

**DTDC FIRST MILE ORDER UPLOAD WS. API Ver. 2.0**

|  |  |  |
| --- | --- | --- |
| **Version** | **Description** | **Release Date** |
| **1.0** | **Pickup & booking details** | **1st April 2019** |
| **2.0** | **RTO address details are added in API** | **23 Sept 2021** |



**Table of Contents**

* **Introduction.........................................................................…..03**
* **About API .......................................................................……….03**
* **Process flow/Validation............................................................03**
* **Access to Server Link.............................................................…03**
* **Structural representation of tag................................................04**
* **Service Input & output Details..................................................08**
* **End of the Doc...........................................................................16**



**Introduction**

This document describes how The Customer can send the pickup request to DTDC by DTDC Web services .

Customer needs to send the individual pickup request to DTDC Web Service, Processing of that request and update back to customer about his request by DTDC.

**About API**

● Customers will submit individual pickup requests to DTDC by Calling DTDC Web Service. DTDC Will share the Web Service details to Customer.

● On every pickup request, DTDC will respond back with a success flag and Relevant detail .

● Json details has been shared in this document under Technical details section

**Process Flow & Validation**

This service is implemented through HTTPRequest. The customer is responsible for making

the HTTPRequest to the specified URL as mentioned in this document. The customer is responsible

for implementing the capability of calling the HTTPRequest in the specified format and get the

DTDC Web Service Response in Json format, mentioned in this document.

# Access to Server Link for api ( Demo as well as Live)

**Demo API Url :**

-

**[http://demodashboardapi.shipsy.in/api/customer/integration/consignment/softdata](http://demodashboard.shipsy.in:3001/api/customer/integration/consignment/softdata#sent/_blank)**

**Production API Url:-**

**[https://dtdcapi.shipsy.io/api/customer/integration/consignment/softdata](https://app.shipsy.in/ap%20%20%20https://https://dtdcapi.shipsy.io/api/customer/integration/consignment/softdatai/customer/integration/consignment/softdata#sent/_blank)**

Add the following header for authentication: **"api-key", "<API KEY>"**

i.e., add a header with name "api-key" and value <API KEY>. The API KEY would have been shared separately.

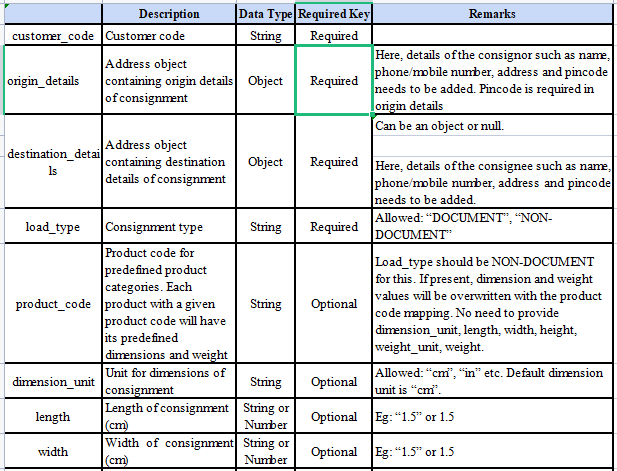
**Set ‘Content-Type’ header to ‘application/json’ Method :- Post**

*There is a limit of 20 consignments per API request, however, multiple parallel requests are allowed.*



Consignments: An array of consignment objects, required

Following is the description, its data type, whether required or not and possible example for every key of consignment,





|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Description** | **Data Type** | **Required Key** | **Remarks** |
| commodity\_id | ID for Content name of the consignment | String | Mandatory in Non DOcs | Eg: “1” for “Laptop”  “2” for “Mobile” etc |
| reference\_number | Reference number/ awb number of consignment | String | Optional | eg: “REF123” |
| cod\_amount | COD amount to be collected for the consignment under consideration | Number or String | Optional | Eg: “2800” or 2800 |
| cod\_collection\_mode | COD collection mode | String | Optional | Allowed: “cash”, “cheque”  “dd”. |
| cod\_favor\_of | Details of the entity in whose name collection is to be made | string | Optional |  |
| declared\_value | Declared value of consignment | Number or String | If load\_type is NON-DOCUMENT, then it is required | Eg: “2800” or 2800 |
| num\_pieces | Number of pieces of Consignment | Number or String | Optional | Eg: “10” or 10 |
| eway\_bill | Eway Bill Number of Consignment | String | Optional | Eg: “12878382929” |
| invoice\_number | Invoice Number of Consignment | String | Optional | Eg: “C3447827878470” |
| invoice\_date | Invoice Date of Consignment | String | Optional | Eg: “25 Jul 2018” |
| consignment\_type | Type of Consignment: Reverse or Forward | String | Optional | ‘reverse’ for reverse pickup and ‘’ for forward pickup. |
| customer\_reference\_number | Customer reference number | String | Optional | Eg: “ABCD” or “1234” or “A1234” |
| consignment\_type | Consignment Type(‘REVERSE’ for reverse shipments) | String | Optional |  |



