled from Word.
- Automatically export these documents to PDF format.
- Develop a simplified service (Windows only) capable of launching multiple instances of document generation.

## Description

This is a prototype I developed that prioritizes the use of Office Word both as a design tool and for merging with data from an Oracle REST ORDS service. I tested several approaches with data obtained in JSON and XML formats.

This project, [available from Github](#), is called w00RDS. It is implemented as a module written in VBA within a Word template. This module handles the following tasks:

- Obtaining REST data from an ORDS server, in JSON or XML format
- Loading a Word template suitable for the document to be generated
- Merging elements using Content Controls
- Exporting the document as a PDF
- Optionally, sending the PDF document via a REST API.

Words relies on a third-party module, vba-json, for reading documents in JSON format.



A command line script, `woords.cmd`, launches Office Word with the needed parameters.

Let's have a look to the way Word is started:

```
WINWORD.EXE /q /p%1 /z%REPDIR%\templates\woords.dotm
```

The tricky part involves calling Word with parameter passing. Specifically, in our case, it is necessary to pass information regarding the template to be used and possibly an ID to select specific data. In command mode, there is no selector to add parameters, but I found that `/p` was not used, so I utilized it to convey the parameters.

The structure of the parameter string is defined as follows:

- `c:config`
- `i:pkid`
- `f:folder Name`

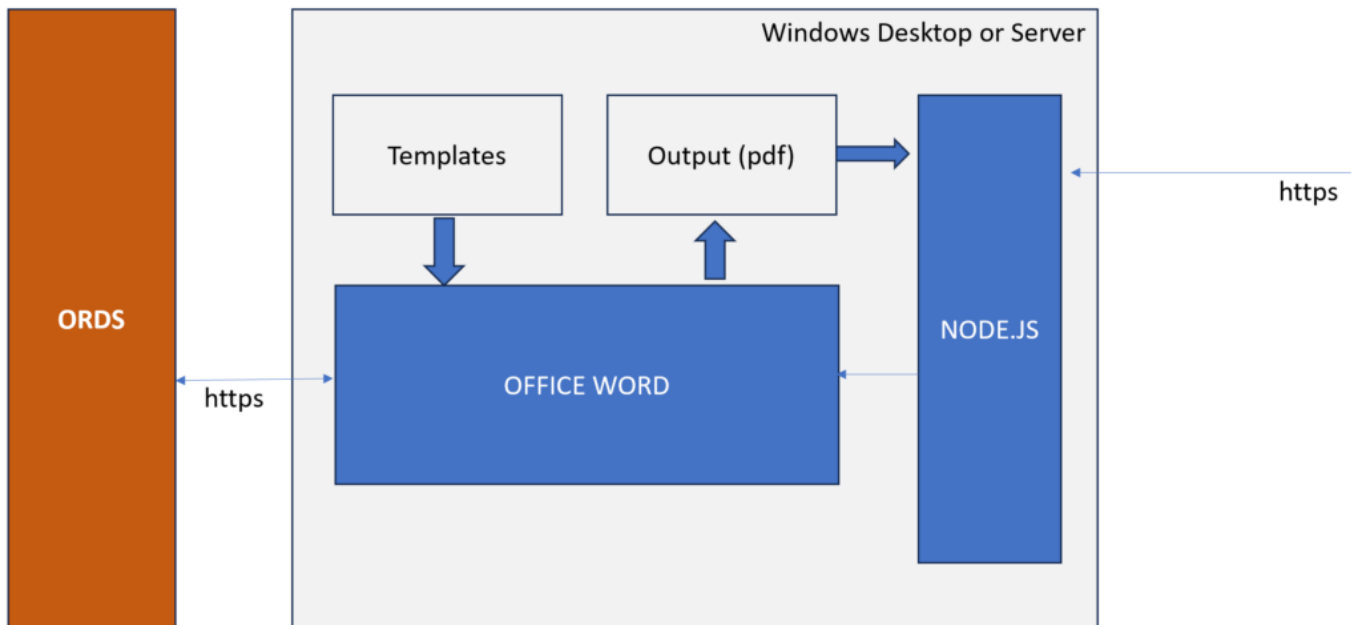
ex: `c:order:i:652:f:zzzzz`

All these details are hidden in the `woords` command script. We just need to launch the Words in the following ways:

- From the command line
- Via a URL call
- Through a WebSocket (not implemented) [see my other post](#)



## Architecture



Important prerequisite: The solution described in this article only works on a physical machine running Windows, as Office Word needs to be activated, or in a Windows VirtualBox under Linux, or in a Windows Docker container under Windows.

In server mode, the HTTP listener receiving the request must be able to trigger the execution of Office Word in command mode, and the listener must be on the same machine as Office Word. Therefore, we can imagine the server composed of an HTTP/HTTPS listener based on Node.js (using the access module), for example, or an Apache HTTP server with PHP installed. Woords includes a server, `swoords.js`, running with Node.js.

