SMM WEB SERVICE

The information in this document is subject to change without notice.

This document contains information protected by copyright. No part of this document may be photocopied or reproduced in any form without prior written consent from San Marino Mail Italia Srl.

© Copyright 2020/5 San Marino Mail Italia.

All rights reserved.

Date Revised: 23 January 2025

Version	Date	Author	Description
1.0	03-02-2012	SMM IT Stuff	Initial version
1.1	10-03-2015	SMM IT Stuff	Updated version
1.2	24-06-2015	SMM IT STUFF	Updated version

Index

1. **Content** 4
2. **Invoking SOAP Service** 4
3. **Invoking SOAP-FLOW** 5
4. **Invoking WEB Service** 9
5. **API Call Reference** 9
6. **Address Info Data** 10
7. **API Call Error messages** 11
8. **TEST PHASE & MIGRATION** 17

1. Content

This document briefly describes how to interact with SMM SOAP / REST Service.

Also are described the operations (i.e. methods) it exposes and response message. This is accomplished by a set of API which interacts with SMM SOAP / REST Service.

2. Invoking SOAP Service

A web service is a set of operations accessible by a standardized web protocol. Each operation can be called by your software to order our shipment products and services. Your software will request a certain operation to be executed with your particular customer shipment data. Our API then responds synchronously by sending back status data and shipment. The data can then be further processed and the fulfillment workflow can be continued.

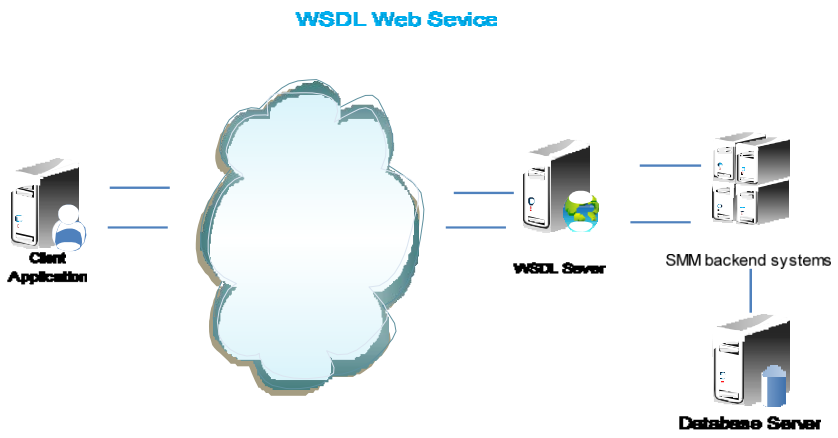


Figure 1 – WSDL Infra

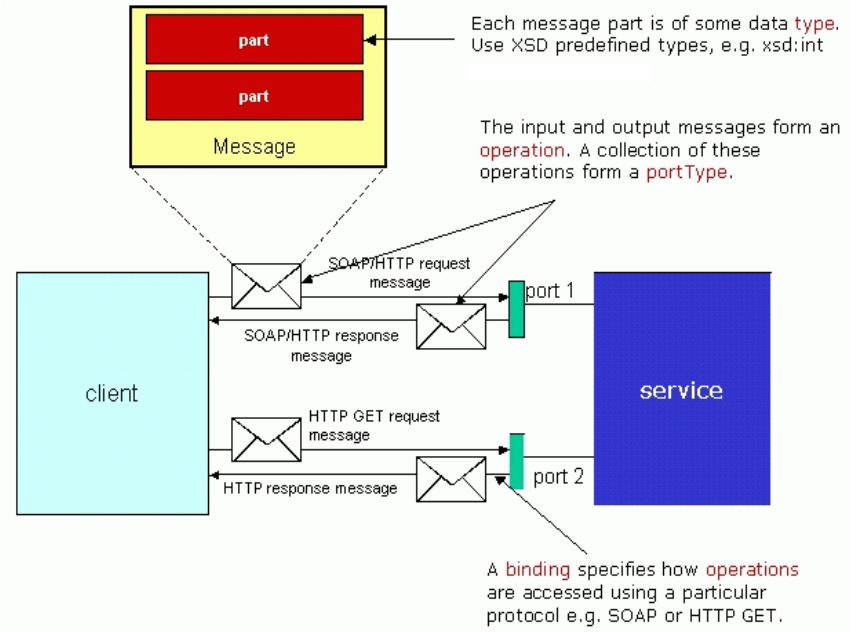


Figure 2 - WSDL terminology used for describing Web services.