**Address Object Origin and Destination:**

Following is the description, its data type, whether required or not and possible example for every key of Address,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Name** | **Description** | **Type** | **Required** | **Remarks** |
| pincode | Pincode details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “122002” |
| name | Name of consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “Nikhil” |
| phone | Phone number of Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “8586999698” |
| address\_line\_1 | Address details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “Carlton Estate” |
| address\_line\_2 | Address details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Optional | Eg: “Phase 5” |
| city |  | String | Optional |  |
| state |  | String | Optional |  |



**Return Address Object:**

Following is the description, its data type, whether required or not and possible example for every key of Address for the Return address details and only applicable when RTO address is different from the origin address in that case customer need to mention the return address of the RTO’d shipment, so RTO will be initiated on return address field.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Name** | **Description** | **Type** | **Required** | **Remarks** |
| pincode | Pincode details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “122002” |
| name | Name of consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “Nikhil” |
| phone | Phone number of Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “8586999698” |
| address\_line\_1 | Address details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “Carlton Estate” |
| address\_line\_2 | Address details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Optional | Eg: “Phase 5” |
| city |  | String | Optional |  |
| state |  | String | Optional |  |
| Country |  | String | optional |  |
| Email |  | String | optional |  |
| Alt Phone |  | String | Required | Eg: “8586999698” |



**Exceptional Return Address Object:**

Following is the description, its data type, whether required or not and possible example for every key of Address for the Exceptional Return address details and only applicable when Central processing center RTO address is different from the origin address and Return address, in that case customer need to mention the exceptional return address of the RTO’d shipment, so RTO will be initiated on Exceptional return address field.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Name** | **Description** | **Type** | **Required** | **Remarks** |
| pincode | Pincode details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “122002” |
| name | Name of consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “Nikhil” |
| phone | Phone number of Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “8586999698” |
| address\_line\_1 | Address details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Required in Origin and Destination | Eg: “Carlton Estate” |
| address\_line\_2 | Address details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Optional | Eg: “Phase 5” |
| city |  | String | Optional |  |
| state |  | String | Optional |  |
| Country |  | String | optional |  |
| Email |  | String | optional |  |
| Alt Phone |  | String | Required | Eg: “8586999698” |



**Piece Details Object:**

Following is the description, its data type, whether required or not and possible example for every key of Address,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Key Name** | **Description** | **Type** | **Required** | **Remarks** |
| description | Piece details of the Consignor (in case of origin) or Consignee (in case of destination) | String | Optional. | Eg: “Notebook” |
| declared\_value | Declared value of Consignment Piece | String or Number | Required in pieces\_details for NON-DOCUMENT | Eg: “15000” |
| weight | Weight of consignment piece (kg) | String or Number | Required in pieces\_details for NON-DOCUMENT | Eg: “1.5” or 1.5 |
| height | Height of consignment piece (cm) | String or Number | Required in pieces\_details for NON-DOCUMENT | Eg: “1.5” or 1.5 |
| length | Length of consignment piece (cm) | String or Number | Required in pieces\_details for NON-DOCUMENT | Eg: “1.5” or 1.5 |
| width | Width of consignment piece (cm) | String or Number | Required in pieces\_details for NON-DOCUMENT | Eg: “1.5” or 1.5 |

