# ETG transfers API integration

[**ETG transfers API integration**](#_9m31hy1k25vh) **1**

[General information](#_p1b63he7344x) 2

[Technical requirements](#_mw9dftgyazs8) 3

[Steps of the integration](#_e4hhb7551xc8) 3

[Transfer categories mapping](#_ovv0wol1j8t4) 3

[Backoffice](#_urk6h24af43f) 4

[Additional information about each method](#_a7avvc1o9xkd) 5

[/search](#_ul2d7axoo0cv) 5

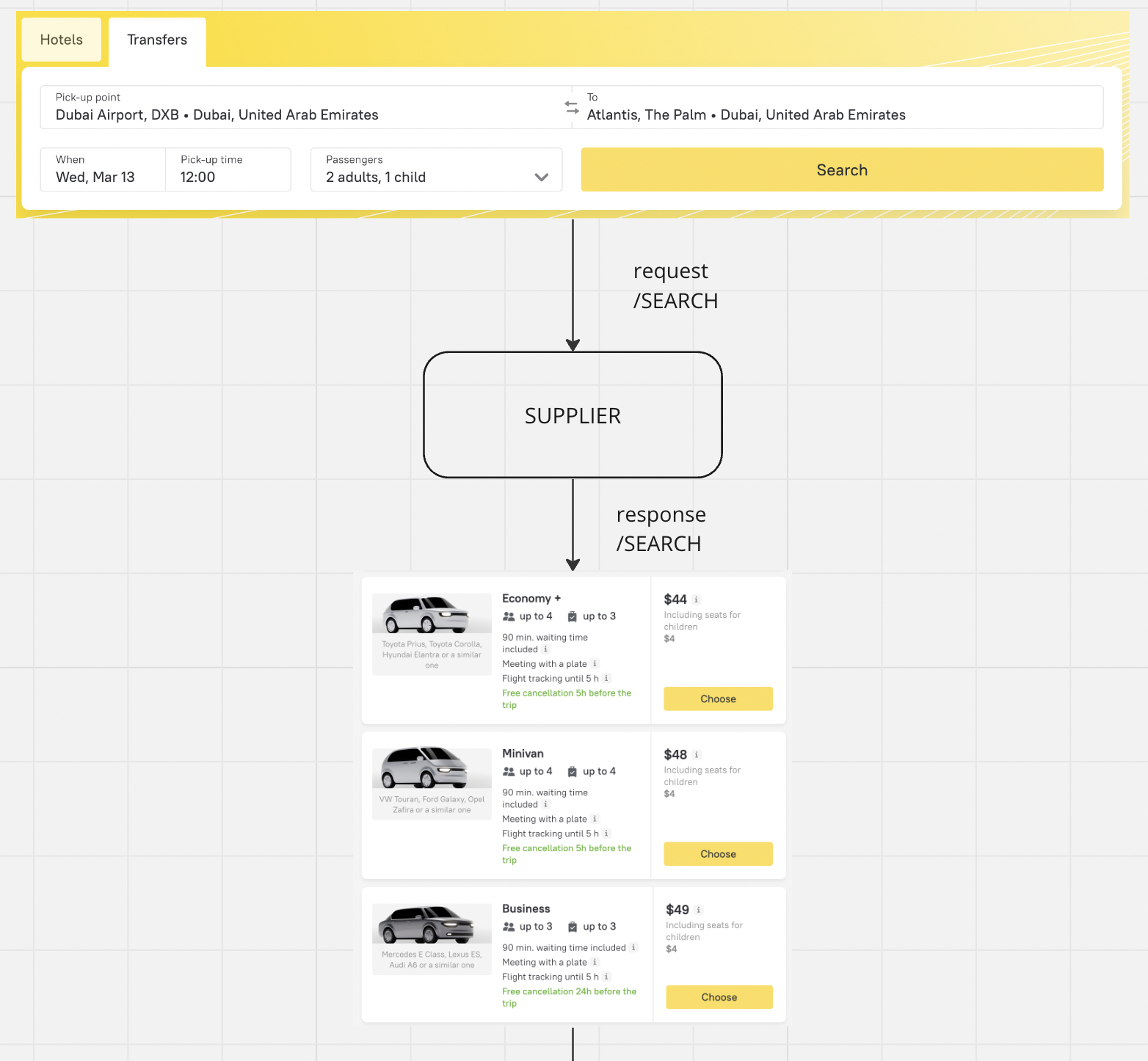
[/book](#_blxn6f16ibxn) 13

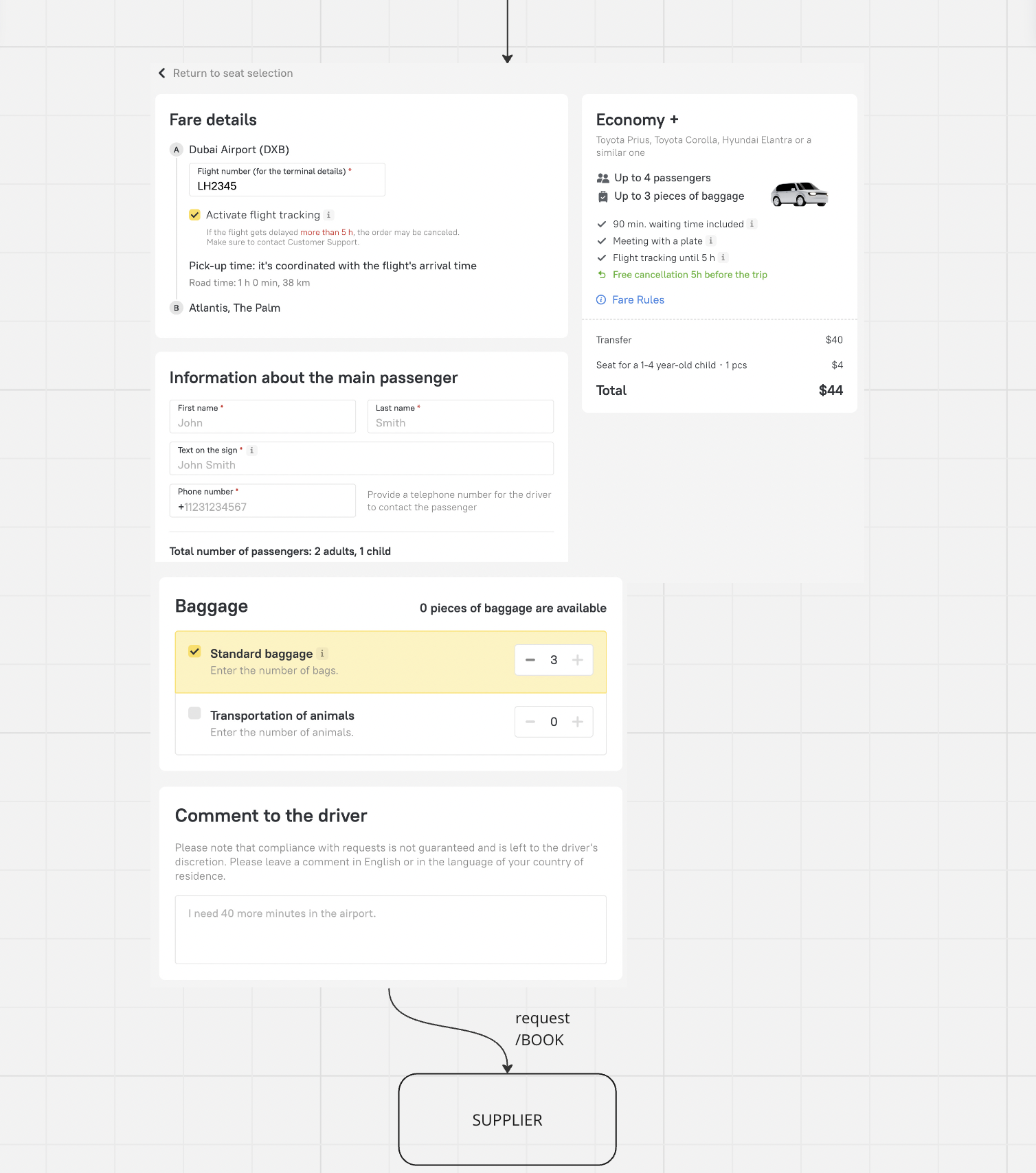
[/status](#_t8nnjrjpcdf) 18

[/cancel](#_3asap8mbq3rb) 20

## General information

Search and book flow:





API documentation <https://openapi.emergingtravel.com/?urls.primaryName=ETG%20Transfers%20API%20v1>

API check up

<https://docs.google.com/document/d/1njoa98aZEe3mmplVOMj_2CbWwlNrJC9raJ1yD9P8DLk/edit?tab=t.0>

Our websites

* <https://www.ratehawk.com/>
* <https://roundtrip.travel/>
* <https://www.zenhotels.com/>

## Technical requirements

1. Your service should be able to handle around 80,000 search requests per day.
2. The response time for the /search method must not exceed 5 seconds for any request. The response time for the /book, /status and /cancel methods must not exceed 30 seconds for any request.

