

SMM WEB SERVICE

SmmWs3 API

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 2024 San Marino Mail Italia.
All rights reserved.

Date Revised: 10th April 2024

Document purpose

This document briefly describes how to interact with SMM SmmWs3 API. Also are described the operations (i.e. methods) that are exposed and response messages.

API description

The API offers these operations:

- Generate Token
- Create Shipment and generate label
- Get Label
- Delete Label
- Manifest
- Upload Document
- Tracking

All these operation except "Generate Token" needs a valid token (available with the "Generate Token" operation) to be able to gain access.

Every data passed and receive with the web-service is based on headers values and JSON strings and everything need to be passed with POST method.

Generate Token

This function allow with valid credentials (http basic authorization) to gain a valid token usable with the other operations.

To be able to use the web servicee is mandatory to add some headers in the call; in the generate Token the headers are:

- Accept: application/json
- Authorization: Basic (USERNAME:PASSWORD base 64 encoded)
- forcenew: true (default false)

If forcenew is set to true the token will be rigerated otherwise the service return the active one if exists.

The URL is: <https://ws3.sanmarinomail.it/SmmWs3/token>

curl example :

```
curl -H "Accept: application/json" -H "Authorization: Basic YWxhZGRpbjpvucGVuc2VzYW1l" -H "forcenew: false" -X POST https://ws3.sanmarinomail.it/SmmWs3/token
```

response examples:

error:

```
{
  "errorDescription": "Username or Password Not Valid",
  "errorCode": 3001,
  "status": -1
}
```

```

}
success:
{
    "token": "{token}d87bb9c6-25fa-4035-ae90-c515554f3cf2",
    "status": 0
}

```

A succesful responce must return a status field = 0;

The token obtained is valid for 24 hours.

Create Shipment

This operation allow to book a shipment and get the related label.

PLEASE NOTE TO UPLOAD ANY NEEDED FILE BEFORE CALLING THE GENERATE LABEL REQUEST (see specific upload method informations).

To be able to use the web service is mandatory to add some headers in the call; in the generate Token the headers are:

- Accept: application/json
- username: testusername
- token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce

If any of the listed mandatory headers is missing an error will be sended and no error label will be generated.

The shipment references for the label to retrive must be passed as JSON string:

Here's a list of all available fields related to the shipment (to be passed in the json object "shipment"):

| | 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 | Y |
| 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- | 10 | N |

| | | | | | |
|----|------------------------|---|--------|----|---|
| | | | MM-dd) | | |
| 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_GDO | GDO delivery | N | 1 | N |
| 84 | DELIVERY_NDS | Neutral delivery service | N | 1 | N |
| 85 | MULTI_PARCEL | set to 1 for multiparcel shipment (default 0) | N | 1 | N |
| 86 | NR_OF_PARCELS | Number of parcels | N | 10 | N |
| 87 | RETURN_LABEL | return label needed | N | 1 | N |
| 88 | RETURN_PRODUCT_CODE | Return label Product code | T | 50 | N |
| 89 | DOCUMENTS_NEEDED | set to 1 if previously uploaded documents must be sent to the carrier (default 0) | N | 1 | N |

Here's a list of all available fields related to the parcel (to be passed in the json object "packages"):