## Use Cases



Woords is suitable for the following types of documents:

- Master/detail style documents with one document per PDF file (invoice, purchase order, employee record, product sheet, etc.)
- Simple lists in table form

Woords isn't for:

- Reports with grouping and page breaks based on break levels

### Using in server mode

Since the operation of a VBA module is single-threaded, it is not feasible to make it a service at its level. However, it is possible to launch multiple instances of Word from a Node.js server. The principle is very basic, as the server launches Word in command line mode as discussed previously.

This allows for a certain level of parallelism. However, it is not advisable to expect intensive use with a large number of simultaneous requests, as this quickly leads to stability issues, resulting in Word instances that remain open.

A simple example server is provided with Woords (`server\swoords.js`). As a prerequisite, it is necessary to have previously installed Node.js along with the dependencies:

- `express`
- `fs`
- `crypto`
- `child_process`



This server provides two endpoints:

- /viewpdf/:config/:pkid? for launching pdf generation and viewing output pdf file
- /pdf/:config/:pkid? for generation only

## Office Word specificities used for merging

To represent variable fields, I chose to use Content Controls (CC) and Repeating Content Controls (RCC).

Each CC has at least the following standard attributes:

- Title
- Tag

I use the Tag attribute to record the name of the JSON or XML node with which it should be merged. In the case where it needs to display an array, the érging will stand between a RCC and a Array type node wich the same name.

# Setup and Use

## Installation

Installation steps

- Download woords01.zip from <https://github.com/patrickmonaco/woords>
- Create a directory (ie: c:\app\woords) and unzip woords.zip



- In Office Word, add this directory to the list of trust locations in the Trust Center
- Complete the content of the seettings.xml file by entering the endpoint for obtaining OAUTH tokens and the endpoint corresponding to the REST module
- Optional: Install Node.js if you want to set up a Woords service
- Start words by entering: bin\words configfile

Trust Center ? X

- Trusted Publishers
- Trusted Locations
- Trusted Documents
- Trusted Add-in Catalogs
- Add-ins
- ActiveX Settings
- Macro Settings
- Protected View
- Message Bar
- File Block Settings
- Privacy Options
- Form-based Sign-in

### Trusted Locations

Warning: All these locations are treated as trusted sources for opening files. If you change or add a location, make sure that the new location is secure.

Path	Description	Date Modified
<b>User Locations</b>		
[REDACTED]		[REDACTED]
C:\[REDACTED]		[REDACTED]
C:\Program Files\Microsoft Office\root\Temp	Word default location: Application Templat...	
C:\Users\patri\AppData\Roaming\Microsoft\T	Word default location: User Templates	
C:\Users\patri\AppData\Roaming\Microsoft\	Word default location: StartUp	
<b>Policy Locations</b>		

Path: C:\[REDACTED]  
Description:  
Date Modified: 05/05/2024 17:46  
Sub Folders: Allowed

Add new location... Remove Modify...

## Access Control

The data is obtained by Word from the VBA code and requests to the REST APIs are made via the MSXML2.XMLHTTP class from VBA.



Access is controlled by OAUTH with a Client Credentials type client.

This means that you must create an OAUTH Client of the *Client Credentials* type at ORDS level (see appendices) and save the Client ID and Client Secret in a settings.xml file accessible by Word. (see appendices)

- Create an OAUTH client (see appendices) and copy the client Id and Secret values
- Adapt the settings.xml file by copying the ID and SECRET.
- You must indicate the endpoint to obtain a valid token and the endpoint of the module. Do not put a trailing slash for the value of the ENDPOINT tag.

```
<CONF>
  <OAUTH_ENDPOINT>https://[REDACTED].adb.eu-
frankfurt-
1.oraclecloudapps.com/[REDACTED]/oauth/token</OAUTH_ENDPOINT>
  <ID>[REDACTED]</ID>
  <SECRET>[REDACTED]</SECRET>
  <ENDPOINT>https://[REDACTED].adb.eu-frankfurt-
1.oraclecloudapps.com/ords/[REDACTED]/sales</ENDPOINT>
  <VERBOSE>N</VERBOSE>
  <SAVEDOC>N</SAVEDOC>
  <DEBUG>Y</DEBUG>
  <REPO>/conf/</REPO>
</CONF>
```