## Test environment requirements

1. Auto-tests must be fully passed on the current test environment.
2. Order availability for 1.5 years from the current date.
3. On search/booking, offers from all available categories should be accessible.
4. There should be offers with all available types of child seats and upsells.
5. Availability of at least 4 routes (IATA -> coordinates, IATA -> IATA, coordinates -> IATA, coordinates -> coordinates).

## Steps of the integration

1) Development on your side

2) Check up (autotests) on your side (+screenshots)

3) You send the credentials to stage and prod connection

4) Our team test corner cases manually and send bug reports

5) Operational questions

6) Launch

## Transfer categories mapping

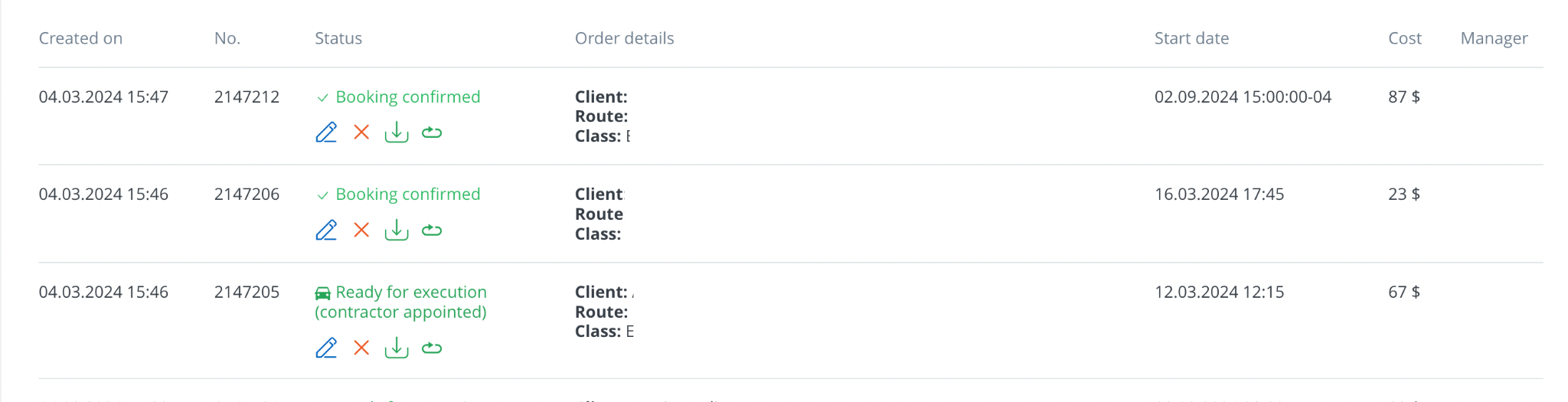
| **API transfer category** | **Interface category name** | **Passengers** | **Models** |
| --- | --- | --- | --- |
| micro | Micro | 2-3 | Opel Corsa, Renault Clio, Skoda Fabia |
| economy | Economy | 3-4 | Toyota Prius, Toyota Corolla, Hyundai Elantra |
| economy\_mpv\_new | MPV | 4 | Mazda MPV, Citroen C4 Picasso, Citroen Grand Tourer, Chevrolet Voyager, Renault Scenic, Mitsubishi Grandis, |
| economy\_mpv | Comfort | 3-4 | VW Passat, Hyundai Sonata, Toyota Camry |
| economy\_van | Minivan | 4-7 | VW Touran, Toyota Previa, Ford Galaxy, Opel Zafira, Toyota Proace, Honda Odyssey,Suzuki Ertiga |
| minibus | Minibus | 7-20 | VW T5, Mercedes Vito, Toyota Hiace, Opel Vivaro, Hyundai H1  Mercedes Sprinter, Toyota Coaster, Ford Transit |
| business | Business | 2-3 | Mercedes E Class, Lexus ES, Audi A6 |
| business\_mpv | Business MPV | 3-4 | Mercedes GL, Chevrolet Suburban, GMC Yukon |
| business\_van | Business Minivan | 4-7 | Mercedes V Class, Mercedes Viano, Toyota Alphard, Mercedes Vito |
| first | First class | 2-3 | Mercedes S Class, Lexus LS, BMW 7 Series |
| bus | Bus | 20-100 | Volvo 9700, Man Lion's city, Mercedes Citaro, Iveco Urbanway, Scania Citywide |
| electro\_micro | Electro Micro | 2-3 | Ora Black Cat, BYD Dolphin, BYD Seagull, Smart For Four Electro |
| electro\_economy | Electro Economy | 3-4 | VW ID3, Nissan Leaf, Ora 03, JAC JS4 Electro |
| electro\_economy\_mpv | Electro MPV | 4-6 | BYD D1, Changan Oshan |
| electro\_comfort | Electro Comfort | 3-4 | Zeekr X, Tesla M3, VW ID4, BMW i3, Polestar 2 |
| electro\_economy\_van | Electro Minivan | 4-7 | Citroen Jumpy SpaceTourer electric, Ford Custom electric, VW ID Buzz |
| electro\_minibus | Electro Minibus | 8-20 | Ford E-Transit Custom, Ford Transit EV Bus, Toyota Joylong E6 |
| electro\_business | Electro Business | 2-3 | Mercedes-Benz EQE, Tesla S, BMW i5 |
| electro\_business\_mpv | Electro Business MPV | 3-4 | Tesla Y, BMW iX, Xiaomi SY7, VW ID6, |
| electro\_business\_van | Electro Business Minivan | 4-7 | Zeekr 009, Mercedes-Benz EQV, Mercedes-Benz eVito, Voyah Dream |
| electro\_first | Electro First class | 2-3 | Mercedes-Benz EQS, Tesla X |
| electro\_bus | Electro Bus | 20-100 | Mercedes-Benz eCitaro, Futian 50 Pure Electric Bus |

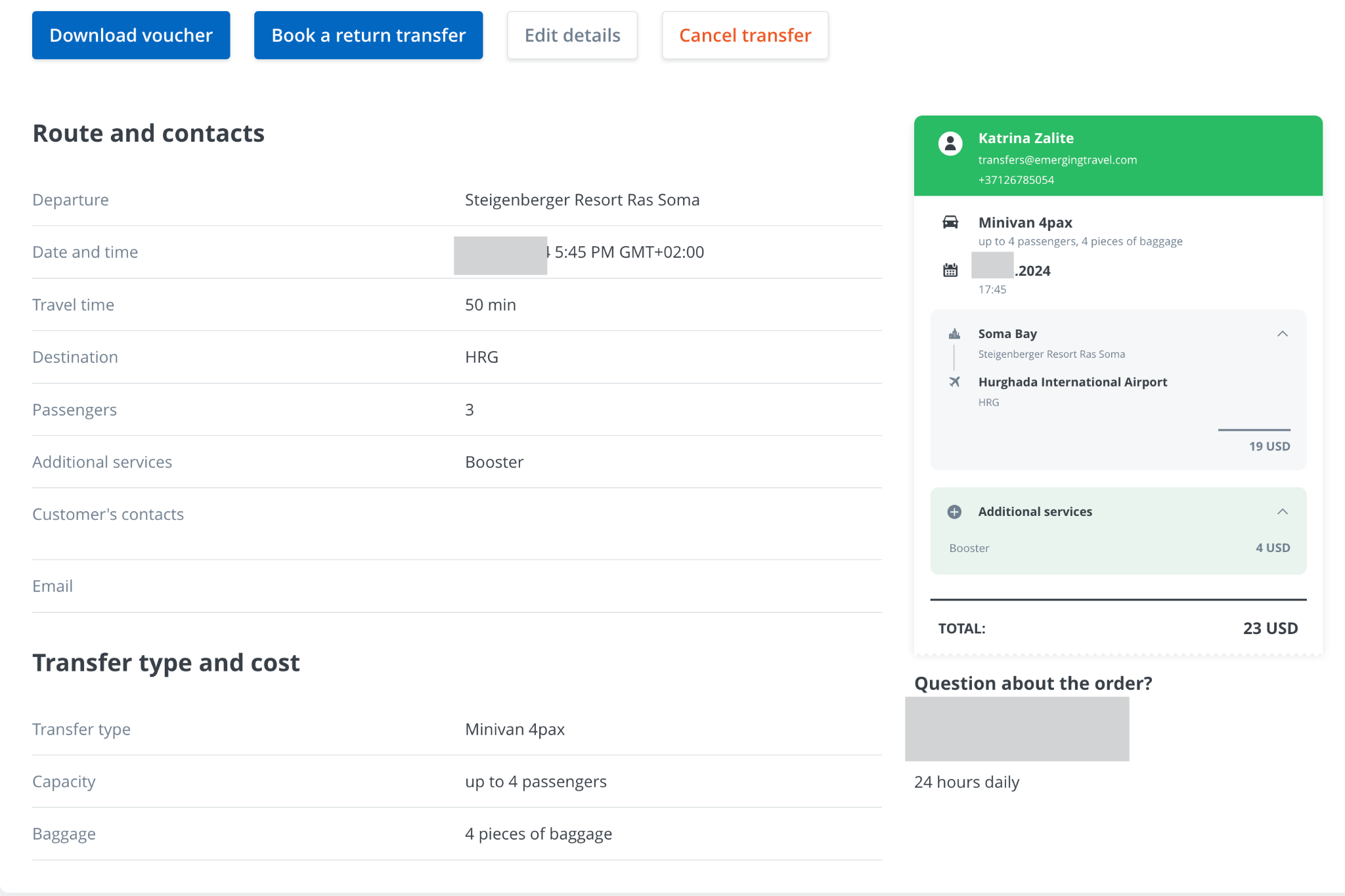
## Backoffice

Our support team works with orders placed by clients through our website. To ensure that our support can effectively manage these orders, it is necessary for you to organize access to the personal account section of your website, where we can view our orders. Our support performs the following actions with the orders:

* Viewing the itinerary, travel details (number of passengers, price, time and date of departure, and additional services)
* Order cancellation
* Free or paid order modification

Example of the backoffice:





## Additional information about each method

### /search

1) There should be only offers in /search response that match all the search query. E.g., these are transfers suitable for the requested number of passengers with specific child seats available for booking.