| | NAME | DESCRIPTION | TYPE | MAX LENGTH | MANDATORY |
|----|-----------------------|---|----------------|------------|-----------|
| 1 | CLIENT_SHIP_REF | Client Shipment Reference | T | 35 | N |
| 2 | WEIGHT | Weight in KG | N | 10 | Y |
| 3 | DIMEN_PCS_LENGTH | Length | N | 10 | Y |
| 4 | DIMEN_PCS_WIDTH | Width | N | 10 | Y |
| 5 | DIMEN_PCS_HEIGHT | Height | N | 10 | Y |
| 6 | CONTENT | Content type | T | 50 | N |
| 7 | COD_AMOUNT | COD amount (otherwise 0) | N | 10 | N |
| 8 | COD_PAYM_METHOD | COD Payment method (CON) | T | 50 | N |
| 9 | COD_CURRENCY | COD currency | T | 10 | N |
| 10 | INSURANCE_VALUE | Insurance value | N | 10 | N |
| 11 | INSURANCE_CURRENCY | Insurance currency | T | 10 | N |
| 12 | GOODS_VALUE | Commercial value | N | 10 | N |
| 13 | GOODS_CURRENCY | Commercial currency | T | 3 | N |
| 14 | HAZMAT | Hazardous goods | N | 1 | N |
| 15 | HAZMAT_CLASS | Hazardous goods class UN (es. LQ) | T | 20 | N |
| 16 | HAZMAT_SHIPPING_NAME | The Proper Shipping Name assigned by ADR, CFR or IATA. | T | 50 | N |
| 17 | HAZMAT_REGULATION_SET | Hazardous goods The Regulatory set (ex ADR) | T | 5 | N |
| | | | | | |
| 18 | INCOTERMS | Incoterms | T | 20 | N |
| 19 | HS_CODE | HS Code | T | 20 | N |
| 20 | INVOICE_NR_DATE | Invoice date | T (yyyy-MM-dd) | 10 | N |
| 21 | NR_PIECES_HAZ | Item Quantity of Hazardous goods | N | 10 | N |
| 22 | INVOICE_NR | Invoice Number | T | 30 | N |
| 23 | HAZMAT_UOM | The unit of measure used for the mass capacity of the regulated good. For Example: ml, L, g, mg, kg, cylinder, pound, pint, quart, gallon, ounce | T | 15 | N |

| | | | | | |
|----|---------------------------------|--|---|-----|---|
| 24 | HAZMAT_ClassDivisionNumber | This is the hazard class associated to the specified commodity. | N | 1 | N |
| 25 | HAZMAT_ChemicalRecordIdentifier | Should be set to 1 | N | 1 | N |
| 26 | HAZMAT_PackagingType | The type of package used to contain the regulated good. (Ex: Fiberboard Box). Required if CommodityRegulatedLevelCode = LQ or FR. Ex. FIBERBOARD BOX, WOOD(EN) BOX, PLASTIC JERRICAN, METAL BOX, STEEL DRUM, OTHER, PLASTIC BOX, PLASTIC DRUM, STYROFOAM BOX, CYLINDERS, ENVIROTAINER, PLYWOOD BOX, ALUMINUM DRUM, ALUMINUM CYLINDERS, PLASTIC PAIL, PLYWOOD DRUM, FIBER DRUM, STEEL JERRICAN, ALUMINUM JERRICAN, STEEL BOX, CARTON, ALUMINUM BOX | T | 255 | N |
| 27 | HAZMAT_IDNumber | This is the ID number (UN/NA/ID) for the specified commodity. Required if CommodityRegulatedLevelCode = LR, LQ or FR and if the field applies to the material by regulation. UN/NA/ID Identification Number assigned to the specified regulated good. (Include the UN/NA/ID as part of the entry). | T | 6 | N |
| 28 | HAZMAT_TransportationMode | Not applicable for ADR regulation set. Required for any other regulation set. Declares that a package was prepared according to ground passenger aircraft or cargo aircraft only. Valid values: Highway=Highway Ground=Ground PAX=Passenger Aircraft Passenger Aircraft=Passenger Aircraft CAO=Cargo Aircraft Only Cargo Aircraft Only=Cargo Aircraft Only Valid entries include: Highway, Ground, PAX, Passenger Aircraft, CAO and Cargo Aircraft Only. | T | 30 | N |

The field SMM_PRODUCT_CODE (not mandatory) if set will be used to identify the SMM product to be used, if the field is not set then SMM algorithm will need to be set up to.

In case of extracee shipment or dangerous good content all the fields in the package object are mandatory.

