How to use the Template
Blue text *is always intended* as instructions, guidelines, explanations, hints, and tips. It must be removed when the document will be finalized in order to provide a consistent representation *of this document to the client.*
Be sure to delete these instructions when you have finished!

DEVELOPMENT & INTERFACE SPECIFICATIONS

DA-3214 : APO - SWO : NEW API CHECK AND VALIDATE ORDERS

Development Identification		
WRICEF Code and Name	**ZGAP or ZFIN Code and Name**	**JIRA Code**
DA-3214 : APO - SWO : NEW API CHECK AND VALIDATE ORDERS		NEWASK-12813

Project Identification			
Project Name	SWO	**Client Name**	DECATHLON
Project Manager		**Decathlon Project Manager**	
CEX for BILLING			

Contact Information and Responsible Party			
Functional Responsible²	Erwan NALEWAJEK	**Third party dev/tech Contact**	
Internal dev/tech contact			
Exploitation team SENDER PART *(HP-Saw Group names and @mail adress)*		**Exploitation team RECEIVER PART** *(HP-Saw Group names and @mail adress)*	
RUN CEX			

Document Identification			
Author	Erwan NALEWAJEK -CROIX-	**Document Location**	SWO
Version	**Status**	**Date (DD/MM/YYYY)**	**Classification**
0.1	Delivered for approval	01/10/2023	Confidential

Revision History			
Version	**Date**	**Sections revised**	**Description**
0.1	01/10/2023	All document	Initialisation of the document

Responsible		
Interlocutor	**Main**	**Backup**
Business		
MOE	Erwan NALEWAJEK -CROIX-	MENET Mathieu
Development Shared Services	SAP DEV	
Support		

Review and Approval			
Name Decathlon Approver	**Approved (Yes/ No)**	**Comment**	**Date (JJ/MM/AA)**
DATA EXCHANGE COMMITTEE	Y	DOCUMENT	05/10/2023

TABLE OF CONTENTS

1°) Purpose of this document

The Specifications are the basis for the developments that will be done. They have to be verified and approved formally.

The purpose of this document is to deliver the detailed functional specifications describing in detail the functional requirements associated to this development.

1.1 Out of scope

The following items are out of scope

1.2 Assumptions

The following assumptions have been taken

1.3 Dependencies

Any dependency to which the business process completion is contingent upon.

The following Dependencies have been taken

1.4 References

References	
Type	**Name**
GAP decision paper	
Others	

2°) Outstanding Issues/open points

Id	Description	R (1)	Planned	Statut (2)

			Closed Date	
	Short description of the open point. Once closed, you may want to record the decision / answer			
01				
02				
03				

(1) : R : person in charge of closing the point

(2) : OPEN / CLOSED

3°) Purpose of the Demand

Summary					
SENDER	**Mapping before sending**	**FLOW technology**	**Mapping in Flow platform**	**Mapping at destination**	**RECEIVER**
SAP : [ ] **EWM** [ ] **WCS** [X] **APO**	No Mapping required	ODATA / API	NO	NO	NON SAP : [ ] **WCS** [X] **OMS**
DESCRIPTION	The API to be developed in this specification will be used to check the Direct Import (SWO) order demands and create Planned Orders in the Planning Book PE_MPS (version 000). OMS will be the system that will consume the new API				