# 



|  |
| --- |
| **Json Source code For DOC shipment**  {  "consignments": [  {  "customer\_code": "GL112",  "reference\_number": "",  "service\_type\_id": "PRIORITY",  "load\_type": "DOCUMENT",  "description": "Notebook",  "cod\_favor\_of": "Ram Manohar",  "cod\_collection\_mode": "Cheque",  "consignment\_type": "Forward",  "dimension\_unit": "",  "length": "",  "width": "",  "height": "",  "weight\_unit": "kg",  "weight": "0.25",  "declared\_value": "",  "cod\_amount": "100",  "num\_pieces": "002",  "customer\_reference\_number": "XF878723",  "is\_risk\_surcharge\_applicable": true,  "origin\_details": {  "name": "Shipsy",  "phone": "9971149561",  "alternate\_phone": "9876543210",  "address\_line\_1": "B-23 Sushant Lok I",  "address\_line\_2": "Opp. Bestech Centre Point Mall",  "pincode": "400063",  "city": "Mumbai",  "state": "Maharashtra"  },  "destination\_details": {  "name": "Shipsy",  "phone": "9971149561",  "alternate\_phone": "9876543210",  "address\_line\_1": "B-23 Sushant Lok I",  "address\_line\_2": "Opp. Bestech Centre Point Mall",  "pincode": "400063",  "city": "Mumbai",  "state": "Maharahstra"  },  "pieces\_detail": [  { "description": "Notebook",  "declared\_value": "100",  "weight": "0.5",  "height": "5",  "length": "5",  "width": "5"  }  ] } ]} |



**Json Source code For NON-DOC shipment**

{

"consignments": [

{

"customer\_code": "GL112",

"reference\_number": "",

"service\_type\_id": "PRIORITY",

"load\_type": "NON-DOCUMENT",

"description": "Notebook",

"cod\_amount": "250",

"cod\_favor\_of": "Rahul",

"cod\_collection\_mode": "cash",

"num\_pieces": "2",

"dimension\_unit": "cm",

"length": "2",

"width": "3",

"height": "4",

"weight\_unit": "kg",

"weight": "1.5",

"declared\_value": "100",

"eway\_bill": "12320012385958",

"invoice\_number": "C378942320",

"invoice\_date": "25 Jul 2018",

"customer\_reference\_number": "XF4787246",

"commodity\_id": "Laptop",

"consignment\_type": "Forward",

"origin\_details": {

"name": "Shipsy",

"phone": "9971149561?",

"alternate\_phone": "9876543210",

"address\_line\_1": "B-23 Sushant Lok I",

"address\_line\_2": "Opp. Bestech Centre Point Mall",

"pincode": "400063",

"city": "Mumbai",

"state": "Maharashtra"

},

"destination\_details": {

"name": "Shipsy",

"alternate\_phone": "9876543210",

"phone": "9971149561?",

"address\_line\_1": "B-23 Sushant Lok I",

"address\_line\_2": "Opp. Bestech Centre Point Mall",

"pincode": "122009",

"city": "Gurgaon",

"state": "Haryana"

},

"return\_details": {

"name": "Abhishek",

"phone": "9971149561",

"alternate\_phone": "9876543210",

"address\_line\_1": "B-23 Sushant Lok I",

"address\_line\_2": "Opp. Bestech Centre Point Mall",

"pincode": "560001",

"city": "udupi",

"state": "karnataka",

"Country":"India",

"email": "rakesh@gmail"

},

"exceptional\_return\_details": {

"name": "Amit",

"phone": "9971149561",

"alternate\_phone": "9876543210",

"address\_line\_1": "B-23 Sushant Lok I",

"address\_line\_2": "Opp. Bestech Centre Point Mall",

"pincode": "560001",

"city": "Udupi",

"state": "Karnataka",

"Country":"India",

"email": "rakesh@gmail"

},

"pieces\_detail": [

{

"description": "Notebook",

"declared\_value": "100",

"weight": "0.5",

"height": "5",

"length": "5",

"width": "5"

},

{

"description": "letter",

"declared\_value": "100",

"weight": "0.5",

"height": "5",

"length": "5",

"width": "5"

