
openapi: 3.0.0

info:

title: Shippit

version: 3.0.20220623

description: The following documentation describes the current Shippit API (version 3).

termsOfService: <https://www.shippit.com/terms-of-service/>

contact:

url: <https://support.shippit.com>

email: support@shippit.com

x-logo:

url: <https://www.shippit.com/wp-content/uploads/Unorganized/>

shippit-logo.png

altText: Shippit

href: <https://www.shippit.com/>

servers:

- url: <https://app.staging.shippit.com/api/3>

description: Shippit API Staging

- url: <https://app.shippit.com/api/3>

description: Shippit API Production

paths:

/orders:

description: ""

post:

requestBody:

description: Passes an Order object under the `order` field.

content:

application/json:

schema:

\$ref: '#/components/schemas/OrderRequest'

examples:

Order with minimum required fields:

value:

order:

courier_type: standard

delivery_address: 1 Union Street

delivery_postcode: "2009"

delivery_state: NSW

delivery_suburb: Pyrmont

authority_to_leave: "Yes"

parcel_attributes:

- qty: 1

weight: 2.1

user_attributes:

email: test@shippit.com

first_name: John

last_name: Smith

Order request with optional fields:

value:

order:

user_attributes:

email: test@shippit.com

```
    first_name: John
    last_name: Smith
    parcel_attributes:
    - depth: 0.13
      length: 0.1
      location: C123
      price: 12.12
      qty: 1
      sku: EWPE123123
      title: Super awesome red tshirt
      weight: 2.1
      width: 0.11
      product_line_id: "1234"
      origin_country_code: CN
      packed: 1
    - depth: 0.13
      length: 0.1
      location: B22
      price: 1.21
      qty: 7
      sku: EWPE123123
      title: Super awesome blue tshirt
      weight: 2.1
      width: 0.11
      dangerous_goods_code: ID8000
      dangerous_goods_text: ID8000 Consumer
```

commodities - Dangerous

```
    Goods as per attached DGD
    product_line_id: "1235"
    packed: 1
    authority_to_leave: "No"
    courier_type: standard
    delivery_address: 1 Union Street
    delivery_postcode: "2009"
    delivery_state: NSW
    delivery_suburb: Pyrmont
    delivery_instructions: Special instructions for
```

the order go here

```
    receiver_contact_number: "0400000000"
    receiver_name: Adam
    retailer_invoice: '#23201005'
    retailer_reference: '#23201005'
    duties: 100.99
    tax_amount: 82.811
    customs_clearance_attributes:
      tax_id_type: FTZ
      tax_id_country_code: GB
      tax_id_number: '*****'
      incoterm: DAP
      export_reason: Sample
      recipient_id: '*****'
      receiver_freight_charge_amount: 7.99
```

Order request using product_attributes:
value:

```
order:
  user_attributes:
    email: test@shippit.com
    first_name: John
    last_name: Smith
  parcel_attributes:
    - weight: 4.4
    - weight: 8.8
  product_attributes:
    - quantity: 10
      title: Super awesome red tshirt
    - quantity: 20
      title: Super awesome blue tshirt
  authority_to_leave: "No"
  courier_type: standard
  delivery_address: 1 Union Street
  delivery_postcode: "2009"
  delivery_state: NSW
  delivery_suburb: Pyrmont
Priority Order request:
value:
  order:
    courier_type: priority
    delivery_address: 1 Union Street
    delivery_postcode: "2009"
    delivery_state: NSW
    delivery_suburb: Pyrmont
    authority_to_leave: "Yes"
    delivery_date: 2016-07-26
    delivery_window: 16:00-19:00
    parcel_attributes:
      - qty: 1
        weight: 2.1
    user_attributes:
      email: test@shippit.com
      first_name: John
      last_name: Smith
Tracking Order without tracking histories:
value:
  order:
    delivery_address: 1 Union Street
    delivery_postcode: "2099"
    delivery_state: NSW
    delivery_suburb: Dee Why
    delivery_country_code: au
    courier_allocation: CouriersPlease
    courier_job_id: SAMPLE-LN-111
    authority_to_leave: "Yes"
    tracking_only: true
    parcel_attributes:
      - qty: 1
        weight: 2.1
        label_number: SAMPLE-LN-111001
    user_attributes:
```

```
        email: test@shippit.com
        first_name: John
        last_name: Smith
Tracking Order with tracking histories:
  value:
    order:
      delivery_address: 1 Union Street
      delivery_postcode: "2099"
      delivery_state: NSW
      delivery_suburb: Dee Why
      delivery_country_code: au
      courier_allocation: CouriersPlease
      courier_job_id: SAMPLE-LN-111
      authority_to_leave: "Yes"
      tracking_only: true
      parcel_attributes:
        - qty: 1
          weight: 2.1
          label_number: SAMPLE-LN-111001
      user_attributes:
        email: test@shippit.com
        first_name: John
        last_name: Smith
      tracking_histories:
        - status: order_placed
          timestamp: 2018-10-10T05:28:12.693Z
        - status: ready_for_pickup
          timestamp: 2018-10-11T05:28:12.693Z
OnDemand Order request:
  value:
    order:
      courier_type: on_demand
      delivery_address: 1 Union Street
      delivery_postcode: "2009"
      delivery_state: NSW
      delivery_suburb: Pyrmont
      delivery_date: 2016-07-26
      authority_to_leave: "Yes"
      parcel_attributes:
        - qty: 1
          weight: 1.1
          width: 0.1
          length: 0.1
          depth: 0.1
      user_attributes:
        email: test@shippit.com
        first_name: John
        last_name: Smith
        receiver_contact_number: "0400000000"
Return Order:
  value:
    order:
      user_attributes:
        email: customer@email.com
```

```

        first_name: John
        last_name: Smith
        parcel_attributes:
        - qty: "1"
          weight: "0.5000"
          length: "0.1500"
          width: "0.1100"
          depth: "0.1000"
          title: Product Title
          price: "50.00"
          sku: "123456789123"
        validate: true
        courier_type: standard
        retailer_invoice: '#ABC123456'
        receiver_contact_number: (123) 456789
        pickup_postcode: "2009"
        pickup_address: 1 Union Street
        pickup_suburb: Pyrmont
        pickup_state: NSW
        return: true
    Order with Apple Pay Wallet Order Tracking:
    value:
    order:
        courier_type: standard
        delivery_address: 1 Union Street
        delivery_postcode: "2009"
        delivery_state: NSW
        delivery_suburb: Pyrmont
        authority_to_leave: "Yes"
        wallet_order_tracking: true
        parcel_attributes:
        - qty: 1
          weight: 2.1
        user_attributes:
        email: test@shippit.com
        first_name: John
        last_name: Smith
    required: true
tags:
- Order
responses:
  "200":
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/OrderResponse'
        examples:
          Example Order Response:
            value:
              response:
                courier_delivery_instructions: special
instructions
        courier_job_id: "30734876324"
        courier_name: eParcel

```

instructions

delivery_address: 1 Union Street
delivery_instructions: test special

delivery_postcode: "2009"
delivery_state: NSW
delivery_suburb: Pyrmont
id: 26599
parcel_attributes:
- depth: 0.13
length: 0.1
qty: 1
weight: 16.8
width: 0.11
products:
- title: Industrial Paint Stripper
price: 29.13
sku: 0.1
tariff_code: "000999"
dangerous_goods_code: ID8000
dangerous_goods_text: ID8000 Consumer

commodities - Dangerous

Goods as per attached DGD
origin_country_code: TH
quantity: 1
price: "0.0"
processing_state: created
receiver_contact_number: "0400000000"
receiver_name: Josh
retailer_invoice: '#23201005'
slug: ppu38wz2tdonj
state: processing
tracking_number: PPu38Wz2TdoNj
user_attributes:
email: test@shippit.com
first_name: jon
last_name: smith
mobile: "0491570006"

Return Order Response:

value:

response:

id: 3020594
tracking_number: PPvd8Ba8g9r3B
state: processing
processing_state: created
delivery_address: 1 Victoria St
delivery_suburb: Melbourne
delivery_postcode: "3000"
delivery_state: VIC
receiver_name: John Smith
receiver_contact_number: (123) 456789
slug: ppvd8ba8g9r3b
price: "0.0"
retailer_invoice: '#ABC123456'
user_attributes:

email: customer@email.com
first_name: John Smith
mobile: (123) 456789

parcel_attributes:

- qty: 1
length: 0.325
width: 0.205
depth: 0.03
weight: 0.5

products:

- price: 50
quantity: 1
title: Product Title
sku: "9354484070976"
origin_country_code: AU

return: true

pickup_address: 1 Union Street

pickup_suburb: Pyrmont

pickup_postcode: "2009"

pickup_state: NSW

Order with Apple Pay Wallet Order Tracking Response:
value:

response:

instructions

courier_delivery_instructions: special

courier_job_id: "30734876324"

courier_name: eParcel

delivery_address: 1 Union Street

instructions

delivery_instructions: test special

delivery_postcode: "2009"

delivery_state: NSW

delivery_suburb: Pyrmont

id: 26599

parcel_attributes:

- depth: 0.13
length: 0.1
qty: 1
weight: 16.8
width: 0.11

products:

- title: Industrial Paint Stripper
price: 29.13
sku: 0.1
tariff_code: "000999"
dangerous_goods_code: ID8000
dangerous_goods_text: ID8000 Consumer

commodities - Dangerous

Goods as per attached DGD

origin_country_code: TH

quantity: 1

price: "0.0"

processing_state: created

receiver_contact_number: "0400000000"

```

        receiver_name: Josh
        retailer_invoice: '#23201005'
        slug: ppu38wz2tdonj
        state: processing
        tracking_number: PPu38Wz2TdoNj
        user_attributes:
            email: test@shippit.com
            first_name: jon
            last_name: smith
            mobile: "0491570006"
        wallet_order_tracking_attributes:
            order_type_identifier:
apple.order.type.identifier
            order_identifier: ppu38wz2tdonj
            web_service_url: https://api.shippit.com/
appleserviceurl
        authentication_token: ABCDEFG123456789
        description: Successful order response
    "400":
        $ref: '#/components/responses/BadRequestErrorResponse'
    "403":
        $ref: '#/components/responses/
InvalidMerchantAccountErrorResponse'
    "422":
        content:
            application/json:
                schema:
                    $ref: '#/components/schemas/
OrderValidationErrorResponse'
        description: |-
            Returned when `validate` is set to `true` and the order
            address is not valid.

            Will specify which input fields failed validation and
            may give a `suggested_address` form if one is found.
    "500":
        $ref: '#/components/responses/UnknownErrorResponse'
    security:
        - AuthToken: []
    operationId: createOrder
    summary: Create an Order
    description: "Submits an Order to be created on Shippit.
\r\n\r\nAt minimum,\
    \ an Order requires a delivery location, user details, and
parcel details.\
    \ Shippit will then generate the order, allocate the
courier, and fill in\
    \ the origin location based on the provided info and
merchant configuration\
    \ on Shippit.\r\n\r\nNote that the required fields for an
Order can vary depending\
    \ on the type of order, the requested courier, whether it is
local or international,\
    \ etcetera.\r\n\r\nThere is a maximum of 1000 parcels per

```



```

quote request.\r\
  \n\r\n\r\n---\r\n\r\n\r\n### Order Lifecycle\r\n\r\n#### Standard
and Express Lifecycle\r\
  \n| Web UI action | API call      | State on completion
| Where seen\
  \ in UI    |
\r\n|-----|-----|-----|-----|
-----|\r\
  \n| Add order      | POST order    | New Order
| New Orders\
  \ page    |\r\n| Confirm order | GET label     | Ready to
Ship      \
  \ | Ready to Ship page |\r\n| Cancel order  | DELETE order
| Cancelled \
  \ | Not visible      |\r\n| Book order
| POST book \
  \ | Booked          | Track page    |
\r\n\r\n\r\n#### Click\
  \ and Collect Lifecycle\r\n| Web UI action | API call      |
State on completion\
  \ | Where seen in UI |
\r\n|-----|-----|-----|-----|
-----|\r\
  \n| N/A            | POST order    | Packing order
| New Orders\
  \ page    |\r\n| Label          | GET label     | Packing
order      \
  \ | New Orders page  |\r\n| Cancel order  | DELETE order
| Cancelled \
  \ | Not visible      |\r\n| Book order
| POST book \
  \ | Booked          | Track page    |
\r\n\r\n\r\nUnlike standard\
  \ / express orders, click and collect orders do not proceed
to the Ready to\
  \ Ship page. They remain in the New Orders page until you
call either the\
  \ Book API or click the Label button.\r\n\r\n\r\n#### Priority
and Ondemand Order\
  \ Lifecycle\r\n| Web UI action | API call      | State on
completion  \
  \ | Where seen in UI |
\r\n|-----|-----|-----|-----|
-----|\r\
  \n| N/A            | POST order    | Packing order
| New Orders\
  \ page    |\r\n| Label          | GET label     | Booked
\
  \ | Track page      |\r\n| Cancel order  | DELETE order
| Cancelled \
  \ | Not visible      |\r\n\r\n\r\nUnlike
standard / express\
  \ orders, priority and ondemand orders do not need the book
call. They are\

```

\ automatically booked by Shippit on the delivery date.
 \r\n\r\nOndemand orders\
 \ are automatically booked by Shippit for an ASAP pickup,
 respecting store\
 \ pick-and-pack time.\r\nPriority and Ondemand orders will
 remain on the New\
 \ Orders page until dismissed by clicking the Label button
 or the GET label\
 \ API call is made.\r\n\r\n#### Return Order Lifecycle\r\n|
 Web UI action\
 \ | API call | State on completion |
 Where seen in\
 \ UI |
 \r\n|-----|-----|-----|
 -----|-----|-----|\r\
 \n| N/A | POST order | New Order
 \
 \ | New Orders page |\r\n| Confirm order | GET label |
 Awaiting drop\
 \ off / Return requested | Track page |\r\n|
 Cancel order | DELETE\
 \ order | Cancelled | Not
 visible |\r\n\
 \r\nUnlike outbound orders, return orders do not proceed to
 Ready to Ship\
 \ page. They move directly to the Track page and change
 their status to Awaiting\
 \ drop off, for AusPost drop offs or Return requested, for
 carrier pickups.\r\n* Carrier pickup – A courier will attend the
 recipient's address and\
 \ collect the product\r\n * Note: Recipient must initiate
 the return by\
 \ opening the new tracking page and pressing a button to
 schedule the pickup.\r\n\ Right now there is no ability for the recipient to choose
 a date or time\
 \ slot for the pickup, as this is not offered by the
 carriers.\r\n* AusPost\
 \ drop off – Your recipient must return the product to an
 AusPost location\r\
 \n\r\n\r\n### Order validation\r\nIf the `validate` field is
 passed and set\
 \ to `true`, then the order is not saved unless Shippit can
 validate the destination\
 \ suburb and postcode to be a valid combination. If a valid
 combination is\
 \ not found, Shippit will return a suggested address.
 \r\n\r\n### Manual or\
 \ Automatic Courier selection\r\nYou can allocate a courier
 to an order in\
 \ one of two ways: you can manually specify the courier as
 part of the order,\r\n\ or you can allow Shippit to automatically allocate a

```

courier for you.\r\n
    \r\nTo manually specify a courier, pass the
`courier_allocation` field with\
    \ the identifier of the courier (see below).\r\n\r\nTo have
Shippit allocate\
    \ the courier for you, pass the `courier_type` field to
specify the type of\
    \ couriers that Shippit can choose from to allocate for you.
This is used\
    \ to stand for a service level such as `standard`,
`express`, `priority`,\
    \ `click_and_collect`.\r\n\r\n\r\n### Specifying Parcels and
Products\r\n\r\n
    Shippit needs to know the specifications of the products you
want shipped\
    \ and what parcels to use in shipping them. There is a
maximum of 1000 parcels\
    \ per quote request.\r\n\r\n\r\nIn the simplest case, this can
be specified by\
    \ a `parcel_attributes` field, which contains a list of
Parcel specifications.\
    \ Each item specifies the number, dimensions, and other
information about\
    \ the products being shipped, and the associated parcel. In
this case, each\
    \ parcel contains one type of product.\r\n\r\n\r\nFor more
complex shipping arrangements,\
    \ you can additionally specify a `product_attributes` field,
which contains\
    \ a list of Product specifications. In this case, the
`parcel_attributes`\
    \ will no longer contain product information such as `qty`.
Each item under\
    \ `product_attributes` would specify the number, dimensions,
and other information\
    \ about the products; while each item under
`parcel_attributes` would separately\
    \ specify the dimensions of the parcels. By mixing parcel
and product specifications,\
    \ you can define arbitrary groupings of products into
parcels.\r\n\r\n\r\nThe\
    \ number of parcels is further determined by the `Allocate
each item in an\
    \ order to a separate carton` setting in the Shippit
administration page.\
    \ If this is enabled, then each item will be assigned a
separate parcel according\
    \ to its dimensions. If this is disabled, then Shippit may
combine multiple\
    \ product or parcel specifications into one.\r\n\r\n\r\n\r\n###
Standard orders\
    \ vs tracking orders\r\n\r\nIn a standard order, Shippit will
book the courier\
    \ given the provided information and provide tracking

```

information and notifications\

- \ for the merchant and recipient on the status of the order.