3. Invoking SOAP - Flow

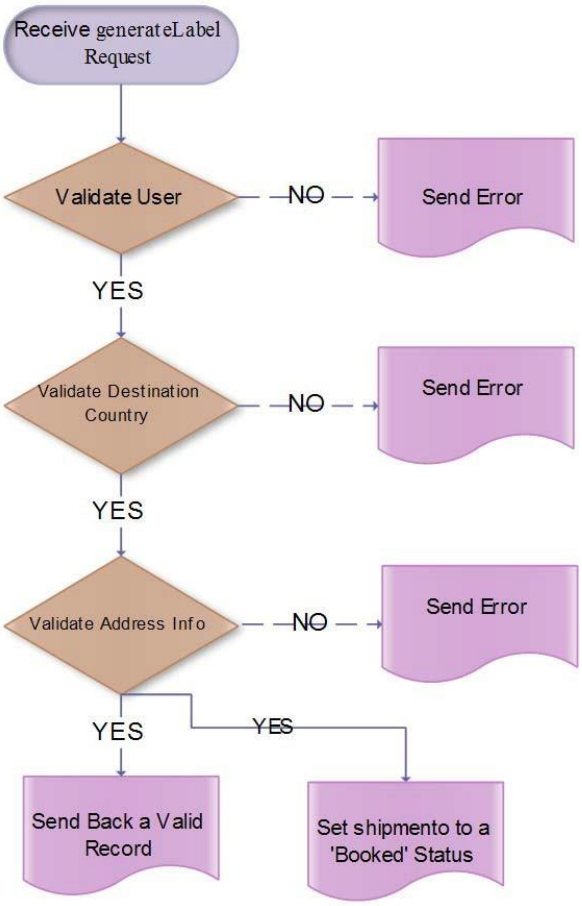
Before using the web service, the following is required:

1. A valid customer ID
2. A valid Login Name and Password
3. Knowledge of the countries where you been authorized to ship

Request and response works basically as fallow:

- Request (Order) a shipments by:
    a. Login into the system and send shipping data
- Respond message:
    a. If the User credentials and shipping data is validated then a record with data is returned and the shipment is set to a 'Booked' Status in our system
    b. If the either the User credentials or shipping data are not validate then an error message is returned with empty data.

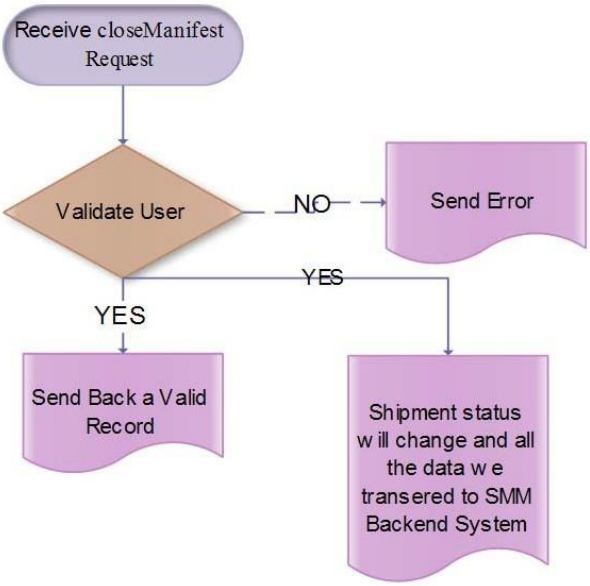
If the request returns a valid record, the shipment is set in our system as ‘booked’ meaning that an order to ship has been made.