2) There should be only offers in /search response that can be guaranteed booked and confirmed during a booking.

E.g., if a transfer cannot be booked less than 24 hours before the trip then such an offer should not be included in a /search response when a transfer is requested 6 hours before the trip.

If a transfer cannot be confirmed during a booking on the supplier's side and requires additional approval for some reason then such an offer should not be included in a/search response.

3) Offers should not be duplicated in a /search response and there also should not be several offers of one category with the same capacity.

E.g., there can be two offers of ‘minibus’ category, one of which is 19pax and another one is 18pax. But no need to send two 19pax minibuses in response.

4) The local time of the pick-up point is always specified without indicating the time zone in the field "start\_date\_time" in the /search request. The date/time is always sent in this format: "2023-07-15T17:00:00Z", with the "Z" symbol at the end of the value. (warning) However in our case "Z" does not indicate the zero time zone. In the response, the "start\_date\_time" field should return exactly the same date/time as indicated in the request but with the correct time zone for the pick-up location.

There is no need to convert the time sent in our request according to the correct time zone of the pick-up location. E.g., if the date/time "2023-07-15T17:00:00Z" is sent in the request and the pick-up location is in Türkiye where the time offset is +3 then the "start\_date\_time" field in the response should contain the value "2023-07-15T17:00:00+03:00".

5) A total number of passengers (including children) is sent in the "passengers" field in request.

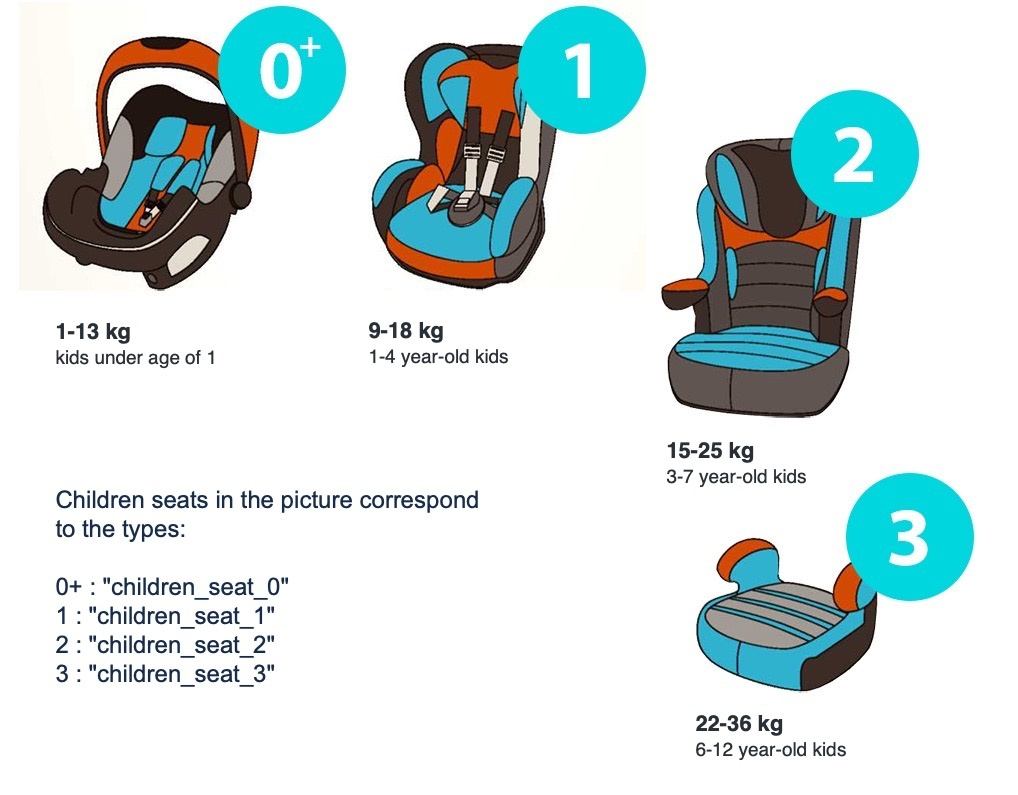
6) Required number of child seats is sent in fields "children\_seat\_0", "children\_seat\_1", "children\_seat\_2", "children\_seat\_3". There should be only offers in response where specified type and number of child seats are available and guaranteed for ordering. If there are no such offers the response should contain an empty array of offers. An example of a response with an empty array:

| {  "start\_date\_time": "2023-11-26T09:00:00+03:00",  "offers": []  } |
| --- |

E.g., if a supplier does not provide child seats of type 0 (children\_seat\_0) at all and when "children\_seat\_0" != 0 in /search request then no need to send any offers in response. Or if a supplier does not provide more than 2 any child seats in any offer and more than 2 child seats are requested in /search then no need to send any offers in response.

*Instructions for mapping children seats provided by supplier with the seats that are presented on our website:*

The types of children seats with the appropriate age groups and weight are shown in the picture:



If the child seats provided by the supplier do not fully correspond to the description indicated in the picture, and the supplier provides a mixed type or convertible type of child seat that covers the weight and age ranges of several child seat types indicated in the picture, it is acceptable to provide them to us, as long as this type of child seat is officially certified and the supplier can guarantee it will be suitable for the weight/age of the child mentioned in the request.

For example, if a supplier provides one convertible type of child seat for children weighing 9-25 kg and aged 0-7 years, this seat's characteristics cover 2 types of seats by weight and age: "children\_seats\_1" and "children\_seat\_2." Thus, if "children\_seats\_1" and "children\_seat\_2" are requested in the /search (with "children\_seat\_1" value != 0 and "children\_seat\_2" value != 0), then offers should be returned in the response.

In this case, the supplier must not relate the provided mixed/convertible type of seat to just one particular type of seat indicated in the picture. In the current example, the fields "children\_seats\_1" and "children\_seat\_2" in the response should have values equal to those in the request ("children\_seat\_1" != 0 and "children\_seat\_2" != 0), and upsells’ values should be returned accordingly (see point 17).

7) Any scripts (Latin, Cyrillic, Arabic etc.), diacritics, special symbols, digits can be sent in the start\_point.address and end\_point.address fields on request. All values in the fields above should be read correctly and there should not be an error in response.

8) Correctly calculated distance and duration of trip should be sent in the "distance" and "estimated\_duration\_minutes" fields in response.

9) Values in the "transfer\_category" field should correspond to the categories list specified in the documentation.

E.g., a ‘Comfort’ category should have the value "economy\_mpv" etc.

10) The value in the "car\_model" field should correspond to the category of an offer. For example, the ‘Business’ category may include such car models as the Mercedes E-Class, Lexus ES, or Audi A6, but not the Toyota Prius or Toyota Hiace.

You may refer to our list of approximate car models that correspond to all transfer categories (see the chapter "Transfer Categories Mapping"). You may use some or all of the car models from that list, or you may propose your own variants.

Please ensure that both the model and the brand of the car are included in the "car\_model" field. For instance, values like "Toyota", "Mercedes", "Corolla" or "V Class" alone are not acceptable. Correct values will be "Toyota Corolla" or "Mercedes V Class"

The value must not contain phrases such as "or similar," "similar," "possible," or any other similar text. The indication of the approximation of car models is managed by ETG. Only specific car model names or a list of car models should be provided in the "car\_model" field.

A value must be sent in Latin script.

A value shouldn’t be sent in uppercase or lowercase only. E.g., "TOYOTA COROLLA, HYUNDAI ELANTRA" or "mercedes v class" are not appropriate. The correct ones will be "Toyota Corolla, Hyundai Elantra" or "Mercedes V Class".

Several examples of appropriate values: "Mercedes S Class, Lexus LS, BMW 7 Series" or "Mercedes Vito", etc.

11) Correct free waiting time should be sent in the "included\_waiting\_time\_minutes" field in response. Waiting time cannot be equal to 0.