\r\n\r\nIf you've\

- \ already booked and allocated a courier to your consignment outside of Shippit,\
- \ however, you can still use Shippit to provide tracking and notifications\
- \ for the order. This is called a ****tracking order****. In this case, you're\
- \ only using Shippit for the tracking and notifications, and not for carrier\
- \ booking.\r\n\r\nCreating a tracking order is the same as creating a standard\
- \ order, only that you have to provide additional attributes so that Shippit\
- \ can match the the courier's internal info to shippit. Note the following\
- \ attributes:\r\n\r\n- `tracking_only` _required_ -- true - this makes this\
- \ order a tracking order.\r\n- `courier_allocation` _required_ -- the name\
- \ of the courier in charge of the item.\r\n- `courier_job_id` -- the ID assigned\
- \ by some carriers to the shipment / consignment.\r\n- `parcel_attributes`\
- \ -> `label_number` _required_ -- the label assigned by the carrier system\
- \ to a parcel, used by Shippit to match the parcel number against the carrier's\
- \ parcel number\r\n- `tracking_histories` -- one or two optional status events\
- \ that will be added to the tracking page for the benefit of the recipient.\r\n

\n\r\n#### Passing tracking histories\r\nIn a tracking order, you can choose\

- \ to pass tracking histories for the benefit of the recipient. These are shown\
- \ to the recipient in the tracking page to inform them of the status of when\
- \ the order has been processed.\r\n\r\nThese items have just two fields, a\
- \ `status` field and a `timestamp` in UTC format, to indicate to the recipient\
- \ events such as `order_placement` and `ready_for_pickup`. The `status` informs\
- \ the user of the Order's state at that point in time and can be either `order_placed`\
- \ or `ready_for_pickup`. The `timestamp` indicates when that change happened.\
- \ The following tracking history entries are recommended to be placed:\r\n\r\n

- \r\n- `order_placed` - set to the time when you received the order on your\
- \ system.\r\n- `ready_for_pickup` - set to the time when you

book the order\
 \ on Shippit.\r\n\r\nIf no `tracking_histories` entries are
passed, Shippit\
 \ will generate an initial tracking history for the order,
set to `ready_for_pickup`\
 \ at the point when Shippit received the API request.
\r\n\r\n### Authority\
 \ to leave\r\n* When using 500g satchels,\r\n * When
specifying ``No``,\
 \ you must have ``Satchel 500g (A5)`` enabled in your
settings (Settings\
 \ > Pick & Pack > Automatic Package Sorting > Carrier
Default Presets). \r\
 \r\n * When specifying ``Yes``, you must have ``Satchel
500g ATL (A5)``\
 \ enabled in your settings (Settings > Pick & Pack >
Automatic Package Sorting\
 \ > Carrier Default Presets).\r\n * If nothing is
specified, we assume ATL\
 \ = ``No``\r\n\r\n* Note that the final ATL value also
depends on how couriers\
 \ are allocated and if users manually override the ATL
option in the UI. It\
 \ is important when confirming quotes that the ATL is set as
desired.\r\n\r\n\r\n### Return orders\r\n\r\nYou can see an example request
for a return order\
 \ on the right hand side under: *Request samples > Example >
Return Order*.\r\
 \r\n\r\nThe following 5 params are essential and `return` must
be true:\r\n\r\n * `pickup_postcode`\r\n\r\n* `pickup_address`\r\n\r\n* `pickup_suburb`\r\n\r\n* `pickup_state`\r\n\r\n* `return`\r\n\r\n\r\nThis will create a new order in the
background, with the\
 \ pickup/recipient data as the origin and the default
merchant address as\
 \ the destination and also set the order to the state
`awaiting_drop_off`\
 \ or `return_requested`, depending on the carrier
allocation. We can also\
 \ get a label for this order via the default label API.
\r\n\r\n\r\nReceivers will\
 \ receive an email with their return label. This email
provides the customer\
 \ instructions on how to return their package/s. It is
automatically triggered\
 \ by Shippit, once a return order is confirmed by the
merchant. \r\n\r\n\r\n###
 \ Limitations\r\n\r\nDespite the existing order API limitations,
the only addition\
 \ for return orders is the selection of the carrier. Only
certain carriers\
 \ allow return orders. \r\n\r\n\r\nIn most cases, the merchant

chooses only AusPost\
 \ drop off as a valid return option.\r\n\r\nA Shippit admin
 can, on behalf\
 \ of a merchant, turn on or off carriers who would instead
 pick up the parcel\
 \ for returns.\r\n\r\nAs this functionality is using the
 default order API,\r\n
 \ we can also create return orders for parcels that were not
 shipped originally\
 \ via Shippit. As long as we supply the necessary pickup
 data in the request.\r\n
 \r\n\r\n### Apple Pay Wallet Order Tracking (coming soon)
 \r\nAfter Apple Pay\
 \ Wallet Order Tracking is successfully activated on your
 account, it can\
 \ be used by providing an additional value
 `wallet_order_tracking` to the\
 \ POST /orders request.\r\n\r\nIf the property
 `wallet_order_tracking: true`\
 \ is attached to the order creation request, the response
 will include an\
 \ additional object `wallet_order_tracking_attributes`,
 which contains the\
 \ necessary data for the [Apple Pay integration](https://
 applepaydemo.apple.com/order-tracking#createOrder):\r\n
 \r\n\r\n```\r\n\r\n{\r\n\r\n "response": {\r\n\r\n "id":
 number,\r\n\r\n "tracking_number":
 : string,\r\n\r\n [...] \r\n\r\n
 "wallet_order_tracking_attributes": {\r\n\r\n
 "order_type_identifier": string,\r\n\r\n
 "order_identifier": string,\r\n\r\n
 "web_service_url": string,\r\n\r\n
 "authentication_token": string\r\n\r\n
 }\r\n\r\n }\r\n\r\n}\r\n\r\n```\r\n\r\n\r\n\r\n---\r\n\r\n\r\n\r\n### Service
 Level Mappings\r\n
 \r\n| Standard | Express |
 Priority |
 \r\n| OnDemand |
 \r\n|-----|-----|
 -----|-----|\r\n
 \r\n| AlliedExpressOvernight | CapitalTransport
 | AlliedExpressSameday\
 \r\n| DoorDashOnDemand | AramexAuNz |
 CouriersPleaseExpress\
 \r\n| Bonds | GrabExpress | \r\n |
 AramexExpress \
 \r\n| DirectCouriers | DoorDash
 \
 \r\n| LalamoveOnDemand | AramexInternational |
 EparcelExpress \
 \r\n| EparcelOnDemand | PandagoOnDemand |
 \r\n | CityLink\
 \r\n| EparcelInternationalExpress |
 QxpressSameday \

	\	UberOndemand	\r\n	CouriersPlease	
FedExInternationalPriority\					
	\	YelloOndemand			\r\n
DhlEcommerce	\				
	\	KerryExpress			
\					
	\	\r\n	DhlEcommerceAsia		
NewZealandPostExpress\					
	\				\r\n
DhlExpress	\				
	\	NinjaVanExpress			
\					
	\	\r\n	DhlExpressInternational		
PbtCourierExpress\					
	\				
\r\n	DirectFreightExpress\				
	\	SekoExpress			
\					
	\	\r\n	Eparcel		
SingPostExpress	\				
	\				\r\n
EparcelInternational\					
	\	StarTrackPremium			
\					
	\	\r\n	Fastway		
TntOvernightExpress	\				
	\				\r\n
FedExInternationalEconomy\					
	\	TollPriority			
\					
	\	\r\n	FlashExpress		
\					
	\				\r\n
FourPXStandard	\				
\					
	\	\r\n	Gdex		
\					
	\				\r\n
HunterExpress\					
	\				
\					
	\	\r\n	Janio		
\					
	\				
\r\n	JanioInternational\				
	\				
\					
	\	\r\n	Jnt		
\					
	\				\r\n
KerryStandard\					
	\				
\					

\ | |\r\n |LyneConnect |
\
|\r\n |LynePlus\ | |
\
|\r\n |Neway |\r\n |Neway
|\r\n |NewZealandCouriers\ | |
| \ |
|\r\n |NewZealandPost |
\
|NinjaVanStandard\ | |\r\n
|\ |
|\r\n |PbtCourier |
\
|\r\n |Pgeon \ | |
\
|\r\n |Pickupp |\r\n |Pickupp
|\r\n |PosMalaysia\ | |
\
|\r\n |Qxpress |\r\n |Qxpress
|\r\n |SekoStandard\ | |
\
|\r\n |SingPost |
\
|\r\n |Skybox\ | |
\
|\r\n |SmartRoutingInternational |\r\n |SmartRoutingInternational
|\r\n |StarTrack\ | |
\
|\r\n |Teleport |\r\n |Teleport
|\r\n |TeleportInternational\ | |
| \


```

\
\
\
Toll
\
\
|\
\
/orders/{tracking_number}:
  delete:
    tags:
      - Order
    responses:
      "200":
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/OrderDeleteResponse'
            examples:
              Example successfully cancelled Order:
                value:
                  response:
                    id: 33
                    tracking_number: PPuqD0J0uLs1M
                    state: cancelled
                    processing_state: ' processing_cancelled'
                    delivery_address: 1 Union Street
                    delivery_suburb: Pymont
                    delivery_postcode: "2009"
                    receiver_name: Francois
                    receiver_contact_number: "0404342342"
                    courier_name: eParcel International
                    slug: ppuqd0j0uls1m
                    price: "0.0"
                    retailer_invoice: S042637
                    courier_job_id: ABC0100023
                    user_attributes:
                      email: test@shippit.com
                      first_name: John
                      last_name: Smith
                    parcel_attributes:
                      - qty: 1
                        length: 0.325
                        width: 0.205
                        depth: 0.03
                        weight: 0.5
                    description: Returns the Order with state = `cancelled`
      "403":
        $ref: '#/components/responses/InvalidMerchantAccountErrorResponse'
      "404":
        $ref: '#/components/responses/ResourceNotFound'
      "422":
        content:
          application/json:

```

```

      schema:
        $ref: '#/components/schemas/
OrderDeleteErrorResponse'
      examples:
        Example failed cancellation of Order:
          value:
            error: order_cannot_be_cancelled
            error_description: The requested order is
currently in a state
              that cannot be cancelled
              order_state: in_transit
            description: "Returns an error indicating that the Order
could not be cancelled,\
              \ and stating the reason why."
      security:
        - AuthToken: []
      operationId: cancelOrder
      summary: Cancel an Order
      description: |-
        Cancels an Order in Shippit using the tracking number.

```

The API first checks if the Order can be cancelled, ie., if its current state allows it, then returns an immediate response. If the Order is successfully cancelled, the API will answer with the Order with its state updated as `cancelled`. If it cannot be cancelled, it will throw a `422` error.

```

      parameters:
        - name: tracking_number
          description: The tracking number of the Order.
          schema:
            type: string
            in: path
            required: true
/orders/{tracking_number}/label:
  get:
    tags:
      - Order
      - Label
    parameters:
      - name: tracking_number
        description: The tracking number of the Order.
        schema:
          type: string
          in: path
          required: true
    responses:
      "200":
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/LabelResponse'
            examples:
              Example Successful Label Response:
                value:

```

```
response:
  id: 5044
  order:
    courier_delivery_instructions: 'Authority to
Leave. '
    courier_job_id: SHP0100002
    delivery_address: 37 Manorvale Pde
    delivery_instructions: ""
    delivery_postcode: "3030"
    delivery_state: VIC
    delivery_suburb: Werribee
    id: 9972
    invoice_number: INV-1384
    parcels:
      - depth: 0.19
        length: 0.38
        name: ""
        weight: 1
        width: 0.29
        label_number: SHP010000201000930803
        courier_data:
          product_code: X123
          pickup_zone: SYD
          dropoff_zone: MEL
    products:
      - title: Industrial Paint Stripper
        price: 29.13
        sku: 0.1
        quantity: 1
        tariff_code: "000999"
        dangerous_goods_code: ID8000
        dangerous_goods_text: ID8000 Consumer
    commodities - Dangerous
      Goods as per attached DGD
      origin_country_code: TH
      price: "15.95"
      receiver_contact_number: ""
      receiver_language_code: EN
      receiver_name: Jane Doe
      retailer_invoice: ""
      slug: ppkfqy44u8nff
      state: completed
      tracking_number: PPKFqy44U8Nff
      user:
        email: jane.doe@shippit.com
        first_name: Jane
        last_name: 'Doe '
        mobile: ""
      customs_documents_require_printing: false
      documents:
        archive_awb:
          url: https://shippit-web-
production.s3.amazonaws.com/uploads/order/archive_document/15870762/
archive_581833317220190318-2852-yi07y0.PDF
```

```

        page_size: a6
        file_type: pdf
        customs_invoice:
            url: https://shippit-web-
production.s3.amazonaws.com/uploads/order/customs_invoice/361690/
commercial_invoice_3073489348620190218-6221-mxrhx4.pdf
        page_size: a3
        file_type: pdf
        dangerous_goods_declaration:
            url: https://shippit-web-
production.s3.amazonaws.com/uploads/order/
dangerous_goods_declaration/361690/
dangerous_goods_declaration_3073489348620140318-6220-11vib5y.pdf
        page_size: a3
        file_type: pdf
        shipping_label:
            url: https://shippit-
production.s3.amazonaws.com/uploads/label/attachment/5044/
PPKFqy44U8Nff.pdf
        page_size: a6
        file_type: pdf
        qualified_url: https://shippit-
production.s3.amazonaws.com/uploads/label/attachment/5044/
PPKFqy44U8Nff.pdf
    description: Returns an Order and related label
information.
    "404":
        content:
            application/json:
                schema:
                    $ref: '#/components/schemas/GenericErrorResponse'
                examples:
                    Example Order not found response:
                        value:
                            error: order_not_found
                            error_description: The requested order could not
be found.
                description: Returns an error indicating that the Order
could not be found
                in Shippit
    "422":
        content:
            application/json:
                schema:
                    $ref: '#/components/schemas/GenericErrorResponse'
                examples:
                    Example due to allocation rules engine:
                        value:
                            error: label_failed_rules_engine
                            error_description: The requested order has not
been allocated
                                a valid courier due to allocation rules.
                Example order is tracking order:
                    value:

```

error: label_failed_tracking_order
error_description: "You have requested a label
for a tracking\
Shippit. Shippit\
orders."

Example invalid address:
value:
error: address_invalid
error_description: The address provided is
invalid

Example order still processing:
value:
error: label_failed_order_processing
error_description: "The requested order is
currently being processed,\
please try again later"

Example order has no courier:
value:
error: label_failed_missing_courier
error_description: The requested order has no
courier assigned

Example invalid quotes:
value:
error: label_failed_invalid_quote
error_description: The requested order has no
valid quotes

Example label still generating:
value:
error: label_failed_still_generating
error_description: "The requested order is
currently generating\
a label, please try again later"

Example label generation request failed:
value:
error: label_failed_failed_request
error_description: "The requested order failed
to generate a label,\
please contact support"

Example courier deprecated:
value:
error: label_failed_courier_deprecated
error_description: The assigned courier has been
deprecated.

Example unprocessable label:
value:
error: order_not_available_to_label
error_description: "The requested order has not
been processed,\
please try again."

description: |-

Returns an error indicating that the Order cannot yet be
processed for labelling.

Typically this is because the Order has yet to be processed or allocated a courier.

```
security:
- AuthToken: []
operationId: getOrderLabel
summary: Get Label information for an Order
description: |-
  Retrieves labelling information for an Order using the
  tracking number.
```

The labelling information for an Order can only be retrieved once the Order has been processed and allocated a courier, which may take some time after the Order has been placed. If the Order is yet to be processed, you will get a `422` Unprocessable response.

```
parameters:
- name: tracking_number
  description: The tracking number of the Order
  schema:
    type: string
  in: path
  required: true
/orders/{tracking_number}/tracking:
get:
  tags:
  - Track
  parameters:
  - name: tracking_number
    description: The tracking number of the Order
    schema:
      type: string
    in: path
    required: true
  responses:
    "200":
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/TrackingResponse'
          examples:
            Example Tracking Response:
              value:
                response:
                  tracking_number: PP39TBECV7QSSPK
                  tracking_url: http://www.shippit.com/track/
                  success: true
                  track:
                    - status: Completed
                      date: 2015-04-03T00:00:00.000Z
                      timestamp: 58985
                      status_owner: Bonds Couriers
                    - status: With Driver
                      date: 2015-04-03T00:00:00.000Z
```

pp39tbecv7qsspk

```

        timestamp: 55385
        status_owner: Bonds Couriers
      - status: In Transit
        date: 2015-04-03T00:00:00.000Z
        timestamp: 55325
        status_owner: Bonds Couriers
      - status: Ready For Pick Up
        date: 2015-04-03T00:00:00.000Z
        timestamp: 49400
        status_owner: Harold's Harpoons
      - status: Despatch In Progress
        date: 2015-04-03T00:00:00.000Z
        timestamp: 36603
        status_owner: Harold's Harpoons
      - status: Order Placed
        date: 2015-04-03T00:00:00.000Z
        timestamp: 07:23:32
        status_owner: Harold's Harpoons
    description: Returns tracking info related to the Order
  "400":
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/TrackingErrorResponse'
        examples:
          Error Tracking Number does not exist:
            value:
              error:
                order_id:
                  - "Sorry, the tracking number you entered does
not exist."
                error_description: The data given to this server
does not meet
                our criteria.
              success: false
            description: Returns an error if the provided tracking
number does not exist
          operationId: trackOrder
          summary: Track Order
          description: |-
            This sends a tracking request to the API to retrieve the
status of an Order.

```

This uses a pull-based model of Order tracking, which sends tracking info on each request. If you would like to use a push-based model where you automatically receive messages on Order status changes, you can subscribe to the `Tracking Webhook` instead.

```

parameters:
  - name: tracking_number
    description: The tracking number of the order
    schema:
      type: string
    in: path
    required: true

```

```

/quotes:
  post:
    requestBody:
      description: ""
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/QuoteRequest'
            required:
              - dropoff_suburb
              - dropoff_postcode
              - parcel_attributes
            properties:
              dropoff_suburb:
                description: The destination suburb for the quote.
                required:
                  - dropoff_suburb
                type: string
                example: Surry Hills
              dropoff_state:
                description: The destination state for the quote.
                type: string
                example: NSW
              dropoff_postcode:
                description: The destination postcode for the
quote.
                type: string
                example: 2010
              dropoff_country_code:
                description: "The destination country code for the
quote in `ISO03166\
\ Alpha-2` format. If unassigned or set to AU,
`dropoff_state`\
\ is required."
                default: AU
                type: string
                example: AU
              order_date:
                description: "Optional date for the quote, this is
only relevant\
\ to priority orders."
                type: string
                example: 2018-08-01T00:00:00.000Z
              return_all_quotes:
                description: "If set to `true`, the system returns
all quotes instead\
\ of only the cheapest or the fastest quote as
per merchant settings."
                default: false
                type: boolean
                example: true

```


dutiable_amount:
description: The declared value for customs when shipping international.

type: number

example: 199.95

parcel_attributes:

description: An array of items for the quote.

****Note**** that only

`qty` and `weight` are required.

type: array

items:

required:

- qty

- weight

type: object

properties:

qty:

description: Number of items in the parcel.

type: integer

example: 1

weight:

description: Parcel weight in kilograms.

type: string

example: 19.4

length:

description: Parcel length in meters.

type: string

example: 0.1

width:

description: Parcel width in meters.

type: string

example: 0.1

depth:

description: Parcel depth in meters.

type: string

example: 0.1

dangerous_goods_text:

description: Dangerous good attributes are

optional strings.

type: string

example:

quote:

dropoff_suburb: Surry Hills

dropoff_state: NSW

dropoff_postcode: 2010

dropoff_country_code: AU

return_all_quotes: true

parcel_attributes:

- qty: 1

weight: 0.5

length: 0.1

width: 0.2

depth: 0.1

examples:

Example Quote Request:

value:

quote:

dropoff_postcode: "2000"

dropoff_state: NSW

dropoff_suburb: Sydney

parcel_attributes:

- qty: 1

weight: 1

Example Priority Order Quote Request:

value:

quote:

dropoff_address: 1 Test Street

dropoff_suburb: Sydney

dropoff_postcode: "2000"

dropoff_state: NSW

dropoff_country_code: AU

return_all_quotes: true

parcel_attributes:

- qty: 1

weight: 1

service_levels:

- priority

Example OnDemand Order Quote Request:

value:

quote:

dropoff_address: 1 Test Street

dropoff_suburb: Sydney

dropoff_postcode: "2000"

dropoff_state: NSW

dropoff_country_code: AU

return_all_quotes: true

parcel_attributes:

- qty: 1

weight: 1

service_levels:

- on_demand

Example International Quote Request:

value:

quote:

dropoff_address: 1 Test Street

dropoff_suburb: Sydney

dropoff_postcode: "2000"

dropoff_state: NSW

dropoff_country_code: AU

parcel_attributes:

- qty: 1

weight: 1

product_attributes:

- title: Socks

tariff_code: 6115.99.00

quantity: 5

price: 10

product_currency: SGD

Example Quote Request With features:

```
value:
  quote:
    dropoff_postcode: "2000"
    dropoff_state: NSW
    dropoff_suburb: Sydney
    parcel_attributes:
      - qty: 1
        weight: 1
    features:
      - calculate_ddp
  required: true
tags:
- Quote
responses:
  "200":
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/QuoteResponse'
          title: Live Quote Response
          properties:
            count:
              description: Number of quotes retrieved.
              type: integer
            response:
              type: object
              properties:
                success:
                  description: XXXX
                  type: boolean
                  example: true
                courier_type:
                  description: Courier name.
                  type: string
                  example: CouriersPlease
                service_level:
                  description: The courier service level
                    offered.
                  type: string
                  example: standard
            quotes:
              type: object
              properties:
                price:
                  description: The courier price for the
                    delivery.
                  type: number
                  example: 7.79
                estimated_transit_time:
                  description: The estimated time for the
                    delivery. This is
                    provided for standard and express
                    deliveries.
```

```
    type: string
    example: 3 business days
  delivery_date:
    type: string
  delivery_window:
    type: string
  delivery_window_desc:
    type: string
  estimated_delivery_time:
    description: The estimated delivery time
    type: string
    example: 2023-10-03T16:21:57.000+11:00
```

example:

```
  count: 4
  response:
    - success: true
      courier_type: CouriersPlease
      service_level: standard
      quotes:
        - price: 7.79
          estimated_transit_time: 1 business day
    - success: true
      courier_type: Eparcel
      service_level: standard
      quotes:
        - price: 17.14
          estimated_transit_time: 2 business days
    - success: true
      courier_type: Bonds
      service_level: priority
      quotes:
        - delivery_date: 2014-08-08T00:00:00.000Z
          delivery_window: 13:29-16:29
          delivery_window_desc: 1PM-4AM
          price: 24.24
        - delivery_date: 2014-08-08T00:00:00.000Z
          delivery_window: 16:00-19:00
          delivery_window_desc: 4PM-7PM
          price: 24.24
    - success: true
      courier_type: DoorDashOndemand
      service_level: on_demand
      quotes:
        - price: 10.99
          estimated_delivery_time:
```