In case of a multiparcel shipment that need parcel informations (es. UPS, DHL etc) the field MULTI_PARCEL must be set to 1 and another JsonArray must be filled up with all the parcels informations (see example below).

options available (not mandatory) are:

- force_reprint_if_exists (default value: true): if a label (with same reference) already exists and the field is set to false the system will answer with "Reference already used" error otherwise if set to true the system will send the previously generated label;
- delete_if_exists (default value: false): if the label already exists and the field is set to true the existing label will be deleted before generating a new one;

- base64 (default value: false): if set to true the output of zpl label will be encoded in base64 otherwise will be sent in clear text;

The optional json object "mirror" will be added to the response as received

The URL is: <https://ws3.sanmarinomail.it/SmmWs3/createshipment>

JSON request example:

```
{
  "shipment":{
    "CLIENT_SHIP_REF":"TEST_NEW_WS_001",
    "SMM_CUST_CODE":"ITIT0000999",
    "DELIVERY_NAME":"Mario Rossi",
    "DELIVERY_ADDR":"via Farini 81",
    "DELIVERY_CITY":"Milano",
    "DELIVERY_ZIP_CODE":"20159",
    "WEIGHT":12,
    "DEL_ISO_COUNTRY_CODE":"IT",
    "PRE_ALERT_DATE":"2020-07-20",
    "DELIVERY_DISTRICT":"MI"
  },
  "options":{
    "force_reprint":false,
    "confirmed":false,
    "delete_if_exists":false,
    "base64":true
  },
  "mirror":{
    "cod_attivita_infolog":"temp",
    "cod_stazione":"temp2"
  }
}
```

curl example :

```
curl -H "Accept: application/json" -H "token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce" -H "username: testusername" -d '{
  "shipment":{
    "CLIENT_SHIP_REF":"TEST_NEW_WS_002",
    "SMM_CUST_CODE":"ITIT0000999",
    "DELIVERY_NAME":"Mario Rossi",
    "DELIVERY_ADDR":"via farini 81",
    "DELIVERY_CITY":"Milano",
    "DELIVERY_ZIP_CODE":"20159",
    "WEIGHT":12,
    "DEL_ISO_COUNTRY_CODE":"IT",
    "PRE_ALERT_DATE":"2020-07-13",
    "DELIVERY_DISTRICT":"MI"
  },
  "options":{
    "force_reprint_if_exists":false,
    "confirmed":false,
    "delete_if_exists":false,
    "base64":true
  },
  "mirror":{
    "field1":"temp"
  }
}' -X POST https://ws3.sanmarinomail.it/SmmWs3/createshipment
```

response examples:

error:

```
{
  "errorDescription": "Token Not Valid",
  "errorCode": 3002,
  "status": -1
}
```