We will not make further action to that record until we are advised that goods have actually left the warehouse, this is done by sending us a manifest.

The manifesting process for us begins when the goods leave the warehouse, shipments created will be buffered for the time being, until they are being manifested. Manifesting can be triggered by the warehouse operator or by company that is handling the line-haul.

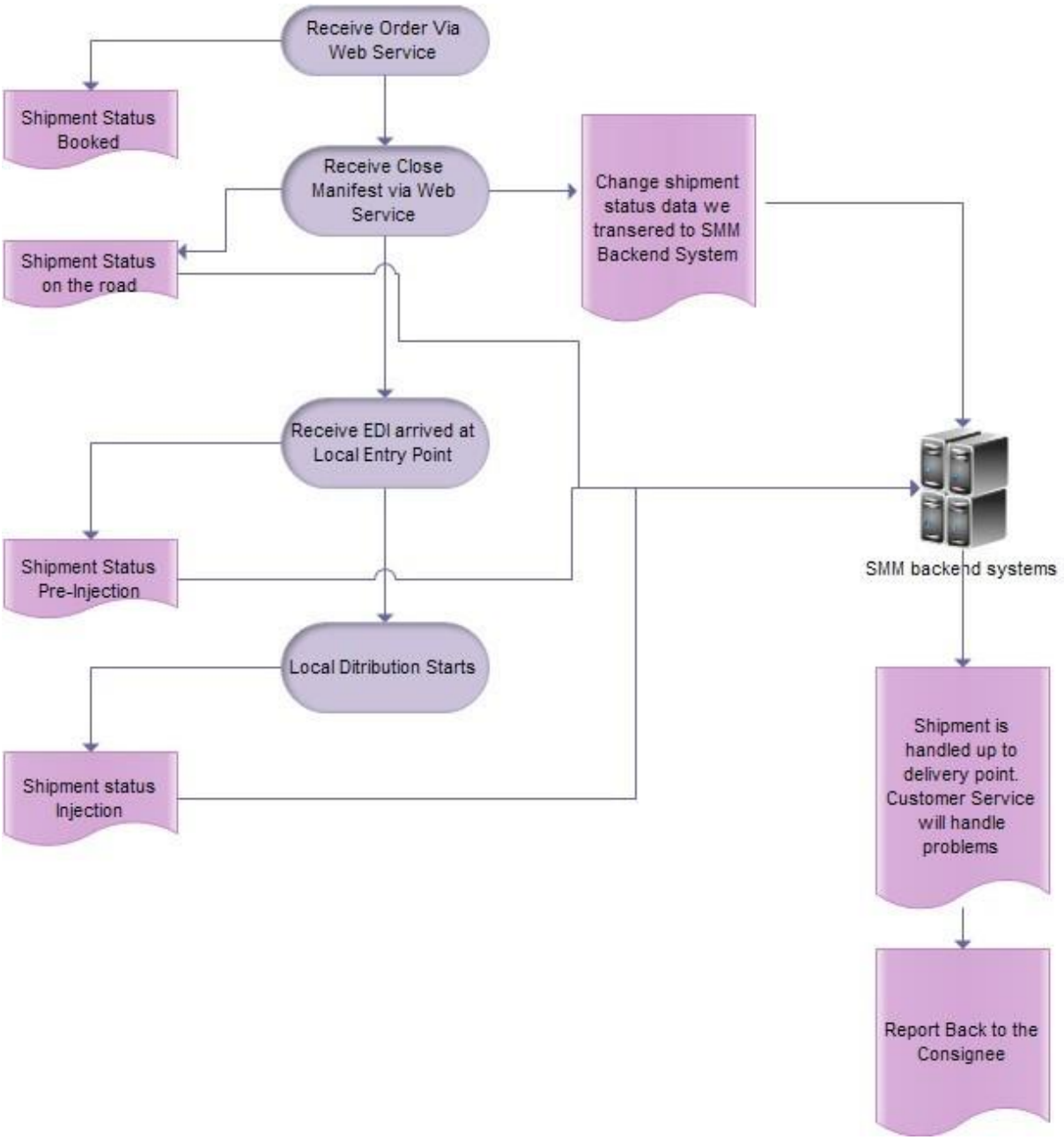
In our case the manifest ID is send in the 'Shipping Data' and it will group a set of shipments. When a closeManifest request is send it will change the shipment(s) status and the data will be moved to SMM Backend systems