2023-10-03T16:21:57.000+11:00

examples:

Successfully Generated Quote:

```
  value:
    response:
      - courier_type: CouriersPlease
        quotes:
          - price: 20.4
            estimated_transit_time: 3 business days
```

```
    service_level: standard
    success: true
  - courier_type: eParcelExpress
    quotes:
      - price: 30.4
        estimated_transit_time: 1 business day
        service_level: express
        success: true
  - courier_type: Priority
    quotes:
      - delivery_date: 2014-09-18T00:00:00.000Z
        delivery_window: 07:00-10:00
        delivery_window_desc: 7AM-10AM
        price: 24.24
        courier_type: Bonds
      - delivery_date: 2014-09-18T00:00:00.000Z
        delivery_window: 10:00-13:00
        delivery_window_desc: 10AM-1PM
        price: 24.24
        courier_type: AlliedExpressP2P
    service_level: priority
    success: true
  - courier_type: DoorDashOndemand
    quotes:
      - price: 10.99
        estimated_delivery_time:
2023-10-03T16:21:57.000+11:00
    service_level: on_demand
    success: true
  count: 4
```

Successfully Generated International Quote:
value:

```
  response:
  - courier_type: SmartRoutingInternational
    quotes:
      - price: 20.4
        estimated_transit_time: 3 business days
        customs_duty: 3.21
        import_tax: 1.22
        import_fee: 0.8
        incoterm: DDP
    service_level: standard
    success: true
  - courier_type: JanioInternational
    quotes:
      - price: 18.3
        estimated_transit_time: 5 business days
        incoterm: DDU
    service_level: standard
    success: true
  - courier_type: AramexInternational
    quotes:
      - price: 30.4
        estimated_transit_time: 4 business day
```

```

        service_level: express
        success: true
    - courier_type: CouriersPlease
      error: cannot transport to country ID.
      success: false
    count: 4
  description: Successful quote response
"400":
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/QuoteErrorResponse'
      text/html: {}
  description: |-
    Returned when a quote request cannot be generated at
all.

```

Note that if a quote request to a carrier `_can_` be generated, but one or more carriers cannot respond to the quote, then the quote request succeeds and this response is not given. The status of each individual quote would then be determined from the response body. This response is only returned when the quote request cannot be processed.

```

    If returned as text/html, will be blank.
"403":
  $ref: '#/components/responses/
InvalidMerchantAccountErrorResponse'
"500":
  $ref: '#/components/responses/UnknownErrorResponse'
security:
- AuthToken: []
operationId: getQuote
summary: Retrieve Quote
description: |-
  Submits a request for Quotes from different couriers on
Shippit.

```

At minimum, a Quote requires a delivery location and information on the parcels being delivered. However, different couriers and delivery methods can require additional fields to satisfy their requirements. There is a maximum of 1000 parcels per quote request.

Listing all quotes vs listing a single quote
By default, Shippit will return a quote from the fastest / cheapest courier. However, for the purposes of showing different options, such as in a courier selection page, you may want to have Shippit display all quotes. To do this, pass the ``return_all_quotes`` field as true.

Note that the quotes returned are filtered based on the requirements given by different couriers. Depending on the info submitted in the order and the carrier settings in the Shippit admin

page, you may not see quotes from all configured carriers.

/merchant:

summary: The merchant settings API allows you to query and update the current

settings for your merchant account.

get:

tags:

– Merchant

responses:

"200":

content:

application/json:

schema:

\$ref: '#/components/schemas/MerchantResponse'

examples:

Example Merchant Response:

value:

response:

store_name: New Store Name

company_name: Harold Pty Ltd

contact_name: Robert

contact_phone: "0400000000"

shipping_cart_method_name: ""

preparation_time: 60

website_url: www.haroldsharpoons.com.au

address_1: 110 Rex Road

suburb: Sydney

state: NSW

postcode: "2000"

country_code: AU

description: Returns an object representing your merchant account settings.

"403":

\$ref: '#/components/responses/

InvalidMerchantAccountErrorResponse'

security:

– AuthToken: []

operationId: getMerchant

summary: Get Merchant settings

patch:

requestBody:

content:

application/json:

schema:

\$ref: '#/components/schemas/MerchantRequest'

examples:

Example Merchant settings change:

value:

merchant:

store_name: New Store Name

required: true

tags:

– Merchant

responses:

```

"200":
  content:
    application/json:
      schema:
        $ref: '#/components/schemas/MerchantResponse'
      examples:
        Updated merchant account:
          value:
            response:
              store_name: Harolds Harpoons
              company_name: Harold Pty Ltd
              contact_name: Robert
              contact_phone: "0400000000"
              shipping_cart_method_name: ""
              preparation_time: 60
              website_url: www.haroldsharpoons.com.au
              address_1: 110 Rex Road
              suburb: Sydney
              state: NSW
              postcode: "2000"
              country_code: AU
            description: The updated merchant account.
"400":
  $ref: '#/components/responses/BadRequestErrorResponse'
"403":
  $ref: '#/components/responses/
InvalidMerchantAccountErrorResponse'
"500":
  $ref: '#/components/responses/UnknownErrorResponse'
operationId: updateMerchant
summary: Update Merchant settings
description: "Given an object with the desired settings to
update, will update\
\ the Merchant settings on Shippit"
/webhook:
  post:
    requestBody:
      description: Updated status info about the Order sent by the
webhook.
    content:
      application/json:
        schema:
          $ref: '#/components/schemas/WebhookRequest'
        examples:
          Example Tracking Webhook Status:
            value:
              tracking_number: PPYvZCTod5bkD
              expected_delivery_date: 2016-04-26T20:46:01+10:00
              tracking_url: http://www.shippit.com/track/
              current_state: completed
              retailer_order_number: WEYFEW232
              retailer_reference: '#OD12345'
              courier_name: Couriers Please

```



```

courier_job_id: CPAVZUZ0001749
delivery_address: 123 Fake Drive
delivery_suburb: Sydney
delivery_postcode: "2000"
delivery_state: NSW
delivery_country_code: AU
delivery_latitude: -33.77
delivery_longitude: 150.91
merchant_url: myshopify.storename.com
status_history:
- status: completed
  time: 2016-04-26T19:36:32.000Z
- status: with_driver
  time: 2016-04-26T18:36:32.000Z
- status: in_transit
  time: 2016-04-26T17:36:32.000Z
- status: ready_for_pickup
  time: 2016-04-26T16:36:11.000Z
- status: despatch_in_progress
  time: 2016-04-26T16:27:35.000Z
- status: order_placed
  time: 2016-04-26T16:25:04.000Z
products:
- quantity: 1
  sku: EWPE123123
  title: Super awesome red tshirt
  product_line_id: "1234"
- quantity: 7
  sku: EWPE123123
  title: Super awesome blue tshirt
  product_line_id: "1235"
required: true
tags:
- Track
responses:
  "200":
    description: The response (success) the webhook expects
from your application
    endpoint.
    operationId: trackOrderHook
    summary: Webhook
    description: "Shippit will send status updates on orders to a
URL you have configured\
  \ in the Shippit admin page.\n\nThis can be configured at
`Settings` -> `Integrations`\
  \ -> `Webhook URL` at the Shippit admin.\n\nIf configured,
Shippit will send\
  \ POST requests to the URL set here with updates on an
Order's status as it\
  \ progresses.\n\n---\n\n### Best practices for using
webhooks\n\n**Event types**\n\
  \nYour webhook implementation should be configured to action
only the types\
  \ of events required by your integration. Actioning for

```

extra events (or all
 \ events) is not recommended as we may add new events at
anytime.\n\n**Handle\
 \ duplicate events**\n\nWebhook endpoints might occasionally
receive the same\
 \ event more than once. We advise you to guard against
duplicated event receipts\
 \ by making your event processing idempotent. One way of
doing this is logging\
 \ the events you've processed, and then not processing
already-logged events.\n\
 \n**Order of events**\n\nShippit does not guarantee delivery
of events in\
 \ the order in which they are generated. \n\n**IP allow
list**\n\nThe webhook\
 \ will originate from one of the following IP addresses:\n-
13.210.185.17\n-
 - 54.153.239.182 [New]\n- 54.66.78.172 [New]\n-
13.237.179.89\n- 13.239.82.79\n-
 - 54.66.178.150\n\nIt is recommended to only allow requests
from these IP\
 \ addresses as part of your firewall configuration."

/book:

description: ""

post:

requestBody:

description: An array of Order tracking numbers to book

content:

application/json:

schema:

\$ref: '#/components/schemas/BookRequest'

examples:

Example Book Request:

value:

orders:

- PPKNpNhR0A5Ud

- PP2ZlACi2vtxNY

- PPKNpNhR0A5Ud

required: true

tags:

- Book

responses:

"200":

content:

application/json:

schema:

\$ref: '#/components/schemas/BookResponse'

examples:

Example Successful Book Response:

value:

response:

- courier: eParcel Express

manifest: ABC0000214

manifest_pdf: https://shippit.com/

```

manifest20160721-96368-1tpowvr.pdf
    order_count: 2
    success: true
    - courier: Fastway
      manifest: FW20160721215
      manifest_pdf: https://shippit.com/
manifest20160721-96368-1tpowvr.pdf
    order_count: 2
    success: true
    - courier: Couriers Please
      manifest: JJJ20160721216
      manifest_pdf: https://shippit.com/
manifest20160721-96368-1tpowvr.pdf
    order_count: 4
    success: true
    - courier: eParcel
      manifest: ABC0000217
      manifest_pdf: https://shippit.com/
manifest20160721-96368-1tpowvr.pdf
    order_count: 2
    success: true
count: 4
Error Book Response - could not be found:
value:
  response:
    - error: orders_not_available_to_book
      error_description: The requested orders could
not be found.
      Please remove and try again.
    orders:
    - invalid-tracking
    success: false
count: 1
Error Manifest failed:
value:
  response:
    - courier: Fastway
      error: Manifest creation failed
      manifest: FW20160721224
      orders:
      - ppd5q24mtuett
      success: false
    - courier: Couriers Please
      error: Manifest creation failed
      manifest: JJJ20160721225
      orders:
      - pp0myuwhnajn4
      success: false
count: 2
description: |-
  Returns the status of the bookings at each courier.

```

This API will validate that all of the requested orders are accurate and ready to be booked, if all of the orders sent in

cannot be booked, the response will detail which orders cannot be found. If this response is received, none of the order numbers sent in will have been booked. If a manifest fails to be generated, the response will include an array of orders on each manifest which were not booked.

```
"400":
  $ref: '#/components/responses/BadRequestErrorResponse'
"403":
  $ref: '#/components/responses/
InvalidMerchantAccountErrorResponse'
"500":
  $ref: '#/components/responses/UnknownErrorResponse'
security:
- AuthToken: []
operationId: bookOrder
summary: Book Order
description: |-
  Given an array of Orders, this will initiate a booking with
  their respective couriers.
```

```
For Orders to be booked, they must first be labelled.
/book/{manifest_id}/document:
  summary: Book Document
  description: ""
  get:
    tags:
    - Book
    parameters:
    - name: manifest_id
      description: The id of the Manifest returned from book call.
      schema:
        type: string
      in: path
      required: true
    responses:
      "200":
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/BookDocumentSuccess'
            description: Success response
      "422":
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/BookingInProgress'
            description: Book in progress
      "424":
        content:
          application/json:
            examples:
              Booking Failed with Courier:
                value:
                  error: booking_failed_with_courier
```

```

        error_description: "The requested manifest
failed to book with\
    \ the courier, please contact support"
    description: Courier Booking Failed
    operationId: getBookDocument
    summary: Book Document
    description: Get Manifest Document from the courier
/returns/order:
  post:
    requestBody:
      description: ""
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ReturnOrderRequest'
          required: true
    tags:
      - Custom Returns Integration
    responses:
      "200":
        description: "The response [success] the webhook expects
from your application\
        \ endpoint."
        operationId: returnsOrder
        summary: Return Order webhook
        description: |-
          A Return Order is created when a Returns Request is approved
and the shipment has been confirmed.

```

Shippit will send a notification when a the Return Order is created to a URL you have configured in your Returns Portal settings in Shippit Admin.

```

/returns/request:
  post:
    requestBody:
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ReturnRequestRequest'
    tags:
      - Custom Returns Integration
    responses:
      "200":
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ReturnRequestResponse'
            description: The return request order was found and has
data available for
            a response.
      "404":
        content:
          application/json: {}
        description: The completed order details could not be

```

found based on the
provided `order_reference` and `email_address`
"502":
description: The return request order details could not be
fetched from
the connected Order Management Platform due to a request
failure when
communicating with the Order Management Platform.
operationId: returnsRequest
summary: Return Request
description: |-
A Return Request provides a mechanism to query the
merchant's platform for details about an order and it's contents.

A Request Request will trigger Shippit to query the
Merchant's platform to retrieve all applicable details required to
be provided as part of the Return Request response.

Upon a successful Return Request response, the return
details are presented to the customer in the Returns Portal,
allowing them to make a selection of the goods to be returned for
the order.

components:
schemas:
QuoteResponse:
title: Root Type for QuoteResponse
description: "Represents a response to an QuoteRequest, which
shows either the\
\ generated Quote object or an error code with a description
of the error."
type: object
allOf:
- type: object
properties:
response:
description: The Quote returned by the carrier
type: array
items:
\$ref: '#/components/schemas/QuoteResult'
count:
description: Number of quote results returned.
type: number
- \$ref: '#/components/schemas/DataResponse'
example: |-
{
"response": [
{
"courier_type": "CouriersPlease",
"error": null,
"quotes": [
{
"price": 20.4,
"estimated_transit_time": "3 business
days"

```

    },
    ],
    "service_level": "standard",
    "success": true
  },
  {
    "courier_type": "eParcelExpress",
    "error": null,
    "quotes": [
      {
        "price": 30.4,
        "estimated_transit_time": "1 business
day"
      }
    ],
    "service_level": "express",
    "success": true
  },
  {
    "courier_type": "Priority",
    "error": null,
    "quotes": [
      {
        "delivery_date": "2014-09-18",
        "delivery_window": "07:00-10:00",
        "delivery_window_desc": "7AM-10AM",
        "price": 24.24,
        "courier_type": "Bonds"
      },
      {
        "delivery_date": "2014-09-18",
        "delivery_window": "10:00-13:00",
        "delivery_window_desc": "10AM-1PM",
        "price": "24.2",
        "courier_type": "AlliedExpressP2P"
      }
    ],
    "service_level": "priority",
    "success": true
  }
],
"count": 3
}
QuoteRequest:
  title: Root Type for QuoteRequest
  description: Represents a request for carrier Quotes from
Shippit.
  required:
  - quote
  type: object
  properties:
    quote:
      $ref: '#/components/schemas/QuoteRequestDetails'
      description: A specification of the queries to include in

```

the quote.

example:

quote:

dropoff_postcode: some text

dropoff_state: some text

dropoff_suburb: some text

parcel_attributes:

qty: 76

weight: 54

MerchantRequest:

title: Root Type for MerchantRequest

description: Represents a request to update a merchant

required:

- merchant

type: object

properties:

merchant:

\$ref: '#/components/schemas/Merchant'

description: Fields to update on a merchant account.

properties:

store_name:

type: string

example: |-

```
{
  "merchant": {
    "store_name": "New Store Name"
  }
}
```

BookRequest:

title: Root Type for BookRequest

description: "The book API accepts an array of orders to book,
this will initiate\

\ a booking with the respective carriers of the orders sent
in. "

required:

- orders

type: object

properties:

orders:

description: "an array of Shippit tracking numbers,
corresponding to the\
\ orders which you would like to book."

type: array

items:

type: string

example: |

```
{
  "orders": [
    "PPKNpNhR0A5Ud",
    "PP2ZlACi2vtxNY",
    "PPKNpNhR0A5Ud"
  ]
}
```

Book:

description: "The book API accepts an array of orders to book,
this will initiate\
 \ a booking with the respective carriers of the orders sent
in. "

type: object
OrderRequest:
 title: Root Type for OrderRequest
 description: Represents a request to submit an order to
shippit.

required:
- order
type: object
properties:
 order:
 \$ref: '#/components/schemas/OrderRequestOrder'
 description: The Order to create in shippit.

example:
 order:
 courier_type: standard
 delivery_address: 1 Union Street
 delivery_postcode: "2009"
 delivery_state: NSW
 delivery_suburb: Pyrmont
 authority_to_leave: "Yes"
 parcel_attributes:
 - qty: 1
 weight: 2.1
 user_attributes:
 email: test@shippit.com
 first_name: John
 last_name: Smith

Merchant:
 title: Root Type for Merchant
 description: Represents a merchant account
 type: object
 properties:
 store_name:
 type: string
 company_name:
 type: string
 contact_name:
 type: string
 contact_phone:
 type: string
 shipping_cart_method_name:
 type: string
 preparation_time:
 format: int32
 type: integer
 website_url:
 type: string
 address_1:
 type: string
 suburb:

```

    type: string
  state:
    type: string
  postcode:
    type: string
  example:
    store_name: New Store Name
    company_name: Harold Pty Ltd
    contact_name: Robert
    contact_phone: "0400000000"
    shipping_cart_method_name: ""
    preparation_time: 60
    website_url: www.haroldsharpoons.com.au
    address_1: 110 Rex Road
    suburb: Sydney
    state: NSW
    postcode: "2000"
  LabelDocument:
    title: Root Type for LabelDocument
    description: "A Label or document in the Labels API,
representing a printable\
\ document available on a public URL."
    type: object
    properties:
      url:
        description: |-
          URL to a printable PDF document.
          This is a [pre-signed URL](https://docs.aws.amazon.com/
AmazonS3/latest/dev/PresignedUrlUploadObject.html), generated by the
Shippit platform to provide access to a secured document. The pre-
signed URL remains valid for 7 days. If you need to access this
document after expiration, please make another request for a new URL
to be issued.
        type: string
      page_size:
        type: string
      file_type:
        description: File type of the label
        type: string
      encoded_label:
        description: "If you have ZPL for your account, printable
ZPL data for the\
\ document."
        type: string
      example: "{\r\n  \"url\": \"https://shippit-web-
production.s3.amazonaws.com/uploads/order/customs_invoice/361690/
commercial_invoice_3073489348620190218-6221-mxrhx4.pdf\"\r\n
, \r\n  \"page_size\": \"a3\", \r\n  \"file_type\":
\"pdf\"\r\n}"
    BookResponse:
      description: "Represents a response to the Book API, which
returns the status\
\ of booking the order at each courier."
      type: object

```

```

properties:
  response:
    description: Array of booking statuses at each courier
    type: array
    items:
      $ref: '#/components/schemas/BookingStatus'
  error:
    description: |-
      Error code returned by Shippit

```

For the book response, there can be errors at the top-level request or for individual orders booked.