12) Correct available luggage places number should be sent in the "luggage\_places" field in response. Luggage places number cannot be equal to 0.

13) Correct available passengers’ seats number should be sent in the "seats" field in response.

Values in the "seats" field should be more or equal (>=) to values of the "passengers" field in request.

E.g., when searching a transfer for 5 passengers only offers with "seats" >= 5 should be sent in response. If transfers for such a number of people are not available then an empty array of offers [] should be sent in response.

14) The "children\_seat\_0", "children\_seat\_1", "children\_seat\_2", "children\_seat\_3" fields should have the values that are specified in the corresponding fields in the request.

E.g., if 0 values were specified in these fields in the request then they should also be 0 in the response (or the fields can be omitted).

If "children\_seat\_0": 1 was specified in the request and such a transfer can be ordered then there should be an offer in the response and the value 1 should be specified in the "children\_seat\_0" field (if such a transfer is not available for ordering an empty array of offers should be in response).

15) A /search request contains the exact coordinates + name of the point/airport code of the route. It also specifies whether a child seat is required (when it is specified). A /search response should include offers with an accurately calculated and fixed price for the specified route, taking into account child seats (if requested). The booking is then made with the same route and set of child seats so the booking price should not change relative to the price in a /search response. The price in the price.amount field cannot be zero.

E.g., if a request is made for a transfer with a specified child seat, and the transfer itself costs 20 EUR while the seat costs 5 EUR then these 5 EUR should be included in the total cost and the price.amount field should have a value of 25. If a seat is available for booking but is not specified in a /search request then the price.amount field should only show the price for the transfer itself (which is 20). The child seat should not be included in price.

16) The price. Сurrency field should contain the correct currency. It should be the currency mentioned in the agreement.

17) A /search response may contain additional services in the upsells array that can be added to an order. Upsells are children seats of all types as well as, for example, a bottled water, an additional stop, an English-speaking drive etc.

Special note: unfortunately at the moment a service of meeting with a sign cannot be processed as an upsell. In case meeting with a sign is a service requiring extra charge, this extra charge should be just included in the total price of every offer at the /search step.

The upsells.id field specifies the internal id of an upsell on the supplier side, which will be then used when adding to the order at the booking stage.

The upsells.price field indicates the price per one piece of this upsell. E.g., if 2 seats of type "children\_seat\_1" have been added to the order then the upsells.price field for the upsell with type=children\_seat\_1 anyway specifies the price per one seat. The total amount for two seats should be added to the total cost which is specified in the price.amount field. The upsells.price value may be zero (meaning that the upsell is free of charge).

The upsells.type field specifies the type of an upsell. In the case of children seats only the types specified in the documentation are allowed: children\_seat\_0, children\_seat\_1, children\_seat\_2, children\_seat\_3 (corresponding to the fields of the seats sent in the request). There are no restrictions on the names of other upsells but the value must reflect the essence of the upsell.

The value in the upsells.count field is the number of already included upsells in the order. At the search stage only upsells of children seats (with type=children\_seat\_0 / children\_seat\_1 / children\_seat\_2 / children\_seat\_3) may be included in the offer and only if they were requested for the order (that is, the value != 0 in a /search request). In this case the upsells.count field should contain the value that was specified in a /search request for this seat. If the seat was not specified in the request (value == 0) then the upsells.count of an upsell of this seat should also be zero. The upsells.count value should always be zero for all other upsells (with type != children\_seat\_0 / children\_seat\_1 / children\_seat\_2 / children\_seat\_3).

If a children seat of any type was specified in a /search request (the value in the children\_seat\_0 / children\_seat\_1 / children\_seat\_2 / children\_seat\_3 fields != 0) then the upsell with the specified children seat must be in the offer. E.g., if there are "children\_seat\_0": 1, "children\_seat\_1": 2 in request then two upsells must be included in the offer: 1) with "type": "children\_seat\_0" + "count": 1 and 2) with "type": "children\_seat\_1" + "count": 2.

If the seat was not requested in /search there may be no upsell for that seat.

The upsells.max\_count field contains the maximum number of available upsells of each type. The value in the field cannot be zero (this would mean that this service is not available for order and in this case such an upsell should not be sent at all). For upsells with type = children\_seat\_0 / children\_seat\_1 / children\_seat\_2 / children\_seat\_3 upsells.max\_count value cannot be less than upsells.count value and it cannot be more than value “offers.seats – 1” of current offer.

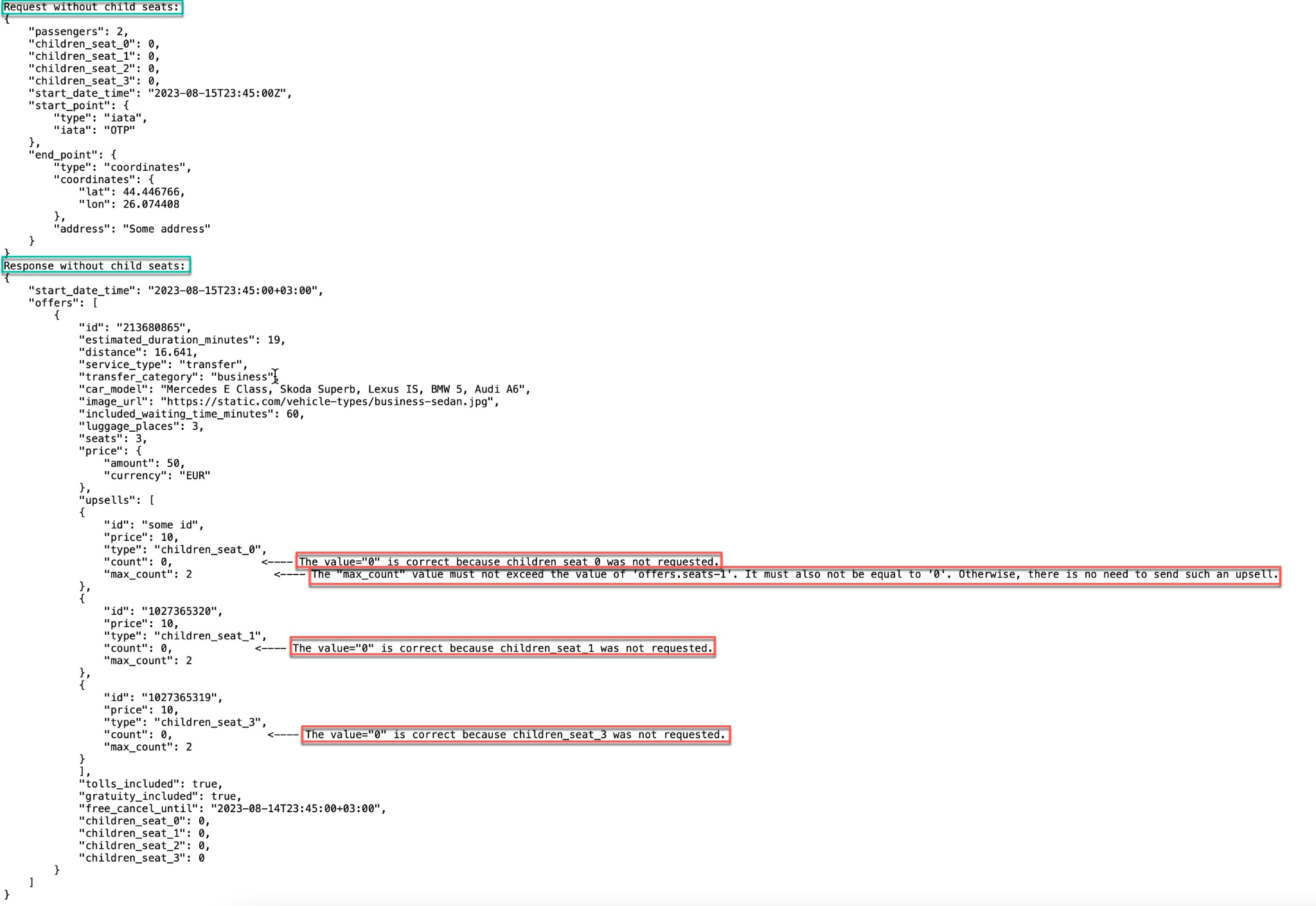
E.g., if the car capacity (offers.seats) == 3 then the upsells.max\_count of the seat should be > 0 and <= 2.

Several examples of how to send upsells:

(1) A request *with* a child seat.



(2) A request without a child seat.



18) The "tolls\_included" field indicates whether tolls are included in the transfer price or not.

19) The "gratuity\_included" field indicates whether gratuity is included in the transfer price or not.

20) The "free\_cancel\_until" field should contain the correct time within which the transfer can be cancelled for free. The correct time zone of the departure location must be specified: it will match the time zone in the "start\_date\_time" field in a current /search response, except for cases when the trip start date/time and the free cancellation date/time are in different time zones (this is possible in places with daylight saving time changes if the trip takes place during the period around the shifting from one time zone to another).