At this point the goods have left the warehouse via line haul company, system will record goods is on the road to the injection point. We call 'Injection Point' where the goods are unloaded and prepared to be delivered to the final destination (B to C). The company that handles the line haul will be sending us a message (to be decided) via a Manifest ID. The shipments will change to a Pre-Injection Status.

At the Injection Point, the single package will be weighted, measured and shipped. The local Distribution Company will notify us via EDI that the parcel has left their warehouse. From now on all the package will be monitored via our system that notifies our internal Customer Service on the action to take when a problem arises.

Overview diagram:



4. Invoking WEB Service

Calling any of the API's operations requires an authorization. It is achieved by passing username (<user>) and password (<signature>) in the SOAP / REST header's authentication field. It is required to have an online user account, and moreover an account number equipped with the needed permissions.

The underlying communication pattern is defined by the Web Service as a request –response message exchange. It is highly recommended to bind your communication to SOAP 1.2 protocol.
The Web Service binding constitutes the message exchange style to be "document" (rather than rpc) and requires use of HTTP as transportation protocol. All request messages shall therefore be bound to SOAP 1.2 protocol and send as HTTP POSTs.

The service url is as follow:

http://labels-api.sanmarinomail.it/SanMarinoMailWeb2/services/SmmWebService?wsdl

5. API Call Reference

WSDL Definition

WSDL URL	http://localhost:8082/SanMarinoMailWeb/services/SmmWebService?wsdl
Namespace	http://ws.web.system
Binding	SmmWebServiceSoap12Binding
SOAP Version	SOAP 1.2
Style	Document
WS-A version	NONE

Definition Parts

SmmWebService?wsdl	http://localhost:8082/SanMarinoMailWeb/services/SmmWebService?wsdl

Operations

Name	Use	One-Way	Action
authenticateUser	Literal	false	urn:authenticateUser
cancelLabel	Literal	false	urn:cancelLabel
checkConsistency	Literal	false	urn:checkConsistency
closeManifest	Literal	false	urn:closeManifest
generateLabel	Literal	false	urn:generateLabel
readLabel	Literal	false	urn:readLabel
readStatus	Literal	false	urn:readStatus
updateStatus	Literal	false	urn:updateStatus
updateTrackArrived	Literal	false	urn:updateTrackArrived
updateTrackLeft	Literal	false	urn:updateTrackLeft

In details you will have:

- **authenticateUser** – Authenticates user and gets a valid Token (username – password);

- **generateLabel** – Generate Label (token – username –shipment info);

- **checkConsistency** - to validate consistency data before label generation request (token-username-shipment info);

- **readLabel** – Reads a Label previously generate (token – username – invoiceecustref or usn - invoiceecompanyId), it could be used to reprint a label in case of paper jam etc.;

- **closeManifest**– Set records to Manifest true, the procedure accept the list of clientShipRef used (token – username – invoiceecustref array - dateTime - manifestId - invoiceecompanyId)

- **cancelLabel**– Deletes records that have not been manifested yet (token – username – invoiceecustref- invoiceecompanyId);

- **updateStatus** – Status Message from customer to SMM (token – username – invoiceecustref- status - invoiceecompanyId) - This is an optional call to notify SMM that the label was succesfully printed;

- **readStatus** – Status Message from SMM to customer (token – username – invoiceecustref - invoiceecompanyId) .