If the error occurs at the top-level request (order format, authorization), then this error code applies. Otherwise, the response will be successful and there will be individual errors applied in the body of each booked item.

```

    type: string
    error_description:
      description: Human-readable description of the top-level
error encountered.
    type: string
    count:
      description: Number of items in the response
      type: integer

```

```

example:
  response:
    - courier: eParcel Express
      manifest: ABC0000214
      manifest_pdf: https://shippit.com/
manifest20160721-96368-1tpowvr.pdf
      order_count: 2
      success: true
    - courier: Fastway
      manifest: FW20160721215
      manifest_pdf: https://shippit.com/
manifest20160721-96368-1tpowvr.pdf
      order_count: 2
      success: true
    - courier: Couriers Please
      manifest: JJJ20160721216
      manifest_pdf: https://shippit.com/
manifest20160721-96368-1tpowvr.pdf
      order_count: 4
      success: true
    - courier: eParcel
      manifest: ABC0000217
      manifest_pdf: https://shippit.com/
manifest20160721-96368-1tpowvr.pdf
      order_count: 2
      success: true

```

```

count: 4
BookingStatus:
  title: Root Type for BookingStatus

```

```

    description: Describes the status of booking an order at a
courier
    type: object
    properties:
      courier:
        description: The courier where the booking was made
        type: string
      manifest:
        description: The Manifest ID of the booking
        type: string
      manifest_pdf:
        description: "A URL to a printable copy of the shipping
manifest. This is\
        \ a [pre-signed URL](https://docs.aws.amazon.com/
AmazonS3/latest/dev/PresignedUrlUploadObject.html),\
        \ generated by the Shippit platform to provide access to
a secured document.\
        \ The pre-signed URL remains valid for 7 days. If you
need to access this\
        \ document after expiration, please make another request
for a new URL\
        \ to be issued."
        type: string
      order_count:
        format: int32
        description: How many orders were included in this
manifest.
        type: integer
      success:
        description: Whether or not the booking succeeded
        type: boolean
      orders:
        description: An array of order tracking numbers included
in this status.
        Used to indicate which orders failed to manifest in the
case where `success`
        = `false`.
        type: array
        items:
          type: string
      error:
        description: Error code that applies to the individual
bookings / orders
        referenced.
        type: string
      error_description:
        description: Human-readable error message applied to the
individual bookings
        / orders referenced.
        type: string
    example: |-
    {
      "courier": "eParcel Express",
      "manifest": "ABC0000214",

```

```

        "manifest_pdf": "https://shippit.com/
manifest20160721-96368-1tpowvr.pdf",
        "order_count": 2,
        "success": true
    }
    QuoteRequestDetails:
        title: Root Type for quote
        description: A specification of the queries to include in the
quote.
        required:
            - parcel_attributes
            - dropoff_postcode
            - dropoff_suburb
        type: object
        properties:
            dropoff_postcode:
                description: destination postcode for the quote
                type: string
            dropoff_state:
                description: destination state for the quote. **Required**
when `dropoff_postcode`
                is unassigned or set to `AU`
                type: string
            dropoff_suburb:
                description: The dropoff suburb
                type: string
            parcel_attributes:
                description: |-
                    An array of parcel specifications to be included in the
quote.

```

Each item can be used to specify the qty, dimensions, and other information about the products to be shipped, and the parcel used to ship it.

```

        At minimum, only the `qty` and `weight` fields are
required. There is a maximum of 1000 parcels per quote request.
        type: array
        items:
            $ref: '#/components/schemas/Parcel'
        product_attributes:
            description: An array of product details to be included in
the quote
            type: array
            items:
                type: object
                properties:
                    title:
                        description: Title or description of the product
                        type: string
                    tariff_code:
                        description: Tariff code for international orders
declarations
                        type: string

```

```

        quantity:
            description: The number of products
            type: integer
        price:
            description: Unit price of each product line item.
Used to calculate
            duties and taxes
            type: number
        order_date:
            format: date
            description: "delivery date for the order, required for
`priority` orders"
            type: string
        return_all_quotes:
            description: "If set to `true`, the system returns all
quotes instead of\
        \ cheapest/fastest quote. Defaults to false."
            type: boolean
        dutiable_amount:
            description: |-
                ```conditional``` Declared value for customs when
shipping international.

 Mandatory for International orders.
 type: number
 product_currency:
 description: Three letter ISO 4217 currency code that
applies to the products
 which will be used for customs.
 type: string
 dropoff_country_code:
 description: "The destination country code for the quote
in `ISO 3166 Alpha-2`\
 \ format. If unassigned, defaults to the country code of
the merchant's\
 \ primary location, or otherwise 'AU'"
 type: string
 dropoff_address:
 description: The dropoff address for the quote. Mandatory
for `OnDemand`
orders.
 type: string
 service_levels:
 description: |-
 An array of service leves to return quotes for. Valid
values include:
 - `standard`
 - `express`
 - `priority`
 - `on_demand`
 type: array
 items:
 type: string
 features:

```

```

 description: |-
 Some carriers provide add-on services (features) which
 can be indicated when requesting a quote. These services are
 identified by a unique string and will depend on the carrier. To
 request these services, add a string to these field.
 Services supported at the moment:
 - `white_glove` (`Neway`, `DesignerTransport`):
 A delivery service that offers more personalised
 and tailored services for shipping high value, large or fragile
 goods.
 - `calculate_ddp` (`SmartRoutingInternational`):
 When requested, the returned quote prices are
 inclusive of import taxes, duties and fees.
 type: array
 items:
 type: string
 authority_to_leave:
 description: "If set to \"Yes\", the system returns all
 quotes where Authority\
 \ to Leave is enabled."
 type: string
 example: |-
 {
 "dropoff_postcode": "2000",
 "dropoff_state": "NSW",
 "dropoff_suburb": "Sydney",
 "parcel_attributes": [
 {
 "qty": 1,
 "weight": 1
 }
]
 }
 QuoteResult:
 title: Root Type for Quote
 description: Represents a Quote returned by a courier
 type: object
 properties:
 courier_type:
 description: The name of the courier or service level
 giving the quote.
 type: string
 error:
 description: "Error response, if any, received from the
 carrier(s) during\
 \ quoting."
 type: string
 quotes:
 description: List of quotes returned for the service level
 or carrier.
 type: array
 items:
 $ref: '#/components/schemas/Quote'
 service_level:

```

description: The service level associated with this quote or group of quotes.

type: string

success:

description: Whether or not this service level was able to obtain a valid

quote from the carrier(s).

type: boolean

```
example: "{\r\n \"courier_type\": \"CouriersPlease\", \r\n \"error\": null, \r\n \r\n \"quotes\": [\r\n {\r\n \r\n \"price\": 20.4, \r\n \r\n \"estimated_transit_time\": \"3 business days\" \r\n }, \r\n \r\n], \r\n \"service_level\": \"standard\", \r\n \"success\": true \r\n }"
```

Quote:

title: Root Type for Quote

description: A quote returned by a carrier.

type: object

properties:

delivery\_date:

format: date

description: Estimated date of delivery for the order

type: string

delivery\_window:

description: "For priority orders, machine-readable estimated range of times\

\ when the delivery could take place."

type: string

delivery\_window\_desc:

description: "For priority orders, human-readable estimated range of times\

\ when the delivery could take place."

type: string

price:

description: Amount quoted by the carrier.

type: number

courier\_type:

description: "`optional` If returned inside a quote response for a service\

\ level (e.g. `standard`), the carrier returning the

quote. May be omitted\

\ if a specific carrier was requested."

type: string

customs\_duty:

description: "Customs duty component for international orders, if applicable.\

\ This amount is `included` in the `price` if the

carrier is configured\

\ to return a DDP quote."

type: number

import\_tax:



```

 description: "Import tax component for international
orders, if applicable.\
 \ This amount is `included` in the `price` if the
carrier is configured\
 \ to return a DDP quote."
 type: number
 import_fee:
 description: "Import fees component for international
orders, if applicable.\
 \ This amount is `included` in the `price` if the
carrier is configured\
 \ to return a DDP quote."
 type: number
 incoterm:
 description: A universal term of trade that defines the
responsibility and
 obligations with the transport of goods. Most commonly
used to indicate
 the responsible party for payment of duties and taxes.
 estimated_transit_time:
 description: Estimated amount of time for the quote to
arrive.
 type: string
 example:
 delivery_date: 2014-09-18T00:00:00.000Z
 delivery_window: 07:00-10:00
 delivery_window_desc: 7AM-10AM
 price: 24.24
 courier_type: Bonds
OrderResponseOrder:
 description: Represents an Order that was created in Shippit
 required:
 - tracking_number
 - slug
 - documents
 type: object
 allOf:
 - $ref: '#/components/schemas/Order'
 properties:
 tracking_number:
 description: Unique random ID assigned by Shippit to an
order. This can
 be used as a reference for future API calls or support
tickets.
 type: string
 slug:
 description: ``tracking_number` in lowercase.'
 type: string
 parcel_attributes:
 $ref: '#/components/schemas/OrderResponseOrderParcel'
 description: An array of parcel specifications that were
created with the
 order.
 products:

```

```

 description: |-
 An array of product items specified in the order.

 If product information was not provided, this array will
be blank.
 type: array
 items:
 $ref: '#/components/schemas/Product'
 tracking_url:
 description: The URL of the order's customer tracking
page.
 type: string
 return:
 description: Whether or not the order is a return order.
 type: boolean
 id:
 description: Shippit internal numerical ID
 type: integer
 processing_state:
 description: Internal order status set and progressed by
Shippit upon creation.
 type: string
 invoice_number:
 description: Internal link between an order and a Shippit
invoice - will
 almost always be null on order creation.
 courier_delivery_instructions:
 description: "Delivery instructions as sent to the
courier. In most cases,\
 \ this is the same as the `delivery_instructions` passed
into the request.\
 \ Some couriers, however, have a specific format
expected when sending\
 \ delivery instructions through their API. This will
contain the actual\
 \ delivery instructions that get sent through the
courier API when booked."
 type: string
 courier_name:
 description: |-
 Human-readable name of the courier that has been
allocated to the order, or `null` if the order hasn't been allocated
yet.

```

If you're creating orders by passing in `courier\_type`, the courier will usually not be yet be assigned at order creation and you will get `null`. Courier allocation will be run as a background process depending on the merchant preferences, available couriers, and quotes returned by the couriers, which may take a few seconds to complete after order creation.

If you're passing in a specific courier keyword in `courier\_allocation`, this will reflect the courier that is specified.

Note that the Shippit web user can also update the courier in between API calls. The final courier that has been assigned can be determined from the LABEL call.

- type: string
- price:
  - description: |-  
The price that will be charged for the allocated order.

If you're creating orders by passing in `courier\_type`, the courier will usually not be yet be assigned at order creation and you will get `0`. Courier allocation will be run as a background process depending on the merchant preferences, available couriers, and quotes returned by the couriers, which may take a few seconds to complete after order creation.

Note that the Shippit web user can also update the courier in between API calls. The final courier and quoted price can be determined from the LABEL call.

If the quoted price is still `0` during the LABEL call, something likely went wrong with the courier selection and the order may need to be amended.

- customs\_documents\_require\_printing:
  - description: "Whether or not customs documents, such as declarations, invoices, \ etc for the Order that should be printed and included with the shipment. \ By default this is `true` for International orders."

- type: boolean
- courier\_type:
  - description: The courier allocated to the order. See the ORDER CREATE `courier\_allocation` API for available keywords.

- type: string
- tracking\_histories:
  - description: An array of Tracking History statuses that are are shown to the recipient as part of the order's history on the tracking page.

- type: array
- items:
  - \$ref: '#/components/schemas/OrderResponseTrackingHistory'
- state:
  - description: The state of the order as progressed by the merchant.

- type: string
- user:
  - \$ref: '#/components/schemas/UserAttributes'
  - description: Represents the recipient of the order
- documents:
  - \$ref: '#/components/schemas/OrderResponseOrderDocumentsHash'

```

 description: Documents attached to the label response
OrderRequestOrder:
 description: Represents parameters that can be used to create
an order in Shippit
 required:
 - parcel_attributes
 type: object
 allOf:
 - $ref: '#/components/schemas/Order'
 properties:
 parcel_attributes:
 description: "An array of parcel specifications to be
included in the order.\n\
\nThere is a maximum of 1000 parcels per request.
\n\nEach item can be\
\n used to specify the qty, dimensions, and other
information about the\
\n products being shipped, and the parcel used to ship
it. If the `Allocate\
\n ech item in an order to a separate carton` setting is
enabled in the\
\n Shippit administration page, then each item will be
assigned a separate\
\n parcel according to the dimensions given. Otherwise,
Shippit may combine\
\n multiple items into a single parcel.\n\nIf the
`product_attributes`\
\n field is present, then product information will be
specified in entries\
\n there, separate from the parcel information. In which
case each parcel\
\n entry represents a single parcel. "
 type: array
 items:
 $ref: '#/components/schemas/
OrderRequestOrderParcelAttributes'
 product_attributes:
 description: |-
 an array of product details. If this is present, each
item in ``parcel_attributes`` represents a separate parcel, while
each line in ``product_attributes`` represents a product line
item.

```

```

 There is a maximum of 1000 parcels per request.
 type: array
 items:
 $ref: '#/components/schemas/Product'
 authority_to_leave:
 description: "whether or not the order can be left
unattended at the delivery\
\n address, options are ``Yes`` or ``No``"
 type: string
 cash_on_delivery_amount:
 description: Used by some carriers (e.g. `NinjaVan`) to

```

```

indicate amount
 to be received on delivery for cash on delivery orders.
 type: number
 courier_type:
 description: "The service level for the order, valid
options are `standard`,\
 \ `express`, `priority`, `click_and_collect`. If you use
`courier_type`\
 \ in your request, then Shippit will allocate the
courier for the job\
 \ at that service level. If you want to specify the
courier in the request\
 \ please use `courier_allocation` attribute instead.
\n\nOne of `courier_allocation`\
 \ or `courier_type` is required. If omitted or invalid
option is provided\
 \ in the request, we will allocate an enabled courier
that matches the\
 \ default service level. The default service level is
`standard` and can\
 \ be configured by Shippit on request for a merchant.
The available service\
 \ level options are ``standard``, ``express`` and
``priority``."
 type: string
 receiver_contact_number:
 description: "``conditional`` -- number of the person
receiving the order,\
 \ may be different than the user who purchased the
order. Mandatory for\
 \ International orders."
 type: string
 receiver_name:
 description: name of the person receiving the order if
different than the
 user who purchased the order
 type: string
 retailer_invoice:
 description: "Merchant invoice number - the customer-
facing sales order\
 \ reference of the order. This would be what the
recipient sees on the\
 \ shipping label, packslip, customs declaration,
receipt, etc."
 type: string
 retailer_reference:
 description: "Merchant reference id - if the merchant uses
a separate internal\
 \ reference id for the order, it can be placed here."
 type: string
 courier_allocation:
 description: "For ``courier_allocation``, the list of
valid options are\
 \ outlined below.\n\n- `AlliedExpressOvernight`\n-

```

```

`AlliedExpressSameday`\n\
 - `AramexAuNz`\n- `AramexExpress`\n-
`AramexInternational`\n- `Bonds`\n\
 - `CapitalTransport`\n- `CityLink`\n-
`ClickAndCollect`\n- `CouriersPlease`\n\
 - `CouriersPleaseExpress`\n- `DhlEcommerce`\n-
`DhlExpress`\n- `DhlEcommerceAsia`\n\
 - `DhlExpressInternational`\n- `DirectCouriers`\n-
`DirectFreightExpress`\n\
 - `DoorDash`\n- `DoorDashOndemand`\n- `Eparcel`\n-
`EparcelExpress`\n\
 - `EparcelInternational`\n-
`EparcelInternationalExpress`\n- `EparcelOndemand`\n\
 - `Fastway`\n- `FedExInternationalEconomy`\n-
`FedExInternationalPriority`\n\
 - `FlashExpress`\n- `FourPXStandard`\n- `Gdex`\n-
`HunterExpress`\n- `Janio`\n\
 - `JanioInternational`\n- `Jnt`\n- `KerryExpress`\n-
`KerryStandard`\n\
 - `LyneConnect`\n- `LynePlus`\n- `Neway`\n-
`NewZealandCouriers`\n- `NewZealandPost`\n\
 - `NewZealandPostExpress`\n- `NinjaVanExpress`\n-
`NinjaVanStandard`\n\
 - `PbtCourier`\n- `PbtCourierExpress`\n- `Pgeon`\n-
`Pickupp`\n- `PlainLabel`\n\
 - `PlainLabelInternational`\n- `PosMalaysia`\n -
`Qxpress`\n- `QxpressSameday`\n\
 - `SekoExpress`\n- `SekoStandard`\n- `SingPost`\n-
`SingPostExpress`\n\
 - `Skybox`\n- `SmartRoutingInternational`\n-
`StarTrack`\n- `StarTrackPremium`\n\
 - `Teleport`\n- `TeleportInternational`\n- `Tnt`\n-
`TntOvernightExpress`\n\
 - `Toll`\n- `TollPriority`\n- `UberOndemand`\n-
`YelloOndemand`\n\nOne\
 \ of `courier_allocation` or `courier_type` is required.
\n\nIf omitted\
 \ or invalid option is provided in the request, we will
allocate an enabled\
 \ courier that matches the default service level. The
default service\
 \ level is `standard` and can be configured by Shippit
on request for\
 \ a merchant. The available service level options are
``standard``,\
 \ ``express`` and ``priority``.
\n\n`courier_allocaiton` is required\
 \ when `tracking_only` is set to `true`\n"
type: string
receiver_language_code:
 description: Two-letter ISO 639-1 language code of the
recipient. Used to
 determine the language of tracking notifications and
other communications.

```

```

 Defaults to ``EN``
 type: string
 validate:
 description: "when true, run validations against the
submitted order and\
 \ only save the order if the supplied parameters are
valid. \nValidations\
 \ run\n - destination suburb and postcode checked
against a list of known\
 \ postcodes / suburbs. \n - If a valid combination is
not found, a suggested\
 \ address will be returned."
 type: boolean
 product_currency:
 description: "Three letter ISO 4217 currency code that
applies to the order,\
 \ which will be sent to the carrier and presented on the
customs invoice.\
 \ Defaults to ``AUD``"
 type: string
 suppress_communications:
 description: "if true, all notifications will be
suppressed for this specific\
 \ order"
 type: boolean
 tracking_only:
 description: |-
 Whether the Order being passed is a tracking order
or not.

```

A **tracking order** is an order where the courier was allocated outside of Shippit, but was loaded into Shippit to take advantage of tracking and notifications. If this is present and set to `true`, this order is a tracking order.

```

 If it is set to ``true``, ``courier_allocation``
must be present.
 type: boolean
 tracking_histories:
 description: An array of Tracking History statuses that
are are shown to
 the recipient as part of the order's history on the
tracking page.
 type: array
 items:
 $ref: '#/components/schemas/
OrderRequestOrderTrackingHistory'
 user_attributes:
 $ref: '#/components/schemas/UserAttributes'
 description: Represents the customer details attached to
the order.
 features:
 description: |-
 Some carriers provide add-on services (features) which

```

can be indicated in the order. These services are identified by a unique string and will depend on the carrier. To request these services, add a string to these field.

Services supported at the moment:

- `white\_glove` (`Neway`, `DesignerTransport`):

A delivery service that offers more personalised and tailored services for shipping high value, large or fragile goods.

- `calculate\_ddp` (`SmartRoutingInternational`):

When requested, the returned quote prices are inclusive of import taxes, duties and fees.

type: array

items:

type: string

customer\_shipping\_fee\_paid:

description: |-

Shipping fee paid by the customer.