21) If there are no transfers available for the search query for any reason the /search response should contain an empty array of offers ('offers': []). No error should be returned in the response.  
Example of how the response should look like when there are no offers for the query:

| {  "start\_date\_time": "2023-11-26T09:00:00+03:00",  "offers": []  } |
| --- |

Please note that the field "start\_date\_time" remains mandatory in the response with an empty array of offers and the value in the field must correspond to the format specified in the documentation. Also, despite the absence of offers, the correct time zone of the pick-up location should be sent in the "start\_date\_time" field.

Pay attention to examples of reasons why transfers for a search query may not be available in order to process such cases correctly:

1. There is no coverage in the requested region.
2. There are no offers for a specific route although there is coverage in the requested region.
3. There are no offers for specific trip dates. E.g., it is a request for a date in 2 years from current date (there is no need to set the maximum allowable search date for a response to our request)
4. There are no offers for the "now" time (if the trip is scheduled for the moment of the search request) or the minimum allowable time to order a transfer for a certain offer has expired.
5. A /search request was made for a "past time" at the departure location. On our website on the search form the nearest valid time for ordering a transfer is determined by the local time of the user's browser, and not by the time of the particular transfer departure location (at this stage we do not have information about the current time at the departure location). Thus in fact the user has the opportunity to order a transfer for a time that is still a valid future time at his own location, but it is already the past for the trip place. Such user behavior is acceptable on our website and sometimes users actually try to find a transfer for the past time by mistake.
6. There are no transfers available for the specified number of passengers (e.g., there is a request for 19 passengers but the supplier does not have minibuses available) (no need to set a maximum allowable number of passengers for our /search response or at least a maximum allowable number should be large).
7. There are no transfers with the specified in the /search request number of child seats.
8. There are no transfers with the specified in the /search request type of child seats.
9. An incorrect route (e.g., IST -> LAX)
10. A route between bordering countries is not allowed.
11. etc.

In **all these** cases, as well as in similar ones, please do not send an error in the response. We expect an empty array of offers.

An error in the response is allowed if some technical server failure or unexpected situation has occurred.

If the /search request from our side does not comply with the documentation an error in the response should be also returned (e.g., the request contains an empty string in the "start\_date\_time" field or the number of "passengers" == 0 in the request).

The error response format must comply with the documentation. Both the "code" and "error" fields are required in the error response and cannot have a *null* value.

A description of an error reason is provided in the "error" field. An error reason must be described correctly and be understandable.

E.g., if there is an empty string in the "start\_date\_time" field in the /search request (which is an invalid value), then the error text in the response should indicate that the error is caused by an invalid value in the "start\_date\_time" field (e.g., message "start\_date\_time is required").

### /book

1) The offer received from the /search response must be guaranteed for booking and confirmation, except in cases of booking errors due to unexpected technical problems (for example, server errors).

E.g. if it is not possible to book non-refundable fares on the supplier side and an error is returned in the /book response when attempting to book such an offer, then the search should not include such offers in the /search response if they have already become non-refundable at the time the search is requested.

2) A lifespan of an offer should be a minimum of 24 hours from the time it was returned in the /search response. During this period, there should not be a recalculation of an offer’s price, and an offer should be available for booking.

Please note that if the search was made less than 24 hours before the start of the trip, then the offer's lifespan will be the time from the moment of the search until the latest time the order can be placed before. E.g., if the minimum time to place an order is 6 hours before the start of the trip, and the search was made 10 hours before the start of the trip, then an offer’s lifespan will be 4 hours. Therefore, in this case, an attempt to place an order after 5 hours from the time of the search will fail, and this is an expected behavior.

3) Any scripts (Latin, Cyrillic, Arabic etc.), diacritics, special symbols, digits can be sent in the start\_point.address and end\_point.address fields in a /book request. All values in these fields should be read correctly and there should not be an error in response. These fields contain the exact location from where the passenger needs to be picked up/dropped off (e.g., the name of a hotel or a train station). This information should also be transmitted to the driver (in addition to the coordinates specified in /search and /book requests).

4) The "supplier\_link" field in a /book response should contain a working link to the booked order in our account on the supplier website (not to a list of all orders from our account).

5) The "order\_id" field should contain a unique identifier for each order made in a supplier's system. The order number sent in the "order\_id" field during the /book step must not be equal to the offer ID sent in the offers.id field during the /search step. It should be designed to be concise and easy to communicate across various channels, including verbal, written, and digital formats, without confusion or misinterpretation. The order number can consist of letters, digits, and some special characters (such as a hyphen) if needed, but it is not mandatory to use all of them at once. The total length should not exceed 15 characters. Please avoid overly complex patterns that could hinder communication.

Examples of acceptable "order\_id" value: "5345362", "7KBNS-4".

Examples of unacceptable "order\_id" value: "3de5268a-3e36-4185-9472-9f77e94ce233", "V015202409262342318760001", "123456789012345678".

6) The "start\_time" field value should match the value of the "start\_date\_time" field from a /search response (i.e., it is the requested date/time of the trip with the time zone of the pick-up location).

7) The values in the "distance" and "estimated\_duration\_minutes" fields should match the values of the corresponding fields for this offer in a /search response.

8) The value in the "included\_waiting\_time\_minutes" field must match the value of the corresponding field for this offer in a /search response.

9) The value in the "passengers" field must match the value of the corresponding field in a /book request. This is the number of passengers for which the transfer is booked (not the total number of seats available in the booked transfer).

10) The value in the "luggage\_places" field must match the value of the corresponding field in a /book request. This is the number of booked luggage places (not the total number of luggage places available in the booked transfer).

11) The value in the "sport\_luggage\_places" field must match the value of the corresponding field in a /book request.

If the "sport\_luggage\_places" field value != 0 in a /book request then this field must be present in the response with the same value as specified in a /book request.

If the "sport\_luggage\_places" field value == 0 in a /book request then this field may not be present in the response and if there is the field its value must also be 0.

12) The value in the "animals" field must match the value in the "animals\_places" field in a /book request.

If the "animals\_places" field value != 0 in a /book request then the "animals" field must be present in the response with the same value as specified in a /book request.

If the "animals\_places" field value == 0 in a /book request then the "animals" field may not be present in the response and if there is the field its value must also be 0.

13) The value in the "wheelchairs\_places" field must match the value of the corresponding field in a /book request.

If the "wheelchairs\_places" field value != 0 in a /book request then the field must be present in the response with the same value as specified in a /book request.

If the "wheelchairs\_places" field value == 0 in a /book request then the field may not be present in the response and if there is the field its value must also be 0.

14) The values in the fields "children\_seat\_0", "children\_seat\_1", "children\_seat\_2", "children\_seat\_3" must match the values of the corresponding fields in a /book request.

If a non-zero value is specified in these fields in a /book request then the fields must be included in a /book response and have the same value as specified in the request. If a value of 0 is specified in these fields in a /book request, then the fields may not be present in a /book response and if there are the fields their value must also be 0.

15) The "comment" field value must totally match the value of the corresponding field in a /book request (it should not be cut or returned in Unicode, no information from a supplier's side should be added etc.). Any scripts (Latin, Cyrillic, Arabic etc.), diacritics, special symbols, digits can be sent in this field in a /book request. All values must be read correctly and the value should not be a reason for an error.

16) The "flight\_number" field value in a /book response must match the value of the corresponding field in a /book request. The value "No flight" is always sent in the "flight\_number" field in a /book request for trips not from an airport (start\_point.type != iata).

For trips from an airport (start\_point.type == iata) the value "No flight" or a flight number (e.g., "LH1234") may be sent.

You can find more detailed information on how flight tracking works in the section “Description of available flight tracking options”.

17) The "shield\_text" field value must totally match the value of the corresponding field in a /book request (it should not be cut or returned in Unicode, no information from a supplier's side should be added, etc.). Any scripts (Latin, Cyrillic, Arabic etc.), diacritics, special symbols, digits can be sent in this field in a /book request. All values must be read correctly and the value should not be a reason for an error.

18) The value in the "price.amount" field in a /book response must match the value of the corresponding field in the /search response for this offer. Since the exact coordinates + name of the point/airport code of the route are specified in a /search request the price for the specified route should be correctly calculated during the search stage. The same points are sent in a /book request so the price should not be recalculated during the booking stage.

Child seats also should be taken into account during the search stage and the price should not change during the booking stage because of presence of child seats in the order (because the same seats are booked in the same quantity as specified in the /search request). The price in the price.amount field cannot be zero.

19) The value in the price.currency field in a /book response must match the value of the corresponding field in the /search response for this offer. The currency of the offer cannot differ between the search and booking stages.

20) The "upsells" array must be necessarily included in a /book response only if additional services have been booked ("upsells" != [] has been sent in a /book request).

The value in the upsells.id field in a /book response must match the value of the corresponding field of the corresponding upsell in the /book request.

The value in the upsells.type field in a /book response must match the value of the corresponding field in the /search response for the corresponding upsell of this offer. As for children seats’ upsells only the types specified in the documentation are allowed for children seats: "children\_seat\_0", "children\_seat\_1", "children\_seat\_2", "children\_seat\_3" (corresponding to the fields of the seats sent in the request).