error:

```

{
  "mirror": {
    "cod_attivita_infolog": "temp",
    "cod_stazione": "temp2"
  },
  "CLIENT_SHIP_REF": "TEST_NEW_WS_003",
  "notes": "",
  "errorDescription": "Error Reference number already used",
  "errorCode": 1063,
  "label":
  "^XA~TA000~JSN^LT0^MNW^MTD^PON^PMN^LH0,0^JMA^PR4,4~SD15^JUS^LRN^CIO
  ^XZ ^XA ^MMT ^PW799 ^LL1199 ^LS0 ^FT753,776^A01,49,36^FH\\^FDSAN MARINO
  MAIL ITALIA SRL^FS ^FO269,741^GB519,0,8^FS ^FT728,694^A01,28,28^FH\\
  ^FDATTENTION ^FS ^FT725,645^A01,28,28^FH\\^FDIF YOU RECEIVE THIS LABEL^FS
  ^FT725,597^A01,28,28^FH\\^FDPLEASE CALL +39900266227799^FS
  ^FT634,553^A01,28,28^FH\\^FDOPS DEPT^FS ^FT733,498^A01,28,28^FH\\^FDE-MAIL
  OPS@SANMARINOMAIL.IT^FS ^FWB ^BY3,3,160 ^FO55,50^BC^FD#ICR#^FS ^PQ1,0,1,Y
  ^FWN ^XZ",
  "status": -1
}
error:
{
  "errorDescription": "Error parsing json input - org.json.JSONException: Expected a
  ':' after a key at 103 [character 104 line 1]",
  "errorCode": 999,
  "status": -1
}

success:
{
  "MASTER_USN_PARCEL_NR": "465162803",
  "mirror": {
    "cod_attivita_infolog": "temp",
    "cod_stazione": "temp2"
  },
  "CLIENT_SHIP_REF": "TEST_NEW_WS_004",
  "notes": "",
  "Haulier": "GLS",
  "USN_PARCEL_NUMBER": "600256057",
  "label": "^XA ^LH0,0 ^FO4,2 ^GB804,798,4 ^FS ^FO4,2 ^GB548,52,4 ^FS ^FO19,12
  ^ADN,36,20 ^FDTEST WEBSERVICE ^FS ^FO548,2 ^GB260,52,4 ^FS ^FO561,16
  ^ADN,36,10 ^FDP/V ^FS ^FO611,16 ^ADN,36,10 ^FD1:3 ^FS ^FO693,16 ^ADN,36,10
  ^FD22/07/20 ^FS ^FO21,48 ^ADN,170,60 ^FDBOLLATE ^FS ^FO593,48 ^ADN,170,60
  ^FD99 ^FS ^FO15,185 ^AQN,10,10 ^FDPer info sul trattamento dati personali www.gls-
  italy.com/privacydest ^FS ^FO4,210 ^GB128,166,4 ^FS ^FO128,210 ^GB680,166,4 ^FS
  ^FO136,220 ^ADN,36,15 ^FDMario Rossi ^FS ^FO136,276 ^ADN,36,10 ^FDvia farini 81
  ^FS ^FO15,243 ^ADN,108,50 ^FDLD ^FS ^FO136,338 ^ADN,36,14 ^FDMilano ^FS
  ^FO580,338 ^ADN,36,15 ^FDMI ^FS ^FO632,210 ^GB176,166,4 ^FS ^FO640,220
  ^ADN,18,5 ^FDSEDI CONTIGUE ^FS ^FO695,246 ^ADN,18,5 ^FDSCN ^FS ^FO632, 273

```

```

^GB176,103,4 ^FS ^FO640,280 ^ADN,18,5 ^FDPeso Kg ^FS ^FO680,318 ^ADN,36,20
^FD12.0 ^FS ^FO632, 372 ^GB176,48,4 ^FS ^FO640,378 ^ADN,18,5 ^FDPorto: ^FS
^FO658,398 ^ADN,18,10 ^FDFRANCO ^FS ^FO4,372 ^GB804,48,4 ^FS ^FO25,383
^ADN,36,10 ^FDNote: ^FS ^FO88,383 ^ADN,36,10 ^FD ^FS ^FO640,650
^XGE:GLSMINI.GRF,1,1 ^FS ^FO606,718 ^GB202,82,4 ^FS ^FO4,718 ^GB606,82,4 ^FS
^FO50,422 ^BY3 ^BCN,260,N,N,N,A ^FVL2600256057010M451^FS ^FO100,687
^AAN,26,16 ^FDL2 60025605701 0 M451 01 ^FS ^XZ",
    "status": 0
}

```

A successful response must return a status field = 0;

Multiparcel JSON request example:

```

{
  "shipment": {
    "CLIENT_SHIP_REF": "TEST_NEW_WS_001",
    "SMM_CUST_CODE": "ITIT0000999",
    "DELIVERY_NAME": "Mario Rossi",
    "DELIVERY_ADDR": "via Farini 81",
    "DELIVERY_CITY": "Milano",
    "DELIVERY_ZIP_CODE": "20159",
    "WEIGHT": 3,
    "DEL_ISO_COUNTRY_CODE": "IT",
    "PRE_ALERT_DATE": "2020-07-20",
    "DELIVERY_DISTRICT": "MI",
    "MULTI_PARCEL": 1,
    "packages": [
      {
        "WEIGHT": 1,
        "DIMEN_PCS_HEIGHT": 10,
        "DIMEN_PCS_LENGTH": 11,
        "DIMEN_PCS_WIDTH": 12
      },
      {
        "WEIGHT": 2,
        "DIMEN_PCS_HEIGHT": 20,
        "DIMEN_PCS_LENGTH": 21,
        "DIMEN_PCS_WIDTH": 22
      }
    ]
  },
  "options": {
    "force_reprint": false,
    "confirmed": false,
    "delete_if_exists": false,
    "base64": true
  }
}