## Organizing models and data sources

The same Word template can be used with different data types and vice versa. For this reason, I introduced the notion of configuration. This is an XML file which indicates the template, the REST endpoint to use as well as miscellaneous informations:



```
<CONF>
  <TEMPLATE>Vision3</TEMPLATE>
  <ROOT_NODE>ORDERS</ROOT_NODE>
  <HANDLER>/ordersxml/</HANDLER>
  <HEADER>HEADER</HEADER>
  <SPLIT>Y</SPLIT>
  <LIST>N</LIST>
  <SAVEDOC>N</SAVEDOC>
  <POST_URL>https://xxx/</POST_URL>
  <DEBUG_LEVEL>0</DEBUG_LEVEL>
</CONF>
```

In the configuration above, we tell Woords to use a template called vision3 and to fetch the data by making a REST call to a handler called /ordersxml/ and which must be suffixed at the endpoint specified in a global settings file. There must be a configuration for each document template. These files are saved in the conf directory but it is also possible to place them in a global directory accessible from a REST service. In this case, you must create a woo\_repo table (see the setup\woords\_createrepository.sql creation script) and you must indicate the corresponding handler in the settings file described below: /conf/. This handler is provided in the gpm.demo.sales module (setup\ORDS\_REST\_DEMO\_gpm.demo.sales)

**SAVEDOC:** Indicates whether you wish to keep the intermediate Word file that was generated.

**POST\_URL:** REST handler to which to post the PDF file or not. NOT IMPLEMENTED.



## Global Settings for Woords

```
<CONF>
  <OAUTH_ENDPOINT>https://xxx/ords/demo/oauth/token</OAUTH_ENDPOINT>
  <ID>XXXXXXXXXXXXXXXXXXXXXXXXXX</ID>
  <SECRET>ZZZZZZZZZZZZZZZZZZ</SECRET>
  <ENDPOINT>yyyyyyyyyyyyyyyy/ords/demo/sales</ENDPOINT>
  <VERBOSE>N</VERBOSE>
  <DEBUG>N</DEBUG>
  <REPO>/conf/</REPO>
</CONF>
```

Comments on the settings file:

- OAUTH\_ENDPOINT corresponds to the URL which provides an OAUTH token (see ORDS doc)
- ID is the Client ID value for Client Credentials
- SECRET is the value of Customer Secret
- ENDPOINT: REST module URL
- VERBOSE: display or not a detailed log
- DEBUG: Display a detailed log
- REPO: optional. REST handler that provides the content of a configuration based on a name passed in the URL. Overrides individual xxx.conf configuration files

## Functional tests

I took as a sample the one supplied with the APEX installations and which



installs tables corresponding to order management.

***A collection of customers, stores, products, and orders. This dataset includes JSON data for the product description, and longitude / latitude for the stores.***

This is sufficient to illustrate common cases.

Reminders for installing this dataset

Sql workshop / Utilities / Sample Datasets.

Chosir *Customer Orders*.

Samples provided with Woords

The following configurations are ready to use for demo/training purpose

- helloworld (basic message)
- invoice (standard invoice with a master/detail)
- order (standard purchase order with a master/detail)
- emp (standard emp table )
- cols (use for report )

## Designing a Word template suitable for merging

To test my application, I used a standard template offered in Word.

It's called *Professional Sales Invoice* and it's included in the Business Blue Design business template series.

The dynamic fields are represented by *Content Controls*.

- Give a tag name to the CC corresponding to a XML tag or to a Json node



to be displayed..

- Implement the multi-line part, corresponding to the order lines, using a *Repeating Content Control* (RCC) and incorporating other *Content Controls* (CC), one per column.
- Set the mapping between a XML tag or JSON node and the CC by using the Tag attribute (cf below).

The tag name in RCC must be the same as the XML « father » tag.  
(ORDER\_ITEMS\_FOP\_V in my example)

Qté.	Num éro d'article	Description	Prix unitaire	Remise	Total de la ligne
ORD 123 QUAI 10	PRODUCT_	PRODUCT_NAME Poussins	UNIT_PRICE 5.00	REMISE REMISE	MNT 50 .00 ORDER_ITEM
				Remise totale	0
				Sous-total	TOTAL 278.00
				Taxe de vente	TVA 13. 90