This is displayed on the commercial invoice for international orders to support accurate duties and tax calculation by customs. This fee is not related to the Shippit quote. When not provided, no shipping fee will be displayed on the commercial invoice.

type: string

description:

description: "Goods description for the order. May appear in labels for\  
\  
international orders. If not provided, defaults to the `type of goods`\  
\  
setting of the merchant account. Description is mandatory for international\  
orders when `validate` is set to `true`."

type: string

duties:

description: "The duty amount for an order. This is displayed on the commercial\  
\  
invoice for international orders to calculate the invoice total. Duties\  
are not returned by Shippit. When not provided, no duties amount will\  
be displayed on the commercial invoice. eg: 12.90"

type: number

tax\_amount:

description: "The tax amount for an order. This is displayed on the commercial\  
\  
invoice for international orders to calculate the invoice total. When\  
not provided, no tax amount will be displayed separately on the commercial\  
\  
invoice. eg: 2.39"

type: number

customs\_clearance\_attributes:

\$ref: '#/components/schemas/OrderRequestOrderCustomsClearanceAttributes'

description: Represents customs clearance parameters for



```

international orders.
 pickup_postcode:
 description: |-
 Pickup postcode for the return order

 * **required** for return orders
 type: string
 pickup_address:
 description: |-
 Pickup address for the return order

 * **required** for return orders
 type: string
 pickup_suburb:
 description: |-
 Pickup suburb for the return order

 * **required** for return orders
 type: string
 pickup_state:
 description: "Pickup state for the return
order\n\nAustralian states valid\
 \ options are: \n``NSW``, ``QLD``, ``ACT``,
``WA``, ``NT``,\
 \ ``SA``, ``VIC``, ``TAS``\n\n* **required** for
return orders"
 type: string
 return:
 description: Whether the order being passed is a return
order or not
 type: boolean
 example:
 user_attributes:
 email: test@shippit.com
 first_name: John
 last_name: Smith
 parcel_attributes:
 - depth: 0.13
 length: 0.1
 location: C123
 price: 12.12
 qty: 1
 sku: EWPE123123
 title: Super awesome red tshirt
 weight: 2.1
 width: 0.11
 product_line_id: "1234"
 origin_country_code: CN
 packed: 1
 - depth: 0.13
 length: 0.1
 location: B22
 price: 1.21
 qty: 7

```

```
 sku: EWPE123123
 title: Super awesome blue tshirt
 weight: 2.1
 width: 0.11
 dangerous_goods_code: ID8000
 dangerous_goods_text: ID8000 Consumer commodities -
Dangerous Goods as per
 attached DGD
 product_line_id: "1235"
 packed: 1
 authority_to_leave: "No"
 courier_type: standard
 delivery_address: 1 Union Street
 delivery_postcode: "2009"
 delivery_state: NSW
 delivery_suburb: Pyrmont
 delivery_instructions: Special instructions for the order go
```

here

```
 receiver_contact_number: "0400000000"
 receiver_name: Adam
 retailer_invoice: '#23201005'
 retailer_reference: '#23201005'
 duties: 100.99
 tax_amount: 82.811
 customs_clearance_attributes:
 tax_id_type: FTZ
 tax_id_country_code: GB
 tax_id_number: '*****'
 incoterm: DAP
 export_reason: Sample
 recipient_id: '*****'
 receiver_freight_charge_amount: 7.99
```

TrackingHistory:

```
 description: |-
 An event that tracks a change in `status` of a parcel's
journey to fulfillment, with a `timestamp` of when the event
happened.
```

This can be either be provided by the merchant as part of a  
\*\*tracking order\*\* request, or returned by a \*\*track\*\* request.

```
 required:
 - status
 - timestamp
 type: object
 properties:
 status:
 description: The status of the order
 type: string
 timestamp:
 description: The date / time when the status change
occurred.
 type: string
 example:
 status: order_placed
```

```

 date: 2020-08-07T00:00:00.000Z
 timestamp: 09:00:00
TrackingResponseTrackingHistory:
 description: |-
 An event that tracks a change in `status` of a parcel's
 journey to fulfillment, with a `timestamp` of when the event
 happened.

 This can be either be provided by the merchant as part of a
 tracking order request, or returned by a **track** request.
 required:
 - date
 - status_owner
 - timestamp
 type: object
 allOf:
 - $ref: '#/components/schemas/TrackingHistory'
 properties:
 date:
 format: date
 description: The `ISO-8601` date as `YYYY-MM-DD` when the
event happened.
 type: string
 status_owner:
 description: The party responsible for the status change.
 type: string
 timestamp:
 description: The time in `HH:MM:SS` that the tracking
event happened.
 type: string
 example:
 status: despatch_in_progress
 date: 2020-08-25T00:00:00.000Z
 timestamp: 41285
 status_owner: Example Store Name
OrderRequestOrderTrackingHistory:
 title: Order Request TrackingHistory
 description: |-
 An event that tracks a change in `status` of a parcel's
 journey to fulfillment, with a `timestamp` of when the event
 happened.

 This can be either be provided by the merchant as part of a
 tracking order request, or returned by a **track** request.
 type: object
 allOf:
 - $ref: '#/components/schemas/TrackingHistory'
 properties:
 timestamp:
 format: date-time
 description: |-
 expected format: `ISO-8601` datetime, such as `YYYY-MM-
 DDThh:mm` or `YYYY-MM-DDThh:mm:ss+TZ`.

```

The date-time when the status change occurred.  
type: string  
example:  
status: With Driver  
timestamp: 2015-04-03T18:20:30.450Z  
Parcel:  
description: |-  
Represents a parcel attached to an order.

## parcels vs products  
How parcels relate to products is controlled by whether or not a `**`product_attributes`**` key is also defined. If a `**`product_attributes`**` is not defined, then the number of cartons is determined by the ``Settings -> Pick & Pack -> Allocate each item in an order to a separate carton`` option in Shippit, and the `**`qty`**` field will now specify either the number of parcels or products.

If a `**`product_attributes`**` key is specified in the request, then the products can be specified separately from the parcels. In which case, the `**`qty`**` field should not be present: each parcel attributes object represents a separate parcel, and the number of products is specified in the `**`product_attributes`**` key. There is a maximum of 1000 parcels per request.

type: object  
properties:  
weight:  
description: Weight of the parcel in kilos.  
type: number  
depth:  
description: Depth of the parcel in metres  
type: number  
length:  
description: Length of the parcel in metres  
type: number  
width:  
description: Width of the parcel in metres.  
type: number  
label\_number:  
description: |-  
Parcel number in the carrier system.

This is used when the order is a tracking order, to match the parcel number in a carrier against the parcel recorded in shippit.

type: string  
qty:  
description: |-  
``conditional`` -- The number of products specified by the entry.

There is a maximum of 1000 parcels per quote request.

Should not be present if a ``product_attributes`` key is

present in the request.

If 'Allocate each item in an order to a separate carton' is enabled in Shippit, a separate parcel is generated for each product listed.

```
 type: number
 example:
 qty: 1
 length: 0.1
 width: 0.2
 depth: 0.2
 weight: 1
 dangerous_goods_code: ID8000
 dangerous_goods_text: ID8000 Consumer commodities -
Dangerous Goods as per
 attached DGD
 OrderResponseTrackingHistory:
 description: |-
 For 'tracking-only' orders, an object representing the
tracking events provided in order creation.
```

The object returned is in the following format:

```
 '{ "status_name": "ISO-8601-TIMESTAMP" }'`.
 type: object
 example:
 despatch_in_progress: 2020-08-06T23:05:00.000Z
 OrderRequestOrderParcelAttributes:
 title: Root Type for ParcelAttributes
 description: |-
 Represents a parcel attached to an order.
```

#### ## parcels vs products

How parcels relate to products is controlled by whether or not a `**`product_attributes`**` key is also defined. If a `**`product_attributes`**` is not defined, then the number of cartons is determined by the 'Settings -> Pick & Pack -> Allocate each item in an order to a separate carton' option in Shippit, and the `**`qty`**` field will now specify either the number of parcels or products.

If a `**`product_attributes`**` key is specified in the request, then the products can be specified separately from the parcels. In which case, the `**`qty`**` field should not be present: each parcel attributes object represents a separate parcel, and the number of products is specified in the `**`product_attributes`**` key. There is a maximum of 1000 parcels per request.

```
 type: object
 allOf:
 - $ref: '#/components/schemas/Parcel'
 properties:
 title:
 description: '``conditional`` -- Title or description of
the product.
```

Mandatory for international orders with specific couriers.'

type: string

packed:

description: |-

The number of products packed; used in partial orders, or otherwise when the number of products in the parcel is less than the total in the order. If not present, set to product `quantity`.

If this is present, at least 1 item should have packed greater than `0`.

For international orders, all items should have packed greater than `0`.

type: integer

sku:

description: "``conditional`` -- Stock Keeping Unit (SKU) code of the\

\ product, for stock keeping purposes. Mandatory on international orders\ \ with specific couriers."

type: string

product\_line\_id:

description: "Product line associated with product. Like SKU, this is also\

\ for stock keeping purposes, and can be used when SKU would not be enough\

\ (e.g. an order containing multiple lines with the same SKU)"

type: string

origin\_country\_code:

description: Two-letter country code (ISO 3166-1 Alpha-2) for the product's

origin country. This is used in customs tracking for International shipments.

Defaults to the home country of the Merchant.

type: string

location:

description: Location of the product

type: string

dangerous\_goods\_code:

description: |-

This is the DG code required when making dangerous goods declarations.

Before you can declare orders as having dangerous goods, you must first request for dangerous goods to be enabled for your account. Contact Shippit Support or your account manager to have this enabled.

DG surcharges are often charged on the weight of the order, so it is recommended to split your order into multiple orders if your order contains either:

– both dangerous and non-dangerous goods, or

- more than one dangerous goods code
- type: string
- dangerous\_goods\_text:
  - description: Additional information related to dangerous goods being shipped
  - type: string
- dangerous\_goods\_class:
  - description: This is the DG class when making dangerous goods declarations.
  - type: string
- price:
  - description: Unit price of each product line item. Used in the packing slip and customs declarations.
  - type: number
- example:
  - qty: 1
  - length: 0.1
  - width: 0.2
  - depth: 0.2
  - weight: 1
  - dangerous\_goods\_code: ID8000
  - dangerous\_goods\_text: ID8000 Consumer commodities - Dangerous Goods as per attached DGD
  - dangerous\_goods\_class: "9"
- Product:
  - title: Product
  - description: Represents a single product line item in an order.
  - type: object
  - properties:
    - title:
    - description: ````conditional`` -- Title or description of the product.
- Mandatory for international orders with specific couriers.'
  - type: string
- packed:
  - description: |-
    - The number of products packed; used in partial orders, or otherwise when the number of products in the parcel is less than the total in the order. If not present, set to product `quantity`.
  - If this is present, at least 1 item should have packed greater than `0`.
  - For international orders, all items should have packed greater than `0`.
  - type: integer
- sku:
  - description: "````conditional`` -- Stock Keeping Unit (SKU) code of the\
    - \ product, for stock keeping purposes. Mandatory on

international orders\  
 \ with specific couriers."  
 type: string  
 product\_line\_id:  
 description: "Product line associated with product. Like  
 SKU, this is also\  
 \ for stock keeping purposes, and can be used when SKU  
 would not be enough\  
 \ (e.g. an order containing multiple lines with the same  
 SKU)"  
 type: string  
 origin\_country\_code:  
 description: Two-letter country code (ISO 3166-1 Alpha-2)  
 for the product's  
 origin country. This is used in customs tracking for  
 International shipments.  
 Defaults to the home country of the Merchant.  
 type: string  
 location:  
 description: Location of the product  
 type: string  
 dangerous\_goods\_code:  
 description: |-  
 This is the DG code required when making dangerous goods  
 declarations.

Before you can declare orders as having dangerous goods,  
 you must first request for dangerous goods to be enabled for your  
 account. Contact Shippit Support or your account manager to have  
 this enabled.

DG surcharges are often charged on the weight of the  
 order, so it is recommended to split your order into multiple orders  
 if your order contains either:

- both dangerous and non-dangerous goods, or
- more than one dangerous goods code

type: string  
 dangerous\_goods\_text:  
 description: Additional information related to dangerous  
 goods being shipped

type: string  
 dangerous\_goods\_class:  
 description: This is the DG class when making dangerous  
 goods declarations.

type: string  
 tariff\_code:  
 description: Tariff code for International orders  
 declarations

type: string  
 quantity:  
 description: The number of products ordered.  
 type: integer

price:  
 description: Unit price of each product line item. Used in



```

the packing slip
 and customs declarations.
 type: number
example:
 quantity: 1
 dangerous_goods_code: ID8000
 dangerous_goods_text: ID8000 Consumer commodities -
Dangerous Goods as per
 attached DGD
 dangerous_goods_class: "9"
WebhookRequest:
 title: Root Type for WebhookRequest
 description: ""
 type: object
 properties:
 tracking_number:
 type: string
 tracking_url:
 type: string
 current_state:
 description: "Possible enumerations are
`await_collection`, `awaiting_collection`,\
 \ `awaiting_drop_off`, `cancelled`, `completed`,
`parcel_completed`, `partially_completed`,\
 \ `customs_awaiting_paym`, `customs_failed`,
`customs_on_hold`, `damaged`,\
 \ `delivery_attempted`, `delivery_failed`, `in_transit`,
`insufficient_address`,\
 \ `invalidated`, `lost`, `pickup_failed`, `processing`,
`ready_for_pickup`,\
 \ `return_booked`, `return_booking_failed`,
`return_requested`, `returned_to_sender`,\
 \ `untrackable`, `with_customs`, `with_driver`"
 type: string
 retailer_order_number:
 type: string
 courier_name:
 type: string
 courier_job_id:
 type: string
 delivery_address:
 type: string
 delivery_suburb:
 type: string
 delivery_postcode:
 type: string
 delivery_state:
 type: string
 merchant_url:
 type: string
 status_history:
 type: array
 items:
 $ref: '#/components/schemas/WebhookTrackingHistory'

```

```
products:
 type: array
 items:
 type: object
 properties:
 quantity:
 format: int32
 type: integer
 sku:
 type: string
 title:
 type: string
 product_line_id:
 type: string
 expected_delivery_date:
 description: ""
 type: string
 retailer_reference:
 description: ""
 type: string
 source_platform:
 description: ""
 type: string
 delivery_country_code:
 description: ""
 type: string
 delivery_latitude:
 description: ""
 type: integer
 delivery_longitude:
 description: ""
 type: integer
 delivery_instructions:
 description: ""
 type: string
example:
 tracking_number: PPYvZCTod5bkD
 expected_delivery_date: 2016-04-26T20:46:01+10:00
 tracking_url: http://www.shippit.com/track/ppyvzctod5bkd
 current_state: completed
 retailer_order_number: WEYFEW232
 retailer_reference: '#0D12345'
 courier_name: Couriers Please
 courier_job_id: CPAVZUZ0001749
 delivery_address: 123 Fake Drive
 delivery_suburb: Sydney
 delivery_postcode: "2000"
 delivery_state: NSW
 delivery_country_code: AU
 delivery_latitude: -33.77
 delivery_longitude: 150.91
 merchant_url: myshopify.storename.com
 status_history:
 - status: completed
```

```

 time: 2016-04-26T19:36:32.000Z
 - status: with_driver
 time: 2016-04-26T18:36:32.000Z
 - status: in_transit
 time: 2016-04-26T17:36:32.000Z
 - status: ready_for_pickup
 time: 2016-04-26T16:36:11.000Z
 - status: despatch_in_progress
 time: 2016-04-26T16:27:35.000Z
 - status: order_placed
 time: 2016-04-26T16:25:04.000Z
products:
 - quantity: 1
 sku: EWPE123123
 title: Super awesome red tshirt
 product_line_id: "1234"
 - quantity: 7
 sku: EWPE123123
 title: Super awesome blue tshirt
 product_line_id: "1235"
Order:
 title: Root Type for Order
 description: Represents an Order to be delivered by Shippit.
 required:
 - delivery_suburb
 - delivery_state
 - delivery_postcode
 - delivery_address
 type: object
 properties:
 delivery_address:
 description: delivery address for the order
 type: string
 delivery_postcode:
 description: delivery postcode for the order
 type: string
 delivery_state:
 description: |-
 delivery state for the order
 Australian States valid options are ``NSW``,
 ``QLD``, ``ACT``, ``WA``, ``NT``, ``SA``, ``VIC``,
 ``TAS``
 type: string
 delivery_suburb:
 description: delivery suburb for the order
 type: string
 delivery_instructions:
 description: "special delivery instructions for the order,
limited to 55\
 \ characters"
 type: string
 receiver_contact_number:
 description: "``conditional`` -- number of the person
receiving the order,\

```

\ may be different than the user who purchased the order. Mandatory for\  
\ International orders."

type: string  
receiver\_name:  
description: name of the person receiving the order if different than the user who purchased the order  
type: string  
retailer\_invoice:  
description: "Merchant invoice number – the customer-facing sales order\  
\ reference of the order. This would be what the recipient sees on the\  
\ shipping label, packslip, customs declaration, receipt, etc."

type: string  
receiver\_language\_code:  
description: Two-letter ISO 639-1 language code of the recipient. Used to determine the language of tracking notifications and other communications.

Defaults to ``EN``

type: string  
delivery\_date:  
format: date  
description: |-  
The date of the order for priority orders.

The recommended format is in ISO-8601 `YYYY-MM-DD`, although the Shippit is flexible enough to parse `YYYY/MM/DD`.

``conditional`` -- specifies the date of the order for priority orders

``delivery\_date`` must be in the future.  
type: string  
delivery\_window:  
description: |-  
expected format: ``00:00-00:00``

``conditional`` -- specifies the time window of the order, to be used when sending a priority order

``delivery\_window`` must be in the future.  
type: string  
courier\_job\_id:  
description: "Used in \*\*tracking\_order\*\*'s, the ID assigned by some carriers\  
\ to the shipment / consignment."  
type: string  
delivery\_country\_code:  
description: "The destination country code for the order in `ISO 3166 Alpha-2`\  
\

\ format. If unassigned, defaults to the country code of the merchant's\

\ primary location, or otherwise 'AU'."

type: string

delivery\_district\_city:

description: Used in some countries to add additional district city information

in addition to the `delivery\_suburb` and `delivery\_state`. Most of this

time this is determined using the `delivery\_suburb` and `delivery\_postcode`.

type: string

example: |-

{

"user\_attributes": {

"email": "test@shippit.com",

"first\_name": "John",

"last\_name": "Smith"

},

"parcel\_attributes": [

{

"depth": 0.13,

"length": 0.1,

"location": "C123",

"price": "12.12",

"qty": 1,

"sku": "EWPE123123",

"title": "Super awesome red tshirt",

"weight": 2.1,

"width": 0.11,

"product\_line\_id": "1234",

"origin\_country\_code": "CN",

"packed": 1

},

{

"depth": 0.13,

"length": 0.1,

"location": "B22",

"price": "1.21",

"qty": 7,

"sku": "EWPE123123",

"title": "Super awesome blue tshirt",

"weight": 2.1,

"width": 0.11,

"dangerous\_goods\_code": "ID8000",

"dangerous\_goods\_text": "ID8000 Consumer

commodities - Dangerous Goods as per attached DGD",

"product\_line\_id": "1235",

"packed": 1

}

],

"authority\_to\_leave": "No",

"courier\_type": "standard",

"delivery\_address": "1 Union Street",

```

 "delivery_postcode": "2009",
 "delivery_state": "NSW",
 "delivery_suburb": "Pyrmont",
 "delivery_instructions": "Special instructions for the
order go here",
 "receiver_contact_number": "0400000000",
 "receiver_name": "Adam",
 "retailer_invoice": "#23201005",
 "retailer_reference": "#23201005"
 }

```

#### OrderDeleteResponse:

description: A response to an Order DELETE request

type: object

allOf:

- \$ref: '#/components/schemas/OrderResponse'

#### TrackingResponseOrder:

title: Root Type for TrackingResponseOrder

description: Order details returned by a tracking request

type: object

properties:

tracking\_number:

description: Tracking number of the Order

type: string

tracking\_url:

description: URL presenting tracking info for the Order.