6. Address Info Data

	NAME	DESCRIPTION	TYPE	MAX LENGTH	MANDATORY
1	SMM_CUST_CODE	Company Id	T	20	Y
2	SMM_PRODUCT_CODE	Product code	T	50	N
3	PRE_ALLERT_DATE	Pre allert Date	T (yyyy-MM-dd)	10	N
4	CLIENT_SHIP_REF	Client Shipment Reference	T	35	Y
5	USN_PARCEL_NUMBER	USN parcel number	T	50	N
6	MASTER_USN_PARCEL_NR	Master USN	T	30	N
7	DELIVERY_NAME	Delivery Name	T	35	Y
8	DELIVERY_STR_TYPE	Delivery Street type	T	30	N
9	DELIVERY_ADDR	Delivery address 1	T	35	Y
10	DELIVERY_ADDR_NO	Delivery address No	T	5	N
11	DELIVERY_ZIP_CODE	Delivery Zip code	T	12	Y
12	DELIVERY_CITY	Delivery City	T	50	Y
13	DELIVERY_DISTRICT	Delivery district (Mandatory for italy)	T	50	Y/N
14	DEL_ISO_COUNTRY_CODE	Delivery iso country	T	3	Y
15	NR_OF_PCS	Number of items	N	10	Y
16	WEIGHT	Weight in KG	N	10	Y
17	PACKAGING_TYPE	Packaging type	T	40	N
18	LDV_PRINTING	Ldv printing (1 , 0) (default 0)	N	1	N
19	ADDRESS_CLEANING	Address cleaning (1,0)	N	1	N
20	CUST_SERVICE_ACTIVITY	Customer service (1,0)	N	1	N
21	WAREHOUSE_TRANSIT	Warehouse Transit (1,0)	N	1	N
22	PARCEL	Parcel Number	N	10	Y
23	SMS_PREADVISE_REQUEST	SMS preadvise	N	1	N
24	EMAIL_PREADVISE_REQUEST	Email preadvise	N	1	N
25	CLIENT_SHIP_REF_2	Client Shipment reference 2	T	35	N
26	DELIVERY_NAME_2	Delivery name 2	T	35	N
27	DELIVERY_ADDR_2	Delivery address 2	T	35	N
28	DELIVERY_ADDR_3	Delivery address 3	T	35	N
29	DELIVERY_ADDR_4	Delivery address 4	T	35	N
30	DELIVERY_PHONE_NR	Delivery phone number	T	50	N
31	DELIVERY_FAX_NR	Delivery fax number	T	50	N
32	DELIVERY_MOBILE_PH	Delivery mobile phone	T	50	N
33	DELIVERY_EMAIL	Delivery email	T	50	N
34	DELIVERY_INSTRUCTION	Delivery Instruction	T	150	N
35	DELIVERY_ADDRESS_ID	Delivery address Id	T	12	N
36	DELIVERY_FISCAL_CODE	Delivery fiscal code	T	50	N
37	DELIVERY_SATURDAY	Delivery saturday (1,0)	N	1	N
38	DIMEN_PCS_LENGTH	Length	N	10	N
39	DIMEN_PCS_WIDTH	Width	N	10	N
40	DIMEN_PCS_HEIGHT	Height	N	10	N
41	TOT_VOLUME	Total volume	N	10	N
42	DECLARED_GOODS_VALUE	Declared goods value	N	10	N
43	CUST_VALUE_CURRENCY	Value currency	T	10	N
44	SENDER_VAT_NUMBER	VAT number	T	50	N
45	SENDER_DOCUMENT_NR	Document Nr	T	50	N
46	INVOICE_NR_FOLLOWING	Invoice Nr	T	50	N
47	COD_AMOUNT	COD amount (otherwise 0)	N	10	N