} ] } ] }

# 

# 

|  |
| --- |
| **Dummy Curl Request to call api**  ?php  $curl = curl\_init();  curl\_setopt\_array($curl, array(  CURLOPT\_URL => "http://demodashboardapi.shipsy.in/api/customer/integration/consignment/softdata",  CURLOPT\_RETURNTRANSFER => true,  CURLOPT\_ENCODING => "",  CURLOPT\_MAXREDIRS => 10,  CURLOPT\_TIMEOUT => 30,  CURLOPT\_HTTP\_VERSION => CURL\_HTTP\_VERSION\_1\_1,  CURLOPT\_CUSTOMREQUEST => "POST",  CURLOPT\_POSTFIELDS => "{ Insert Json source code here }",  CURLOPT\_HTTPHEADER => array(  "Authorization: Basic aGltZ3VwOg==",  "Postman-Token: c096d7ba-830d-440a-9de4-10425e62e52f",  "api-key: x5zn!w740gwZOJNzHt5hF$!r9uGzz!n8",  "cache-control: no-cache",  "customerId: 259",  "organisation-id: 1"  ),  ));  $response = curl\_exec($curl);  $err = curl\_error($curl);  curl\_close($curl);  if ($err) {  echo "cURL Error #:" . $err;  } else {  echo $response;  } |



Response Codes

|  |  |
| --- | --- |
| Response Code | Remarks |
| 200 | Each consignment will be processed independently. For each consignment, “success” key will be either true or false. If success is true for a consignment, then the consignment is successfully entered into shipsy system. If success is false, then that consignment is not entered (in case of false, the response contains an error message and reason) |
| 400 | There is some validation error in the overall request format. In this case, complete request is rejected |
| 401 | There is authentication error |

## 



**Sample Response for 200 - I**

When response is 200, it means that there is no error in the structure of the request. But there can be validation errors in individual consignments. Below is the sample response for a request in which two consignments, ‘E12345678’ and ‘E87654321’ are sent.

|  |
| --- |
| {  "status": "OK",  "data": [  {  "success": true,  "reference\_number": "E12345678",  "pieces": [   {  "reference\_number": "E12345678001"  },  {  "reference\_number": "E12345678002"  }  ]  },  {  "success": true,  "reference\_number": "E87654321"   },  {  "success": true,  "reference\_number": "E87654400"   }  ] } |
|  |

The above response means that:

* For consignment ‘E12345678’, there are multiple pieces.
* For consignment ‘E87654321’, there is a single piece.
* For cconsignment ‘E87654400’, there is no reference\_number provided.

## **Sample Response for 200 - II**

When response is 200, it means that there is no error in the structure of the request. But there can be validation errors in individual consignments.

Below is the sample response for a request in which two consignments, ‘SHIPSYTEST1’ and ‘SHIPSYTEST2’ are sent

|  |
| --- |
| {  "status": "OK",  "data": [  {  "success": false,  "message": "Service type not found",  "reason": "SERVICE\_TYPE\_NOT\_FOUND",  "reference\_number": "SHIPSYTEST1",  "should\_retry": false  },  {  "success": true,  "reference\_number": "SHIPSYTEST2"  },  {  "reason": "Invalid Declared Price",  "success": false,  "message": "Invalid Declared Price",  "should\_retry": false  }  ]  } |
|  |

**The above response means that:**

* For consignment ‘SHIPSYTEST1’, success is false. The error code is SERVICE\_TYPE\_NOT\_FOUND and error message is Service type not found. Success is false means that request for ‘SHIPSYTEST1’ is rejected
* For consignment ‘SHIPSYTEST2’, success is true. It means that ‘SHIPSYTEST2’ is successfully processed
* For consignment with no reference\_number, success is false. It means consignment was not created so virtual series reference\_number is not assigned.

## 

## **Sample Response for 400**

{

"error": {

"message": 'Number of consignments in a single request should not be greater than 20

"reason": "MAX\_MANIFEST\_LIST\_LENGTH\_EXCEEDED"

}

}

There is an “error” key, which has the error object. Error object contains

* “message”: Error description
* “reason”: Error code

## **Sample Response for 401**

* If API key is not present

{

"error": {

"message": "API key should be present",

"reason": "NO\_API\_KEY"

}

}

* If Wrong Api key is used

{

"error": {

"message": "Wrong api key",

"statusCode": 401,

"reason": "WRONG\_API\_KEY"

}

}

There is an “error” key, which has the error object. Error object contains

* “message”: Error description
* “reason”: Error code