This can be shown

to the recipient to give them status updates on the

Order.

type: string

success:

description: Whether or not this tracking request was

successful - should

always be `true` as tracking errors will be returned at

the top-level

response.

type: boolean

error:

description: Errors associated with the tracking request - should always

be `null` as tracking errors will be returned at the

top-level response.

type: string

track:

description: "A list of historical tracking statuses for the Order. This\

\ will at minimum list the order status, along with the date and time\

\ when it changed, and who was responsible for the status change."

type: array

items:

\$ref: '#/components/schemas/

#### TrackingResponseTrackingHistory'

example:

```
tracking_number: PPdNSOmJxxfzz
tracking_url: https://app.staging.shippit.com/track/
ppdnsomjxxfzz
```

```
success: true
track:
- status: despatch_in_progress
 date: 2020-08-28T00:00:00.000Z
 timestamp: 48332
 status_owner: test-store-name
- status: order_placed
 date: 2020-06-19T00:00:00.000Z
 timestamp: 36243
 status_owner: test-store-name
```

TrackingResponse:

description: |-  
Represents a response to the Tracking API, which provides tracking info for a specific order.

If successful, will return a response object with tracking information. Otherwise, will return an error code at the top-level.

```
type: object
allOf:
- type: object
 properties:
 response:
 $ref: '#/components/schemas/TrackingResponseOrder'
 description: "If the tracking request is successful,
order details relevant\
\ to tracking."
- $ref: '#/components/schemas/DataResponse'
```

example:

```
response:
tracking_number: PP39TBECV7QSSPK
tracking_url: http://www.shippit.com/track/pp39tbecv7qsspk
success: true
track:
- status: completed
 date: 2015-04-03T00:00:00.000Z
 timestamp: 58985
 status_owner: Bonds Couriers
- status: with_driver
 date: 2015-04-03T00:00:00.000Z
 timestamp: 55385
 status_owner: Bonds Couriers
- status: in_transit
 date: 2015-04-03T00:00:00.000Z
 timestamp: 55325
 status_owner: Bonds Couriers
- status: ready_for_pickup
 date: 2015-04-03T00:00:00.000Z
 timestamp: 49400
 status_owner: Harold's Harpoons
- status: despatch_in_progress
 date: 2015-04-03T00:00:00.000Z
```

```

 timestamp: 36603
 status_owner: Harold's Harpoons
 - status: order_placed
 date: 2015-04-03T00:00:00.000Z
 timestamp: 07:23:32
 status_owner: Harold's Harpoons
DataResponse:
 description: Response message containing data.
 type: object
MerchantResponse:
 title: Root Type for MerchantResponse
 description: "Represents a response to the Merchant request
API, which shows\
 \ either the generated Merchant object, or an error code
with a description\
 \ of the error."
 type: object
 allOf:
 - type: object
 - $ref: '#/components/schemas/DataResponse'
 properties:
 response:
 $ref: '#/components/schemas/Merchant'
 description: Object representing the merchant account
settings
 example:
 response:
 store_name: New Store Name
 company_name: Harold Pty Ltd
 contact_name: Robert
 contact_phone: "0400000000"
 shipping_cart_method_name: ""
 preparation_time: 60
 website_url: www.haroldsharpoons.com.au
 address_1: 110 Rex Road
 suburb: Sydney
 state: NSW
 postcode: "2000"
 country_code: AU
LabelResponse:
 description: "Represents a response to the Label API, which
provides an Order\
 \ and related label information"
 type: object
 allOf:
 - type: object
 - $ref: '#/components/schemas/DataResponse'
 properties:
 response:
 $ref: '#/components/schemas/Label'
 description: Object representing the Label in Shippit
 example: "{\r\n \"response\": {\r\n \"data\": null,\r\n\r\n
\r\n \"id\": 5044,\r\n \"order\": {\r\n

```



```

\"courier_delivery_instructions\"\\
 : \"Authority to Leave. \",\r\n
null,\r\n
 \"courier_job_id\": \"SHP0100002\", \r\n
\"delivery_address\"\\
 : \"37 Manorvale Pde\", \r\n
 \"delivery_date\":
null,\r\n
 \"delivery_instructions\": \"\", \r\n
\"delivery_postcode\"\\
 : \"3030\", \r\n
 \"delivery_state\": \"VIC\", \r\n
\"\\
 delivery_district_city\": null, \r\n
\"delivery_suburb\": \"Werribee\"\\
 , \r\n
 \"delivery_window\": null, \r\n
\"id\": 9972, \r\n
 \"invoice_number\": \"INV-1384\", \r\n
\"parcels\"\\
 : [\r\n
0.19, \r\n
 { \r\n
 \"length\": 0.38, \r\n
 \"name\": \"\", \r\n
 \"weight\": 1.0, \r\n
 \"width\": 0.29, \r\n
 \"label_number\":
\"SHP010000201000930803\", \r\n
 \"courier_data\": { \r\n
\"product_code\"\\
 : \"X123\", \r\n
 \"pickup_zone\":
\"SYD\", \r\n
 \"dropoff_zone\":
\"MEL\" \r\n
 } \r\n
], \r\n
\"products\": [\r\n
 { \r\n
 \"title\": 'Industrial
Paint Stripper', \r\n
 \"price\": 29.13, \r\n
 \"sku\": 0.1, \r\n
 \"quantity\": 1, \r\n
 \"tariff_code\"\\
 : \"000999\", \r\n
 \"dangerous_goods_code\": \"ID8000\"\\
 , \r\n
 \"dangerous_goods_text\": \"ID8000
Consumer commodities\\
 \ - Dangerous Goods as per attached DGD\", \r\n
 \"origin_country_code\"\\
 : \"TH\" \r\n
 } \r\n
], \r\n
\"price\": \r\n
 \"15.95\", \r\n
 \"receiver_contact_number\":
 \"\", \r\n
 \"receiver_language_code\": \"EN\", \r\n
\"receiver_name\"\\
 : \"Jane Doe\", \r\n
 \"retailer_invoice\": \"\",
\r\n
 \"slug\": \"ppkfgy44u8nff\", \r\n
 \"state\":

```

```

{"completed": true,
 "tracking_number": "PPKFqy44U8Nff",
 "user": {
 "email": "jane.doe@shippit.com",
 "first_name": "Jane",
 "id": 7808,
 "last_name": "Doe",
 "mobile": ""
 },
 "customs_documents_require_printing": false,
 "documents": {
 "archive_awb": {
 "url": "https://shippit-web-production.s3.amazonaws.com/uploads/order/archive_document/15870762/archive_581833317220190318-2852-yi07y0.PDF",
 "page_size": "a6",
 "file_type": "pdf"
 },
 "customs_invoice": {
 "url": "https://shippit-web-production.s3.amazonaws.com/uploads/order/customs_invoice/361690/commercial_invoice_3073489348620190218-6221-mxrhx4.pdf",
 "page_size": "a3",
 "file_type": "pdf"
 },
 "dangerous_goods_declaration": {
 "url": "https://shippit-web-production.s3.amazonaws.com/uploads/order/dangerous_goods_declaration/361690/dangerous_goods_declaration_3073489348620140318-6220-11vib5y.pdf",
 "page_size": "a3",
 "file_type": "pdf"
 },
 "shipping_label": {
 "url": "https://shippit-production.s3.amazonaws.com/uploads/label/attachment/5044/PPKFqy44U8Nff.pdf",
 "page_size": "a6",
 "file_type": "pdf"
 }
 },
 "qualified_url": "https://shippit-production.s3.amazonaws.com/uploads/label/attachment/5044/PPKFqy44U8Nff.pdf"
}

```

```

OrderResponseOrderDocumentsHash:
 title: OrderResponseOrderDocumentsHash
 description: A keyword hash of documents generated for an
order. Usually populated
 during the LABEL call.
 type: object
 properties:
 archive_awb:

```

```

 $ref: '#/components/schemas/LabelDocument'
 description: Archive Airway Bill
 customs_invoice:
 $ref: '#/components/schemas/LabelDocument'
 description: Customs invoice. Generated for International
orders.
 dangerous_goods_declaration:
 $ref: '#/components/schemas/LabelDocument'
 description: Dangerous goods declaration – generated for
orders that have
 dangerous_goods_properties specified.
 packing_slip:
 $ref: '#/components/schemas/LabelDocument'
 description: Packing slip for the order.
 shipping_label:
 $ref: '#/components/schemas/LabelDocument'
 description: |-
 Shipping label for the order.

```

If the merchant account was configured to have the label and packing slip combined into a single document, the `shipping\_label` will print the label only without the packing slip.

If the merchant account was configured to enable ZPL output, can contain an `encoded\_label` field containing ZPL output.

```

example:
 archive_awb:
 page_size: a4
 file_type: pdf
 customs_invoice:
 page_size: a4
 file_type: pdf
 dangerous_goods_declaration:
 page_size: a4
 file_type: pdf
 packing_slip:
 url: https://shippit-web-staging.s3-ap-
southeast-2.amazonaws.com/uploads/packing_slip/attachment/123456/
PP12345678.pdf
 page_size: a4_pickslip_and_label
 file_type: pdf
 shipping_label:
 url: https://shippit-web-staging.s3-ap-
southeast-2.amazonaws.com/uploads/label/attachment/123456/
PP12345678.pdf
 page_size: a6
 file_type: pdf
 encoded_label: ZPLDATA
OrderResponseOrderParcel:
 title: Order Response Parcel specification
 description: Represents a parcel attached to an order.
 type: object
 allOf:
 - type: object

```

- \$ref: '#/components/schemas/Parcel'  
properties:

qty:  
description: |-  
The number of parcels specified by the entry.

Depending on how the merchant account is configured, the number of parcels generated may not match the number of parcel attributes passed in. By default, Shippit will try to combine all parcel attributes into a single parcel with the sum of the weights.

However, if `Allocate each item in an order to a separate carton` is enabled, then Shippit will generate one parcel entry for each parcel entry sent in.

type: number

example:

qty: 1  
length: 0.1  
width: 0.2  
depth: 0.2  
weight: 1  
dangerous\_goods\_code: ID8000  
dangerous\_goods\_text: ID8000 Consumer commodities -

Dangerous Goods as per  
attached DGD

OrderValidationErrorResponse:

title: Root Type for OrderValidationErrorResponse  
description: Response message for order create requests with  
`validate` flag  
set to true.

type: object

allOf:

- type: object

properties:

delivery\_postcode:

description: List of error messages related to the  
postcode.

type: array

items:

type: string

delivery\_suburb:

description: List of error messages related to the  
suburb.

type: array

items:

type: string

suggested\_address:

description: "Suggested address, if one is found."

type: object

properties:

suburb:

type: string

postcode:

type: string

```

 state:
 type: string
 country_code:
 type: string
 delivery_country_code:
 description: List of error messages related to the
country code.
 type: array
 items:
 type: string
 delivery_state:
 description: List of error messages related to the
state.
 type: array
 items:
 type: string
 products:
 description: List of error messages related to the
provided product information.
 Typically appears in international orders.
 type: array
 items:
 type: string
 - $ref: '#/components/schemas/GenericErrorResponse'
 example:
 error: invalid
 error_description: The details provided are invalid
 delivery_postcode:
 - is invalid for Australia and should follow the pattern
9999
 delivery_suburb:
 - Invalid suburb name for the postcode 20261
 suggested_address:
 suburb: bondi beach
 postcode: "2026"
 state: NSW
 country_code: au
 OrderDeleteErrorResponse:
 description: ""
 type: object
 allOf:
 - required:
 - order_state
 type: object
 properties:
 order_state:
 description: The current state of the order should a
DELETE request fail
 type: string
 - $ref: '#/components/schemas/GenericErrorResponse'
 TrackingErrorResponse:
 title: Root Type for TrackingErrorResponse
 description: Error response returned in tracking requests
 type: object

```

```

allOf:
- required:
 - error
 type: object
 properties:
 error:
 $ref: '#/components/schemas/
TrackingErrorResponseDetails'
 description: Error code returned by Shippit
 properties:
 order_id:
 type: array
 items:
 type: string
 success:
 description: Returns `false`
 type: boolean
 error_description:
 description: Human-readable description of error
encountered.
 type: string
- $ref: '#/components/schemas/ErrorResponse'
example:
 error:
 order_id:
 - "Sorry, the tracking number you entered does not exist."
 error_description: The data given to this server does not
meet our criteria.
 success: false
TrackingErrorResponseDetails:
 title: Root Type for TrackingResponseErrors
 description: |-
 A Hash of error codes returned by tracking response.

 Each error code is returned as a key, with an array of
strings for each instance of the error.
 type: object
 properties:
 order_id:
 description: Indicates an order that cannot be found.
 type: array
 items:
 type: string
 example:
 order_id:
 - "Sorry, the tracking number you entered does not exist."
QuoteErrorResponse:
 title: Root Type for QuoteErrorResponse
 description: Request-level error response for the Quote API.
 type: object
allOf:
- type: object
 properties:
 success:

```

```

 description: Returns `false`.
 type: boolean
 - $ref: '#/components/schemas/GenericErrorResponse'
 example:
 error: bad_request
 error_description: Dutiable amount must be greater than 0
for international
 orders
 success: false
GenericErrorResponse:
 title: ErrorResponse
 description: Response message containing an error.
 type: object
 allOf:
 - required:
 - error
 type: object
 properties:
 error:
 description: Error code returned by Shippit
 type: string
 error_description:
 description: Human-readable description of error
encountered.
 type: string
 - $ref: '#/components/schemas/ErrorResponse'
 example:
 error: system
 error_description: An unknown error has occurred.
ErrorResponse:
 title: ErrorResponse
 description: Response message containing an error.
 required:
 - error
 type: object
 properties:
 error:
 description: Error code returned by Shippit
 error_description:
 description: Human-readable response
 type: string
 example:
 error: system
 error_description: An unknown error has occurred.
OrderRequestOrderCustomsClearanceAttributes:
 title: Root Type for
OrderRequestOrderCustomsClearanceAttribute
 description: Represents customs clearance parameters for
international orders.
 type: object
 properties:
 tax_id_type:
 description: |-
 The Shipper's tax id type which is passed to a carrier

```

for the purpose of customs clearance. Valid values include:

- `EORI`
- `IOSS`
- `LVG`
- `OSR`
- `VOEC`
- `VAT/GST`
- `FTZ`
- `DAN`
- `TAN`
- `DTF`
- `EIN`
- `SSN`
- `DUN`
- `FED`
- `STA`
- `CNP`
- `GBVAT`
- `NZ IRD`

maxLength: 10

type: string

tax\_id\_country\_code:

description: The issuing country of the tax id number in  
in ISO 3166 Alpha-2

format. eg. GB for Great Britain

maxLength: 2

type: string

tax\_id\_number:

description: The tax id number that is declared against  
this order.

maxLength: 15

type: string

incoterm:

description: |-

A universal term of trade that defines the  
responsibility and obligations with the transport of goods. Most  
commonly used to indicate the responsible party for payment of  
duties and taxes. Valid values include:

- `DDU`
- `DDP`
- `EXW`
- `FCA`
- `CPT`
- `CFR`
- `CIP`
- `CIF`
- `DAF`
- `DAT`
- `DAP`
- `DPU`
- `DES`
- `DEQ`
- `FAS`
- `FOB`



```

 maxLength: 10
 type: string
 export_reason:
 description: |-
 A term used to indicate the reason for export of
 dutiable goods. Selected options can affect how your shipment is
 cleared and how much tax is imposed on your shipment during customs
 clearance. Example values:
 - `Permanent`
 - `Temporary`
 - `Return For Repair`
 - `Used Exhibition Goods To Origin`
 - `Intercompany Use`
 - `Commercial Purpose Or Sale`
 - `Personal Belongings or Personal Use`
 - `Sample`
 - `Gift`
 - `Return To Origin`
 - `Warranty Replacement`
 - `Diplomatic Goods`
 - `Defence Material`
 maxLength: 40
 type: string
 recipient_id:
 description: "A free text field to provide a
 identification number for a\
 \ recipient. For example, a citizen identification
 number or passport\
 \ number which is required for customs clearance for
 some international\
 \ destinations. eg: PA1992991"
 maxLength: 20
 type: string
 receiver_freight_charge_amount:
 description: The amount charged to the receiver for
 freight services. This
 represents the cost that the receiver is responsible for
 paying as part
 of the freight or shipping charges.
 type: number
 example:
 tax_id_type: FTZ
 tax_id_country_code: GB
 tax_id_number: '*****'
 incoterm: DAP
 export_reason: Sample
 recipient_id: '*****'
 receiver_freight_charge_amount: 7.99
 UserAttributes:
 title: Root Type for UserAttributes
 description: |
 Represents the customers details attached to the order.
 required:
 - first_name

```

- email

properties:

- email:
  - description: the customer's email address. Must match regular expression as defined in <https://html.spec.whatwg.org/multipage/input.html#valid-e-mail-address>
  - type: string
- first\_name:
  - description: |
    - the customer's first name. If ``last\_name`` is not provided, then it is used as the customer's full name.
  - type: string
- last\_name:
  - description: customer's last name
  - type: string
- mobile:
  - description: Mobile number of the user / receiver.
  - type: string

example:

- email: john.smith@shippit.com
- first\_name: John
- last\_name: Smith

LabelResponseOrder:

- description: Represents an Order that is returned in Label response
- required:
  - tracking\_number
  - slug
  - documents
- type: object
- properties:
  - tracking\_number:
    - description: Unique random ID assigned by Shippit to an order. This can be used as a reference for future API calls or support tickets.
    - type: string
  - slug:
    - description: ``tracking\_number`` in lowercase.
    - type: string
  - parcel\_attributes:
    - \$ref: '#/components/schemas/OrderResponseOrderParcel'
    - description: An array of parcel specifications that were created with the order.
  - products:
    - description: |-
      - An array of product items specified in the order.
    - If product information was not provided, this array will be blank.
    - type: array
    - items:

```

 $ref: '#/components/schemas/Product'
tracking_url:
 description: The URL of the order's customer tracking
page.
 type: string
return:
 description: Whether or not the order is a return order.
 type: boolean
id:
 description: Shippit internal numerical ID
 type: integer
processing_state:
 description: Internal order status set and progressed by
Shippit upon creation.
 type: string
invoice_number:
 description: Internal link between an order and a Shippit
invoice - will
 almost always be null on order creation.
courier_delivery_instructions:
 description: "Delivery instructions as sent to the
courier. In most cases,\
 \ this is the same as the `delivery_instructions` passed
into the request.\
 \ Some couriers, however, have a specific format
expected when sending\
 \ delivery instructions through their API. This will
contain the actual\
 \ delivery instructions that get sent through the
courier API when booked."
 type: string
price:
 description: |-
 The price that will be charged for the allocated order.

```

If you're creating orders by passing in `courier\_type`, the courier will usually not be yet be assigned at order creation and you will get `0`. Courier allocation will be run as a background process depending on the merchant preferences, available couriers, and quotes returned by the couriers, which may take a few seconds to complete after order creation.

Note that the Shippit web user can also update the courier in between API calls. The final courier and quoted price can be determined from the LABEL call.