```

succesfull response:

```
{
  "MASTER_USN_PARCEL_NR": "1ZXXXXXXXXXXXXXXXXXX",
  "CLIENT_SHIP_REF": "TESTMC3_05120006",
  "notes": "",
  "SMM_CUST_CODE": "ITIT0000999",
  "SMM_PRODUCT_CODE": "INT_UPS_EXPRESS",
  "packages": [
    {
      "MASTER_USN_PARCEL_NR": "TESTMC3_05120006",
      "CLIENT_SHIP_REF": "TESTMC3_05120006",
      "USN_PARCEL_NUMBER": "1ZXXXXXXXXXXXXXXXXXX",
      "label":
"Cl5YQQ0KXkxSTg0KXk1OWQ0KXk1GTixODQpeTEgxMCwxMg0KXk1DWQ0KXlBPSQ0K
XlBXODEyDQpeQ0kyNw0KXkZPMTUsN15BME4sMjAsMjReRlZDTElFTlRFIFRFU1ReRlMN..
.....peWfoNCg=="
    },
    {
      "MASTER_USN_PARCEL_NR": "TESTMC3_05120006",
      "CLIENT_SHIP_REF": "TESTMC3_05120006",
      "USN_PARCEL_NUMBER": "1ZXXXXXXXXXXXXXXXXXX",
      "label":
"Cl5YQQ0KXkxSTg0KXk1OWQ0KXk1GTixODQpeTEgxMCwxMg0KXk1DWQ0KXlBPSQ0K
XlBXODEyDQpeQ0kyNw0KXkZPMTUsN15BME4sMjAsMjReRlZDTElFTlRFIFRFU1ReRlMN
Cl5GTzE1LDI3XkEwTiwyMCwyNF5GVjAyNjYyMjc3OTleRlMNCI5GTzE1LDQ3XkEwTiwyMC
wyNF5GVkNMSUVOVEUgVEVTVF5GUw0KXkZPMTUsNjdeQTBOLDIwLDI0XkZWVklBIENB
.....KXkRODQpeWfoNCg=="
    }
  ],
  "Haulier": "PAITITINT0999",
  "status": 0
}
```

Get Label

This operation allow to retrieve a label for a referenced shipment; to be able to gain access is mandatory to have a valid token.

To be able to use the web service is mandatory to add some headers in the call; in the generate Token the headers are:

- Accept: application/json
- username: testusername
- token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce

the shipment reference for the label to retrive must be passed as JSON string:

the mandatory fields are:

- SMM_CUST_CODE: San Marino Mail client id (string)
- CLIENT_SHIP_REF: shipment reference (string)

the optional fields are:

- base64: ask for a label string encoded in base64 or not; (boolean default is true)

JSON request example:

```
{
  "CLIENT_SHIP_REF": "testreference",
  "SMM_CUST_CODE": "TEST_ID",
  "base64": false,
  "mirror":{
    "cod_attivita_infolog":"temp",
    "cod_stazione":"temp2"
  }
}
```

The URL is: <https://ws3.sanmarinomail.it/SmmWs3/retrievelabel>

curl example :

```
curl -H "Accept: application/json" -H "token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce" -H "username: testusername" -d '{"CLIENT_SHIP_REF":"testreference","SMM_CUST_CODE":"TEST_ID","base64":false}.....' -X POST https://ws3.sanmarinomail.it/SmmWs3/retrievelabel
```

response examples:

error:

```
{
  "errorDescription": "No shipments found for reference testreference",
  "errorCode": 1031,
  "status": -1,
  "mirror":{
    "cod_attivita_infolog":"temp",
    "cod_stazione":"temp2"
  }
}
```

success:

```
{
  "CLIENT_SHIP_REF": "testreference",
  "SMM_CUST_CODE": "TEST_ID",
  "label": "^XA^PON^PQ1^MUd^PW79.....Special Instruction:^FS^XZ",
  "status": 0,
}
```

```

    "usn":"xxxxyyyzzzz",
    "mirror":{
      "cod_attivita_infolog":"temp",
      "cod_stazione":"temp2"
    }
  }
}

```

A successful response must return a status field = 0;

Delete Label

This operation allow to delete a generated label; to be able to gain access is mandatory to have a valid token.