The value in the upsells.count field must match the value of the corresponding field of the corresponding upsell in a /book request. The field indicates the number of booked upsells and cannot be equal to 0 (if an upsell is not booked it would not have been sent in a /book request at all).

The value in the upsells.price field in a /book response must match the value of the corresponding field in the /search response for the corresponding upsell of this offer. The field indicates the price per one piece of this upsell. The upsells.price value can be zero (meaning that the upsell is free).

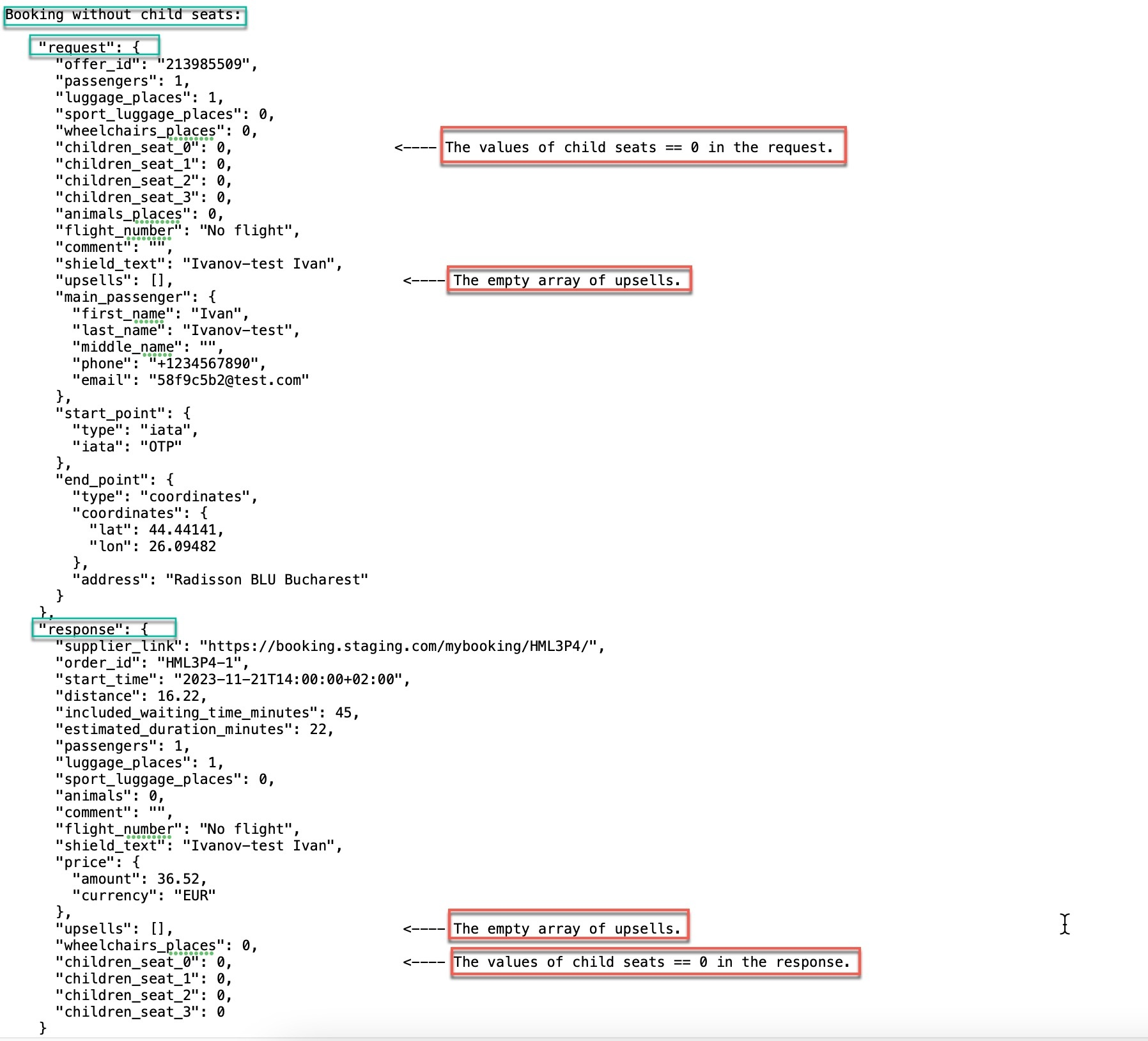
If additional services have not been booked (an empty array "upsells": [] is sent in a /book request), this array must not be included in the /book response. In this case it is only acceptable for the "upsells" array to be completely absent or to have a value of null or [].

Several examples of how to send upsells:

(1) A request with a child seat.



(2) A request without a child seat.



21) The "meeting\_instructions" field may contain additional text instructions for the passenger's meeting point. The value should be in English only. There should not be a mention of the supplier in the value. General information should be provided only.

E.g., the instruction can look like: *"Meeting point is at arrival hall after exit of baggage claim and customs."*

If the supplier does not provide additional text meeting instructions the response may be without this field or this field may contain the value *null* or *"" (empty string)*.

22) If there are pictures indicating the meeting point for the passenger they can be sent in the "meeting\_images" field. The field is an array of strings. Each string of an array can contain one picture.

An example of how the value in the "meeting\_images" field in the /book response should look like:  


If the supplier does not provide mentioned pictures the response may be without this field or this field may contain the value *null* or *[]*.

23) In case of a booking error the error format should be according to the documentation. Both the "code" and "error" fields are required in the error response and cannot have a *null* value.

The reason for an error should be specified in the "error" field. The reason should be described correctly and be understandable.

E.g., if an empty string is sent in the "offer\_id" field in a /book request (which is an invalid value) the error message should indicate that the reason for this error is an invalid value in the "offer\_id" field (e.g. "offer\_id is required").

### /status

1) A /status response should include a current status of an order in the "status" field: completed or cancelled. Any other values are not allowed. Orders do not have storage expiration dates; they should always be accessible.

2) The value in the "order\_id" field in a /status response should match the value of the corresponding field in a /status request.

3) The value in the "start\_time" field may differ from the value of the corresponding field in a /book response (if the date and/or time of the order has changed) but it should be sent in the correct format with the correct time zone of the departure location.

4) The value in the price.amount field represents the actual booking price. It is strictly required that the price of the order is always accurate and remains up-to-date. If no changes have been made to the order or if changes were free of charge, the price must remain the same as it was at the time of booking. However, if any modifications result in a price change (e.g., route changes, car category changes, etc.), the updated price must be immediately reflected in the API response without exception.

The price in the price.amount field cannot be zero.

5) The value in the price.currency field should match the value of the corresponding field for this offer in a /book response. The currency of the offer cannot be changed after booking.

6) If at the time of request of a /status method a driver has not yet been assigned to the order the driver\_info structure may not be included in the response or may be with a null value. However, the structure should not be empty (the value should not be equal to {}).

If there is the "driver\_info" structure in a /status response and it is not null the driver\_info.phone field must be present in the response. It should contain a correct phone number of a driver/carrier company. The field may also contain several numbers separated by commas.

The value in the driver\_info.first\_name field should only be in Latin characters (if there is the value).

The value in the driver\_info.last\_name field should only be in Latin characters (if there is the value).

The value in the driver\_info.carrier\_company\_name field should only be in Latin characters (if there is the value).

7) If at the time of request of a /status method the car has not yet been assigned to the order, the car\_info structure may not be included in the response or may be with a null value. However, the structure should not be empty (the value should not be equal to {}). There also should not be a structure with empty values in mandatory fields.

If there is the "car\_info" structure in a /status response and it is not null the car\_info.car\_model field must be present in the response. We expect in this field a value of one particular car model that will be assigned to the order. Please ensure that both the model and the brand of the car are included in the "car\_model" field. The value in the field should only be in Latin characters.

If there is the "car\_info" structure in a /status response and it is not null the car\_info.plate\_number field must be present in the response.

The value in the car\_info.color field should only be in English (if there is the value). E.g., there should be values like "black", "white" etc.

8) You can provide instructions for the passenger's meeting point or update the instructions that were sent during the booking in the "meeting\_info" structure.