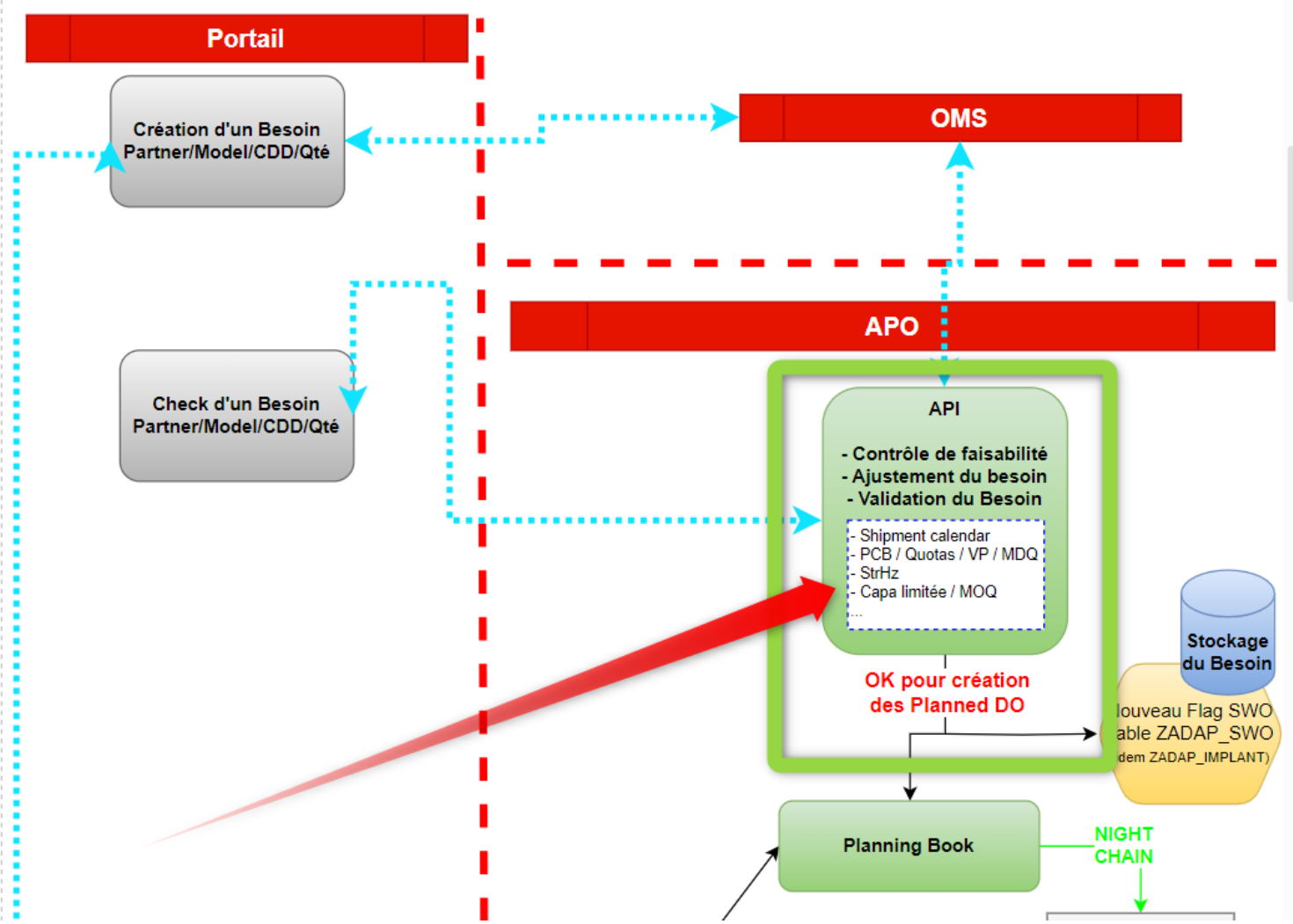
4°) Detailed FLOW Specifications

FLOW SPECIFICATION				
WHAT	**OBJECTIVES**	**Team**	**MANDATORY (Yes/No)**	**FILLED Yes/No)**
FUNCTIONAL DESCRIPTION	Check and Validate orders coming from SWO portail	**Project**	**Yes**	**YES**
BUSINESS IMPACT				
FLOW FEATURES				
TECHNICAL INFORMATIONS				
TESTS SETS				
EXPLOITATION				

4.1 Functional description of the flow (description fonctionnelle du flux)

Please insert the description of the process here
<Example: SAP use direct webservice with osmose for their pre-order fonctionnal process. They want to use PO to call SAP webservice (as for this existing flow)>
<TO FILL>

The following capture highlights the most important steps that concerns our logic:



Main Flow : https://app.diagrams.net/#G1yoYXiwVSAgtEMq2Ak_VeMtGM0qFz6pLu

The portail will call the API on APO by sending a list of SKU (i.e. products/materials) / Delivery Date / Quantities.
The goal is to check the consistency of this datas and return to the portail the feasibility of the demand, and eventually adapt the quantities and the delivery date.

If all the lights are green, the portail will send the creation order demand to OMS. This one will create a new order number and send to APO the creation order demand.
A last check is done by APO, in case the datas have been modified by the portail.

This API has two functionalities :

- Return to the portail the feasibility of its demand according to our Supply constraints on APO.
- Create Planned DOs on APO if OMS send us a “creation request” for all lines of its demand.

4.2 Business impact

INFORMATIONS RELATED TO THE KINEMATIC *(Informations liées à la cinématique)*	
SENDER (Application name, not project name)	APO
SENDING PROCEDURE	On the fly
SENDER TECHNOLOGY	ODATA-API
RECEIVING APPLICATION NAME (not platform)	Portail or OMS
RECEIVER TECHNOLOGY *Techno destinataire*	ODATA-API
EXCHANGE FORMAT	JSON
TRANSFORMATION NEEDED *Besoin en transformation / Mapping ?*	NO
Migration?	NO
Standard format? *format standard B2B?*	NO

4.4 Flow Features (caractéritiques du Flux)

FLOW FEATURES (Caractéristiques du flux)	
SLA	*2h*
Frequency *Fréquence*	*10 / day*
Avg Size of message *Taille d'1 message*	

Min & Max Size of message (if variable)	*N/A*
Peak requirements *Besoin d'engagement en pic*	*200 000 materials in case of initial load or mass change*
Peak Period	*Rodrigues*
Acceptable Response time (from a business point of view)	*Updates will run every 15min*
Recovery time Objective (After a major crash)	*12h*
Legal archiving required	*No*
Data Confidentiality required	*No*
Routing (*Routage)*	*No*
Filtering (*Filtrage)*	*No*
Volumetry repartition (multi sender/receiver)	*N/A*
Maintain duplicates (Gestion des doublons)	*No*
Resend mode (*Mode de rejeu)*	*If needed, EWM will manage the resending via ELOG472*
Business Tags in follow up tool (GoldenEyes)	*No*
Initialization (*Phase d'initialisation)*	*Yes, managed out of EWM*
Estimated Lifetime *durée de vie estimée*	*20 years*
Load Mode (Dataservice)	*<Full / Delta>*
Retention period	

4.6 Technical Informations (Informations techniques)

Systems and Applications (Serveurs et applis)		
PROD	**Sender server PROD** (Serveurs emission)	**PA1**
	Sender Servers load balanced or switched to alternate @IP?	
	Reception server PROD (Serveurs reception)	**Portail/OMS**
	Reception Server load balanced or switched to alternate @IP?	
Pre-PROD	**Sender server PrePROD** (Serveurs emission)	**RA2**
	Reception server PrePROD (Serveurs reception)	**Portail/OMS**
Qual	**Sender server Qual**(Serveurs emission)	**QA2**
	Reception server Qual(Serveurs reception)	**Portail/OMS**

