

# Orders

Once you've searched for flights by creating an offer request, and you've chosen which offer you want to book, you'll then want to create an order.

To create an order, you just need to provide the offer ID, payment details and some additional information on the passengers (e.g. their full name and date of birth).

## Schema

[Expand all](#) 

### ▶ **airline\_initiated\_changes** `list`

The airline-initiated changes for this Order

### ○ **available\_actions** `string[]`

The available actions you can take on this order through our API. It's a list of zero or more of the following values:

- `"cancel"` : You can **cancel** the order.
- `"change"` : You can **change** the order's slices.
- `"update"` : You can **update** some of the order's fields.

Example: `["cancel","update"]`

### ○ **base\_amount** `string` `nullable`

The base price of the order for all flights and services booked, excluding taxes

Example: `"60.60"`

The currency of the `base_amount`, as an ISO 4217 currency code. It will match your organisation's billing currency unless you're using Duffel as an accredited IATA agent, in which case it will be in the currency provided by the airline (which will usually be based on the country where your IATA agency is registered).

Example: `"GBP"`

#### ○ **booking\_reference** `string`

The airline's reference for the order, sometimes known as a "passenger name record" (PNR) or "record locator". Your customers can use this to check in and manage their booking on the airline's website. Usually, this is made up of six alphanumeric characters, but airlines can have their own formats (for example, easyJet's booking references are 7 alphanumeric characters long and LATAM's references are made up of 13 alphanumeric characters beginning with `LA`.)

Example: `"RZPNX8"`

#### ▶ **booking\_references** `list`

All carrier booking references (PNRs) associated with this order. This includes the primary booking reference as well as any additional references from other carriers involved in the booking.

A booking reference represents a passenger name record (PNR) or record locator issued by a carrier for an order

#### ▶ **cancellation** `object` `nullable`

The confirmed cancellation associated with this Order (if this Order has been cancelled)

#### ○ **cancelled\_at** `datetime` `nullable`

The ISO 8601 datetime at which the order was cancelled, if it has been cancelled

Example: `"2020-04-11T15:40:11-04:37"`

▶ **changes** `list`

The passenger-initiated changes for this Order

An order change represents alterations to an existing order with an airline

▶ **conditions** `object`

The conditions associated with this order, describing the kinds of modifications you can make to it and any penalties that will apply to those modifications. This information assumes the condition is applied to all of the slices and passengers associated with this order - for information at the slice level (e.g. "what happens if I just want to change the first slice?") refer to the `slices`. If a particular kind of modification is allowed, you may not always be able to take action through the Duffel API. In some cases, you may need to contact the Duffel support team or the airline directly.

○ **content** `enum`

Whether the order is Self-Managed or Managed

Possible values: `"self-managed"` or `"managed"`

○ **created\_at** `datetime`

The ISO 8601 datetime at which the order was created

Example: `"2020-04-11T15:48:11.642Z"`

▶ **documents** `list`

The documents issued for this order

○ **id** `string`

- **live\_mode** `boolean`

Whether the order was created in live mode. This field will be set to `true` if the order was created in live mode, or `false` if it was created in test mode.

Example: `false`

- **metadata** `object` `nullable`

Metadata contains a set of key-value pairs that you can attach to an object. It can be useful for storing additional information about the object, in a structured format. Duffel does not use this information. You should not store sensitive information in this field.

Example: `{"customer_prefs": "window seat", "payment_intent_id": "pit_00009htYpSCXrwaB9DnUm2"}`

- **offer\_id** `string`

The offer ID used at the time of booking.

Example: `"off_00009htYpSCXrwaB9DnUm0"`

- ▶ **owner** `object`

The airline who owns the order

- ▶ **passengers** `list`

The passengers who are travelling

- ▶ **payment\_status** `object`

The payment status for this order

▶ **services** `list`

The services booked along with this order

▶ **slices** `list`

The `slices` that make up the itinerary of this order. One-way journeys can be expressed using one slice, whereas return trips will need two.