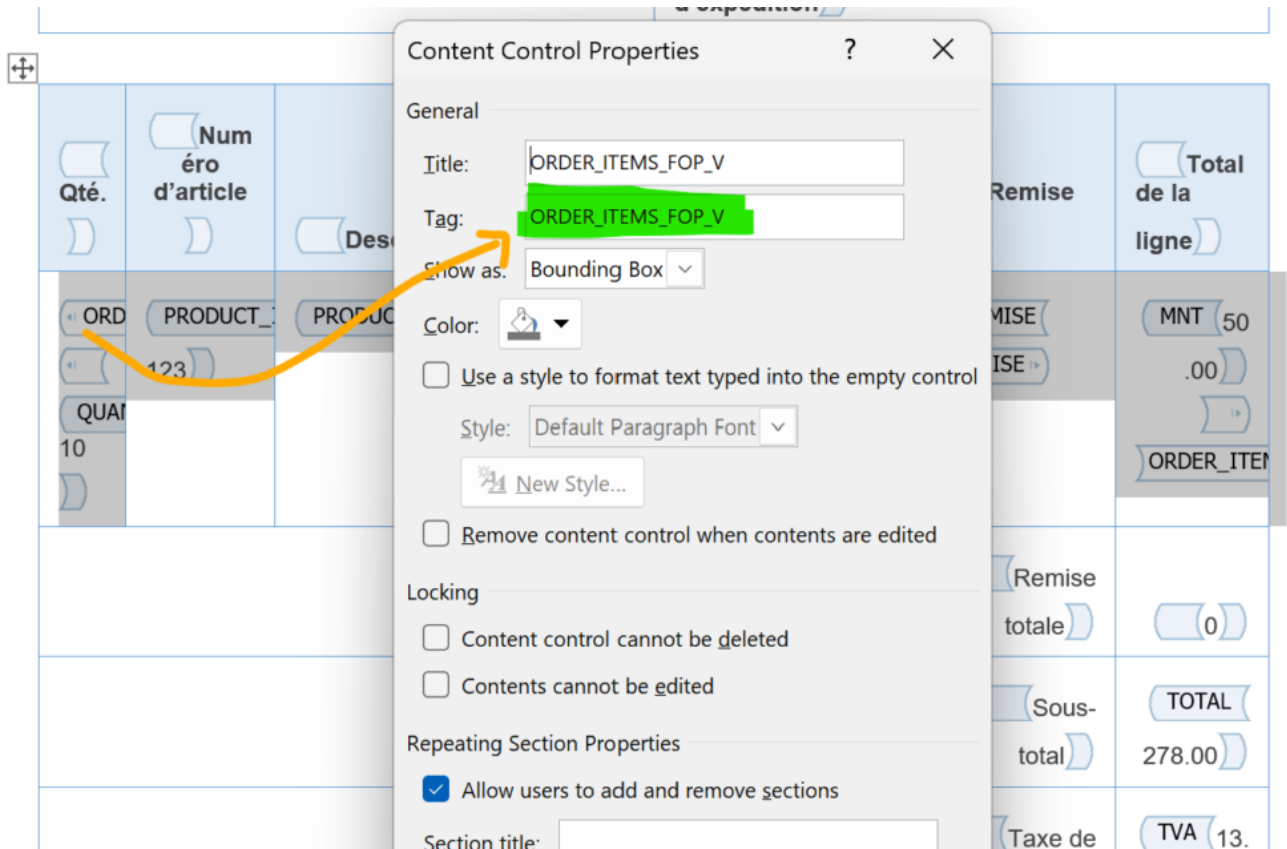


The screenshot shows a Word document with a table. The table has columns for 'Qté.', 'Num éro d'article', and 'Description'. The 'Description' column contains a content control with the text 'PRODUCT\_NAME' and a 'Skirt' label. A 'Content Control Properties' dialog box is open, showing the 'General' tab. The 'Title' and 'Tag' fields are both set to 'PRODUCT\_NAME'. The 'Show as' dropdown is set to 'None'. The 'Color' dropdown is set to a blue color. The 'Use a style to format text typed into the empty control' checkbox is unchecked. The 'Style' dropdown is set to 'Default Paragraph Font'. The 'Remove content control when contents are edited' checkbox is unchecked. The 'Locking' section has two unchecked options: 'Content control cannot be deleted' and 'Contents cannot be edited'. The 'OK' and 'Cancel' buttons are visible at the bottom of the dialog box. An orange arrow points from the 'Tag' field in the dialog box to the 'PRODUCT\_NAME' content control in the table.