To be able to use the web service is mandatory to add some headers in the call; in the generate Token the headers are:

- Accept: application/json
- username: testusername
- token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce

the shipment reference for the label to retrieve must be passed as JSON string:

the mandatory fields are:

- SMM_CUST_CODE: San Marino Mail client id (string)
- CLIENT_SHIP_REF: shipment reference (string)

JSON request example:

```

{
  "CLIENT_SHIP_REF": "testreference",
  "SMM_CUST_CODE": "TEST_ID"
}

```

The URL is: <https://ws3.sanmarinomail.it/SmmWs3/deletelabel>

curl example :

```

curl -H "Accept: application/json" -H "token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce" -H "username: testusername" -d '{"CLIENT_SHIP_REF":"testreference","SMM_CUST_CODE":"TEST_ID"}.....' -X POST https://ws3.sanmarinomail.it/SmmWs3/deletelabel

```

response examples:

error:
{

```

    "CLIENT_SHIP_REF": "TEST_NEW_WS_003",
    "errorDescription": "no label to delete with reference TEST_NEW_WS_003",
    "errorCode": 3003,
    "status": -1
}

success:
{
    "CLIENT_SHIP_REF": "TEST_NEW_WS_004",
    "notes": "Label deleted succesfully",
    "SMM_CUST_CODE": "ITIT0000999",
    "status": 0
}

```

A succesful responce must return a status field = 0;

Manifest

This operation allow to confirm a list of shipments and retrieve a PDF (in base64 format) report.

To be able to use the web service is mandatory to add some headers in the call; in the generate Token the headers are:

- Accept: application/json
- username: testusername
- token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce

the request must be passed as JSON string:

the mandatory fields are:

- SMM_CUST_CODE: San Marino Mail client id (string)
- CLIENT_SHIP_REF_LIST: shipment reference list json array

JSON request example:

```

{
  "manifest": {
    "SMM_CUST_CODE": "ITIT0000999",
    "CLIENT_SHIP_REF_LIST": [
      "TEST_NEW_WS_003",
      "TEST_NEW_WS_004"
    ]
  }
}

```

The URL is: <https://ws3.sanmarinomail.it/SmmWs3/manifest>

curl example :

```
curl -H "Accept: application/json" -H "token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce" -H "username: testusername" -d '{ "manifest": { "SMM_CUST_CODE": "ITIT0000999", "CLIENT_SHIP_REF_LIST": ["TEST_NEW_WS_003","TEST_NEW_WS_004"]}}' -X POST https://ws3.sanmarinomail.it/SmmWs3/manifest
```

response examples:

error:

```
{  "errorDescription": "Error No shipments to be manifested",  "errorCode": 1065,  "status": -1}
```

success:

```
{  "SMM_CUST_CODE": "ITIT0000999",  "manifestPdf": "JVBERi0xLjQKJfbk/N8KMSAwIG9iago8PAovVHlwZSAvQ2F0YWxvZ.....",  "status": 0}
```

A succesful responce must return a status field = 0;

Document upload

This operation allow to upload a file (in base64 format) related to a shipment. When using the "Create Shipment" method populating with a 1 the "DOCUMENTS_NEEDED" field every document related to the shipment (related with the CLIENT_SHIP_REF field) will be uploaded to the carrier.

PLEASE NOTE TO UPLOAD THE FILES BEFORE CALLING THE GENERATE LABEL REQUEST.