If the quoted price is still `0` during the LABEL call, something likely went wrong with the courier selection and the order may need to be amended.

```

customs_documents_require_printing:
 description: "Whether or not customs documents, such as
declarations, invoices,\
 \ etc for the Order that should be printed and included
with the shipment.\

```

```

 \ By default this is `true` for International orders."
 type: boolean
 courier_type:
 description: The courier allocated to the order. See the
ORDER CREATE `courier_allocation`
 API for available keywords.
 type: string
 tracking_histories:
 description: An array of Tracking History statuses that
are are shown to
 the recipient as part of the order's history on the
tracking page.
 type: array
 items:
 $ref: '#/components/schemas/
OrderResponseTrackingHistory'
 state:
 description: The state of the order as progressed by the
merchant.
 type: string
 user:
 $ref: '#/components/schemas/UserAttributes'
 description: Represents the recipient of the order
 documents:
 $ref: '#/components/schemas/
OrderResponseOrderDocumentsHash'
 description: Documents attached to the label response
 courier_job_id:
 description: The ID assigned by some carriers to the
shipment/consignment.
 type: string
 OrderResponse:
 title: OrderResponse
 description: "Represents a response to an OrderRequest, which
shows either the\
 \ generated order object or an error code with a description
of the error."
 type: object
 allOf:
 - $ref: '#/components/schemas/DataResponse'
 properties:
 response:
 $ref: '#/components/schemas/OrderResponseOrder'
 description: Object representing the Order in shippit.
 example:
 response:
 courier_delivery_instructions: special instructions
 courier_job_id: "30734876324"
 courier_name: eParcel
 delivery_address: 1 Union Street
 delivery_instructions: test special instructions
 delivery_postcode: "2009"
 delivery_state: NSW
 delivery_suburb: Pyrmont

```

```

id: 26599
parcel_attributes:
- depth: 0.13
 length: 0.1
 qty: 1
 weight: 16.8
 width: 0.11
products:
- title: Industrial Paint Stripper
 price: 29.13
 sku: "0.1"
 tariff_code: "000999"
 dangerous_goods_code: ID8000
 dangerous_goods_text: ID8000 Consumer commodities -
Dangerous Goods as
 per attached DGD
 origin_country_code: TH
 quantity: 1
 price: "0.0"
 processing_state: created
 receiver_contact_number: "0400000000"
 receiver_name: Josh
 retailer_invoice: '#23201005'
 slug: ppu38wz2tdonj
 state: processing
 tracking_number: PPu38Wz2TdoNj
 user_attributes:
 email: test@shippit.com
 first_name: jon
 last_name: smith
 mobile: "0491570006"
BookDocumentSuccess:
 title: Root Type for BookDocumentSuccess
 description: ""
 type: object
 properties:
 response:
 type: object
 properties:
 manifest:
 type: string
 manifest_id:
 format: int32
 type: integer
 courier:
 type: string
 manifest_pdf:
 type: string
 order_count:
 format: int32
 type: integer
 orders:
 type: array
 items:

```

```

 type: string
 success:
 type: boolean
 example:
 response:
 manifest: SHP12345
 manifest_id: 123123
 courier: eParcel
 manifest_pdf: https://shippit.com/
manifest20160721-96368-1tpowvr.pdf
 order_count: 1
 orders:
 - PPKNpNhR0A5Ud
 success: true
BookingInProgress:
 title: Root Type for BookingInProgress
 description: ""
 type: object
 properties:
 error:
 type: string
 error_description:
 type: string
 example:
 error: generation_in_progress
 error_description: "The requested manifest has not been
booked with the carrier,\
 \ please try again later"
Label:
 title: Root Type for Label
 description: Represents an Order and related labelling
information
 required:
 - order
 - qualified_url
 type: object
 properties:
 data:
 type: object
 id:
 format: invalid
 description: Shippit internal ID of the generated label.
 type: integer
 order:
 $ref: '#/components/schemas/LabelResponseOrder'
 description: The Order being labelled.
 qualified_url:
 description: |-
 URL to a printable label for the Order.
 This is a [pre-signed URL](https://docs.aws.amazon.com/
AmazonS3/latest/dev/PresignedUrlUploadObject.html), generated by the
Shippit platform to provide access to a secured label. The pre-
signed URL remains valid for 7 days. If you need to access this
label after expiration, please make another request for a new URL to

```



```

 : \"PPKFqy44U8Nff\",\\r\\n \"user\": {\\r\\n
\"email\": \"jane.doe@shippit.com\"\\
,\\r\\n \"first_name\": \"Jane\",\\r\\n
\"id\": 7808,\\r\\n\\
 \"last_name\": \"Doe \",\\r\\n
\"mobile\": \"\"\\r\\n\\
 },\\r\\n
\"customs_documents_require_printing\": false,\\r\\n\\
 \"documents\": {\\r\\n \"archive_awb\":
{\\r\\n \\
 \"url\": \"https://shippit-web-
production.s3.amazonaws.com/uploads/order/archive_document/15870762/
archive_581833317220190318-2852-yi07y0.PDF\"\\
,\\r\\n \"page_size\": \"a6\",\\r\\n
\"file_type\":\\
 \\ \"pdf\"\\r\\n },\\r\\n
\"customs_invoice\": {\\r\\n \\
 \"url\": \"https://shippit-web-
production.s3.amazonaws.com/uploads/order/customs_invoice/361690/
commercial_invoice_3073489348620190218-6221-mxrhx4.pdf\"\\
,\\r\\n \"page_size\": \"a3\",\\r\\n
\"file_type\":\\
 \\ \"pdf\"\\r\\n },\\r\\n
\"dangerous_goods_declaration\"\\
 : {\\r\\n \"url\": \"https://shippit-web-
production.s3.amazonaws.com/uploads/order/
dangerous_goods_declaration/361690/
dangerous_goods_declaration_3073489348620140318-6220-11vib5y.pdf\"\\
,\\r\\n \"page_size\": \"a3\",\\r\\n
\"file_type\":\\
 \\ \"pdf\"\\r\\n },\\r\\n
\"shipping_label\": {\\r\\n \\
 \"url\": \"https://shippit-
production.s3.amazonaws.com/uploads/label/attachment/5044/
PPKFqy44U8Nff.pdf\"\\
,\\r\\n \"page_size\": \"a6\",\\r\\n
\"file_type\":\\
 \\ \"pdf\"\\r\\n }\\r\\n }\\r\\n },\\r\\n
\"qualified_url\"\\
 : \"https://shippit-production.s3.amazonaws.com/uploads/
label/attachment/5044/PPKFqy44U8Nff.pdf\"\\
 \\r\\n}\"

```

ReturnRequestBillingAddress:

title: Root Type for ReturnRequestBillingAddress

description: The root of the ReturnRequestBillingAddress

type's schema.

required:

- country
- country\_code
- street
- suburb
- postcode

type: object

properties:



```
id:
 description: The ID of the record within the integrated
order platform.
 type: integer
api_reference:
 description: The API Reference provided by the
Intergration Platform
 type: string
firstname:
 description: The addressee's firstname
 type: string
middlename:
 description: The addressee's middlename
 type: string
lastname:
 description: The addressee's lastname
 type: string
company:
 description: The addressee company name
 type: string
phone:
 description: The addressee's phone number
 type: string
street:
 description: The street level address details
 minItems: 1
 type: array
 items:
 type: string
suburb:
 description: The suburb of the address
 type: string
postcode:
 description: The postcode of the address
 type: string
region:
 description: The region of the address
 type: string
region_code:
 description: The region code of the address (ISO-3166-2
Subdivision)
 type: string
country:
 description: The country name (ISO-3166 Country Name)
 type: string
country_code:
 description: The country code (ISO-3166-2 Alpha 2)
 type: string
example: |-
{
 "id": 1291,
 "api_reference": null,
 "firstname": "John",
 "middlename": null,
```

```

 "lastname": "Wick",
 "company": null,
 "phone": null,
 "street": [
 "10 Flinders Street"
],
 "suburb": "Sydney",
 "postcode": "2000",
 "region": "New South Wales",
 "region_code": "NSW",
 "country": "Australia",
 "country_code": "AU"
 }
}
ReturnRequestCustomer:
 title: Root Type for ReturnRequestCustomer
 description: The root of the ReturnRequestCustomer type's
schema.
 required:
 - email
 type: object
 properties:
 id:
 description: The ID of the record within Shippit Connect
 type: integer
 api_reference:
 description: The API Reference provided by the
Intergration Platform
 type: string
 firstname:
 description: The addressee's firstname
 type: string
 middlename:
 description: The addressee's middlename
 type: string
 lastname:
 description: The addressee's lastname
 type: string
 email:
 description: The email of the customer.
 type: string
 example: |-
 {
 "id": 1348,
 "api_reference": "1047141580863",
 "email": "test@shippit.com",
 "firstname": "John",
 "middlename": null,
 "lastname": "Wick"
 }
}
ReturnRequestItem:
 title: Root Type for ReturnRequestItem
 description: The root of the ReturnRequestItem type's schema.
 required:
 - qty

```

- qty\_ordered
- id
- price

type: object

properties:

- id:
  - description: The ID of the item record in Shippit Connect
  - type: integer
- api\_reference:
  - description: "The API Reference of the Item, according to the Integration\ \ Platform."
  - type: string
- sku:
  - description: The item's stock keeping unit (Item SKU)
  - type: string
- name:
  - description: The item's name
  - type: string
- weight:
  - format: float
  - description: "The item's weight measurement, in kilograms"
  - type: number
- depth:
  - format: float
  - description: "The item's depth measurement, in centimeteres"
  - type: number
- length:
  - format: float
  - description: "The item's length measurement, in centimeteres"
  - type: number
- width:
  - format: float
  - description: "The item's width measurement, in centimeteres"
  - type: number
- options:
  - description: "An array of options applicable to the item (ie: colour or\ \ size)"
  - type: array
  - items: {}
- qty:
  - description: |-
 

The qty of the item that could be returned by the customer.

This value takes into consideration all available data from the Integration Platform as to the "returnable amount".

In all instances, Connect will attempt to determine a suitable amount that is available for return, considering both the

qty that was originally ordered, qty shipped to the customer, along with taking into consideration any existing returns, refunds or cancellations.

```
 type: integer
 qty_ordered:
 description: "The qty of the item that was originally
ordered, according\
\ to the Integration Platform"
 type: integer
 qty_invoiced:
 description: "The qty of the item that has been invoiced,
according to the\
\ Integration Platform"
 type: integer
 qty_shipped:
 description: The qty of the item that has been marked as
shipped according
to the Integration Platform
 type: integer
 qty_cancelled:
 description: |-
 The qty of the item that has been cancelled, according
to the Integration Platform.
```

A cancelled item is considered as an item that was ordered, but not yet paid for by the customer.

Note: Not all platforms support the cancellation of an item. In cases where this data is not available, a value of `NULL` will be returned.

```
 type: integer
 qty_refunded:
 description: |-
 The qty of the item that has been refunded, according to
the Integration Platform.
```

A refunded item is considered as an item that was ordered and invoiced to the customer.

Note: Not all platforms support the refund of an item. In cases where this data is not available, a value of `NULL` will be returned.

```
 type: integer
 qty_returned:
 description: |-
 The qty of the item returned, according to the
integration platform.
```

A return item is considered as an item that was shipped to the customer and returned baack to the merchant.

Note: Not all platforms support the return of an item. In cases where this data is not available, a value of `NULL` will be returned.

type: integer  
price:  
description: "The item's price, after any item-level discounts applied"  
type: number  
original\_price:  
format: float  
description: |-  
The original price of the item, before any discounts.

Note: Some integration platforms do not support providing data relating to the original price of the item at the time the order was placed. In these instances, a `NULL` value is returned.

type: number  
discount\_amount:  
format: float  
description: "The discount amount applicable to the item.  
Note: Only item\  
level discounts are considered here. Subtotal & Order level discounts\  
are not considered in this response value."

type: number  
tax:  
format: float  
description: The tax amount applicable to the item  
type: number  
is\_price\_incl\_tax:  
description: "Indicates whether the price fields  
(`original\_price`, `price`)\  
includes tax as part of it's value."

type: boolean  
image\_url:  
description: The URL of the product image  
type: string  
is\_virtual:  
description: |-  
Indicates whether the item is considered as a "virtual" item.

Virtual items are considered non-physical items that do not require shipping. ie: Virtual Gift Cards and Downloadable Content.

type: boolean  
non\_return\_reason:  
description: The reason why the item cannot be returned.  
type: string  
example:  
id: 1890  
api\_reference: "1936837640255"  
sku: TACO  
name: Single Taco  
weight: 0.5  
options:

```

 - api_reference: "2380278235199"
 label: Title
 value: Default Title
 qty: 1
 qty_ordered: 1
 qty_shipped: 0
 qty_refunded: 0
 non_return_reason: This product cannot be returned because
it is a sale item
 price: 20
 original_price: 20
 discount_amount: 0
 tax: 2
 is_price_incl_tax: true
 image_url: https://cdn.shopify.com/s/files/1/0083/3798/1503/
products/Taco.jpeg?v=1554966383
 is_virtual: false
 ReturnRequestMerchant:
 title: Root Type for ReturnRequestMerchant
 description: The root of the ReturnRequestMerchant type's
schema.
 required:
 - merchant_id
 type: object
 properties:
 merchant_id:
 description: The Shippit Merchant ID
 type: integer
 items:
 description: The items that the Merchant is to be
considered for
 type: array
 items:
 $ref: '#/components/schemas/ReturnRequestMerchantItem'
 address:
 $ref: '#/components/schemas/ReturnRequestMerchantAddress'
 description: The address of the Merchant
 properties:
 api_reference:
 type: object
 firstname:
 type: object
 middlename:
 type: object
 lastname:
 type: object
 company:
 type: string
 phone:
 type: object
 street:
 type: array
 items: {}
 suburb:

```

```

 type: object
 postcode:
 type: string
 region:
 type: string
 region_code:
 type: string
 country:
 type: object
 country_code:
 type: object
example:
 merchant_id: 7
 items:
 - {}
 - {}
 address:
 api_reference: {}
 firstname: {}
 middlename: {}
 lastname: {}
 company: some text
 phone: {}
 street:
 - {}
 - {}
 suburb: {}
 postcode: some text
 region: some text
 region_code: some text
 country: {}
 country_code: {}
ReturnRequestMerchantAddress:
 title: Root Type for ReturnRequestMerchantAddress
 description: The root of the ReturnRequestMerchantAddress
type's schema.
 required:
 - country
 - country_code
 - street
 - postcode
 - suburb
 type: object
 properties:
 id:
 description: The ID of the record within the order
platform
 type: integer
 api_reference:
 description: The API Reference provided by the
Intergration Platform
 type: string
 firstname:
 description: The merchant's firstname

```

```
 type: string
 middlename:
 description: The merchant's middlename
 type: string
 lastname:
 description: The merchant's lastname
 type: string
 company:
 description: The merchant's company name
 type: string
 phone:
 description: The merchant's phone number
 type: string
 street:
 description: The street level address details
 minItems: 1
 type: array
 items:
 type: string
 suburb:
 description: The suburb of the address
 type: string
 postcode:
 description: The postcode of the address
 type: string
 region:
 description: The region of the address
 type: string
 region_code:
 description: The region code of the address (ISO-3166-2
Subdivision)
 type: string
 country:
 description: The country name (ISO-3166 Country Name)
 type: string
 country_code:
 description: The country code (ISO-3166-2 Alpha 2)
 type: string
 example:
 id: 56
 api_reference: some text
 firstname: some text
 middlename: some text
 lastname: some text
 company: some text
 phone: some text
 street:
 - some text
 - some text
 suburb: some text
 postcode: some text
 region: some text
 region_code: some text
 country: some text
```



```

 country_code: some text
ReturnRequestMerchantItem:
 title: Root Type for ReturnRequestMerchantItem
 description: The root of the ReturnRequestMerchantItem type's
schema.
 required:
 - id
 - sku
 - api_reference
 - name
 type: object
 properties:
 id:
 description: The ID of the item record in Shippit Connect
 type: integer
 api_reference:
 description: "The API Reference of the Item, according to
the Integration\
\ Platform."
 type: string
 sku:
 description: The item's stock keeping unit (Item SKU)
 type: string
 name:
 description: The item's name
 type: string
 qty:
 description: The qty of the item that should be considered
for this merchant.
 type: integer
 example:
 api_reference: some text
 sku: some text
 name: some text
 qty: 71
ReturnRequestShippingAddress:
 title: Root Type for ReturnRequestShippingAddress
 description: The root of the ReturnRequestShippingAddress
type's schema.
 required:
 - id
 - country
 - country_code
 - postcode
 - street
 - suburb
 type: object
 properties:
 id:
 description: The ID of the record within the integrated
order platform.
 type: integer
 api_reference:
 description: The API Reference provided by the

```

```

Intergration Platform
 type: string
 firstname:
 description: The addressee's firstname
 type: string
 middlename:
 description: The addressee's middlename
 type: string
 lastname:
 description: The addressee's lastname
 type: string
 company:
 description: The addressee company name
 type: string
 phone:
 description: The addressee's phone number
 type: string
 street:
 description: The street level address details
 minItems: 1
 type: array
 items:
 type: string
 suburb:
 description: The suburb of the address
 type: string
 postcode:
 description: The postcode of the address
 type: string
 region:
 description: The region of the address
 type: string
 region_code:
 description: The region code of the address (ISO-3166-2
Subdivision)
 type: string
 country:
 description: The country name (ISO-3166 Country Name)
 type: string
 country_code:
 description: The country code (ISO-3166-2 Alpha 2)
 type: string
 example: |-
 {
 "id": 1291,
 "api_reference": null,
 "firstname": "John",
 "middlename": null,
 "lastname": "Wick",
 "company": null,
 "phone": null,
 "street": [
 "10 Flinders Street"
],

```

```

 "suburb": "Sydney",
 "postcode": "2000",
 "region": "New South Wales",
 "region_code": "NSW",
 "country": "Australia",
 "country_code": "AU"
 }
ReturnOrder:
 title: Root Type for ReturnOrder
 description: ""
 required:
 - shipping_address
 - return_address
 - shipping_method
 - items
 - date
 - customer
 - return_request_id
 - shippit_api_reference
 type: object
 properties:
 return_request_id:
 description: The id of the corresponding return request
that was returned
 by the ReturnRequest
 type: string
 order_reference:
 description: The reference of the order returned from the
corresponding
 ReturnRequest.
 type: string
 order_api_reference:
 description: The api_reference of the order returned from
the corresponding
 ReturnRequest.
 type: string
 shippit_api_reference:
 format: int32
 description: Shippit's ID for the return order.
 type: integer
 date:
 format: date-time
 description: The date the return order was confirmed.
 type: string
 shipping_method:
 $ref: '#/components/schemas/ReturnOrderShippingMethod'
 description: The details of the shipment itself.
 properties:
 code:
 type: string
 source:
 type: string
 carrier:
 type: string