48	COD_PAYM_METHOD	COD Payment method (CON)	T	50	N
49	COD_CURRENCY	COD currency	T	10	N
50	INSURANCE_VALUE	Insurance value	N	10	N
51	INSURANCE_CURRENCY	Insurance currency	T	10	N
52	PICKUP_NAME	Pickup name	T	50	N
53	PICKUP_NAME2	Pickup name 2	T	50	N
54	PICKUP_ADDR	Pickup address 1	T	100	N
55	PICKUP_ADDR_NO	Pickup address No	T	5	N
56	PICKUP_ADDR_2	Pickup address 2	T	100	N
57	PICKUP_ADDR_3	Pickup address 3	T	100	N
58	PICKUP_ADDR_4	Pickup address 4	T	100	N
59	PICKUP_ZIP_CODE	Pickup zipcode	T	10	N
60	PICKUP_CITY	Pickup city	T	50	N
61	PICKUP_PHONE_NR	Pickup phone nr	T	50	N
62	PICKUP_FAX_NR	Pickup fax nr	T	50	N
63	PICKUP_MOBILE_PH	Pickup mobile nr	T	50	N
64	PICKUP_EMAIL	Pickup email	T	50	N
65	PICKUP_ISO_COUNTRY_CODE	Pickup ISO country	T	3	N
66	PICKUP_ADDR_ID	Pickup address Id	T	12	N
67	DELIVERY_AT_FLOOR	Delivery at floor (1,0)	N	1	N
68	DELIVERY_APPOINTMENT	Delivery appointment (1,0)	N	1	N
69	DELIVERY_EVENING	Delivery evening (1,0)	N	1	N
70	PICKUP_USED_RAEE	Used RAEE (1,0)	N	10	N
71	PICKUP_DATE	Pickup date	T (yyyy-MM-dd)	10	N
72	OPENING_HOUR_MORNING	Opening hour morning	N	1	N
73	CLOSING_HOUR_MORNING	Closing hour morning	N	1	N
74	OPENING HOUR_AFTERNOON	Opening afternoon	N	1	N
75	CLOSING HOUR_AFTERNOON	Closing afternoon	N	1	N
76	PICKUP_WINDOW	Pickup window	T	2	N
77	PICKUP_DISTRICT	Pickup district	T	2	N
78	CONTENT	Content type	T	50	N
79	GOODS_VALUE	Commercial value	N	10	N
80	GOODS_CURRENCY	Commercial currency	T	3	N
81	CUSTOM_CODE	Customs code	T	12	N
82	OUTPUT_TYPE	OutputType (0=ZPL,1=PDF)	N	1	Y
83	DELIVERY_POINTRELAYID	PUDO Identifier (Mandatory only for PUDO delivery)	T	50	Y/N