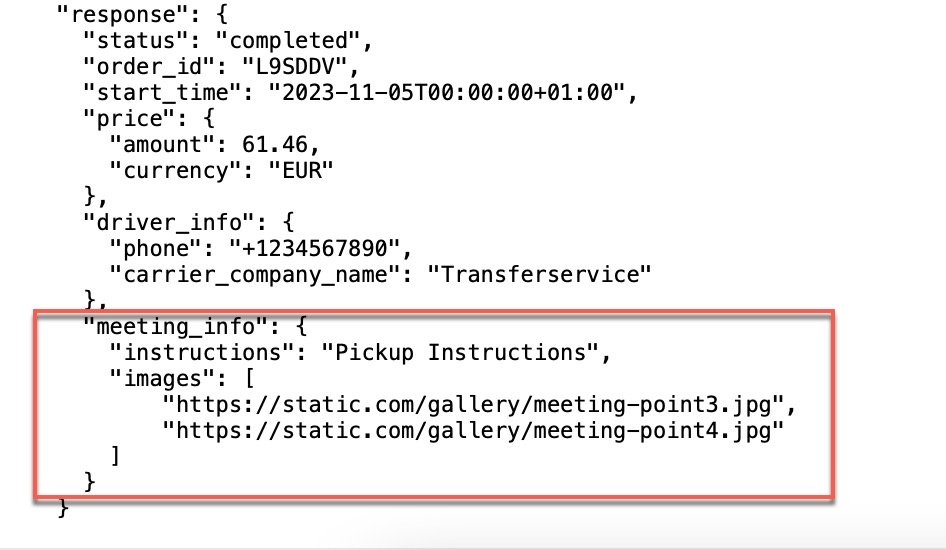
The meeting\_info.instructions field may contain text instructions for the passenger's meeting point. The value should be in English only. There should not be a mention of supplier in the value. General information should be provided only.

E.g., the instructions can look like: *"Meeting point is at the arrival hall after the exit of baggage claim and customs."*

Pictures indicating the passenger's meeting point may be provided in the meeting\_info.images field. The field is an array of strings. Each string of an array can contain one picture.

Please note that the fields with a text and an image of the meeting point instructions should be sent exactly within the "meeting\_info" structure in the /status response (not as separate fields outside the structure as it is in the /book response).

An example of how the value in the "meeting\_info" structure in the /status response should look like:



The "meeting\_info" structure as well as the particular fields inside are not required to send. It may be sent if there is something to send.

9) You can update the main passenger’s details that were sent during the booking in the "main\_passenger" structure. The values in the "main\_passenger.first\_name", "main\_passenger.last\_name", and "main\_passenger.phone\_number" fields must exactly match either the values of the corresponding fields in a /book request (if the original values have not been changed by a user) or the precise new values updated by a customer via a support request. The values should not be truncated or returned in Unicode, and no information from the supplier’s side should be added in these fields. If the "main\_passenger" structure is provided, "main\_passenger.first\_name", "main\_passenger.last\_name" and "main\_passenger.phone\_number" fields are required.

10) You can update the included waiting time that was sent during the booking in the "included\_waiting\_time\_minutes" field. The value in the "included\_waiting\_time\_minutes" field must exactly match either the value of the corresponding field in a /book response or the new value that could be updated due to changes in the originally booked route or transfer category requested by a customer via support. The value must be greater than 0.

11) You can update the number of passengers that was sent during the booking in the "passengers" field. The value in the "passengers" field must exactly match either the value of the corresponding field in a /book request (if the originally booked passengers number wasn't updated by a user) or the precise passengers number updated by a customer via a support request. This is the number of passengers for which the transfer was booked (not the total number of seats available in the booked transfer). If provided, the value must be greater than 0.

12) You can update the number of booked luggage places that was sent during the booking in the "luggage\_places" field. The value in the "luggage\_places" field must exactly match either the value of the corresponding field in a /book request (if the originally booked luggage places number wasn't updated by a user) or the precise luggage places number updated by a customer via a support request. This is the number of booked luggage places (not the total number of luggage places available in the booked transfer).

13) You can update the number of booked sport luggage places that was sent during the booking in the "sport\_luggage\_places" field. The value in the "sport\_luggage\_places" field must exactly match either the value of the corresponding field in a /book request (if the originally booked sport luggage places number wasn't updated by a user) or the precise sport luggage places number updated by a customer via a support request.

14) You can update the number of animal places that was sent during the booking in the "animals" field. The value in the "animals" field must exactly match either the value of the corresponding field in a /book request (if the originally booked animals number wasn't updated by a user) or the precise animals number updated by a customer via a support request.

15) You can update the comment that was sent during the booking in the "comment" field. The "comment" field value must exactly match either the value of the corresponding field in a /book request (if the original comment wasn't updated by a user) or the precise value updated by a customer via a support request. It should not be truncated or returned in Unicode, and no information from the supplier’s side should be added. This field must support any script (Latin, Cyrillic, Arabic, etc.), diacritics, special symbols, and digits.

16) You can update the shield text that was sent during the booking in the "shield\_text" field. The "shield\_text" field value must exactly match either the value of the corresponding field in a /book request (if the original shield text wasn't updated by a user) or the precise value updated by a customer via a support request. It should not be truncated or returned in Unicode, and no information from the supplier’s side should be added. This field must support any script (Latin, Cyrillic, Arabic, etc.), diacritics, special symbols, and digits.

17) You can update the transfer category that was sent during the booking in the "transfer\_category" field. If provided, values should correspond to the categories list specified in the documentation.

E.g., a ‘Comfort’ category should have the value "economy\_mpv" etc.

18) You can update the time within which the transfer can be canceled for free that was sent during the search in the "free\_cancel\_until" field. If provided, the value should be represented in the RFC 3339 format, and be between the current time and the new start time of the transfer, otherwise it will be ignored. The correct time zone of the departure location must be specified.

19) In case of an error in /status the error format should be according to the documentation. A reason for an error should be specified in the "error" field. The reason should be described correctly and be understandable.

### /cancel

1. The order should be cancelled successfully even if it is non-refundable.
2. The order should be cancelled successfully even if a carrier/car has already been assigned.
3. In case of free cancellation the value in the penalty.amount field should be equal to 0.

When canceling a non-refundable order the penalty for cancellation should be specified in the penalty.amount field. In this case the value in the penalty.amount field should not exceed the order price (i.e., it should not exceed the price.amount).

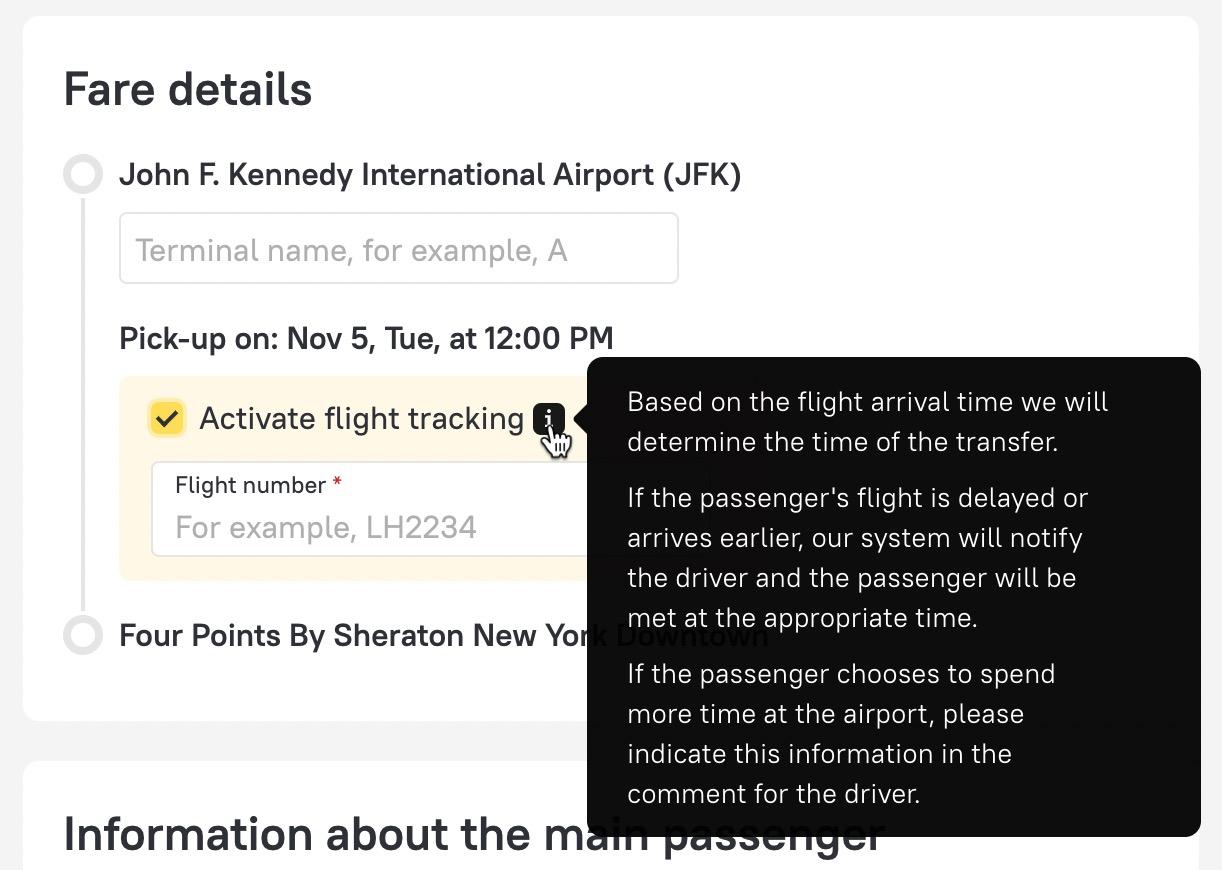
1. The value in the penalty.currency field should match the value in the price.currency field for this offer in /book response. The offer currency cannot be changed after booking.
2. In case of cancellation error, the error message format should comply with the documentation. The reason for an error should be specified in the "error" field. The reason should be described correctly and be understandable.