```

```
 label:
 type: string
 tracking_number:
 type: string
 customer:
 $ref: '#/components/schemas/ReturnRequestCustomer'
 description: The details of the customer making the
return.
 properties:
 email:
 type: string
 phone:
 type: string
 firstname:
 type: string
 lastname:
 type: string
 return_address:
 $ref: '#/components/schemas/ReturnOrderReturnAddress'
 description: The delivery address to where the order is
being returned.
 properties:
 firstname:
 type: string
 lastname:
 type: string
 company:
 type: string
 phone:
 type: string
 street:
 type: array
 items:
 type: string
 suburb:
 type: string
 postcode:
 type: string
 region:
 type: string
 region_code:
 type: string
 country:
 type: string
 country_code:
 type: string
 shipping_address:
 $ref: '#/components/schemas/ReturnOrderShippingAddress'
 description: The address from where the order is being
shipped from.
 properties:
 firstname:
 type: string
 lastname:
```

```

 type: string
 phone:
 type: string
 street:
 type: array
 items:
 type: string
 suburb:
 type: string
 postcode:
 type: string
 region:
 type: string
 region_code:
 type: string
 country:
 type: string
 country_code:
 type: string
 items:
 description: The items to be returned in the order.
 type: array
 items:
 $ref: '#/components/schemas/ReturnOrderItem'
 preferred_refund_method:
 description: The refund method selected by the customer.
 The available string
 values are configured from the Returns settings in the
 Shippit Admin.
 type: string
 example:
 return_request_id: aaaa-1111-bbbb-2222
 order_reference: "123456"
 order_api_reference: abcdef
 shippit_api_reference: 180663
 date: 2024-05-03T11:31:45+10:00
 preferred_refund_method: Store credit
 shipping_method:
 code: standard
 source: shippit
 carrier: eparcel
 label: AusPost Standard Post
 tracking_number: PPsometracking123
 customer:
 email: test@shippit.com
 phone: "0410123456"
 firstname: John
 lastname: Wick
 return_address:
 firstname: Robert
 lastname: Johns
 company: Harold's Harpoons
 phone: "0400000000"
 street:

```

```
 - 110 Rex Road
 suburb: Sydney
 postcode: "2000"
 region: New South Wales
 region_code: NSW
 country: Australia
 country_code: AU
 shipping_address:
 firstname: John
 lastname: Wick
 phone: "0411111111"
 street:
 - Level 10
 - 1 Union Street
 suburb: Pyrmont
 postcode: "2000"
 region: New South Wales
 region_code: NSW
 country: Australia
 country_code: AU
 items:
 - api_reference: "1980"
 sku: HARPOON
 name: Pointy Harpoon
 qty: 1
 price: 20
 reason: Changed mind
 condition: Unused
 - api_reference: "3001"
 sku: NET
 name: Fishing Net
 qty: 3
 price: 11
 reason: Faulty
 condition: Used
ReturnOrderItem:
 title: Root Type for ReturnOrderItem
 description: ""
 type: object
 properties:
 api_reference:
 type: string
 sku:
 type: string
 name:
 type: string
 qty:
 format: int32
 type: integer
 price:
 format: int32
 type: integer
 reason:
 type: string
```

```
 condition:
 type: string
 example:
 api_reference: "1980"
 sku: TACO
 name: Single Taco
 qty: 1
 price: 20
 reason: Changed mind
 condition: Uneaten
ReturnOrderRequest:
 title: Root Type for ReturnOrder
 description: Details of the returns order that has been
confirmed.
 required:
 - return_order
 type: object
 properties:
 return_order:
 $ref: '#/components/schemas/ReturnOrder'
 properties:
 return_request_id:
 type: string
 order_reference:
 type: string
 customer_email:
 type: string
 order_api_reference:
 type: string
 shippit_api_reference:
 format: int32
 type: integer
 date:
 format: date-time
 type: string
 shipping_method:
 type: object
 properties:
 code:
 type: string
 source:
 type: string
 carrier:
 type: string
 label:
 type: string
 tracking_number:
 type: string
 customer:
 type: object
 properties:
 email:
 type: string
 phone:
```

```
 type: string
 firstname:
 type: string
 lastname:
 type: string
return_address:
 type: object
 properties:
 firstname:
 type: string
 lastname:
 type: string
 company:
 type: string
 phone:
 type: string
 street:
 type: array
 items:
 type: string
 suburb:
 type: string
 postcode:
 type: string
 region:
 type: string
 region_code:
 type: string
 country:
 type: string
 country_code:
 type: string
shipping_address:
 type: object
 properties:
 firstname:
 type: string
 lastname:
 type: string
 phone:
 type: string
 street:
 type: array
 items:
 type: string
 suburb:
 type: string
 postcode:
 type: string
 region:
 type: string
 region_code:
 type: string
 country:
```



```

 type: string
 country_code:
 type: string
example:
 return_order:
 return_request_id: aaaa-1111-bbbb-2222
 order_reference: "123456"
 customer_email: test@shippit.com
 order_api_reference: abcdef
 shippit_api_reference: 180663
 date: 2024-05-03T11:31:45+10:00
 shipping_method:
 code: standard
 source: shippit
 carrier: eparcel
 label: AusPost Standard Post
 tracking_number: PPsometracking
 customer:
 email: test@shippit.com
 phone: "0410123456"
 firstname: John
 lastname: Wick
 return_address:
 firstname: Robert
 lastname: Johns
 company: Harold's Harpoons
 phone: "0400000000"
 street:
 - 110 Rex Road
 suburb: Sydney
 postcode: "2000"
 region: New South Wales
 region_code: NSW
 country: Australia
 country_code: AU
 shipping_address:
 firstname: John
 lastname: Wick
 phone: "0411111111"
 street:
 - Level 10
 - 1 Union Street
 suburb: Pyrmont
 postcode: "2000"
 region: New South Wales
 region_code: NSW
 country: Australia
 country_code: AU
ReturnOrderReturnAddress:
 title: Root Type for ReturnRequestMerchantAddress
 description: The root of the ReturnRequestMerchantAddress
type's schema.
 required:
 - country

```

- country\_code
- street
- postcode

type: object

properties:

- firstname:
  - description: The merchant's firstname
  - type: string
- lastname:
  - description: The merchant's lastname
  - type: string
- company:
  - description: The merchant's company name
  - type: string
- phone:
  - description: The merchant's phone number
  - type: string
- street:
  - description: The street level address details
  - minItems: 1
  - type: array
  - items:
    - type: string
- suburb:
  - description: The suburb of the address
  - type: string
- postcode:
  - description: The postcode of the address
  - type: string
- region:
  - description: The region of the address
  - type: string
- region\_code:
  - description: The region code of the address (ISO-3166-2 Subdivision)
  - type: string
- country:
  - description: The country name (ISO-3166 Country Name)
  - type: string
- country\_code:
  - description: The country code (ISO-3166-2 Alpha 2)
  - type: string

example:

- id: 56
- api\_reference: some text
- firstname: some text
- middlename: some text
- lastname: some text
- company: some text
- phone: some text
- street:
  - some text
  - some text
- suburb: some text

```

 postcode: some text
 region: some text
 region_code: some text
 country: some text
 country_code: some text
ReturnOrderShippingAddress:
 title: Root Type for ReturnRequestBillingAddress
 description: The root of the ReturnRequestBillingAddress
type's schema.
 required:
 - street
 - postcode
 - suburb
 - country
 - country_code
 type: object
 properties:
 firstname:
 description: The addressee's firstname
 type: string
 middlename:
 description: The addressee's middlename
 type: string
 lastname:
 description: The addressee's lastname
 type: string
 phone:
 description: The addressee's phone number
 type: string
 street:
 description: The street level address details
 type: array
 items:
 type: string
 suburb:
 description: The suburb of the address
 type: string
 postcode:
 description: The postcode of the address
 type: string
 region:
 description: The region of the address
 type: string
 region_code:
 description: The region code of the address (ISO-3166-2
Subdivision)
 type: string
 country:
 description: The country name (ISO-3166 Country Name)
 type: string
 country_code:
 description: The country code (ISO-3166-2 Alpha 2)
 type: string
 example: |-

```

```

{
 "id": 1291,
 "api_reference": null,
 "firstname": "John",
 "middlename": null,
 "lastname": "Wick",
 "company": null,
 "phone": null,
 "street": [
 "10 Flinders Street"
],
 "suburb": "Sydney",
 "postcode": "2000",
 "region": "New South Wales",
 "region_code": "NSW",
 "country": "Australia",
 "country_code": "AU"
}
ReturnOrderShippingMethod:
 title: Root Type for ReturnOrderShippingMethod
 description: ""
 type: object
 properties:
 code:
 description: "The service level of the shipment, can be
'standard' or 'express'."
 type: string
 source:
 description: Always 'shippit'.
 type: string
 carrier:
 description: The carrier used to deliver the return.
 type: string
 label:
 description: The human readable name of the carrier used
to deliver the
 return.
 type: string
 tracking_number:
 description: 'The tracking number for the return shipment.
This is used
 to receive status updates for the shipment. '
 type: string
 example:
 code: standard
 source: shippit
 carrier: eparcel
 label: AusPost Standard Post
 tracking_number: PPsometracking123
ReturnRequestResponse:
 title: Root Type for ReturnRequest
 description: The root of the ReturnRequest type's schema.
 required:
 - return_request

```

```
type: object
properties:
 return_request:
 $ref: '#/components/schemas/ReturnRequest'
 description: The Return Request object built by the order
platform.
```

```
example:
 return_request:
 id: 1355
 reference: '#1001'
 api_reference: "845450641471"
 date: 2019-04-11T03:30:30+00:00
 customer:
 id: 1348
 api_reference: "1047141580863"
 email: test@shippit.com
 firstname: John
 lastname: Wick
 billing_address:
 id: 1291
 firstname: John
 lastname: Wick
 street:
 - 10 Flinders Street
 suburb: Sydney
 postcode: "2000"
 region: New South Wales
 region_code: NSW
 country: Australia
 country_code: AU
 shipping_address:
 id: 1290
 firstname: John
 lastname: Wick
 street:
 - 10 Flinders Street
 suburb: Sydney
 postcode: "2000"
 region: New South Wales
 region_code: NSW
 country: Australia
 country_code: AU
 items:
 - id: 1890
 api_reference: "1936837640255"
 sku: TACO
 name: Single Taco
 weight: 0.5
 options:
 - api_reference: "2380278235199"
 label: Title
 value: Default Title
 qty: 1
 qty_ordered: 1
```

qty\_shipped: 0  
qty\_refunded: 0  
non\_return\_reason: This product cannot be returned  
because it is a sale

item  
price: 20  
original\_price: 20  
discount\_amount: 0  
tax: 2  
is\_price\_incl\_tax: true  
image\_url: https://cdn.shopify.com/s/files/  
1/0083/3798/1503/products/Taco.jpeg?v=1554966383  
is\_virtual: false  
currency\_code: AUD  
subtotal: 20  
tax\_amount: 2  
shipping\_amount: 10  
grand\_total: 32  
merchants: []  
created\_at: 2019-04-12T00:07:19+00:00  
updated\_at: 2019-04-12T00:07:19+00:00

ReturnRequestRequest:

title: Root Type for ReturnRequestPost  
description: Post request body for a ReturnRequest  
type: object  
properties:  
  order\_reference:  
    type: string  
  email\_address:  
    type: string

example:

  order\_reference: ORDER-10001  
  email\_address: test@shippit.com

ReturnRequest:

title: Root Type for ReturnRequest  
description: The root of the ReturnRequest type's schema.  
required:  
- customer  
- shipping\_address  
- id  
- date  
- items  
- reference

type: object

properties:

  id:

    description: "The ID of the return request, this is used  
to correlate the\

  \ Return Request with the Return Order"

    type: integer

  reference:

    description: The order reference number. This is the  
friendly order reference

    communicated to end-customers in relation to their

```

order.
 type: string
 api_reference:
 description: "The API Reference of the Order, according to
the Integration\
 \ Platform."
 type: string
 date:
 format: date-time
 description: The date the order was created. The date
format is a valid
 ISO8601 DateTime
 type: string
 customer:
 $ref: '#/components/schemas/ReturnRequestCustomer'
 description: Details relating to the customer of record
for the order
 properties:
 id:
 format: int32
 type: integer
 api_reference:
 type: string
 email:
 type: string
 firstname:
 type: string
 middlename:
 type: object
 lastname:
 type: string
 billing_address:
 $ref: '#/components/schemas/ReturnRequestBillingAddress'
 description: The Billing Address attached to the order
 properties:
 id:
 format: int32
 type: integer
 api_reference:
 type: object
 firstname:
 type: string
 middlename:
 type: object
 lastname:
 type: string
 company:
 type: object
 phone:
 type: object
 street:
 type: array
 items: {}
 suburb:

```

```

 type: string
 postcode:
 type: string
 region:
 type: string
 region_code:
 type: string
 country:
 type: string
 country_code:
 type: string
shipping_address:
 $ref: '#/components/schemas/ReturnRequestShippingAddress'
 description: The Shipping Address attached to the order
 properties:
 id:
 format: int32
 type: integer
 api_reference:
 type: object
 firstname:
 type: string
 middlename:
 type: object
 lastname:
 type: string
 company:
 type: object
 phone:
 type: object
 street:
 type: array
 items: {}
 suburb:
 type: string
 postcode:
 type: string
 region:
 type: string
 region_code:
 type: string
 country:
 type: string
 country_code:
 type: string
 items:
 description: |-
 The items contained in the order.

```

Note: All items from the original order will be included in the response, including items that do not have a qty available for return.

```

 type: array
 items:

```



```

 $ref: '#/components/schemas/ReturnRequestItem'
 currency_code:
 description: The ISO-4217 Currency Code of the Order
 type: string
 subtotal:
 format: float
 description: The subtotal of the order.
 type: number
 tax_amount:
 format: float
 description: The tax amount applicable to the order.
 type: number
 shipping_amount:
 format: float
 description: The shipping costs applicable to the order.
 type: number
 grand_total:
 format: float
 description: "The grand total of the order, after any
discounts, applicable\
 \ taxes and shipping costs"
 type: number
 merchants:
 description: "The Shippit Merchant accounts involved.
\n\nNote: This field\
 \ is optional and only provided in some integrations.
\n\nIt's purpose\
 \ is to provide a list of the Shippit merchant accounts
that are to be\
 \ considered for the return, based on a merchant's
details in relation\
 \ to the order that has been queried. "
 type: array
 items:
 $ref: '#/components/schemas/ReturnRequestMerchant'
 updated_at:
 format: date-time
 description: The date the return request was last updated
at. The date format
 is a valid ISO8601 DateTime
 type: string
 customer_return_charge:
 format: float
 description: The amount charged to the customer to process
the return.
 type: number
 customer_return_charge_currency:
 description: |
 Three letter ISO 4217 currency code that applies to
`customer_return_charge`.

 If not provided, the merchant's currency will be used.
 type: string
 example:

```

```
id: 1355
reference: '#1001'
api_reference: "845450641471"
date: 2019-04-11T03:30:30+00:00
customer:
 id: 1348
 api_reference: "1047141580863"
 email: test@shippit.com
 firstname: John
 lastname: Wick
billing_address:
 id: 1291
 firstname: John
 lastname: Wick
 street:
 - 10 Flinders Street
 suburb: Sydney
 postcode: "2000"
 region: New South Wales
 region_code: NSW
 country: Australia
 country_code: AU
shipping_address:
 id: 1290
 firstname: John
 lastname: Wick
 street:
 - 10 Flinders Street
 suburb: Sydney
 postcode: "2000"
 region: New South Wales
 region_code: NSW
 country: Australia
 country_code: AU
items:
- id: 1890
 api_reference: "1936837640255"
 sku: TACO
 name: Single Taco
 weight: 0.5
 options:
 - api_reference: "2380278235199"
 label: Title
 value: Default Title
 qty: 1
 qty_ordered: 1
 qty_shipped: 0
 qty_refunded: 0
 non_return_reason: This product cannot be returned because
```

it is a sale

```
 item
 price: 20
 original_price: 20
 discount_amount: 0
```

```
 tax: 2
 is_price_incl_tax: true
 image_url: https://cdn.shopify.com/s/files/
1/0083/3798/1503/products/Taco.jpeg?v=1554966383
 is_virtual: false
 currency_code: AUD
 subtotal: 20
 tax_amount: 2
 shipping_amount: 10
 grand_total: 32
 merchants: []
 created_at: 2019-04-12T00:07:19+00:00
 updated_at: 2019-04-12T00:07:19+00:00
```

WebhookTrackingHistory:

description: The tracking history returned by the webhook

required:

- status

- time

type: object

properties:

status:

description: Order status

type: string

time:

description: Timestamp

type: string

responses:

UnknownErrorResponse:

content:

application/json:

schema:

\$ref: '#/components/schemas/GenericErrorResponse'

examples:

Unknown Error:

value:

error: system

error\_description: An unknown error has occurred.

description: |-

Returned when an unknown error occurs.

May be returned if Shippit has trouble with the combination of the input or merchant account configuration.

BadRequestErrorResponse:

content:

application/json:

schema:

\$ref: '#/components/schemas/GenericErrorResponse'

examples:

Generic Bad Request:

value:

error: bad\_request

does not meet our  
criteria.

```
text/html:
 schema:
 type: string
description: |-
 Returned when Shippit has a problem with the input
parameters.
```

Not all input errors can result in this error. Depending on the request, the error may be parsed as an unknown error or as part of the response body.

If returned as text/html, will be blank.

```
InvalidMerchantAccountErrorResponse:
 content:
 application/json:
 schema:
 $ref: '#/components/schemas/GenericErrorResponse'
 examples:
 Invalid Merchant Account:
 value:
 error: invalid_merchant_account
 error_description: An unknown error has occurred.
 description: Returned when the API key given does not match a
valid merchant
account.
```

```
ResourceNotFound:
 content:
 application/json:
 schema:
 $ref: '#/components/schemas/GenericErrorResponse'
 examples:
 Not Found:
 value:
 error: not_found
 error_description: The requested resource could not
be found.
 description: Returned when the given id could not be found.
```

```
securitySchemes:
 AuthToken:
 type: apiKey
 description: |-
 To make requests to the API, an authentication token is
 required. This should be passed in the header containing
 `Authorization: Bearer xxxx`, substituting the authentication token
 in the place of `xxxx`.
```

To view your authentication token, login to your Shippit account, then visit `Settings` -> `Integrations`. You should see the token labelled as `API Key`.

### ## Rate Limiting

All our APIs are rate limited to 60 requests per rolling 60 seconds per merchant (based on your ``API Key``). If you exceed this rate you'll receive a ``429`` HTTP error

Should you need a higher rate, please contact our support.

## ## Additional Request Headers

To assist Shippit's support of our custom built API consumers we ask that requests to our HTTP APIs supply additional information in the request headers. The headers are as follows:

Request Header	Purpose/Usage	Example	Type/Limits
Mandatory			
user-agent	A string to help identify technical information about the integration. Useful information includes software library names, release versions/dates.	Shippit_Shipping for Magento2 v1.5.3	200 chars   Optional, but strongly encouraged
x-shippit-partner	A string to identify the entity developing and maintaining the integration. This could be a business name for self-maintained integrations, or the name of a technical partner maintaining the integration.	Shopify, Wallymart	200 chars   optional
x-shippit-platform	A string identifying the platform/software making the calls. This will help us to identify when all integrations on the same platform require action, or who to contact about known issues with a platform type.	Magento v2, CustomShop	2000   200 chars   Optional, but strongly encouraged

Whilst this information in the request headers is currently optional, it is highly encouraged as it will allow Shippit to better support your integration by helping us to more easily identify which systems are making which API requests, and who to contact with pertinent information about specific integrations and/or platforms.

name: Authorization  
in: header

SSLCertificate:  
scheme: digest  
type: http

description: |-

Both Shippit Staging and Production environment are protected by a HTTPS layer.

This will ensure the data is in the encrypted connection.

The SSL certificates are issued by Amazon which is a trusted CA(Certificate authority), and will be auto renewed every 60 days before the expiry date.

Note that due to this renewal mechanism, certificate pinning is not recommended.

tags:

- name: Quote

description: The Quote API will return a quote given a destination location and

parcel attributes.

- name: Order

- description: The Order API will submit an order to Shippit
- name: Merchant
  - description: The merchant settings API will allow you to query the current settings for your merchant account and to make updates to this via API.
- name: Track
  - description: The Track API will allow you to request tracking information or subscribe to our webhook.
- name: Book
  - description: The book API initiates a booking with the respective carriers.
- name: Label
  - description: The label API will return a URL to a label for an order.
- name: Custom Returns Integration
  - description: The returns request and returns order motification can be implemented to support the returns portal.