DEV	**Sender server DEV** (Serveurs emission)	**N/A**
	Reception server DEV (Serveurs reception)	**N/A**

4.7 Tests Sets

example of url and json

TEST Sets	
Test set	
Test description	

4.8 Trace

	YES	NO
ZAPI Framework		X

4.9 Transcodification

No Transcodification is required

4.10 Support / Exploitation Part

Errors processing
<Description of errors and how correct them>
< La description des erreurs qui peuvent se produire, et plus important, comment ils doivent être traités>

DATA ERRORS (*Erreur de données*)			
Error	**Emergency** ***Urgence***	**Description**	**Action**
<Ex1: Error message> Message erreur>	<High/Medium/Low *Haut/Moyen/Bas*>	<The customer number is not found in the message *Le numéro du client n'est pas dans le message*>	<Customer number is compulsory.==> keep message in error queue + send error in Goldeneyes + wait for manual correction & resubmit>

SYSTEM ERRORS (*Erreurs systèmes*)			
Error	**Urgence**	**Description**	**Action**
Example: BODI-1241021	*High/Medium/Low* *Haut/Moyen/Bas*	*Job server : xxx – no response*	*Contact the team system to restart job server*

APPLICATION ERRORS (Erreurs Applicatives)
<All errors regarding application/functional / *Toutes les erreurs applicatifs/fonctionnels*
- The customer doesn't exist in SAP ECC / *Le client n'existe pas dans SAP ECC*
- No stock / *Il n'y a plus de stock*

Erreur	Urgence	Description	Action
Example: Wrong measure unit	High Haut Medium Moyen Low Bas	The measure unit is unknown in SAP system. Order can't be created	*- Inform the partner. He has to change the unit and send a new message.*

4.11 Testing Phase

a) Test Scenarios
Unit testing
SOAPUI to test the call of webservice

Unit testing (Waiting error)
Flow testing (Waiting success)
Flow testing / *Tests de bout en bout* (Waiting error / *attente d'échec*)

b) Test results
Scores
Check list

Description	OK	Remarks
Connection Sender system		
Connection Receiver system		
No mapping errors		
Application errors caught properly		
System errors caught properly		
Data successfully received from Sender in PI		
Data successfully delivered to Receiver from PI		
Routing in PI is successful (if applicable)		
Security checks out ok (if applicable)		
Interface is completely documented (including mapping)		
All test scenarios run		
End-to-end test in processes are all succesful		

5°) Detailed Specifications CALLER part.

DESTINATION		
SAP		

Summary		
System	**Mdt**	**Technology**
OMS		
Title		
Description		
Running Mode		
Performance Considerations		
Complexity		
Priority		

OBJECT DETAILS				
NEW / CHANGE	OBJECT NAME	OBJECT TYPE	Description of Object	Complexity (Simple / Medium / Complex)

5.1 Development's Objectives

This section is used to provide a general description of the development to be performed

Objectives

5.2 Fields

Field EWM	Field name	Type	Size	Comment

5.2 Overall process diagram

This section lists the principal development steps

Process Diagram
-

5.3 Logic Flow

<This section is used to describe how the receiving system reacts upon receiving the message.>

LOGIC FLOW

Description

Rules *(règles de gestion)*	
RG001	
RG002	

Error Messages (messages d'erreur)	
ER001	
ER002	

5.4 Traces to ZAPI Framework

	YES	NO
Send a trace To API framework		

5.5 MAPPING

MAPPING FIELDS TO SOURCE / TARGET

If the Trans codification is required you need to fill-up this part.

5.6 Translation

N/A

5.7 TESTS

UNIT TESTS

~~Scenario #~~	~~Input Selection Criteria~~	~~Expected Result~~

6°) Detailed Specifications RECEIVER part.

Summary		
System	**Mdt**	**Technology**
APO	*Mdt*	ODATA / API
Title		**Check order creation demand and create planned DO in the Planning Book**
Description and detailed specification		**Name of the API : planned_delivery_orders** The portail and OMS systems can call the new API in APO, with a given list of SKU/Qty/Delivery Date and a Partner. From now on, any application that would need to consume the API will be called the "*API Caller*". [SKU = material = product = ITEM_CODE] For a given list of materials (ITEM_CODE), this new API is meant to: - Read the master data and check the feasibility of the demand (**called the "*Check Process*" or API Verb "POST"**) ⇒ **url : api/v1/planned_delivery_orders/simulations** - Create the corresponding Planned DOs in the planning Book (**called the "*Creation Process*" or API Verb "POST"**) ⇒ **url : api/v1/planned_delivery_orders** - Delete a PARTNER_REQUEST_NUMBER and its Planned DOs if it is not yet sent to ECC (**called the "*Deletion Process*" or API Verb "DELETE"**) ⇒ **url api/v1/planned_delivery_orders/"&partner_request_number"** - Render both these results to the *API Caller* The "*API Caller*" needs to call the new API by providing **these input parameters:** - **A table input parameter** containing one or multiple SKU/Qty/Date for a given Partner : **INPUT table parameters** **Name of the API Input parameter of the Input table** / **Field meaning** / **Business Rule to respect** partner_request_number / Partner Request Number / *Format : CHAR20* ⇒ Should be empty in the "*Check Process*", ⇒ Must be filled if the "*Creation Process*" or in the