Repeating Content Control setup



```
▼ <DATASET>
  ▶ <HEADER>
  ...
  </HEADER>
  ▼ <ORDERS>
    ▼ <ORDERS_REC>
      <ORDER_ID>652</ORDER_ID>
      <ORDER_DATETIME>25/08/2023</ORDER_DATETIME>
      <CUSTOMER_ID>374</CUSTOMER_ID>
      <ORDER_STATUS>COMPLETE</ORDER_STATUS>
      <STORE_ID>1</STORE_ID>
      <CUSTOMERS_CUSTOMER_ID>374</CUSTOMERS_CUSTOMER_ID>
      <CUSTOMERS_FULL_NAME>Irene Moore</CUSTOMERS_FULL_NAME>
      <CUSTOMERS_EMAIL_ADDRESS>irene.moore@internalmail</CUSTOMERS_EMAIL_ADDRESS>
      <STORES_STORE_ID>1</STORES_STORE_ID>
      <STORES_STORE_NAME>Online</STORES_STORE_NAME>
      <STORES_WEB_ADDRESS>https://www.example.com</STORES_WEB_ADDRESS>
      ▼ <ORDER_ITEMS_FOP_V>
        ▼ <ORDER_ITEMS_FOP_V_ORDERS_REC>
          <ORDER_ID>652</ORDER_ID>
          <PRODUCT_ID>13</PRODUCT_ID>
          <PRODUCT_NAME>Boy's Hoodie (Grey)</PRODUCT_NAME>
          <QUANTITY>2</QUANTITY>
          <UNIT_PRICE> 12.64</UNIT_PRICE>
          <MNT> 25.28</MNT>
        </ORDER_ITEMS_FOP_V_ORDERS_REC>
        ▼ <ORDER_ITEMS_FOP_V_ORDERS_REC>
          <ORDER_ID>652</ORDER_ID>
          <PRODUCT_ID>44</PRODUCT_ID>
          <PRODUCT_NAME>Women's Coat (Black)</PRODUCT_NAME>
          <QUANTITY>3</QUANTITY>
          <UNIT_PRICE> 39.32</UNIT_PRICE>
          <MNT> 117.96</MNT>
        </ORDER_ITEMS_FOP_V_ORDERS_REC>
      </ORDER_ITEMS_FOP_V>
      ▼ <ORDER_FOOTER_FOP_V>
        ▼ <ORDER_FOOTER_FOP_V_ORDERS_REC>
          <ORDER_ID>652</ORDER_ID>
          <TOTAL> 143.24</TOTAL>
          <TVA> 28.65</TVA>
          <TTC> 171.89</TTC>
        </ORDER_FOOTER_FOP_V_ORDERS_REC>
      </ORDER_FOOTER_FOP_V>
    </ORDERS_REC>
  </ORDERS>
```



## Content Controls types to be used for repeating sections

When adding a Content Control inside a repeating section, it is important to note that the oblong symbol designating the CC must be light blue and not with a pictogram of arrows pointing to the right or the left. If this second case occurs (which I consider to be an anomaly, but for which I have no explanation), an additional CC must be created just before. If we leave things as is (with « arrows »), we lose the formatting and alignment that we could have given to CCs containing numbers.

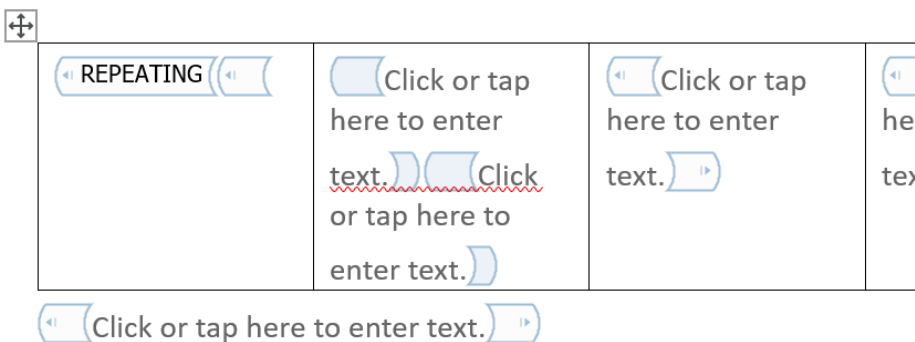
## Conditional displaying

It is possible to condition the display of the content of an RCC according to the presence or absence of data.



For example, if there are no direct reports at the level of a manager, we neutralize the display of the corresponding RCC. To do this, you must encapsulate the RCC in another RCC with the same tag name but prefixed with the “?” sign.

If the RCC tag = `direct_reports`, the RCC container tag will be: `?direct_reports`



## Producing REST data

JSON based on a collection Query

XML Data input

I have already developed a [tool to produce SQL queries that generate XML](#). The constraint is that I cannot use a Collection Query handler. Instead, I have to use a Media Resource query, which results in the loss of pagination elements.

The [XML results generated](#) by these queries can also be converted to JSON. Thus, a Media Resource handler with a MIME type of `application/json` can be used. Since `vba-json` is case sensitive, it is better to generate column aliases in lowercase.



## PDF files and logs

Each Woords session generates a random label that is used to create a folder in the pdf directory and to create a log file in the logs directory.

It is the user's responsibility to purge these directories and files.

## Performances

Launching 1 to 5 Word instances. During the test with 5 instances, one ended with an error and a second one stopped after two documents. Only three threads survived.

The ramp-up was two seconds between each launch. Conclusion: the concurrency level is a maximum of three. Beyond that, there are risks of abrupt termination.

The maximum throughput is approximately 1.7 PDF documents per second.