○ **synced\_at** `datetime`

Airlines are always the source of truth for orders. The orders returned in the Duffel API are a view of those orders. This field is the [ISO 8601](#) datetime at which the order was last synced with the airline. If this datetime is in the last minute you can consider the order up to date.

Example: `"2020-04-11T15:48:11Z"`

○ **tax\_amount** `string` `nullable`

The amount of tax payable on the order for all the flights booked

Example: `"30.20"`

○ **tax\_currency** `string` `nullable`

The currency of the `tax_amount`, as an [ISO 4217](#) currency code. It will match your organisation's billing currency unless you're using Duffel as an accredited IATA agent, in which case it will be in the currency provided by the airline (which will usually be based on the country where your IATA agency is registered).

Example: `"GBP"`

○ **total\_amount** `string`

The total price of the order for all the flights and services booked, including taxes

- **total\_currency** `string`

The currency of the `total_amount`, as an [ISO 4217](#) currency code. It will match your organisation's billing currency unless you're using Duffel as an accredited IATA agent, in which case it will be in the currency provided by the airline (which will usually be based on the country where your IATA agency is registered).

Example: `"GBP"`

- **type** `enum` `nullable`

The type of this order

Possible values: `"instant"` or `"hold"`

- **users** `string[]`

The `ids` of users that are allowed to manage the order.

Example: `["icu_00009htyDGjIfajdNBZRLw"]`

- **void\_window\_ends\_at** `datetime` `nullable` `preview`

Some paid orders are eligible to be voided, typically within the first 24 hours after payment. An order that is cancelled within this window is eligible for a full refund. Note that orders created near to the initial departure date i.e. within 7 days may not be eligible for void. This field is the [ISO 8601](#) datetime at which the shall no longer be eligible to be void, after which any refunds shall be subject to the conditions on the order.

Example: `"2020-04-11T15:48:11Z"`

a sorting parameter is provided.

## Query parameters

[Expand all](#)

- **after** `string`

A cursor pointing to the previous page of records. For more information on how to paginate through records, see the [Pagination](#) section.

Example: `"g2wAAAACbQAAABBBZXJvbWlzdC1LaGFya2l2bQAAAB="`

- **before** `string`

A cursor pointing to the next page of records. For more information on how to paginate through records, see the [Pagination](#) section.

Example: `"g2wAAAACbQAAABBBZXJvbWlzdC1LaGFya2l2bQAAAB="`

- **limit** `integer`

The maximum number of records to return per page. Defaults to `50`. May be set to any integer between `1` and `200`. For more information on how to paginate through records, see the [Pagination](#) section.

Example: `1`

Default value: `50`

- **booking\_reference** `string`

Filters orders by their `booking reference`.

The filter requires an exact match but is case insensitive.

Example: `"RZPNX8"`

Filters orders by the offer ID used at the time of booking.

Example: `"off_00009htYpSCXrwaB9DnUm0"`

- **awaiting\_payment** `boolean`

Whether to filter orders that are awaiting payment or not. If not specified, all orders regardless of their payment state will be returned.

Example: `false`

- **sort** `enum`

By default, orders aren't returned in any specific order. This parameter allows you to sort the list of orders by the `payment_required_by`, `total_amount`, `created_at`, or `next_departure`. If you wish to sort in descending order, a `-` prefix will be needed (e.g. `-payment_required_by`). When sorting by next departure (`next_departure`), flown orders will be last.

Possible values: `"payment_required_by"`, `"total_amount"`, `"created_at"`, or `"next_departure"`

- **owner\_id[]** `string[]`

Filters the returned orders by `owner.id`. Values must be valid `airline.id`s. Check the [Airline schema](#) for details.

Example: `["arl_00009VME7DBKeMags5CliQ", "arl_00009VME7DC0aPRQvNhcmu"]`

- **origin\_id[]** `string[]`

Filters the returned orders by `origin`. Values must be valid origin identifiers. Check the [Order schema](#) for details.

Orders will be included if **any** of their slices matches the given criteria.

Example: `["arp_lhr_gb", "arp_jfk_us"]`



Filters the returned orders by `destination`. Values must be valid destination identifiers. Check the [Order schema](#) for details.

Orders will be included if **any** of their slices matches the given criteria.

Example: `["arp_lhr_gb","arp_jfk_us"]`

▶ **departing\_at** `object`

Filters the returned orders by `departure datetime`. Orders will be included if **any** of their segments matches the given criteria.

▶ **arriving\_at** `object`

Filters the returned orders by `arrival datetime`. Orders will be included if **any** of their segments matches the given criteria.

▶ **created\_at** `object`

Filters the returned orders by `creation datetime`.

○ **passenger\_name[]** `string[]`

Filters the returned orders by passengers' `family_name` and `given_name`.

Orders will be included if **any** of their passengers matches **any** of the given names. Matches are case insensitive, and include partial matches.

For example, `?passenger_name[]=roger` will return orders for both `Anna Rogers` and `Roger Smith`.

Example: `["Earhart","Smith"]`

○ **requires\_action** `boolean`

## ENDPOINT



## REQUEST



## RESPONSE



... [View full sample](#)

## Create an order

Creates a booking with an airline based on an offer.

Orders are usually paid at the time of creation, but can be held, and paid for at a later date if the offer supports it (see

`offer.payment_requirements.requires_instant_payment`).

To create an order and pay for it at the same time, specify a `payments` key with an appropriate value. Also, specify the `type` key with a value of `"instant"`.

To hold the order and pay for it later, specify the type as `"hold"`, omit the `payments` key. You'll then need to complete payment after creating the order through the [Create a payment endpoint](#).

Airlines limit the number of passengers in a single order to  $\leq 9$ .

When presenting an order confirmation to your customers (e.g. on screen or in an email), you should include the `booking_reference` and details of the full itinerary and show the full name of the operating carrier of each segment (`slices[].segments[].operating_carrier.name`) in order to comply with US regulations.

Please see the [Order creation response handling](#) documentation for more information on possible responses and how to handle them.

## Validation errors

Field	Code	Description
<code>payments</code>	<code>payments_not_allowed_for_order_type</code>	You've included a <code>payments</code> key for a <code>hold</code> order, but the key should be omitted.
<code>services</code>	<code>services_not_allowed_for_order_type</code>	You've included <code>services</code> for a <code>hold</code> order, but they're not currently

`_offer`

selected offer because either `requires_instant_payment` is `true` or the `payment_required_by` is in the past.

## Body parameters

[Expand all](#)

### ○ **metadata** `object`

Metadata contains a set of key-value pairs that you can attach to an object. It can be useful for storing additional information about the object, in a structured format. Duffel does not use this information. You should not store sensitive information in this field.

The metadata is a collection of key-value pairs, both of which are strings. You can store a maximum of 50 key-value pairs, where each key has a maximum length of 40 characters and each value has a maximum length of 500 characters.

Keys must only contain numbers, letters, dashes, or underscores.

Example: `{"payment_intent_id":"pit_00009htYpSCXrwaB9DnUm2"}`

### ▶ **passengers** `list` `required`

The personal details of the passengers, expanding on the information initially provided when creating the offer request

### ▶ **payments** `list` `required`

The payment details to use to pay for the order. This key should be omitted when the order's `type` is `hold`.

### ○ **selected\_offers** `string[]` `required`

Note that you can only book one offer per offer request.

Example: `["off_00009htyDGjIfajdNBZRlw"]`

▶ **services** `list`

The services you want to book along with the first selected offer.

○ **type** `enum`

The type of order. You can only use `hold` with offers where `payment_requirements.requires_instant_payment` is `false`.

When booking an offer with type `hold`, do not specify `payments` key.

Possible values: `"instant"` or `"hold"`

○ **users** `string[]`

The `ids` of users that would be allowed to manage the order.

Example: `["icu_00009htyDGjIfajdNBZRlw"]`

ENDPOINT



POST `https://api.duffel.com/air/orders`

REQUEST

curl



```
1 curl -X POST --compressed "https://api.duffel.com/air/orders" \  
2   -H "Accept-Encoding: gzip" \  
3   -H "Accept: application/json" \  
4   -H "Content-Type: application/json" \  
5   -H "Duffel-Version: v2" \  
6   -H "Authorization: Bearer <YOUR_ACCESS_TOKEN>" \  
7   -d '{
```

```
11     ],
12     "type": "instant",
13     "services": [
14       {
15         "quantity": 1,
16         "id": "ase_00009hj8USM7Ncg31cB123"
17       }
18     ],
19     "selected_offers": [
20       "off_00009htyDGjIfajdNBZRLw"
21     ]
22   }
```

... [View full sample](#)

## RESPONSE



```
{
  "data": {
    "void_window_ends_at": "2020-04-11T15:48:11Z",
    "users": [
      "icu_00009htyDGjIfajdNBZRLw"
    ],
    "type": "instant",
    "total_currency": "GBP",
    "total_amount": "90.80",
    "tax_currency": "GBP",
    "tax_amount": "30.20",
    "synced_at": "2020-04-11T15:48:11Z",
    "slices": [
      {
        "segments": [
          {
            "stops": [
              {
                "id": "sto_00009htYpSCXrwaB9Dn456",
                "duration": "PT02H26M",
                "departing_at": "2020-06-13T16:38:02"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

... [View full sample](#)

## List available services for an order

Retrieves a list of available services for an order.

Note this feature is currently not supported for order's of type `hold` .

### URL parameters

- **id** `string` required

Duffel's unique identifier for the resource

Example: `"ord_00009hthhsUZ8W4LxQgkjo"`

#### ENDPOINT



GET `https://api.duffel.com/air/orders/{id}/available_services`

#### REQUEST

curl



```
1 curl -X GET --compressed "https://api.duffel.com/air/orders/{id}/available_services"
2   -H "Accept-Encoding: gzip" \
3   -H "Accept: application/json" \
4   -H "Duffel-Version: v2" \
5   -H "Authorization: Bearer <YOUR_ACCESS_TOKEN>"
```

#### RESPONSE



```
{
  "data": [
    {
      "type": "baggage",
      "total_currency": "GBP",
      "total_amount": "15.00",
      "segment_ids": [
        "seg_00009hj8USM7Ncg31cB456"
      ],
      "passenger_ids": [
```

```
    "maximum_quantity": 1,  
    "id": "aso_00009UhD4ongolulWd9123"  
  }  
]  
}
```

## Get a single order

Retrieves an order by its ID

### URL parameters

○ **id** `string` required

Duffel's unique identifier for the resource

Example: "ord\_00009hthhsUZ8W4LxQgkjo"

#### ENDPOINT



GET `https://api.duffel.com/air/orders/{id}`

#### REQUEST

curl



```
1 curl -X GET --compressed "https://api.duffel.com/air/orders/{id}" \  
2   -H "Accept-Encoding: gzip" \  
3   -H "Accept: application/json" \  
4   -H "Duffel-Version: v2" \  
5   -H "Authorization: Bearer <YOUR_ACCESS_TOKEN>"
```

#### RESPONSE



```
{  
  "data": {  
    "void_window_ends_at": "2020-01-11T15:48:11Z"
```



```

    ],
    "type": "instant",
    "total_currency": "GBP",
    "total_amount": "90.80",
    "tax_currency": "GBP",
    "tax_amount": "30.20",
    "synced_at": "2020-04-11T15:48:11Z",
    "slices": [
      {
        "segments": [
          {
            "stops": [
              {
                "id": "sto_00009htYpSCXrwaB9Dn456",
                "duration": "PT02H26M",
                "departing_at": "2020-06-13T16:38:02"
              }
            ]
          }
        ]
      }
    ]
  }
}

```

... [View full sample](#)

## Update a single order

Some order fields are updateable. Each field that can be updated is detailed in the request object.

### URL parameters

○ **id** `string` required

Duffel's unique identifier for the resource

Example: `"ord_00009hthhsUZ8W4LxQgkjo"`

### Body parameters

[Expand all](#)

○ **metadata** `object`

useful for storing additional information about the object, in a structured format. Duffel does not use this information. You should not store sensitive information in this field.

The metadata is a collection of key-value pairs, both of which are strings. You can store a maximum of 50 key-value pairs, where each key has a maximum length of 40 characters and each value has a maximum length of 500 characters.

Keys must only contain numbers, letters, dashes, or underscores.

To clear this field, set it to an empty object ( `{}` ).

Example: `{"customer_prefs": "window seat", "payment_intent_id": "pit_00009htYpSCXrwaB9DnUm2"}`

#### ○ **users** `string[]`

The `ids` of customer users that would be allowed to manage the booking.

Example: `["icu_00009htyDGjIfajdNBZRlw"]`

#### ENDPOINT



PATCH `https://api.duffel.com/air/orders/{id}`

#### REQUEST

curl



```
1 curl -X PATCH --compressed "https://api.duffel.com/air/orders/{id}" \
2   -H "Accept-Encoding: gzip" \
3   -H "Accept: application/json" \
4   -H "Duffel-Version: v2" \
5   -H "Authorization: Bearer <YOUR_ACCESS_TOKEN>" \
6   -d '{
7     "data": {
8       "users": [
9         "icu_00009htyDGjIfajdNBZRlw"
10      ],
11     "metadata": {
12       "customer_prefs": "window seat",
13       "payment_intent_id": "pit_00009htYpSCXrwaB9DnUm2"
```

16 }'

## RESPONSE



```
{
  "data": {
    "void_window_ends_at": "2020-04-11T15:48:11Z",
    "users": [
      "icu_00009htyDGjIfajdNBZRlw"
    ],
    "type": "instant",
    "total_currency": "GBP",
    "total_amount": "90.80",
    "tax_currency": "GBP",
    "tax_amount": "30.20",
    "synced_at": "2020-04-11T15:48:11Z",
    "slices": [
      {
        "segments": [
          {
            "stops": [
              {
                "id": "sto_00009htYpSCXrwaB9Dn456",
                "duration": "PT02H26M",
                "departing_at": "2020-06-13T16:38:02".
```

... [View full sample](#)

## Add a service to an order

Adds services to an order.

Note this feature is currently not supported for order's of type `hold` .

## URL parameters

Example: `"ord_00009hthhsUZ8W4LxQgkjo"`

## Body parameters

[Expand all](#)

▶ **add\_services** `list` required

The services you want to add to the order.

▶ **payment** `object`

The payment details to pay for the services.

### ENDPOINT



POST `https://api.duffel.com/air/orders/{id}/services`

### REQUEST

`curl`

```
1 curl -X POST --compressed "https://api.duffel.com/air/orders/{id}/servi
2   -H "Accept-Encoding: gzip" \
3   -H "Accept: application/json" \
4   -H "Content-Type: application/json" \
5   -H "Duffel-Version: v2" \
6   -H "Authorization: Bearer <YOUR_ACCESS_TOKEN>" \
7   -d '{
8     "data": {
9       "payment": {
10         "type": "balance",
11         "currency": "GBP",
12         "amount": "30.20"
13       },
14       "add_services": [
15         {
16           "quantity": 1,
17           "id": "ase_00009hj8USM7Ncg31cB123"
18         }
19       ]
20     }
21   }'
```

 

## RESPONSE



```
{
  "data": {
    "void_window_ends_at": "2020-04-11T15:48:11Z",
    "users": [
      "icu_00009htyDGjIfajdNBZRLw"
    ],
    "type": "instant",
    "total_currency": "GBP",
    "total_amount": "90.80",
    "tax_currency": "GBP",
    "tax_amount": "30.20",
    "synced_at": "2020-04-11T15:48:11Z",
    "slices": [
      {
        "segments": [
          {
            "stops": [
              {
                "id": "sto_00009htYpSCXrwaB9Dn456",
                "duration": "PT02H26M",
                "departing_at": "2020-06-13T16:38:02".
```

[... View full sample](#)**Need some help?**

We all do sometimes; code is hard.

Visit our [Help Centre](#) or contact us directly—we're here to help.