To be able to use the web service is mandatory to add some headers in the call; in the generate Token the headers are:

- Accept: application/json
- username: testusername
- token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce

the request must be passed as JSON string:

the mandatory fields are:

- action: "upload"
- SMM_CUST_CODE: San Marino Mail client id (string)
- CLIENT_SHIP_REF: shipment reference

- category: document category (see list below)
- documentdata64: base64 encoding string of file
- filename: filename
- filetype: filetype (ex. pdf, tiff, png....)

Document category list:

| | SMM_code | DESCRIPTION |
|----|----------|-------------------------------|
| 1 | PDC001 | Commercial invoice |
| 2 | PDC002 | CIN Commercial invoice |
| 3 | PDC003 | Pro forma invoice |
| 4 | PDC004 | Certificate of origin |
| 5 | PDC005 | NAFTA certificate |
| 6 | PDC006 | Custom Declaration |
| 7 | PDC007 | HWB |
| 8 | PDC008 | OTHER |
| 9 | PDC009 | ETD |
| 10 | PDC010 | AUTH FORM |
| 11 | PDC011 | Export Accompanying doc |
| 12 | PDC012 | Export License |
| 13 | PDC013 | Import Permit |
| 14 | PDC014 | Power of Attorney |
| 15 | PDC015 | Packing list |
| 16 | PDC016 | Sed document |
| 17 | PDC017 | shipper letter of instruction |

Non mandatory Fields:

- SMMProductCode

JSON request example:

```
{
  "document": {
    "action": "upload",
    "SMM_CUST_CODE": "ITIT0000999",
    "CLIENT_SHIP_REF": "TEST_PPP2",
    "category": "0B",
    "filename": "test.pdf",
    "filetype": "pdf",
    "documentdata64":
    "JVBERi0xLjYNJeLjz9MNCjM3IDAgb2JqIDw8L0xpbmVhcml6ZWQgMS9MIDIwNTk3L08gNDAvRSAXNDExNS9OIDEvVCAxOTc5NS9II...."
  }
}
```

The URL is: <https://ws3.sanmarinomail.it/SmmWs3/documents>

curl example :

```
curl -H "Accept: application/json" -H "token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce" -H "username: testusername" -d '{
  "document": {
    "SMM_CUST_CODE": "ITIT0000999",
    "CLIENT_SHIP_REF": "TEST_PPP2",
    "category": "0B",
    "filename": "test.pdf",
    "filetype": "pdf",
    "documentdata64":
    "JVBERi0xLjYNJeLjz9MNCjM3IDAgb2JqIDw8L0xpbmVhcml6ZWQgMS9MIDIwNTk3L08gNDAvRSAXNDExNS9OIDEvVCAxOTc5NS9II...."
  }
}' -X POST
https://ws3.sanmarinomail.it/SmmWs3/documentupload
```

response examples:

error:

```
{
  "errorDescription": "missing mandatory fields",
  "errorCode": 3030,
  "status": -1
}
```

success:

```
{
  "CLIENT_SHIP_REF": "TEST_doc",
  "SMM_CUST_CODE": "ITIT0000999",
  "DOCUMENT_ID": "ITIT0000999_00000000003",
  "category": "0B",
  "status": 0
}
```

A succesful rponse must return a status field = 0;

Document delete

This operation allow to delete a previously uploaded file.

To be able to use the web service is mandatory to add some headers in the call; in the generate Token the headers are:

- Accept: application/json
- username: testusername
- token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce

the request must be passed as JSON string:

the mandatory fields are:

- action: "delete"
- SMM_CUST_CODE: San Marino Mail client id (string)
- CLIENT_SHIP_REF: shipment reference

- DOCUMENT_ID: identificativo documento ricevuto in risposta alla chiamata di upload

JSON request example:

```
{
  "document": {
    "action": "delete",
    "SMM_CUST_CODE": "ITIT0000999",
    "CLIENT_SHIP_REF": "TEST_PPP2",
    "DOCUMENT_ID": "ITIT0000999_00000000003"
  }
}
```

The URL is: <https://ws3.sanmarinomail.it/SmmWs3/documents>

curl example :

```
curl -H "Accept: application/json" -H "token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce" -H "username: testusername" -d '{ "document": {
  "SMM_CUST_CODE": "ITIT0000999",
  "CLIENT_SHIP_REF": "TEST_PPP2",
  "category": "0B",
  "filename": "test.pdf",
  "filetype": "pdf",
  "documentdata64":
  "JVBERi0xLjYNJeLjz9MNCjM3IDAgb2JqIDw8L0xpbmVhcml6ZWQgMS9MIDIwNTk3L08gNDAvRSxNDEvVCAxOTc5NS9II..." } }' -X POST
https://ws3.sanmarinomail.it/SmmWs3/documentupload
```

response examples:

error:

```
{
  "errorDescription": "missing mandatory fields",
  "errorCode": 3030,
  "status": -1
}
```

success:

```
{
  "CLIENT_SHIP_REF": "TEST_doc",
  "SMM_CUST_CODE": "ITIT0000999",
  "DOCUMENT_ID": "ITIT0000999_00000000003",
  "category": "0B",
  "status": 0
}
```

A succesful rponse must return a status field = 0;

Tracking

This operation allow to retrieve all the events for a referenced shipment; to be able to gain access is mandatory to have a valid token.

To be able to use the web service is mandatory to add some headers in the call; in the generate Token the headers are:

- Accept: application/json
- username: testusername
- token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce

the shipment reference for the label to retrieve must be passed as JSON string:

the mandatory fields are:

- CLIENT_SHIP_REF: shipment reference (string) or usn: shipment number
- SMM_CUST_CODE: San Marino Mail client id (string)

JSON request example:

```
{
  "clientshipref": "testreference",
  "SMM_CUST_CODE": "TEST_ID"
}
```

The URL is: <https://ws3.sanmarinomail.it/SmmWs3/tracking>

curl example :

```
curl -H "Accept: application/json" -H "token: {token}TEST-TOKEN-a0a2-47b0-b1ab-923d2ec4b8ce" -H "username: testusername" -d '{"clientshipref":"testreference","SMM_CUST_CODE": "TEST_ID"}' -X POST https://ws3.sanmarinomail.it/SmmWs3/tracking
```

response examples:

error:

```
{
  "errorDescription": "Shipment not found",
  "errorCode": 3003,
  "status": -1
}
```

success:

```
{
  {
    "usn": "XXXXXXXXXXXXXXXXXX",
    "country": "United Kingdom",
    "productCode": "GGGGGGGG",
    "invoiceecustref": "testreference",
    "countryiso": "GB",
  }
}
```

```

"clientshipref": "testreference",
"status": 0,
"events": [
  {
    "date": "06/12/19",
    "eventcode": "104",
    "eventdescription": "shipment has been delivered",
    "time": "12:14",
    "eventname": "delivered"
  },
  {
    "date": "06/12/19",
    "eventcode": "141",
    "eventdescription": "shipment has been given to driver for delivery",
    "time": "09:20",
    "eventname": "out for delivery"
  },
  {
    "date": "06/12/19",
    "eventcode": "124",
    "eventdescription": "shipment is in transit to final destination",
    "time": "04:21",
    "eventname": "in transit to final destination"
  },
  {
    "date": "02/12/19",
    "eventcode": "102",
    "eventdescription": "shipment has left for the local distribution
network",
    "time": "15:55",
    "eventname": "departure"
  },
  {
    "date": "02/12/19",
    "eventcode": "751",
    "eventdescription": "SHIPMENT ARRIVED P.M.: IN CHARGE IN SMM
MIL",
    "time": "08:28",
    "eventname": "SHIPMENT ARRIVED P.M.: IN CHARGE IN SMM MIL"
  },
  {
    "date": "28/11/19",
    "eventcode": "0",
    "eventdescription": "shipment has been pre alerted by sender",
    "time": "15:14",
    "eventname": "shipment booked"
  }
]
}

```

}

A succesful responce must return a status field = 0;