#Threads	Sec	Doc	Doc/sec
1	38	50	1,32
2	70	100	1,43
3	85	150	1,76
5	86	152	1,77

## Annexes



## Module ORDS pour la gestion des commandes

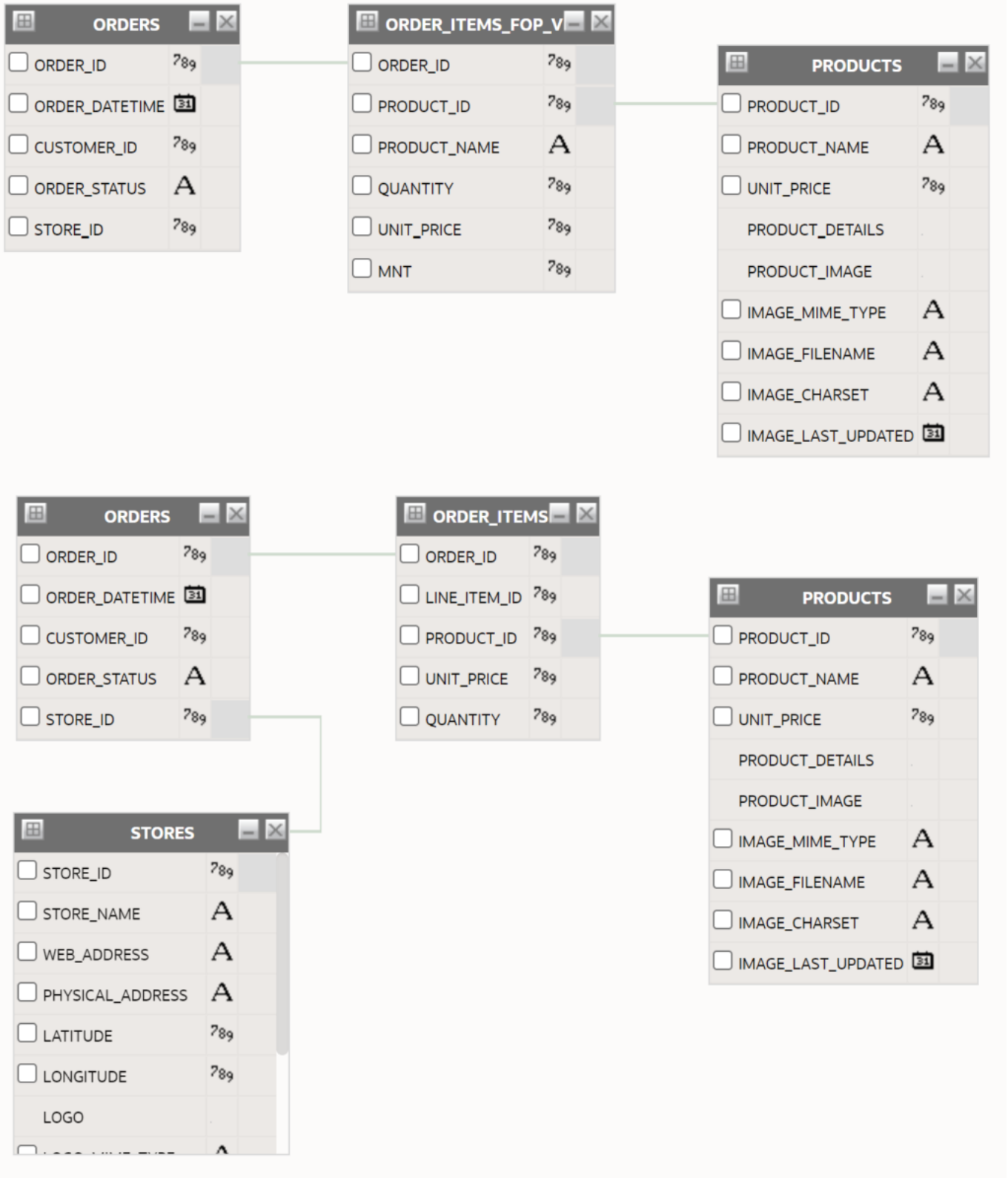


- ▼  gpm.demo.sales
  - >  orders
  - >  orders/:id
    -  orders/:id/footer
  - >  orders/:id/header
  - >  orders/:id/lines
  - >  ordersxml/:pkid?
  - >  pdf/:pid
  - >  view/:pid

The module gpm.demo.sales is protected by a privilege named orders.priv.

One must create a CLIENT\_CREDENTIAL client and get back the Client id and Client secret. These informations must be stored in the settings.xml file.

ORDERS data model





## Vues ajoutées

In order to add formatting and computing, it's better to minimize the code in the REST handler code, and move this task to a sql view level.

### ORDER\_ITEMS\_FOP\_V

This view calculates the amounts for each order line

```
CREATE OR REPLACE FORCE EDITIONABLE VIEW "ORDER_ITEMS_FOP_V"
("ORDER_ID", "PRODUCT_ID", "PRODUCT_NAME", "QUANTITY", "UNIT_PRICE",
"MNT",
FOREIGN KEY ("ORDER_ID")
REFERENCES "ORDERS" ("ORDER_ID") DISABLE) AS
select order_id, i.product_id, product_name,
quantity, i.unit_price ,(quantity * i.unit_price) mnt
from order_items i,
products p
where i.product_id = p.product_id;
```

### ORDER\_FOOTER\_FOP\_V

This view provides the global amounts for the footer.

```
CREATE OR REPLACE FORCE EDITIONABLE VIEW "ORDER_FOOTER_FOP_V"
("ORDER_ID", "TOTAL", "TVA", "TTC", "SHIPPING",
FOREIGN KEY ("ORDER_ID")
REFERENCES "ORDERS" ("ORDER_ID") DISABLE) AS
select order_id, sum(quantity * i.unit_price) total,
round(sum(quantity * i.unit_price * 0.2),2) tva,
```



```
round(sum(quantity * i.unit_price * 1.2),2) ttc,  
100 as "shipping"  
from order_items i,  
products p  
where i.product_id = p.product_id  
group by order_id;
```

## XML statement giving orders

In the sample module, see the ordersxml handler

## JSON Collection Query

In order to keep the advantages provided by the use of a query collection (pagination, among others), we can create a handler with a main query and secondary links to be able to search the data via an additional REST call.

```
select CUSTOMERS.CUSTOMER_ID as CUSTOMER_ID,  
CUSTOMERS.FULL_NAME as FULL_NAME,  
CUSTOMERS.EMAIL_ADDRESS as EMAIL_ADDRESS,  
ORDERS.ORDER_DATETIME as ORDER_DATETIME,  
ORDERS.ORDER_ID as ORDER_ID,  
'../orders/' || ORDERS.ORDER_ID || '/lines' AS "$order_lines"  
from  
ORDERS ORDERS,  
CUSTOMERS CUSTOMERS  
where CUSTOMERS.CUSTOMER_ID = ORDERS.CUSTOMER_ID
```



```
and ORDER_ID = :id
```

```
{
  "items": [
    {
      "customer_id": 374,
      "full_name": "Irene Moore",
      "email_address": "irene.moore@internalmail",
      "order_datetime": "2023-08-25T03:30:13Z",
      "order_id": 652,
      "links": [
        {
          "rel": "order_lines",
          "href": "https://[redacted].com/ords/demo/sales/orders/orders/652/lines"
        }
      ]
    }
  ],
  "hasMore": false,
  "limit": 25,
  "offset": 0,
}
```

Pour la technique d'insertion d'un [lien relatif, cf article Oracle-Base](#).

## Ajout d'un client OAUTH

```
BEGIN
  OAUTH.create_client(
    p_name          => 'word_client',
    p_grant_type    => 'client_credentials',
    p_owner         => 'GPM Factory',
    p_description   => 'A client Office Word for Orders printing',
    p_support_email => 'pmo@email.com',
    p_privilege_names => 'orders.priv'
  );
COMMIT;
```



```
END;
```

## Starting words in command line

Enter `bin\words`

Available Word selectors used:

- `/q` Word trigger in silent mode
- `/p` selector not used by Word and added to allow passing a parameter when launching Word which will be used by a VBA macro. The character string which immediately follows the `/p` selector responds to a precise syntax, specific to words and which is indicated earlier in this document.
- `/z` Opens a new document based on a Word template (`words.dotm`) and TRIGGERS a possible macro named `Autonew`.

## Documents samples

Purchase order from [TemplateLAB](#) site



TemplateLAB



# PURCHASE ORDER

**Bill To:**  
 Diana Fowler  
 Diana Fowler  
 diana.fowler@internalmail  
 [City, ST ZIP Code]  
 [Phone]

**Ship to:**  
 Diana Fowler  
 Customer  
 diana.fowler@internalmail  
 [City, ST ZIP Code]  
 [Phone]

**PURCHASE ORDER #**  
653

**ORDER DATE**  
25/08/2023

**Notes:**  
 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero.

Qty	Fabric Description Yarn type/Width of fabric	Dyed? Dye Type	Finishing	Packing Type*	Unit Price	Line Total
3	<b>Men's Hoodie (Red)</b> Ring spun yarn, 1200mm	YES Sulfur	Pre-washed	A	\$10.24	\$30.72
3	<b>Women's Trousers (Blue)</b> Ring spun yarn, 1200mm	YES Sulfur	Pre-washed	A	\$49.12	\$147.36
5	<b>Boy's Socks (White)</b> Ring spun yarn, 1200mm	YES Sulfur	Pre-washed	A	\$10.48	\$52.40

\*) Packing types:

A - Rolled on tube and then each roll in an un tear able Polly bag.  
 B - Folded at one yard length and each piece in an un tear able Polly bag and then so many pieces in a 3-ply carton.

Shipping Date: 12/05/2022

<b>SUBTOTAL:</b>	<b>\$230.48</b>
<b>TAX:</b>	<b>\$46.10</b>
<b>SHIPPING:</b>	<b>\$100.00</b>
<b>TOTAL:</b>	<b>\$276.58</b>



Invoice from Microsoft templates library

**VISION CORP**

26 April 2024

**FACTURE N° 652**

Facturer à		Expédier à	
<b>Client</b>	Irene Moore	<b>Destinataire</b>	Irene Moore
<b>ID client</b>	374	<b>Adresse</b>	123 Avenue A, Métropole
<b>Adresse</b>	123 Avenue A, Métropole	<b>Téléphone</b>	(123) 456-7890
<b>Téléphone</b>	(123) 456-7890		
<b>Paiement dû</b>	30 décembre 2022	<b>Date de livraison</b>	7 décembre 2022
<b>Vendeur</b>	Luca Richter	<b>Mode d'expédition</b>	Transport
<b>Conditions de paiement</b>	Espèces ou chèque	<b>Conditions d'expédition</b>	Retours non acceptés

Qté.	Numéro d'article	Description	Prix unitaire	Remise	Total de la ligne
2	13	Boy's Hoodie (Grey)	12.64		25.28
3	44	Women's Coat (Black)	39.32		117.96
				Remise totale	0
				Sous-total	143.24
				Taxe de vente	28.65
				<b>Total</b>	<b>171.89</b>

**Nous vous remercions de votre confiance !****Vision Corp**

456 N'importe où, n'importe où | siteintéressant.com  
t. (123) 987-6543 | f.(123) 987-6542 | le plus heureux@exemple.com



## EMP table with two childs

The following sample illustrates the rendering of a master with two detail parts:

- Direct reports
- Other people working in the same department

cf emp template and /emp/handler

# SCOTT

## ANALYST

Content  
[Direct reports](#)  
[Employees in RESEARCH](#)  
[History](#)  
[Skills](#)

### Direct reports

ADAMS	CLERK	12/01/1983
-------	-------	------------

Total: **1**

### Employees in RESEARCH

JONES	MANAGER	02/04/1981
FORD	ANALYST	03/12/1981
SMITH	CLERK	17/12/1980
ADAMS	CLERK	12/01/1983

Simple lists



# Orders

ID	ORDER ID	PRODUCT	QUANTITY	URL	PRICE	MNT
25/08/2023	652	Irene Moore	374		1	Online
25/08/2023	653	Diana Fowler	293		1	Online
25/08/2023	654	Desmond Lifschitz	169		1	Online
25/08/2023	655	Hung Pedroncelli	197		1	Online
25/08/2023	656	Santiago Vautus	70		1	Online
26/08/2023	657	Florencia Kohler	322		3	Seattle
26/08/2023	658	Brooks Craker	112		1	Online
26/08/2023	659	Shamira Jones	58		1	Online
27/08/2023	660	Kristin Brooks	297		1	Online

# Liste des colonnes

COL	NAME	TABLE	TYPE	URL	NULLABLE	ID
owner	XMLTYPE001	GFM_DATA	XMLTYPE		Y	NO
	GEOMETRY	EBA_SAMPLE_MAP_STATES	SDO_GEOMETRY		Y	NO
	GEOMETRY	EBA_SAMPLE_MAP_SIMPLE_STATES	SDO_GEOMETRY		Y	NO
	GEOMETRY	EBA_SAMPLE_MAP_AIRPORTS	SDO_GEOMETRY		Y	NO
3	ID	FAN_SETTINGS	NUMBER		N	NO
1	DATEP	FAN_SETTINGS	DATE		Y	NO
15	CODE	FAN_SETTINGS	VARCHAR2		Y	NO
8	LOC	DEPT	VARCHAR2		Y	NO
10	DNAME	DEPT	VARCHAR2		Y	NO
3	DEPTNO	DEPT	NUMBER		N	NO
4	SAL	EMP	NUMBER		Y	NO
4	MGR	EMP	NUMBER		Y	NO
8	JOB	EMP	VARCHAR2		Y	NO
8	HIREDATE	EMP	DATE		Y	NO
6	ENAME	EMP	VARCHAR2		Y	NO
4	EMPNO	EMP	NUMBER		N	NO
3	DEPTNO	EMP	NUMBER		Y	NO
2	COMM	EMP	NUMBER		Y	NO
46	ROUND_PIC	FAN_MEMBERS	VARCHAR2		Y	NO
6	PID	FAN_MEMBERS	VARCHAR2		Y	NO



## Author



[Patrick](#)

GPM Factory