(*)ZPL=0, PDF=1, IMG=2, NONE=3

7. API Call Error messages

```
-1 => generic / Unknown Error
Error: Generic Error – Call SMM

public static int ROUTING_NOT_FOUND = 1059;
Error: Generic Error on CAP+Address+City+Province field. Destination
       information are not valid / not found
Solution: check destination information

public static int INVOCIEECOMPANYID_NOT_VALID = 1110;
Error: Error on InvoiceeCompanyId field. InvoiceeCompanyId information
       are not valid / not found
Solution: check InvoiceeCompanyId information

public static int CAP_LENGTH_NOT_VALID = 1008;
Error: Generic Error on CAP field. It is missing
Solution: fill zipCode information

public static int CAP_NOT_VALID = 1009;
Error: Error on CAP. CAP information is present but is not found inside
       CAP archive
Solution: check ZipCode information

public static int CAP_DB_ERROR = 1023;
Error: Internal ZipCode DB error – Call SMM

public static int CITY_NOT_VALID = 1024;
Error: Error on City. City information is missing
Solution: check city information


public static int PROVINCE_NOT_VALID = 1025;
Error: Error on Province. Province information is missing
Solution: check province information

public static int NAME_NOT_VALID = 1026;
Error: Error on Name field. Name information is missing
Solution: check name information

public static int ADDRESS_NOT_VALID = 1027;
Error: Error on Address. Address information is missing
Solution: check address information

public static  int CONTRASSEGNO_NOT_VALID = 1028;
Error: Error on COD. COD information is missing. You set cod=1, but and
codPaymentMethod is missing
Solution: check COD information

public static int SERVICE_COMBINATION_NOT_VALID = 1029;
Error: Error on Service combination. You ask for an add-on service, but
you are not allowed to use it
Solution: check Service data information
```

```
public static int WEIGHT_NOT_VALID = 1030;
Error: Error on Weight. Weight information is missing or exceeds
max.value. Please check commercial agreement
Solution: check Weight information. It must be a double

public static int SHIPMENT_TYPE_NOT_VALID = 1031;
Error: Error on Service/Country combination. Not authorized to using a
service for a country
Solution: check Service data information

public static int COLLI_NOT_VALID = 1032;
Error: Error on Parcel number. It must be =1
Solution: check parcels data information

public static int PRODUCT_DIMENSION_EXCEEDED = 1034;
Error: Error on Dimension. Dimension information exceeds max value.
Solution: check Dimension information and commercial agreement

public static int DIMENSION_INFO_NOT_VALID = 1035;
Error: Error on Dimension. Dimension information is missing.
Solution: check Dimension information. Must be in cm

public static int DECLAREDVALUE_NOT_VALID = 1036;
Error: Error on Declared Value. Declared Value information is missing.
Solution: check declared value information. It must be a double. It's
mandatory for parcel

public static int INSTRUCTION_NOT_VALID = 1037;
Error: Error on Delivery instruction. Delivery instruction information
is missing. Solution: check Delivery instruction information.


public static int DELIVERY_PHONE_NOT_VALID = 1038;
Error: Error on delivery Phone/ Mobile phone. Both Phone/Phone2/Mobile
phone information are missing.
Solution: At least one between Phone /Mobile phones must be available.
This applies to special deliveries

public static int PICKUP_PHONE_NOT_VALID = 1175;
Error: Error on pickup Phone. Pickup Phone information are needed for
selected product and are missing.
Solution: Pickup Phone must be filled. This applies to special
deliveries

public static int SERVICE_WEIGHT_NOT_VALID = 1039;
Error: Error on Weight. Weight field exceed max value for additional
service. Please check commercial agreement
Solution: check Weight information. It must be a double

public static int ISO_NOT_VALID = 1041;
Error: Error on ISO. ISO field is missing
Solution: check ISO information.

public static int EXT_COUNTRY_NOT_VALID = 1042;
Error: Error on country Name. Country name field is missing.
Solution: check Country name information.
```

```
public static int REFERENCE_NOT_VALID = 1043;
Error: Error on clientShipRef. clientShipRef field is missing or already
used.
Solution: check clientShipRef information.

public static int SMM_PRODUCT_ID_NOTFOUND = 1044;
Error: Error on Request. Please check SMM product code value
Solution: check SMM product code value based on commercial agreement.

public static int COUNTRY_NOTFOUND = 1045;
Error: Error on ISO. ISO field is present, but not valid
Solution: check ISO information.

public static int INVOICEECOMPANYID_NOT_MATCH = 1111;
Error: Error on clientShipRef Request. clientSHipref set up inside
ShipmentData is not yours
Solution: Change clientShipRef

public static int LDV_PRINTING_NOT_VALID = 1061;
Error: Error on ldvPrinting field . It must be 0 or 1.
Solution: Check ldvPrinting field

public static int EMAIL_NOT_VALID= 1072;
Error: Error on delivery email field . It must be a valid email address.
Solution: Check delivery_email field

public static int REFERENCE_ALREADY_USED= 1063;
Error: Error on clientShipRef field . It must be unique.
Solution: Check clientShipRef field

public static int FIELD_LENGTH_NOT_VALID= 1064;
Error: Error on field length.
Solution: Check all fields field

public static int PICKUP_NAME= 1167;
Error: You are using a product where pickup information are needed.
Pickup Name field is missing
Solution: Check Pickup Name field

public static int PICKUP_ADDRESS= 1168;
Error: You are using a product where pickup information are needed.
Pickup Address field is missing
Solution: Check Pickup Address field

public static int PICKUP_CITY= 1169;
Error: You are using a product where pickup information are needed.
Pickup city field is missing
Solution: Check Pickup city field

public static int PICKUP_DISTRICT= 1170;
Error: You are using a product where pickup information are needed.
Pickup District field is missing
Solution: Check Pickup District field

public static int PICKUP_EMAIL= 1171;
Error: You are using a product where pickup information are needed.
Pickup Email field is missing
Solution: Check Pickup Email field
```

```
public static int PICKUP_ISOCODE= 1172;
Error: You are using a product where pickup information are needed.
Pickup Iso code field is missing
Solution: Check Pickup Iso code field

public static int PICKUP_ZIPCODE= 1174;
Error: You are using a product where pickup information are needed.
Pickup Zipcode field is missing
Solution: Check Pickup Zipcode field

public static int PICKUP_WINDOW= 1173;
Error: You are using a product where pickup information are needed.
Pickup window information fields are missing
Solution: Check Pickup window information fields

public static int PICKUP_COUNTRY= 1181;
Error: You are using a product where pickup information are needed.
Pickup country is missing
Solution: Check Pickup country field

public static int LABEL_OUTPUT_TYPE_INVALID= 1182;
Error: Output type is not valid. It must be 0 for ZPL, 1 for PDF, 2 for
IMG.
Solution: Check outputType field

public static final int USN_NOT_VALID = 1183;
Error: USN / USNM is not valid. It must be filled if LDV_PRINTING is 0.
Solution: Check USN and USNM field

public static final int MANIFESTID_NOT_VALID = 1191;
Error: Manifest Id is not valid.
Solution: Check manifestId field

public static final int MANIFESTDATE_NOT_VALID = 1192;
Error: Manifest date is not valid.
Solution: Check manifest date field

public static final int MANIFESTREFLIST_NOT_VALID = 1193;
Error: Client ship ref list is not valid or is empty
Solution: Check Client Shipment Reference list field

public static final int INVALID_CHARACTERS = 1194;
Error: Some fields contain invalid characters
Solution: Check fields and remove invalid
characters

public static final int PARCEL_NOTVALID=1195;
Error: Parcel field is not valid. It must be 1=PARCEL, 0=NON PARCEL
Solution: Check Parcel field

public static final int FEATURE_NOT_SUPPORTED = 1196;
Error: A feature requested is not available for selected product
Solution: Check features

public static final int CUST_VALUE_NOT_VALID = 1197;
Error: CUST_VALUE_CURRENCY field is empty or not valid
Solution: Check CUST_VALUE_CURRENCY
```

```
public static final int GOODS_CURRENCY_NOT_VALID = 1198;
Error: GOODS_CURRENCY field is empty or not valid
Solution: Check GOODS_CURRENCY

public static final int ADDRESS_NO_NOT_VALID = 1199;
Error: DELIVERY_ADDRESS_NO /PICKUP_ADDRESS_NO is required and is missing
or not valid
Solution: Check DELIVERY_ADDRESS_NO / PICKUP_ADDRESS_NO
```

8. TEST PHASE & MIGRATION

The endpoint above, can be used also for testing purposes. The condition useful to identify a call as a test, is to add at the beginning of the CLIENT_SHIP_REF field the word **TEST**; Any call sent to the system without the word **TEST** will be considered as a real shipment and will be invoiced.

During test phase both systems (the old one and the new one) will be available, but with an important limitation: the first call performed to the new system without the word "TEST" at the beginning of the CLIENT_SHIP_REF field, will cause the deactivation the account on the old system.