		"*Deletion Process*", else Return API REST 400 with the text Err_016
partner_gln	Partner Number	*Format : CHAR20* Must be filled in the "*Creation Process*" or in the "*Check Process*", else Return API REST 400 with the text Err_001
item_lines	Array	**All lines below are in this structure**
item_code	Material Number	*Format : CHAR40* Must be filled in the "*Creation Process*" or in the "*Check Process*", else Return API REST 400 with the text Err_002
requested_delivery_date	Delivery Date	*Format : CHAR10 - "YYYY-MM-DD"* Must be filled in the "*Creation Process*" or in the "*Check Process*", else Return API REST 400 with the text Err_003
requested_quantity	Quantity to order	*Format : INT* Must be filled in the "*Creation Process*" or in the "*Check Process*", else Return API REST 400 with the text Err_004

Here is an example of the JSON :

```
{
 "partner_request_number": "1234567890",
 "partner_gln": "VADA0027839",
 "item_lines":[{
  "item_code": "2467489",
  "requested_delivery_date": "2023-11-30",
  "requested_quantity": 10000
 }]
}
```

The API will need to use these input parameters to return **a table** with the **following output parameters** respecting these rules:

OUTPUT table parameters		
Name of the Output parameter of the Output table returned by the API	**Field meaning**	**Business Rule to respect**
partner_request_number	OMS Order Number	*Format : CHAR 20*
partner_gln	Partner Number	*Format : CHAR20*
destination_description	Description of the location	*Format : CHAR40*

is_request_accepted	Request accepted	*Format : CHAR 2* Values : true or false
item_lines	Array	**All lines below are in this structure**
item_code	Material Number	*Format : CHAR40*
~~destination_description~~	~~Description of the location~~	~~*Format : CHAR40*~~
minimum_delivery_quantity	MDQ (model level)	*Format : INT*
adjusted_delivery_date	Delivery Date	*Format : CHAR10 -* *"YYYY-MM-DD"*
is_delivery_date_accepted	Accepted Delivery Date	*Format : CHAR 1* Values : true or false
adjusted_quantity	Quantity to order	*Format : INT*
is_quantity_accepted	Accepted Quantity	*Format : CHAR 1* Values : true or false
~~is_request_accepted~~	~~Request accepted~~	~~*Format : CHAR 2*~~ ~~Values : true or false~~

Here is an example of the JSON :

```
{
  "partner_request_number": "1234567890",
  "partner_gln": "VADA0027839",
  "destination_description": "Walmart Port Shangai",
  "is_request_accepted": "X",
  "item_lines":[{
    "item_code": "2467489",
    "minimum_delivery_quantity": 50,
    "adjusted_delivery_date": "2023-11-30",
    "is_delivery_date_accepted": "X",
    "adjusted_quantity": 10000,
    "is_quantity_accepted": "X",
  }]
}
```

For information, when an API call is received, it can be one of these three processes :

- "*Deletion Process*" ⇒ the "*API Caller*" is OMS
- "*Creation Process*" ⇒ the "*API Caller*" is OMS
- "*Check Process*" ⇒ the "*API Caller*" is the Portail

DELETION PROCESS

In the "*Deletion Process*", we have to delete the Planned DO previously created by the OMS order.

Search for all the records in the table ZAPLA_SWO with PARTNER_REQUEST_NUMBER = INPUT-PARTNER_REQUEST_NUMBER.

If already deleted (ZAPLA_SWO-DELETED = 'X'), Return API REST 201.

If the flag ZAPLA_SWO-SENT_TO_ECC is not empty, the reason is that the Planned DO has already been transformed to a ECC PO Order or is in progress, so it's too late to delete the objects.
Return API REST 400 with the text Err_005.

If there are no records in ZAPLA_SWO, Return API REST 404 with the text Err_006.

If the lines are found in ZAPLA_SWO, put all the ORDID in the function module BAPI_POSRVAPS_DELMULTI like it is done in this program :

Include ZADAP_TRANSF_MTS_DEP_TO_DO_F03 Active

```
288      lv_version = p_vrsid.
289
290      CALL FUNCTION 'BAPI_POSRVAPS_DELMULTI'
291        DESTINATION 'NONE'
292        EXPORTING
293          logical_system   = lv_logsys
294          commit_control   = 'E'
295          planning_version = lv_version
296          order_type       = '1' "For Purchase Requisitions
297          event_control    = lc_event_control_1
298        TABLES
299          order_keys       = lt_deployables_to_del
300          return           = lt_bapi_del_logs.
301
```

If no error in the RETURN table, put all the lines concerning this PARTNER_REQUEST_NUMBER in the OUTPUT table, and Return API REST 201 and IS_REQUEST_ACCEPTED = "true".
Update fields DELETED, DEL_DATE and DEL_TIME in the table ZAPLA_SWO.
Deleted corresponding rows in ZADAP_FIRM_DO.

Else Return API REST 400 with the text Err_007.

CHECK PROCESS

In the "*Check Process*", we have to check the feasibility of the Planned DO creation.

For each line of the INPUT table, we have to do these actions :

SKU Format
Convert the ITEM_CODE number in the APO MATNR format with the function module CONVERSION_EXIT_MATN1_INPUT.

Control of the couple Entry Point / Product existence
With the PARTNER_GLN, read all the ENTRY_POINT in the table ZAPLA_SWO_EP.
The INPUT-ITEM_CODE must exist only on one of these ZAPLA_SWO_EP-ENTRY_POINT.
Read the view /SAPAPO/V_MATLOC with MATNR = INPUT-ITEM_CODE and LOCNO = ZAPLA_SWO_EP-ENTRY_POINT.
If more than one ENTRY_POINT has a line in /SAPAPO/V_MATLOC, Return API REST 400 with the text Err_008.
If no ENTRY_POINT has a line in /SAPAPO/V_MATLOC, Return API REST 400 with the text Err_009.

Control of the Vendor Plant existence
Search the Vendor Plant in the table ZADAP_SUPPLY with WERKS = ZAPLA_SWO_EP-ENTRY_POINT and SKU = INPUT-ITEM_CODE.
We must find only one line.
If more than one line has been found, Return API REST 400 with the text Err_010.
If no line has been found in the table ZADAP_SUPPLY, Return API REST 400 with the text Err_011.

Grouping rules
The Portail can send to APO a global file with multiple MODEL and multiple Delivery Date.
First of all, we have to group these lines by the couple MODEL/REQUESTED_DELIVERY_DATE.
All the rules below must be done for this couple, because the finality of this flow is to create a Purchase Order which must be grouped by Entry Point / Vendor Plant / Model / Delivery Date.
For information the MODEL value can be found in the table ZADAP_SUPPLY read previously.

Control of the PCB
You can find the PCB value in the field /SAPAPO/MATLOTSZ-BSTRF (joined with /SAPAPO/V_MATLOC with LSZID).
Check if INPUT-REQUESTED_QUANTITY is a multiple of BSTRF.
If yes, put the INPUT-REQUESTED_QUANTITY in the OUTPUT-ADJUSTED_QUANTITY.
If no, increase the INPUT-REQUESTED_QUANTITY until it is a multiple of the PCB, then put this new value in OUTPUT-ADJUSTED_QUANTITY and put "false" in OUTPUT-IS_QUANTITY_ACCEPTED.

Control of the MDQ (ie Minimum Delivery Quantity)
You can find the MDQ value in the table ZADAP_SUPPLY read previously. It must be the same for all SKU of the ZADAP_SUPPLY-MODEL (the MDQ value is concerning the sum of all the SKU of the MODEL), and must be applied for one Delivery Date.
For all the SKU of the INPUT table (grouped by the MODEL / Delivery Date), do the sum of INPUT-REQUESTED_QUANTITY.
If the sum is lower than ZADAP_SUPPLY-MDQ, we have to increase the REQUESTED_QUANTITY of each ITEM_CODE, proportionally to the value of each ITEM_CODE, and by respecting the PCB of each ITEM_CODE, until the sum becomes higher or equal than the MDQ value.
For each ITEM_CODE, put this new value in OUTPUT-ADJUSTED_QUANTITY and put "false" in OUTPUT-IS_QUANTITY_ACCEPTED.
If the sum is higher than ZADAP_SUPPLY-MDQ, put the INPUT-REQUESTED_QUANTITY in the OUTPUT-ADJUSTED_QUANTITY.

Control of the Transportation Lane (will be called TLane below)
The Tlane of a couple Entry Point / Vendor Plant must exist in APO, else the delivery can't be done.
Read the view /SAPAPO/V_TRM with LOCNOFR = ZADAP_SUPPLY-VENDOR_PLANT, LOCNOTO = ZADAP_SUPPLY-WERKS, and TRNAME = "000".
If no line is found, Return API REST 400 with the text Err_012.
If multiple lines are found, Return API REST 400 with the text Err_013.

Control of the Delivery Date
First, the Delivery Date must be higher than the first day of the week of the date ZADAP_SUPPLY-DO_FIRM_DATE.
If it's not the case, change the OUTPUT-ADJUSTED_DELIVERY_DATE value by the first day of the week of the date ZADAP_SUPPLY-DO_FIRM_DATE.
(if the date has been changed, don't forget to put "false" in OUTPUT-IS_DELIVERY_DATE_ACCEPTED when we will send the result of the API)

Then, we have to check if a shipment calendar exists on the couple Entry Point / Vendor Plant.
To do this, read first the table ZADAP_SHIP_CAL with LOCNOFR = ZADAP_SUPPLY-VENDOR_PLANT, LOCNOTO = ZADAP_SUPPLY-WERKS, and TTYPE = /SAPAPO/V_TRM-TTYPE.
If a shipment calendar has been found, read the table /SAPAPO/TCALPERI with TSTRID = ZADAP_SHIP_CAL-TSTRID.
Sort the lines by PPBIS descending and check if at least one value is higher than (SY-DATUM + 2). If not, there is no active shipment calendar.

We have to calculate the Start Date of the delivery :
The field /SAPAPO/V_TRM-DURAT contains the transport duration (in hours).
So we must divide it by 24 to have the number of days (the divider is 240000 with the format of DURAT, not directly 24).
Then pull off this value from the INPUT-REQUESTED_DELIVERY_DATE to obtain the Start Date.
If this Start Date is lower than (SY-DATUM + 2), put SY-DATUM + 2 in the Start Date.
Then search for the first higher date after the Start Date in the table TCALPERI read before.

If found, add the DURAT to this date, put it into OUTPUT-ADJUSTED_DELIVERY_DATE and “true” in OUTPUT-IS_DELIVERY_DATE_ACCEPTED (if adjusted=original_requested date) or “false” in OUTPUT-IS_DELIVERY_DATE_ACCEPTED (if adjusted<>original_requested date).

If no shipment calendar has been found, put directly the INPUT-REQUESTED_DELIVERY_DATE in the OUTPUT-ADJUSTED_DELIVERY_DATE and “true” in OUTPUT-IS_DELIVERY_DATE_ACCEPTED.

IF ANY ADJUSTMENT HAS BEEN DONE for the delivery date and/or the quantity, Return API REST 200 and IS_REQUEST_ACCEPTED = “false” for all lines

ELSE Return API REST 200 and IS_REQUEST_ACCEPTED = “true” for all lines

MDQ and Destination Description
In all OUTPUT lines returned by the API, we have to fulfill the fields *minimum_delivery_quantity* and *destination_description* :

- minimum_delivery_quantity : put the value found in ZADAP_SUPPLY-MDQ
- destination_description : put the value of DESCR40 found in the table /SAPAPO/LOCT readed with SPRAS = ‘E’ and LOCID = /SAPAPO/V_MATLOC-LOCID (of the ENTRY_POINT : ie the part **Control of the couple Entry Point / Product existence**)

CREATION PROCESS

If the PARTNER_REQUEST_NUMBER is filled, we have to check the feasibility of the Planned DO creation (done in the “*Check Process*”, so this part must be also executed), and then create the Planned DO if no adjustment has been done (i.e. all lines of the OUTPUT table have IS_REQUEST_ACCEPTED = “true“)

For the Planned DO creation, we will use the function module BAPI_POSRVAPS_CREATESNPORDER, like it is done in this program :

```
lude                    ZADAP_SPLIT_PR_MULTI_ITEMS_F01  Active
210   * Create the new Planned DOs
211     CALL FUNCTION 'BAPI_POSRVAPS_CREATESNPORDER'
212       EXPORTING
213         logical_system         = lv_logsys
214         application            = 'SNP'
215         ext_number_assignment  = ' '  "To force the system to create
216         commit_control         = 'E'
217         plng_version           = lv_version
218         event_control          = '1'  "To force the system to create
219         check_source_existence = 'X'
220       TABLES
221         order_data             = it_dos_2_creat_of_multitem_prs
222         mapping_data           = ct_new_monoitem_dos_created
223         return                 = ct_creation_dos_logs.
224
```

Attention : the parameter CHECK_SOURCE_EXISTENCE must be equal to ABAP_FALSE.

Here are the actions to execute for the "*Creation Process*" :

Check OMS order number existence
If the PARTNER_REQUEST_NUMBER already exists in the table ZAPLA_SWO, Return API REST 400 with the text Err_015.

Check process
The whole part of the "*Check Process*" above must be executed and the result must be OK for all the lines and without any adjustment.

Fill the BAPI table and launch it to create Planned DO
Each line of INPUT must create a Planned DO.
The ORDER_DATA table must be filled like this :

ORDER_DATA Field	Value	Notes
ORDER_NUMBER	Concatenate "&" with an increment type N. *(exemple : you create an abap variable TYPE N(6), you concatenate and obtain the value &000001 then &000002, etc…)*	
PRODUCT	INPUT-ITEM_CODE	Converted in the APO MATNR format with the

		function module CONVERSION_EXIT_MATN1_INPUT
LOCATION_FROM	ZADAP_SUPPLY-VENDOR_PLANT	
LOCTYPE_FROM	1001	
LOCATION_TO	ZADAP_SUPPLY-WERKS	
LOCTYPE_TO	1002	
TTYPE	/SAPAPO/V_TRM-TTYPE	
FIXED	“X”	
QUANTITY	Quantity calculated in the “*Check Process*”	
ORDER_START_DATE	Start Date calculated in the “*Check Process*”	
ORDER_END_DATE	Delivery Date calculated in the “*Check Process*”	
ATPCAT_TO	“ZG”	
ATPCAT_FROM	“ZH”	

If an error occurs during the creation by the BAPI, all the Planned DO will be rollbacked, so if the RETURN table contains any line with an error A or E, Return API REST 400 with the text Err_014.

If no error occurs, Return API REST 201 and IS_REQUEST_ACCEPTED = “true”.

Save the Creation in the table SWO

The list of the Planned DO created is in the MAPPING_DATA table of the BAPI_POSRVAPS_CREATESNPORDER.

For each line of OUTPUT, we have to create a line in the table ZAPLA_SWO.

Note : Read the table MAPPING_DATA with ORDER_NUMBER = the value given to the BAPI in the field ORDER_NUMBER (exemple : “&000001”)

ZADAP_SWO	Value

PARTNER_REQUEST_NUMBER	INPUT-PARTNER_REQUEST_NUMBER
ORDID	MAPPING_DATA-ORDID
LOCID_FROM	Read the table /SAPAPO/LOC with LOCNO = ZADAP_SUPPLY-VENDOR_PLANT and get LOCID value
LOCID_TO	Read the table /SAPAPO/LOC with LOCNO = ZADAP_SUPPLY-WERKS and get LOCID value
MATID	/SAPAPO/V_MATLOC-MATID
ORDNO	MAPPING_DATA-INT_ORDER_NUMBER
VENDOR_PLANT	ZADAP_SUPPLY-VENDOR_PLANT
ENTRY_POINT	ZADAP_SUPPLY-WERKS
MATNR	INPUT-ITEM_CODE *Converted in the APO MATNR format with the function module CONVERSION_EXIT_MATN1_INPUT*
ERDAT	SY-DATUM
ERZET	SY-UZEIT

Then add as line as orders in the table ZADAP_FIRM_DO :

ZADAP_FIRM_DO	Value
ORDERNUMBER	MAPPING_DATA-ORDID
BUCKET	Launch the function module DATE_GET_WEEK with the delivery date calculated in the "*Check Process*"
LOCATION	ZADAP_SUPPLY-WERKS
PRODUCT	INPUT-ITEM_CODE
ERNAM	SY-UNAME
ERDAT	SY-DATUM
ERZET	SY-UZEIT

Here are the texts of the messages Err_xxx used in this document :

Error code	API REST (RFC 2616)	Error message
Err_001	400	GLN Partner Number is empty.
Err_002	400	Item code is empty.
Err_003	400	Requested Delivery Date is empty.
Err_004	400	Requested Quantity is empty.
Err_005	400	This Order is already converted into a Purchase Order. Contact your support to delete it.
Err_006	404	No orders found on APO. Contact your support.
Err_007	400	Error occurred when deleting orders on APO. Contact your support.
Err_008	400	Multiple Entry Point found for the Product &1. Contact your support.
Err_009	400	No Entry Point found for the Product &1. Contact your support.
Err_010	400	Multiple Vendor Plant found for the Product &1. Contact your s upport.
Err_011	400	No Vendor Plant found for the Product &1. Contact your support.
Err_012	400	No Transportation Lane found. Contact your support.
Err_013	400	Multiple Transportation Lane found. Contact your support.

	<table><tr><td>Err_014</td><td>400</td><td>An error occurs during the creation of orders. Contact your support.</td></tr><tr><td>Err_015</td><td>400</td><td>Partner Request Number has already been created on APO.</td></tr><tr><td>Err_016</td><td>400</td><td>Partner Request Number is requested.</td></tr></table>
Running Mode	Real Time
Performance Considerations	On the fly
Complexity	Simple
Priority	High

Here are the new tables to create in the DDIC :

OBJECT DETAILS					
NEW / CHANGE	OBJECT NAME	OBJECT TYPE	Description of Object	Complexity (Simple / Medium / Complex)	Manageable (SM30)
NEW	ZAPLA_SWO	TABLE	SWO Orders	S	Y
NEW	ZAPLA_SWO_EP	TABLE	SWO Entry Point Mapping	S	Y

Here are the objects to create :

ZAPLA_SWO (The table ZAPLA_SWO have to be managed by SM30)				
FIELD	KEY	Data element	Description of Object	Length
MANDT	X	MANDT	Client	CLNT 3
PARTNER_REQ	X	ZADAP_PARTNER_REQ	OMS Number	CHAR 20

UEST_NUMBER		UEST_NUMBER		
ORDID	X	/SAPAPO/OM_ORDERUID	Internal Number (UID) for an Order in APO	CHAR 22
LOCID_FROM	X	/SAPAPO/LOCID	Internal Location Number	CHAR 22
LOCID_TO	X	/SAPAPO/LOCID	Internal Location Number	CHAR 22
MATID	X	/SAPAPO/MATID	Internal Number (UID) for Product	CHAR 22
ORDNO		/SAPAPO/OM_ORDNO	Order Number	CHAR 12
VENDOR_PLANT		/SAPAPO/LOCNO	Location	CHAR 20
ENTRY_POINT		/SAPAPO/LOCNO	Location	CHAR 20
MATNR		/SAPAPO/MATNR	Product Number	CHAR 40
ERDAT		ERDAT	Creation Date	DATS 8
ERZET		UZEIT	Creation Time	TIMS 6
SENT_TO_ECC		/SAPAPO/FLAG	Sent to ECC Indicator	CHAR 1
SENT_DATE		/SAPAPO/SENDDATE	Sending Date	DATS 8
SENT_TIME		/SAPAPO/SENDTIME	Sending Time	TIMS 6
DELPR		/SAPAPO/OM_DELNR	Order Number from Connected OLTP System	CHAR 12
DELPS		/SAPAPO/OM_DELPS	Order Item	NUMC 6
PO_DATE		ERDAT	Creation Date	DATS 8
PO_TIME		UZEIT	Creation Time	TIMS 6
DELETED		/SAPAPO/DELETION_FLAG	Deletion Flag	CHAR 1
DEL_DATE		ERDAT	Creation Date	DATS 8
DEL_TIME		UZEIT	Creation Time	TIMS 6

ZAPLA_SWO_EP (The table ZAPLA_SWO_EP have to be managed by SM30)				
FIELD	KEY	Data element	Description of Object	Length
MANDT	X	MANDT	Client	CLNT 3
PARTNER_GLN	X	ZADAP_PARTNER_GLN	Partner Number (GLN)	CHAR 20
ENTRY_POINT	X	/SAPAPO/LOCNO	Entry Point	CHAR 20
MAIN_CAC	X	/SAPAPO/LOCNO	Main CAC	CHAR 20

6.1 Development's Objectives

This section is used to provide a general description of the development to be performed

Objectives

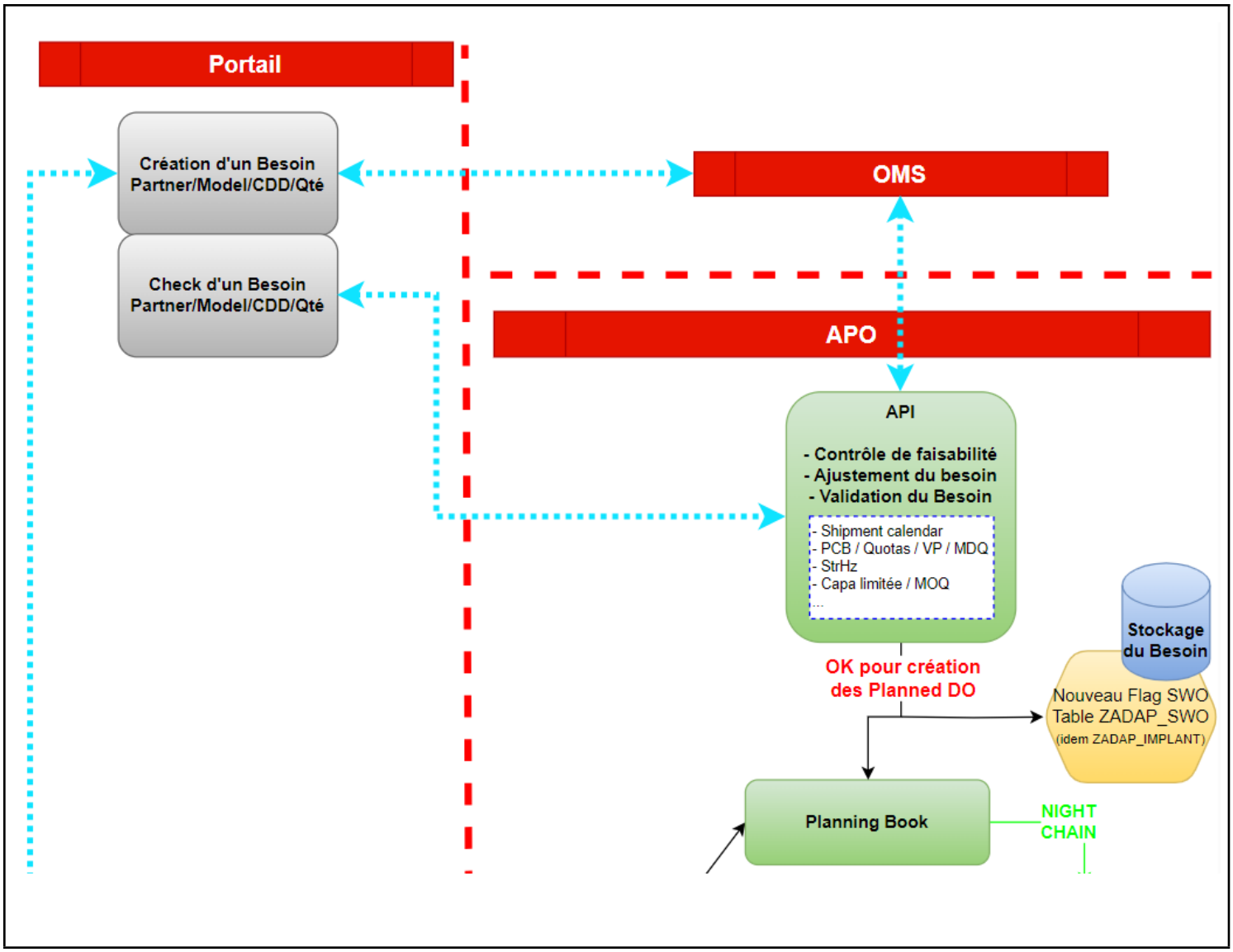
6.2 Fields

Field EWM	Field name	Type	Size	Comment

6.3 Overall process diagram

This section lists the principal development steps

Process Diagram



6.4 Traces to ZAPI Framework

	YES	NO
Send a trace To API framework		X

6.5 MAPPING

MAPPING FIELDS TO SOURCE / TARGET

CF the “Summary” explanation in the above paragraph “6°) Detailed Specifications”

If the Trans codification is required you need to fill-up this part.

N / A

6.6 Translation

Translation				
What	**Reference**	**EN**	**FR**	**XX**
Program Title	Z_TEST	SEND DATA #1	Envoi des données n°1	EXAMPLE
Text element	T001	Name	Nom	EXAMPLE
Transaction Code	ZT01	VIEW EMPLOYEES	Visualisation Salarié	EXAMPLE
TABLE Name	ZAUTH01	Authorization for xxxx	Autorisation pour xxx	EXAMPLE

6.7 TESTS

UNIT TESTS

DA-3214 Test Scenario

Scenario #	Input Selection Criteria	Expected Result

6.12 Security

Security		
	Descritpion of criticity	Protection means
Confidentiality of transported datas: Does the flow contain Stratégic datas? Personnal datas? Is this flow internal, or going through public network? Define if we must set a resticted access to these datas. Crypt?		
Integrity of datas: In case of database restoration of the target system, will you use this flow to resynchronize your system? Do you have to be able to replay your flow, and detect if datas has already be received or not, to avoid treat it several times? *This can be useful in case you make a database restoration, and if you have to resynchronize easily your system.*		
Availability: Is your flow vital for business? What are the business consequences in case this flow is interrupted? delayed?		

For questions, please contact jerome.decamps@decathlon.com

7°) MEMO for a good specification (page to be deleted when OK)

Checklist:

- ➢ My document is understandable to my interlocutors?
- ➢ If I make copy-past, do I cut line? Forget word? Spellchecker?
- ➢ Read back next day
- ➢ Do I see all most critical points?
- ➢ Study of risk for a go back?
- ➢ Datasets are in the good server
  All customizing is fine
- ➢ The summary page is updated?