## Description of available flight tracking options

### Full Flight Tracking

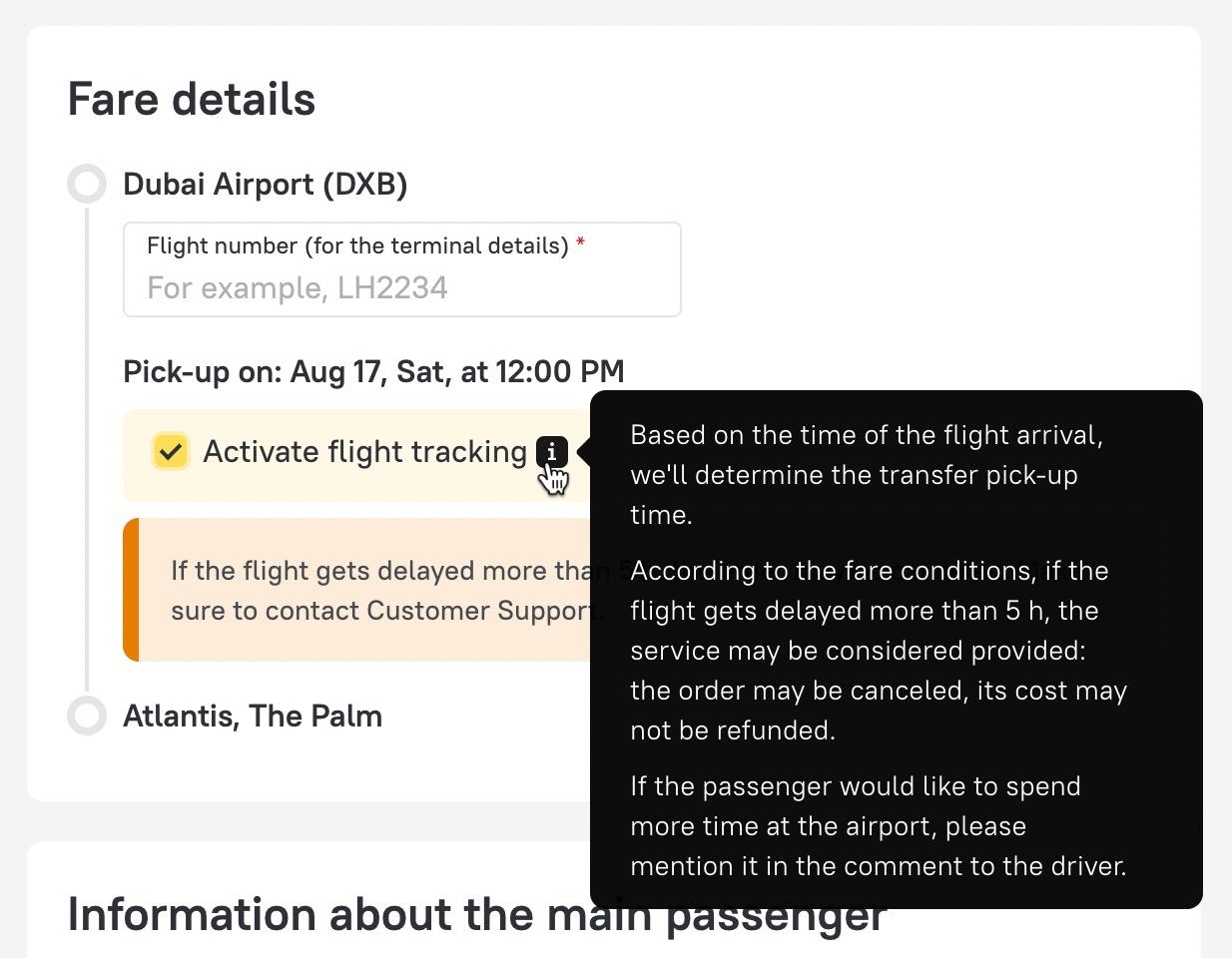
When a client enables flight tracking on the booking form, they expect that the transfer will arrive at the time of the flight's actual arrival, regardless of whether the flight arrives early or late in comparison to the schedule, and regardless of how much the plane was delayed or arrived early. In this scenario, the flight number will be sent in the "flight\_number" field of the /book request (for example, "flight\_number": "LH1234"). If the client wishes to specify a terminal for the meeting, they can send this information in the comment field with the detail /airport terminal: [text] (terminal information is not mandatory for the user to fill out).

If the client declines flight tracking and wants the transfer to be provided exactly at the time they specified, we will send "flight\_number": "No flight" in the /book request.

How the option with full tracking is displayed on the site:

### Partial Flight Tracking

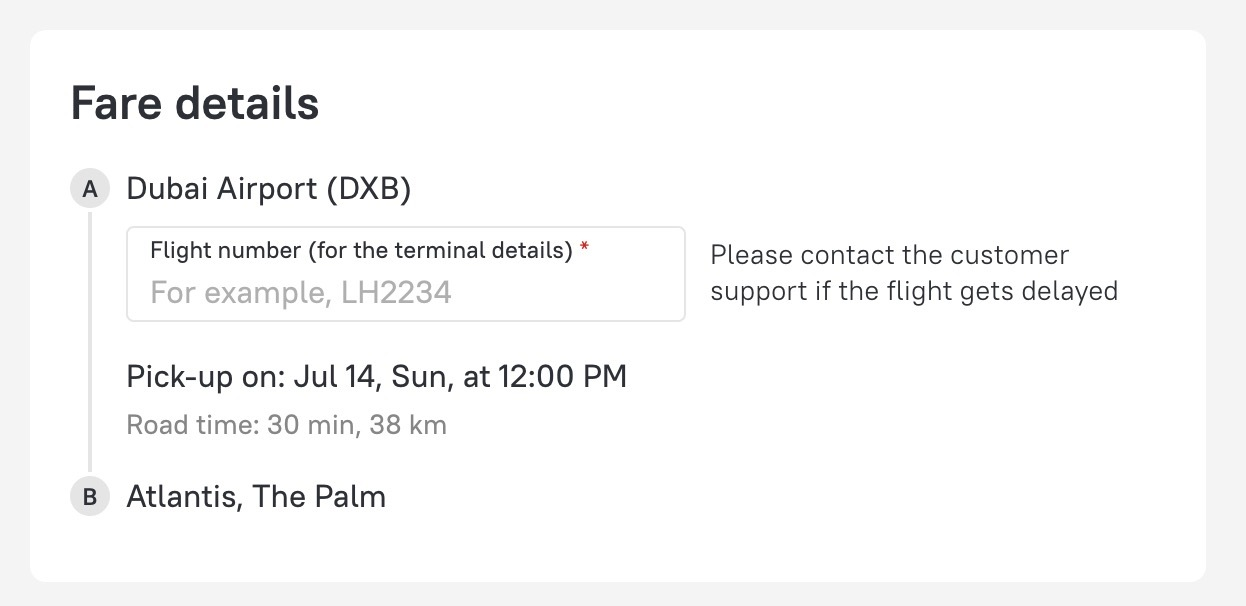
The client can enable or disable this option just like with full tracking.

If the client enables flight tracking on the booking form, they expect the transfer to arrive at the actual arrival time of the flight. However, there is a limitation: tracking is only carried out within the supplier-provided time window (the time is specified by the supplier), and this is shown to the client on the booking form. With this type of tracking, in the /book request, the "flight\_number" field will always contain the flight number (for example, "flight\_number": "LH1234") regardless of whether the client has enabled or disabled flight tracking. Whether the client has enabled flight tracking can be understood by an indicator: in the case of DISABLED partial tracking, the text “NO FLIGHT TRACKING” will be received in the 'comment' field; in the case of ENABLED partial tracking, there will NOT be any “NO FLIGHT TRACKING” text in the 'comment' field. How partial flight tracking is displayed on the site:

### Completely Without Flight Tracking

On the booking form, the client does not have the option to enable/disable flight tracking, but they must still provide the flight number to clarify the terminal (the supplier may use this information but is not obliged to). In this case, the flight number will always be sent in the "flight\_number" field of the /book request (for example, "flight\_number": "LH1234"). However, the presence of the flight number in the field does not imply that we expect flight tracking.

How it looks on the site when no flight tracking is available from the supplier:



It should be noted that when selecting a particular option for flight tracking, we will be sending accordingly the values in certain fields in the /book request, and it's necessary to be able to process them in such a way. There is no separate additional field to indicate the enabling/disabling of flight tracking.

### Non-airport pick up point

On the booking form, the client does not have the option to mention a flight number and to enable/disable flight tracking. In this case, the value "No flight" will always be sent in the "flight\_number" field of the /book request. How it looks on the site when the trip is not from an airport:

