

Submit Amendment : (Full Refund)

This is used to apply an amendment on a booking. After applying, an “amendmentId” will be returned in response which will be used to check/poll the status of the amendment.

Note - Please note that using the below checklist in the remarks would help automate the process of full refunds.

1. Flight Cancelled by Airline
2. Airline rescheduled flight, revised timings are not suitable
3. Already cancelled by directly contacting airline customer support team
4. Airline confirmed, refund is already processed
5. Refund under DGCA policy
6. Personal loss or bereavement
7. Passenger is medically unfit for travel
8. Refund under empowerment policy

URL: <https://apitest.tripjack.com/oms/v1/air/amendment/submit-amendment>

Method: POST

Sample Payloads : [Click Here to take examples.](#)

Request Structure :

- **bookingId**
- **type** - FULL_REFUND
- **remarks**- Mandatory to send some remarks to submit an amendment.
- **trips** : ArrayList of trips for which amendment is to be raised
 - ◆ **src** - departure airport code
 - ◆ **dest** - arrival airport code
 - ◆ **departureDate** - Departure travel date(YYYY-MM-DD)
 - ◆ **travellers** - list of travellers from a particular trip in a booking for which amendment is to be raised.
 - **fn** - first name of traveller
 - **ln** - last name of traveller

Submit amendment request for complete booking: Submitting only the booking id with remark will apply amendment on the complete trip.

```
1  {
2    "bookingId": "XXXXXXXX",
3    "type": "FULL_REFUND",
4    "remarks": "YYYYYYYYYYYYY"
5  }
```

Submit amendment request example in case of DGCA -

```
1  {
2    "bookingId": "TJS10450000651327",
3    "type": "FULL_REFUND",
4    "remarks": "Refund under DGCA policy"
5  }
```

In case of DGCA, add the string “Refund under DGCA policy” in the remarks of the request to automate the process of refund.

Submit amendment request for some selected travellers : Submit travellers details along with their trip details. The amendment will be applicable only on the given trip and given travellers.

```
1  {
2    "bookingId": "XXXXXXXXXX",
3    "type": "FULL_REFUND",
4    "remarks": "YYYYYYYYYY",
5    "trips": [
6      {
7        "src": "AAA",
8        "dest": "BBB",
9        "departureDate": "YYYY-MM-DD",
10       "travellers": [
11         {
12           "fn": "XXXXXXXX",
13           "ln": "YYYYYYYY"
14         }
15       ]
16     },
17     {
18       "src": "BBB",
19       "dest": "CCC",
20       "departureDate": "YYYY-MM-DD",
21       "travellers": [
22         {
23           "fn": "XXXXXXXX",
24           "ln": "YYYYYYYY"
25         }
26       ]
27     }
28   ]
29 }
```

Submit amendment request for a particular trip: If amendment needs to be submitted for all pax types in a particular trip then submit trip details without any traveller details.

```

1  {
2    "bookingId": "XXXXXXXXXX",
3    "type": "FULL_REFUND",
4    "remarks": "YYYYYYYYYY",
5    "trips": [
6      {
7        "src": "AAA",
8        "dest": "BBB",
9        "departureDate": "YYYY-MM-DD",
10       "travellers": [
11         {
12           "fn": "XXXXXXXX",
13           "ln": "YYYYYYYY"
14         },
15         {
16           "fn": "AAAAAAA",
17           "ln": "BBBBBBBB"
18         }
19       ]
20     }
21   ]
22 }

```

Response Structure :

- **bookingId**
- **amendmentId** - Should be used for polling current amendment status

Sample Response

```

1  {
2    "bookingId": "TJ10890000016409",
3    "amendmentId": "900000016410",
4    "status": {
5      "success": true,
6      "httpStatus": 200
7    }
8  }

```



Void Api DOCUMENTATION

© Copyright 2023, Tripjack (Powered By Atlas Travels Online).

This document, which contains confidential material, is private and confidential and is the property and copyright of Tripjack. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording or otherwise without the prior written permission of Tripjack.

Revision History

Version	Publish date	Revisions
1.0	Jan 18, 2024	Document Creation

1. Amendment API

This method is used to add the VOIDED status to a booking. It consists of three APIs:

1. **Get Amendment Charges (Optional)** -To obtain the charges associated with a void booking. It will not apply amendments to the booking.
2. **Submit Amendment** - To make changes to a booking, you must use this API. It will return an Amendment ID, which may be used to check/poll the status of the amendment.
3. **Get Amendment Details** - To obtain the status and other records of an applied amendment.

1.1 Get Amendment Charges:

This is an optional API request that will return the VOID amendment fees. For any modifications, the booking must be in the SUCCESS state. It will offer amendment fees and refundable amounts for each passenger in the booking, or if the charges are not available, it will respond with "Please contact the support team to know the amendment charges".

URL : [{{host}}](#)/oms/v1/air/amendment/amendment-charges

Method: POST

Sample Payloads: [Link](#)

Request Structure :

→ **bookingId**

→ **Type**

→ **Trips** - ArrayList of trips for which amendment is to be raised (required only if a certain journey is void)

- ◆ **src**: departure airport code
- ◆ **dest**: arrival airport code
- ◆ **departureDate**: Departure travel date (YYYY-MM-DD)
- ◆ **travelers**: list of travelers from a particular trip in a booking for which amendment is to be raised.
 - **fn** - first name of traveler
 - **ln** - last name of traveler

1.1.1 Request amendment charges for the whole booking: Submitting just the booking id will result in prices for all pax categories for the whole booked journeys.

▼ object {2}

bookingId : {{bookingId}}

type : VOIDED

1.1.2 Request amendment charges for specific travelers: Submit traveler information along with trip information. Charges will only be provided for the specified passenger types and trips.

```
▼ object {3}
  bookingId : {{bookingId}}
  type : VOIDED
  ▼ trips [2]
    ▼ 0 {4}
      src : DEL
      dest : SIN
      departureDate : 2024-03-22
      ▼ travellers [1]
        ▼ 0 {2}
          fn : P1
          ln : P1
    ▼ 1 {4}
      src : SIN
      dest : DEL
      departureDate : 2024-02-24
      ▼ travellers [2]
        ▼ 0 {2}
          fn : P1
          ln : P1
        ▼ 1 {2}
          fn : P2
          ln : P2
```

```
▼ object {3}
  bookingId : {{bookingId}}
  type : VOIDED
  ▼ trips [1]
    ▼ 0 {4}
      src : SIN
      dest : DEL
      departureDate : 2024-02-24
      ▼ travellers [2]
        ▼ 0 {2}
          fn : P1
          ln : P1
        ▼ 1 {2}
          fn : P2
          ln : P2
```

```
▼ object {3}
  bookingId : {{bookingId}}
  type : VOIDED
  ▼ trips [1]
    ▼ 0 {4}
      src : DEL
      dest : SIN
      departureDate : 2024-03-22
      ▼ travellers [1]
        ▼ 0 {2}
          fn : P1
          ln : P1
```

1.1.3 Request amendment charges for a particular trip: If charges are required for all passengers in a trip, provide trip data without traveler details.

```
▼ object {3}
  bookingId : {{bookingId}}
  type : VOIDED
  ▼ trips [1]
    ▼ 0 {3}
      src : DEL
      dest : SIN
      departureDate : 2024-03-22
```

Response Structure :

→ **bookingId**

→ **Trips**

- ◆ **Src** - source airport code
- ◆ **dest** - destination airport code
- ◆ **departureDate** - Departure travel date(YYYY-MM-DD)
- ◆ **flightNumbers** - flight numbers list (includes connecting flights flight numbers)
- ◆ **airlines** - airlines code
- ◆ **amendmentInfo** - (Map of PaxType,PaxAmendmentCharges)
 - **amendmentCharges** - Amendment Charges for Particular paxtype
 - **refundAmount** - Refundable Amount for Particular paxtype
 - **totalFare** - Total Fare for Particular paxtype

Sample Response :

```
▼ object {3}
  bookingId : TJS11780000077091
  ▼ trips [1]
    ▼ 0 {6}
      src : DEL
      dest : SIN
      departureDate : 2024-03-22T21:50
      ▼ flightNumbers [1]
        0 : 403
      ▼ airlines [1]
        0 : SQ
      ▼ amendmentInfo {2}
        ▼ CHILD {3}
          amendmentCharges : 950
          refundAmount : 7824
          totalFare : 8774
        ▼ ADULT {3}
          amendmentCharges : 950
          refundAmount : 10307
          totalFare : 11257
      ▼ status {2}
        success : ☒ true
        httpStatus : 200
```

Note: total fare provided in the response will be (total fare) / (number of pax of a particular type in the journey).

1.2 Submit Amendment (VOID) :

This is used to apply an amendment to a booking. After submitting the request, you will receive an "amendmentId" that may be used to check/poll the status of the amendment.

URL: [{{host}}](#)/oms/v1/air/amendment/submit-amendment

Method: POST

Sample Payloads: [Link](#)

Request Structure :

- **bookingId**
- **Type**
- **Remarks** - Comment regarding the amendment.
- **Trips** - ArrayList of trips for which amendment is to be raised (required only If a certain journey is void)
 - ◆ **src**: departure airport code
 - ◆ **dest** : arrival airport code
 - ◆ **departureDate**: Departure travel date (YYYY-MM-DD)
 - ◆ **travelers**: list of travelers from a particular trip in a booking for which amendment is to be raised.
 - **fn** - first name of traveler
 - **ln** - last name of traveler

Sample Request :

```
▼ object {4}
  bookingId : {{booking id}}
  ▼ trips [1]
    ▼ 0 {4}
      src : BOM
      dest : DEL
      departureDate : 2020-10-19
      ▼ travellers [4]
        ▼ 0 {2}
          fn : TESTA
          ln : TESTA
        ▼ 1 {2}
          fn : TESTD
          ln : TESTD
        ▼ 2 {2}
          fn : TESTB
          ln : TESTB
        ▼ 3 {2}
          fn : TESTc
          ln : TESTc
      type : VOIDED
      remarks : this is test void
```

Note : The request structure will remain the same only adding one Key remarks.

[1.1.1 Request amendment charges for the whole booking : Submitting just the booking id will result in prices for all pax categories for the whole booked journeys.](#)

[1.1.2 Request amendment charges for specific travelers : Submit traveler information along with trip information. Charges will only be provided for the specified passenger types and trips.](#)

[1.1.3 Request amendment charges for a particular trip : If charges are required for all passengers in a trip, provide trip data without traveler details.](#)

Response Structure :

- **bookingId**
- **amendmentId** - Should be used for polling current amendment status.

Sample Response :

```
▼ object {3}
  bookingId : bookingId
  amendmentId : amendmentId
  ▼ status {2}
    success : ☒ true
    httpStatus : 200
```

1.3 Get Amendment Details :

This request is used to check the status of amendment, amendment, and refund amount. An amendmentId must be returned with the submit amendment response. If the amendment status is REQUESTED, polling must be done 4-5 times with a 10-second interval to obtain the status. (If the amendment state is still REQUESTED, contact the support staff to find out more about its status).

URL: [{{host}}](#)/oms/v1/air/amendment/amendment-details

Method: POST

Sample Payloads: [Link](#)

Request Sample :

▼ object {1}

amendmentId : 800000009202

Request Structure :

→ **amendmentId** - AmendmentId (from submit amendment response)

Response Structure :

▼ object {6}

bookingId : TJ11310000001169

amendmentId : 520000009115

amendmentStatus : SUCCESS

refundableAmount : 59504

▼ trips [1]

▼ 0 {6}

src : DEL

dest : SIN

departureDate : 2024-02-17T09:55

▼ flightNumbers [1]

0 : 401

▼ airlines [1]

0 : SQ

▼ travellers [2]

▼ 0 {5}

fn : t1

ln : t1

amendmentCharges : 350

refundableamount : 29752

totalFare : 30102

▼ 1 {5}

fn : t2

ln : t2

amendmentCharges : 350

refundableamount : 29752

totalFare : 30102

▼ status {2}

success : ☒ true

httpStatus : 200

Response Structure : It consists of the amendment details

- **bookingId** - Booking Id
- **amendmentId** - Amendment Id
- **amendmentStatus** - Status of amendment
 - ◆ REQUESTED - Amendment is in requested state
 - ◆ REJECTED - Amendment is rejected.
 - ◆ SUCCESS - Amendment modification has been successful with all changes.
 - ◆ PENDING - Amendment has been in progress.
- **refundableAmount** - Total Refund Amount of selected trips
- **totalFare** - Total fare of Booking
- **Trips** - It is a list consisting of the tripwise amendment details.
 - ◆ **src** - source airport code
 - ◆ **dest** - destination airport code
 - ◆ **departureDate** - Departure travel date(YYYY-MM-DD)
 - ◆ **flightNumbers** - flight numbers list (includes connecting flights flight numbers)
 - ◆ **airlines** - airlines list
 - ◆ **Travelers** - list of traveler with amendment charges details
 - **fn** - traveler first name
 - **ln** - traveler last name
 - **amendmentCharges** - traveler amendment charges for particular trip
 - **refundAmount** - traveler refund amount for particular trip
 - **totalFare** - traveler total fare for particular trip

{{host}}

Testenv <https://apitest.tripjack.com>

Prodenv <https://tripjack.com>

Auto - Reissue :

Below are the available service URL & definition

UAT Environment: Protocol : https

Domain: apitest.tripjack.com

SERVICE NAME	URL
SEARCH	https://apitest.tripjack.com/fms/v1/reissue/poll/searchquery-list
SEARCH	https://apitest.tripjack.com/fms/v1/reissue/search
REVIEW	https://apitest.tripjack.com/fms/v1/reissue/review
BOOK	https://apitest.tripjack.com/oms/v1/air/amendment/auto-reissue

Production Environment: Protocol : https

Domain: tripjack.com

SERVICE NAME	URL
SEARCH	https://tripjack.com/fms/v1/reissue/poll/searchquery-list
SEARCH	https://tripjack.com/fms/v1/reissue/poll/search
REVIEW	https://tripjack.com/fms/v1/reissue/review
BOOK	https://tripjack.com/oms/v1/air/amendment/auto-reissue

Sample Payloads: [Link](#)

These APIs need to be invoked in a mention order.

- SEARCH (SearchQuery Listing)
- SEARCH (with request id)
- REVIEW
- BOOK

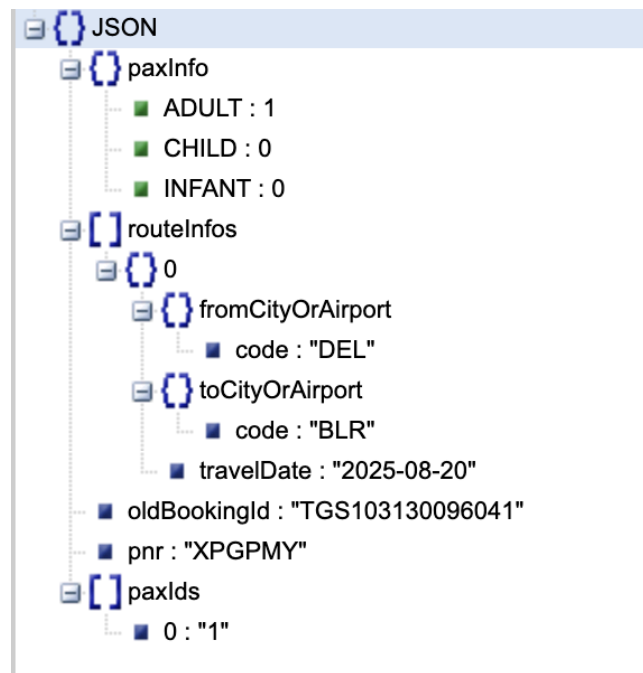
1. SEARCH API

This method provides availability information with the cheapest fare option available to be rescheduled. The request includes a single query for all cases (Oneway, Return, Multi-City). When the request is requested, the system starts validating and the reissue search engine will return the request id to retrieve the reissue results.

Note : One trip can be reissued at a time and partial auto-reissue is supported.

URL : <https://apitest.tripjack.com/fms/v1/reissue/poll/searchquery-list>

Request Method : POST



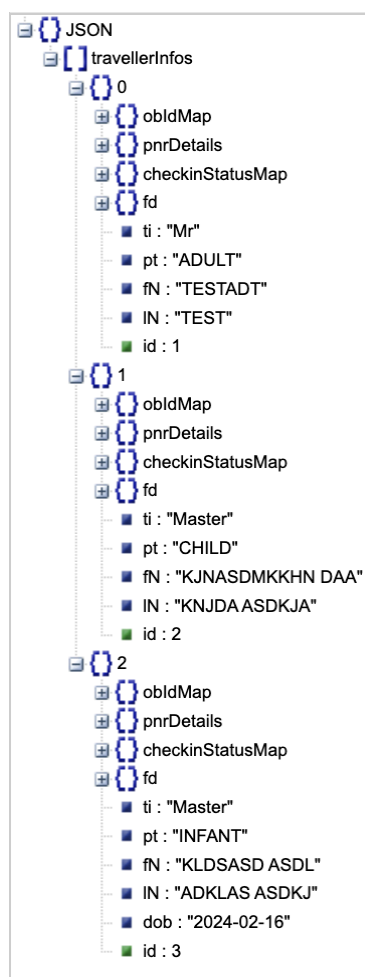
*** Travel Date Format : YYYY-MM-DD**

Request Format Table -

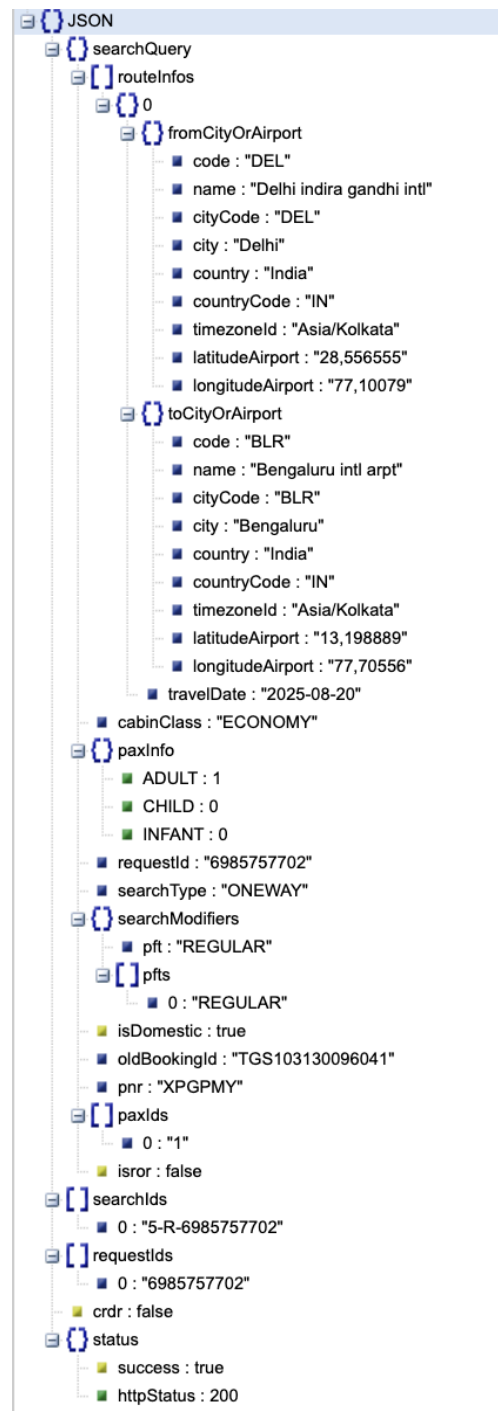
Key	Parent	Type	Description	Mandatory/ Optional
paxInfo	searchQuery	Object	Pax count	Mandatory
ADULT	paxInfo	String		Mandatory
CHILD	paxInfo	String		
INFANT	paxInfo	String		
routeInfos	searchQuery	Object		Mandatory
fromCityOrAirport	routeInfos	String		Mandatory

code	fromCityOrAirport	String		Mandatory
toCityOrAirport	routeInfos	String		Mandatory
code	toCityOrAirport	String		Mandatory
travelDate	routeInfos	String	New travel date of the booking	Mandatory
pnr	searchQuery	String	PNR of the booking to be reschedule	Mandatory
oldBookingId	searchQuery	String	BookingId to be reschedule	Mandatory
paxIds	searchQuery	ArrayList of String	Pax Ids of Pax to be rescheduled	Mandatory

NOTE - Pax ids can be found in booking-details API under traveller Info as “id”.



SEARCHQUERY LISTING RESPONSE STRUCTURE :



Response Format Table -

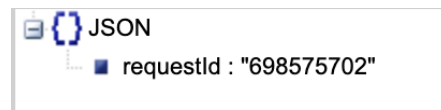
Key	Parent	Type	Description
searchQuery	root	Object	
routeInfos	paxInfo	ArrayList	

fromCityOrAirport	routeInfos	Object	
code	fromCityOrAirport	String	
name	fromCityOrAirport	String	
cityCode	fromCityOrAirport	String	
city	fromCityOrAirport	String	
country	fromCityOrAirport	String	
countryCode	fromCityOrAirport	String	
toCityOrAirport	routeInfos	Object	
travelDate	routeInfos	String	New travel date of the booking
cabinClass	searchQuery	String	
paxInfo	searchQuery	String	
requestId	searchQuery	String	
searchType	searchQuery	String	Type of the journey
searchModifiers	searchQuery	Object	
pft	searchModifiers	String	Preferred Fare Type
pfts	searchModifiers	ArrayList	List of Preferred Fare Type
isDomestic	searchQuery	Boolean	
oldBookingId	searchQuery	String	Booking which is going to be reschedule
pnr	searchQuery	String	PNR of the original booking

searchIds	response	ArrayList	
status	response	Object	

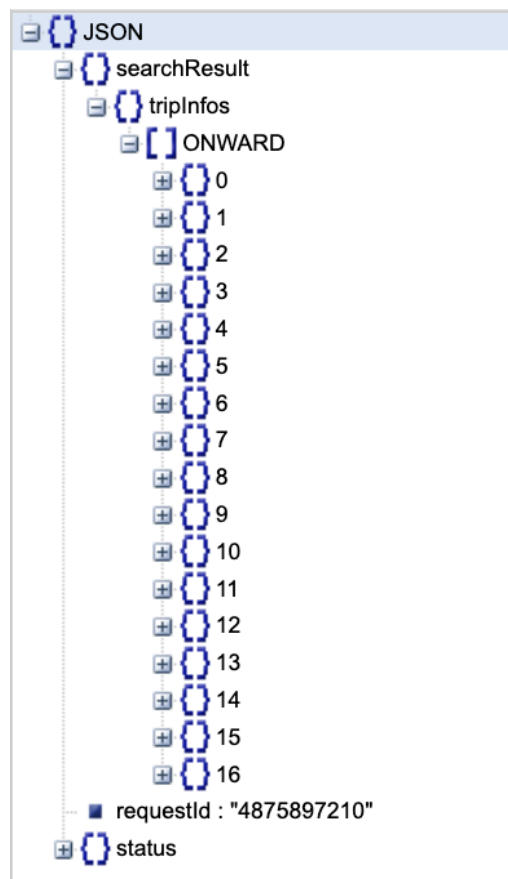
2. SEARCH API (with request Id)

Request -



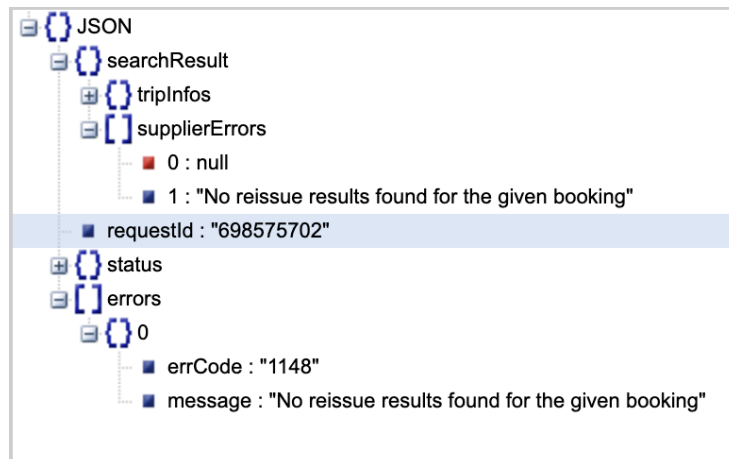
SEARCHQUERY LISTING RESPONSE STRUCTURE :

Response -



Note : Ancillary applied on original booking will be refundable based on the airline behaviour. If refundable, the average of total ancillary applied on booking will be populated in adults and children.

In Case Of No Search Result From Supplier , then following response will be returned



Single Trip Info in response -



TripInfos → Includes List of SegmentInfos

sl (Segment Information) - List<Segments> : which has control of the fields.

- **fd** (Flight Designator) - which has Marketing Airline Information , Airline Equipment type, flight number).
 - **al** - airline info which has code (IATA Airline Code, followed with Airline name).
 - **fN** - Marketing Airline Flight Number
 - **eT** - Marketing Airline Equipment Type
- **duration** - Segment Journey duration
- **da** - Departure Airport Information (which has IATA Airport code,name,country,Terminal)
- **aa** - Arrival Airport Information (which has IATA Airport code,name,country,Terminal)
- **dt** - Departure Time of this segment (YYYY-MM-SS HH:MM Format)
- **at** - Arrival time of this segment (YYYY-MM-SS HH:MM Format)
- **isRs** - isReturnSegment will be true and this field is present only when the segment is returning flight.
- **sN** - segment number

Total Price List in TripInfo -



totalPriceList - Under TripInfo - This is Summarised PriceInformation about the trip

- **fareIdentifier** - what type of fare is this. PUBLISHED, SPECIAL_RETURN.
- **Id** - Unique Trip price Id (which is used for reviewing/revalidating the price which is going to be sold and booked) in upcoming requests and valid for 30 Minutes from search.
- **msri** - matched special return identifier
- **lcca** - is change class allowed
- **fD** - Fare Detail in Per PaxType Wise - (Example : 2Adult, 1 Child Search then have to multiply by 2 for Adult Fare and multiply by 1 for Child Fare)
- **ADULT/CHILD/INFANT** - will be key (Each PayType - Per Pax Fare)
- **fC** - fare Components of Fare to PaxType -
 - TAF - Taxes and Fees
 - NF - Net Fare
 - BF - Difference of New and Old Base Fare
 - TF - Total Fare
 - OTF - Total Fare of original booking to be rescheduled
 - RSSR- Old ancillary amount which has to be refunded
 - AFS- Reissue Fees applied on booking
- **afc** - additional fareComponents in Detail - Break Up of fC Component
- **OT** - Difference of old and new taxes

AGST - Difference of old and new Airline GST

- **ARF** - Airline Reissue fees applied by airline
- **ARFT** - Airline Reissue fees Tax
- **sR** - Seats Remaining
- **cc** - Cabin Class Fare
 - ECONOMY
 - PREMIUM_ECONOMY
 - BUSINESS
 - FIRST
- **cB** - Class Of Booking
- **fB** - Fare Basis
- **rfc** - fare Components regarding reissue
 - **RBF** - The base fare you are reissuing into without difference
 - **RTX** - The total tax you are reissuing into without difference (Does not contain ARFT tax)
 - **OBF** - Original booking base Fare
 - **OTX** - Original Total Tax

NOTE - TC is submission of all the components i.e. difference in BF and Taxes, Airline reissue fee and taxes, refundable ancillaries (if any).

3. REVIEW API

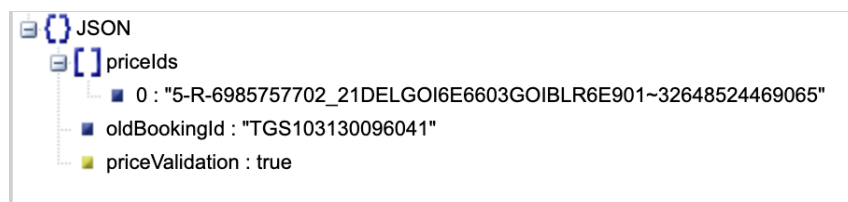
This method provides selected flight information with revalidated fare available from the

supplier. When the request is requested, the system starts validating and the review engine starts validating the selected flight from the supplier and returns the confirmed fare for particular session time with Tripjack Booking Reference Created and returned Tripjack booking reference (reviewed price id) valid for next mentioned seconds in AirReissueReviewResponse (conditions::st).

URL: <https://apitest.tripjack.com/fms/v1/reissue/review>

Request Method : POST

REQUEST



Request Payload

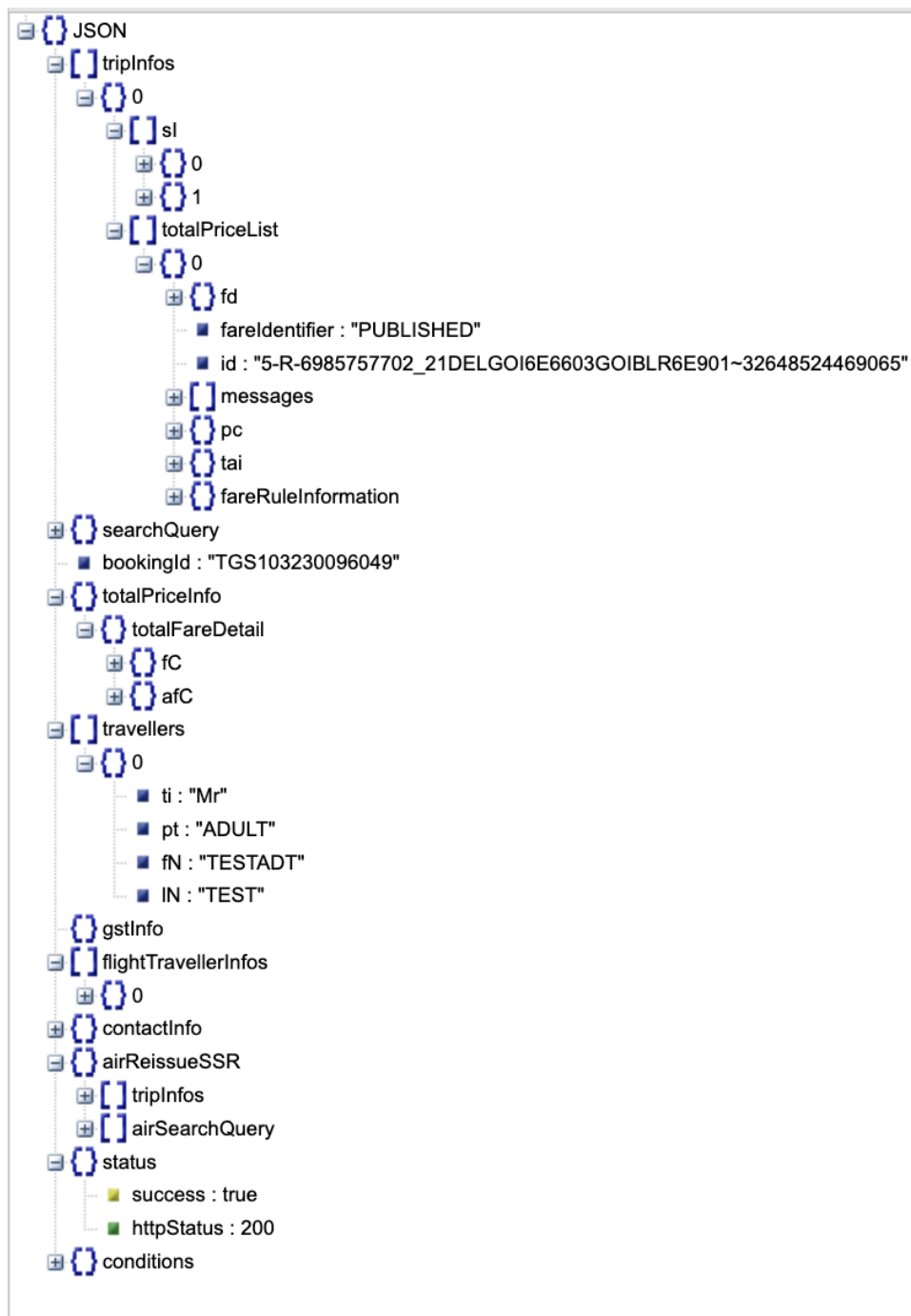
Key	Parent	Type	Description	Mandatory/Optional
pricelds	root	ArrayList	List of Priceld for which itinerary has to be reviewed(price validated).	Mandatory
oldBookingId	root	String	Booking id which is going to be reschedule	Mandatory
priceValidation	root	boolean		Mandatory

Note: A single price id will be passed in the request as one trip can be reissued at a time.

Case - 1 : If pricelds keys are expired (if keys sent in review request are wrong or if expired from system). It will give “Keys Passed if the request is already expired. Please pass valid keys” with httpStatus 400.

Case - 2 : If Selected Flight (Price Id is not available from supplier due to booking class not available,other reasons) then AirReviewResponse will have errors (List<errors> which give description and httpStatus 400).

RESPONSE -



TripInfo Brief : - Each TripInfo which has “sl” (updated segments information and totalPriceList of trip).

1. **sl (Segment Information)** - List<Segments> : which has control of the fields.

→ **Id** - Segment ID - which is used for SSR if any to be selected.

→ **fd** (Flight Designator) - which has Marketing Airline Information , Airline

Equipment

type, flight number).

- **al** - airline info which has code (IATA Airline Code, followed with Airline name).

- **fN** - Marketing Airline Flight Number

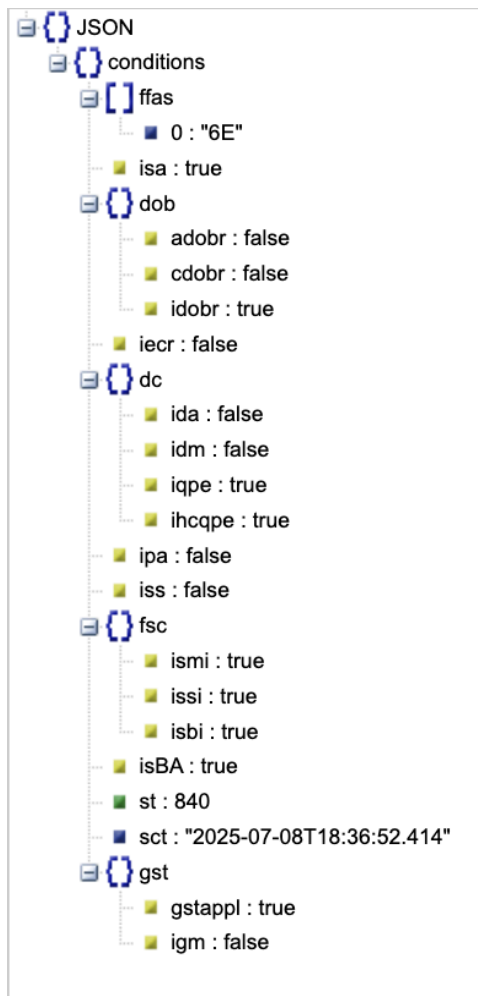
- **eT** - Marketing Airline Equipment Type

- **stops** - Number of Stops in this segment.
- **duration** - Segment Journey duration
- **isRs** - isReturnSegment will be true when the segment is returning flight and in other cases it will be false.
- **sN** - segment number
- **dt** - Departure Time of this segment (YYYY-MM-SS HH:MM Format)
- **at** - Arrival time of this segment (YYYY-MM-SS HH:MM Format)
- **aa** - Arrival Airport Information (which has IATA Airport code, name, country, Terminal)
- **da** - Departure Airport Information (which has IATA Airport code,name,country,Terminal)
- **so** - Stop Over airports (which has IATA Airport code,name,country)
- **cT** - Connecting Time (LayOver time)
- **ssrInfo** - Baggage, Meal, Extra Services .. (explained below)

2. **totalPriceList** : Journey Summarised fare

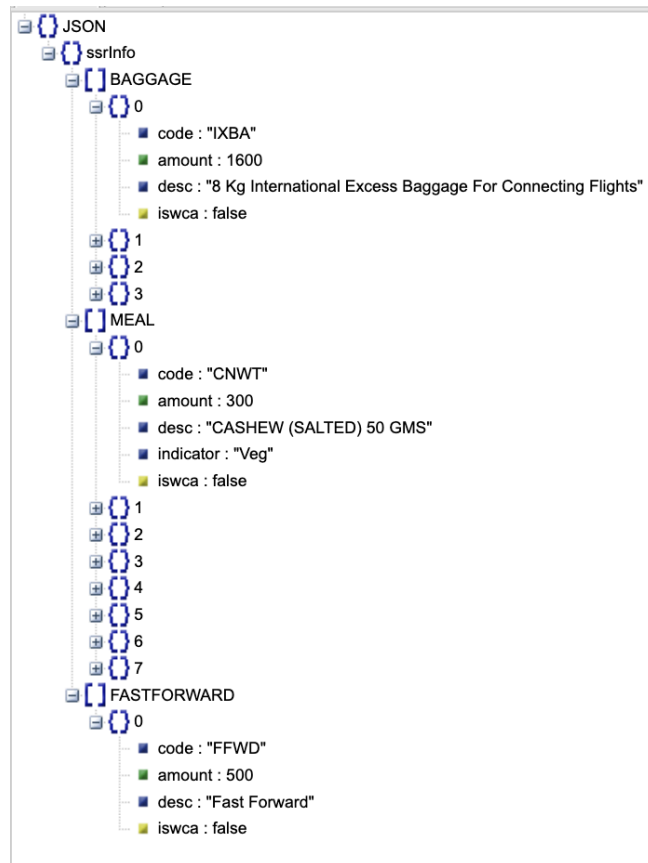
- **fd** - fareDetails of Requested Passenger Types
- **fareIdentifier** - what kind of fare PUBLISHED,CORPORATE,SPECIAL_RETURN.
- **messages** - if any supplier or airline remarks (Object type is list)
 - ◆ **message** - The content of the message
- **id** - unique id of price
- **pc** - Plating Carrier of journeyFare Detail : - PaxType Wise Fare Break Up, Booking class and fare related information.
- **ADULT/CHILD/INFANT** - will be key
- **fc** - fare Components of Fare to PaxType
- **sR** - Seats Remaining
- **bl** - Baggage information (Hand Baggage & Checking baggage Related)
 - ◆ **iB** - Checking Baggage (No format specific example :20Kg , 20 Kg, 1 Piece , 1 Piece)
 - ◆ **cB** - Cabin Baggage
- **cc** - Cabin Class Fare
 - ◆ ECONOMY
 - ◆ PREMIUM_ECONOMY
 - ◆ BUSINESS
 - ◆ FIRST
- **cB** - Class Of Booking
- **fB** - Fare Basis

Conditions -



- **ffas** - Frequent Flier Airlines
- **isa** - is Seat Applicable on selected flight (if this true ,another api to fetch seat information)
- **isBA** - is Hold (Block) option applicable for selected flight
- **st** - Session Time In Seconds (Reviewed Price Id is valid for mentioned seconds to make booking)
- **sct** - Session Created Time.
- **dob** - Date Of Birth Required or Not in PaxType wise
 - **adobr** - Adult Date of Birth Required
 - **cdobr** - Child Date of Birth Required
 - **idobr** - Infant Date of Birth Required
- **gst** - GST is Applicable and mandatory information
 - **igm - GST Mandatory (true)** - Valid GST Number and Registered Name and GST Address, GST mobile number has to be passed in Booking request.
 - **gstappl - GST Applicable (true)** - GST Can be passed in this booking but not mandatory.

SSR Info Brief -



Applicable SSR's are under segment information : BAGGAGE, MEAL, EXTRASERVICES. Each SSR Information has detailed information that needs to be shown and passed to the next request.

→ **code** : which is a unique code of that SSR Received from supplier and that's needed to be sent in booking request if customer selected any.

→ **amount** : Amount of Corresponding SSR.

→ **desc** - Description of Corresponding SSR that needs to be shown to Customer for clear information.

- **In Case connecting Segments , Baggage (SSRType , amount field will not be present) - which means - whatever selected on the first segment of journey applicable to that corresponding journey. (Baggage always works as journey wise, when the amount is present segment wise, then segment wise Baggage applicable, it can be varied).**

- **In Case of Meal (SSR type) - Each segment will have a unique Meal SSR which is applicable only on that segment.**

- **In Case of Meal (SSR type, if the amount field is empty , then any meal can be selected at free of cost).**

If SSR is being selected for any segment then corresponding segment (segment "id" as key has to be passed in booking request field key) - which is explained in below booking flows.

Response Structure -

Key	Parent	Type	Description
tripInfos	root	Array of tripInfo	Array of tripInfo object. Each Object contains details of each trip.
searchQuery	root	Object	Contains details of Search made.
bookingId	root	String	Booking Id generated after setting up the session.
totalPriceInfo	root	Object	Contains the total fare details of the booking
travellers	root	ArrayList of traveller	Pax data of old booking
conditions	root	Object	Details like passport , DOB,GST.
pcs	conditions	Object	Only in case of International search.Passport components
pped	pcs	Boolean	Passport Expiry Date
pid	pcs	Boolean	Passport Issue Date
pm	pcs	Boolean	Passport Mandatory
dobe	pcs	Boolean	DateOfBirth Required

ffas	conditions	ArrayList	Frequent Flier
dob	conditions	Object	Whether date of birth of ADULT, CHILD and INFANT is required or not during booking
adobr	dob	Boolean	Is Adult date of Birth is required in Passengers details
cdobr	dob	Boolean	Is Child date of Birth is required in Passengers details
idobr	dob	Boolean	Is Infant date of Birth is required in Passengers details
isBA	conditions	Boolean	Blocking Allowed (Booking can be put on HOLD)
st	conditions	Integer	Review Session Valid Time in seconds
sct	conditions	String	Session Created Time

gst	conditions	Object	Whether GST is required or not during booking
gstappl	gst	String	GST applicable for current booking.
igm	gst	String	GST Mandatory
dc	conditions		Conditions for document Id
ida	dc	Boolean	Is documentID applicable
idm	dc	Boolean	Is documentID mandatory
status	root	Object	Status of the API call

4. BOOK API -

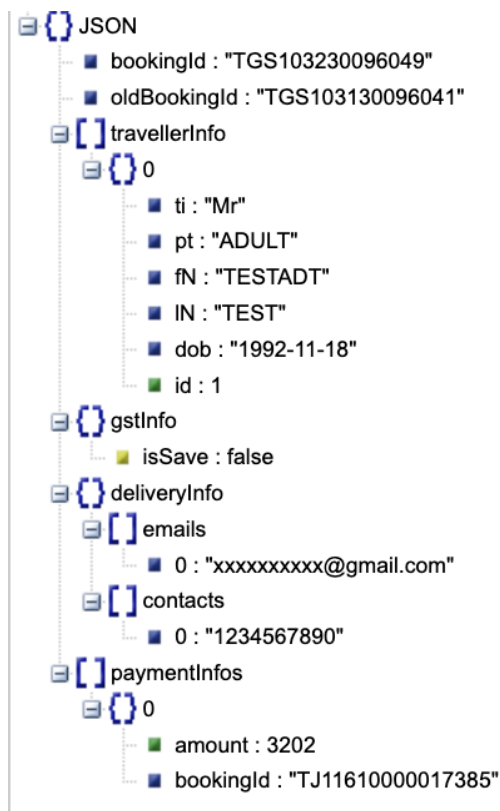
Book method is Direct Ticketing the reissued booking. After submitting the passenger & contact details for the selected itinerary ,BookingResponse will be given with bookingId and status of submitted request.

1. While submitting the reissue BookingRequest, it will be validated, then taking forward to booking, in case of any validation failure, Error Message will be given in response.
2. After validation,Booking will be created in the Tripjack System by deducting Payment, and the Supplier Side will be done.

URL : <https://apitest.tripjack.com/oms/v1/air/amendment/auto-reissue>

Request Method : POST

REQUEST -



Note - The amount sent in the Book Request should be Gross Fare (TF from AirReissueReviewResponse).

Notes on Booking Request :

Case :1 : SSR - List of Meal, Baggage - In Request code which is selected by customer Vs on segmentkey (AirReissueReviewResponse -> "tripInfos" under segment will be having unique key called id which is key)

Case :2 : In Case of GST (all fields of GST have to be passed) with Valid and as IATA

Standard have to be used.

Case 3 : In case of “iecr” is mandatory in Review Response, in book requests contactInfo (Emergency contact name,contact email and contact number) are mandatory to be passed in request.

Case 5 : Pass the correct data in travellerInfo.This data should match with passenger data of the original booking. It should be the same as returned in the review response.

Description :

- **bookingId** : Booking Id from AirReviewResponse
- **oldBookingId** : old Booking Id which is going to be reschedule
- **paymentInfos** - Payment for the booking to Tripjack (List<PaymentInfos>)
 - **amount** - total payable amount to Tripjack for the booking (TF from AirReviewResponse)
- **deliveryInfo** - Delivery Details Related to Booking (which need to be passed to supplier)
 - **emails** - Booking Email Id (It should be valid email id)
 - **contacts** - Valid Contact Number (Followed with country code and valid mobile number) Example (+919500112233)
- **travellerInfo** - List of Traveller which flight to be booked
- **ti** - Title of the Traveller
 - **Adult**: Mr, Mrs, Ms
 - **Child**: Ms, Master
 - **Infant** : Ms, Master
- **pt** - PaxType - ADULT, CHILD, INFANT (Values should be Caps)
- **fN** - First Name of the Traveller
- **IN** - Last Name of the Traveller (Key : lowercase “L”N)
 - **dob** - Traveller Date Of Birth (YYYY-MM-DD Format)
- **pNum** - Passport Number
- **eD** - Expiry Date of the passport (YYYY-MM-DD Format)
- **pNat** - Passport Nationality (Passport Issued Country) - IATA Country Code(2-Letter)
- **pid** - Passport Issue Date (YYYY-MM-DD Format)
- **ssrBaggageInfos** (List of baggage on segment selected)
 - **key** : Segment Key from AirReviewResponse
 - **code** : Customer Selected SSR Code on Corresponding Segment
- **ssrMealInfos** (List of meal on segment selected)
 - **key** : Segment Key from AirReviewResponse
 - **code** : Customer Selected SSR Code on Corresponding Segment
- **ssrSeatInfos** (List of seat on segment selected)
 - **key** : Segment Key from AirReviewResponse
 - **code** : Customer Selected SSR Code on Corresponding Segment
- **gstInfo** - GST Which needs to be passed for billing and supplier - all fields under GST are mandatory.
 - **gstNumber** - Valid 15 digit GST Number (mandatory)
 - **registeredName** - GST Registered Name (maximum 35 characters as IATA)

standard) (mandatory)

- **Email** - GST Email
- **Mobile** - GST Mobile
- **Address** - GST Registered Address (maximum 70 characters as IATA standard)

Request Payload

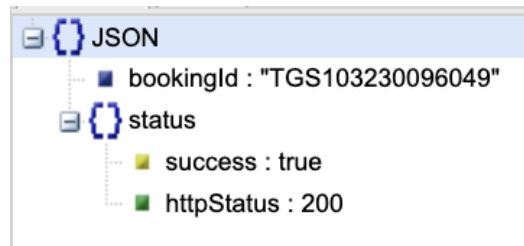
Key	Parent	Type	Description	Mandatory/Optional
bookingId	root	String	Booking id which is generated at Review.	Mandatory
oldBookingId	root	String	Old booking id which is going to be reschedule	Mandatory
paymentInfos	root	Array of Objects	Object containing the payment details of Booking.	Mandatory
amount	paymentInfos	number	Total cost for the Booking	Mandatory
travellerInfo	root	Array of Objects	List of traveller details object	Mandatory
gstInfo	root	Object	Object containing GST info	Mandatory
gstNumber	gstInfo	String	GST number	Mandatory
registeredName	gstInfo	String	GST name	Mandatory
contactInfo	root	Object	Object containing the details of delivery emails	Optional
deliveryInfo	root	Object	Object containing the emergency contact details (emails, mobile numbers and contact name)	Mandatory

Response Payload -

Key	Parent	Type	Description	Mandatory/Optional
bookingId	root	String	Booking id which is generated at Review.	Mandatory

status	root	Object	Status of the API call	Mandatory
--------	------	--------	------------------------	-----------

Response -



5. FAQ s-

1. How the rescheduled booking amount adjusts the old booking amount in the fare calculation.

Ans - The previous base fare is subtracted from the new Base Fare, and the difference is added to the total. The same is done for the tax components. Then the Airline Reissue Fee and its respective tax components are added to the total.

An example breakdown of this adjustment.

Original Booking Fare Distribution -

BF - 3000

TAX - 1000

Reissue Fare Distribution -

BF - 3200

TAX - 1100

Reissue Fee - 2500

Reissue Fee Tax - 100

TripJack Reissue Fee - 50 (TripJack Reissue Charge)

Total - Difference of Base Fare + Difference of Tax + Reissue Fee + Reissue Fee Tax + TripJack Reissue Fee

- 200 + 100 + 2500 + 100 + 50 = 2900

Any refundable ancillary charges are also subtracted from the total charge.

2. What are paxIds in search requests?

Ans - It is a list of all travellers to be reissued. The specific id for all travellers can be retrieved from the booking-details API under TravellerInfo as "id".

3. Date restrictions for onward reissue in a round trip :

The doc does not state any restriction that forbids reissuing to a date earlier than the original departure or between the original onward and return dates.

It explicitly says:

"One trip can be reissued at a time and partial auto-reissue is supported".

Since you pass the new travel Date in the SEARCH request, the system uses that to check availability and price.

Whether a pre-departure change (like 1-Dec → 28-Nov) is allowed will ultimately depend on airline fare rules — the API will return availability and cost if the airline permits it.

Interpretation:

28-Nov — technically possible in API, but may be rejected if the fare rules prohibit moving to an earlier date.

14-Dec — possible if fare rules allow; still before the return date.

4. Domestic vs. international applicability: Yes, reissues are permitted for both international and domestic journeys, regardless of the airline.

5. Airline restrictions: The doc does not list specific airlines excluded from auto-reissue. However, all functionality depends on airline system support — so certain low-cost or non-participating carriers may fail to return results.

6. Partial reissue: Yes — partial reissue is supported:

“Partial auto-reissue is supported”.

If the onward journey is already flown or cancelled, you can still reissue the remaining journey (like the return leg).

7. Partial reissue for combined round-trip international bookings : The doc does not exclude international cases from partial reissue. So combined round-trip international should also be supported if the airline allows.

8. Selecting seats, meals, or baggage: Yes — SSR (Special Service Request) items are supported during reissue:

Seats, meals, baggage can be selected during the BOOK request, using SSR codes from the REVIEW response.

9. Changing cabin class during reissue : Yes — the search and review results display both the cabin class and booking class in the fare. If Business Class is available and permitted

by the fare rules, you may reissue into it. Upgrading is allowed, but downgrading to a lower cabin class is not permitted.

10. Number of reissues: No, once a booking has been reissued, it cannot be reissued again.

11. Does totalPrice include airline penalty?

Yes — the doc states:

“Total – Difference of Base Fare + Difference of Tax + Reissue Fee + Reissue Fee Tax + TripJack Reissue Fee”

This means the airline penalty (Airline Reissue Fee) is already part of totalPrice.

12. Status of the original booking: You’d have to check via the booking-details API before starting the reissue to know whether it’s “Success,” etc.

13. Processing time: After BOOK, Tripjack issues the new ticket instantly if validations pass.

Status updates for both the old and new bookings happen immediately in the Tripjack system, but airline systems may have a short delay.



AIR API DOCUMENTATION - POST-BOOKING ANCILLARIES

© Copyright 2023, Tripjack (Powered By Atlas Travels Online).

This document, which contains confidential material, is private and confidential and is the property and copyright of Tripjack. No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, chemical, photocopy, recording or otherwise without the prior written permission of Tripjack.

Revision History

REVISION	DATE	REVISION
1.0	22-06-2023	POST-ANCILLARIES

Below are the available service URL & definition

UAT Environment:

Protocol : https

Domain: apitest.tripjack.com

SERVICE NAME	URL
FETCH SSR	https://apitest.tripjack.com/fms/v1/ancillaries/fetch/ssr
FETCH SEAT	https://apitest.tripjack.com/fms/v1/ancillaries/fetch/seat
ADD SSR	https://apitest.tripjack.com/oms/v1/air/amendment/add/ssr
AMENDMENT DETAILS	https://apitest.tripjack.com/oms/v1/air/amendment/amendment-details

Production Environment:

Protocol : https

Domain: tripjack.com

SERVICE NAME	URL
FETCH SSR	https://tripjack.com/fms/v1/ancillaries/fetch/ssr
FETCH SEAT	https://tripjack.com/fms/v1/ancillaries/fetch/seat
ADD SSR	https://tripjack.com/oms/v1/air/amendment/add/ssr
AMENDMENT DETAILS	https://tripjack.com/oms/v1/air/amendment/amendment-details

1.0 POST BOOKING ANCILLARIES

These APIs need to be invoked in a mention order.

- FETCH SSR
- FETCH SEAT
- ADD SSR
- AMENDMENT DETAILS

1.0.1 FETCH SSR API

This method retrieves the SSR (MEAL/BAGGAGE) information after the booking process is finalized and the booking status is marked as SUCCESS.

→ For meals, they can only be added once per passenger per segment. If a meal has already been selected during pre-booking, it cannot be added again for the same passenger for that segment.

→ For baggage, it can be added per passenger per trip, and multiple baggage items can be selected(Once at a time).


Note : Infants are not allowed to add SSR

URL: <https://apitest.tripjack.com/fms/v1/ancillaries/fetch/ssr>

Request Method : POST

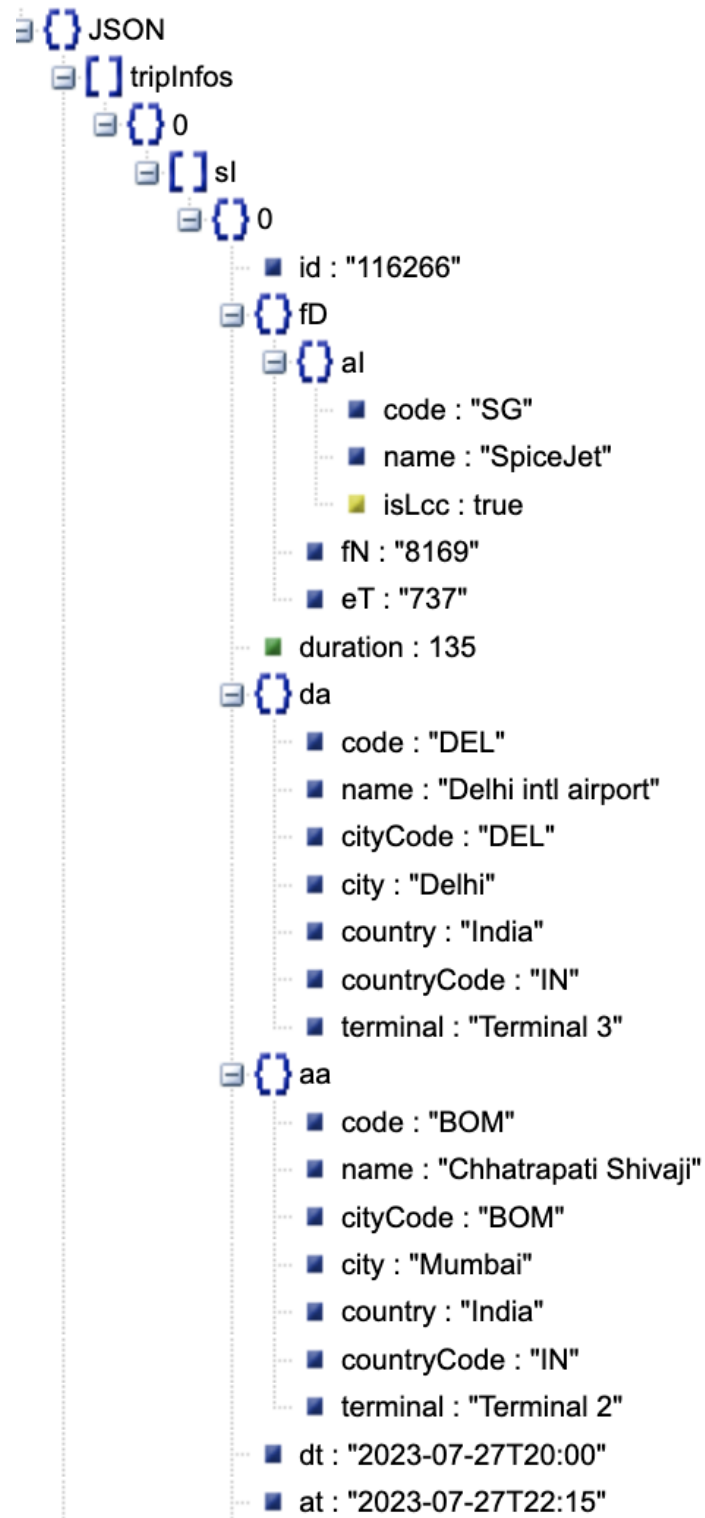
Sample Payloads : will attach soon

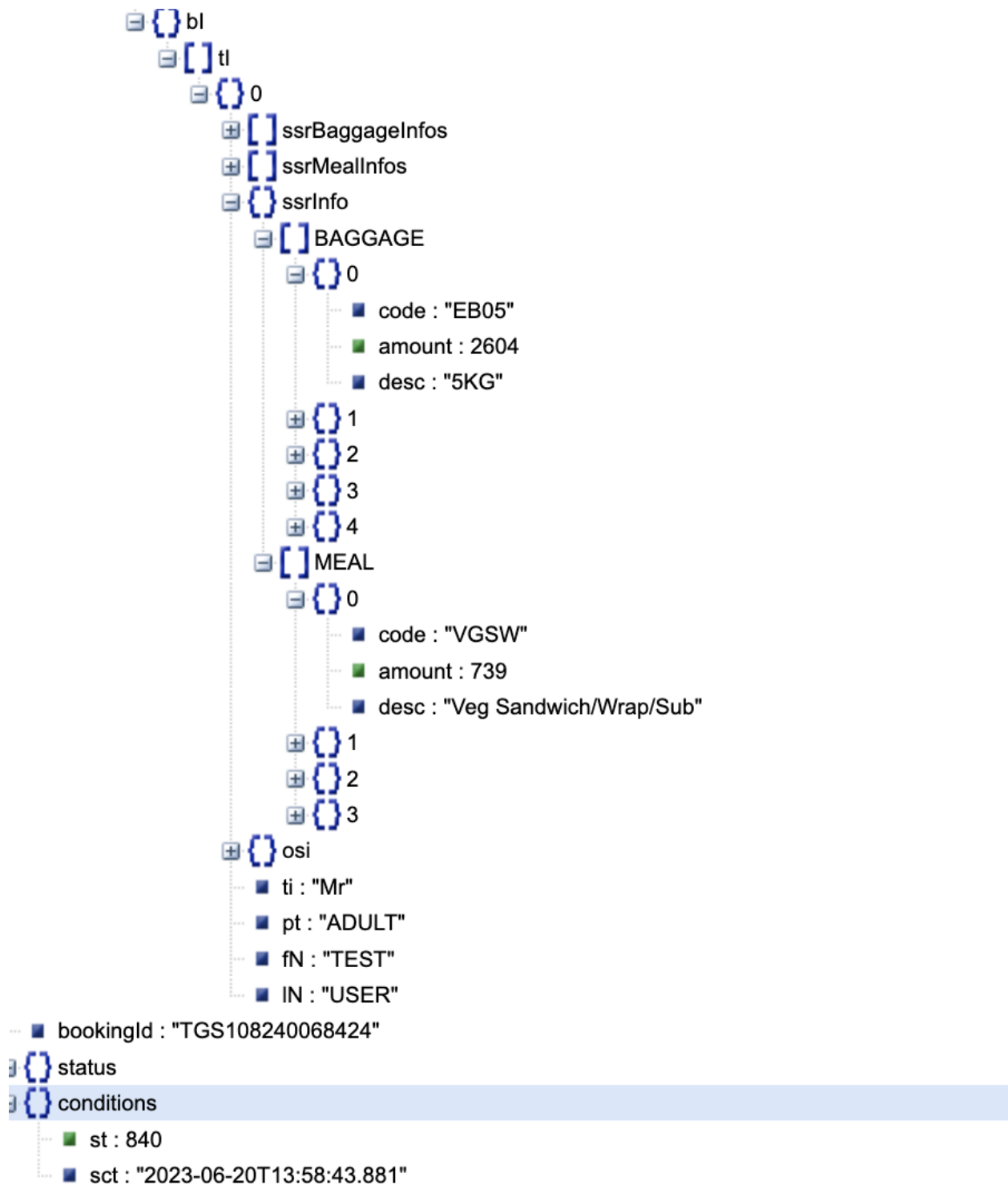
FETCH SSR REQUEST STRUCTURE :

 JSON

```
■ bookingId : "TGS108370066489"
```


FETCH SSR RESPONSE STRUCTURE :





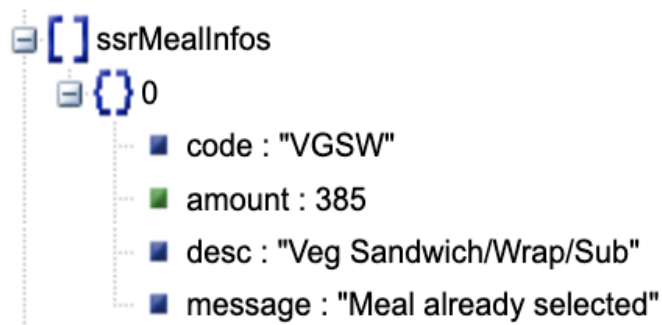
TriplInfos → Includes List of SegmentInfos

sl (Segment Information) - List<Segments> : which has control of the fields.

- **Id** - Segment ID - which is used for SSR if any to be selected.
- **fD** (Flight Designator) - which has Marketing Airline Information , Airline Equipment type, flight number).
 - **al** - airline info which has code (IATA Airline Code, followed with Airline name).
 - **fN** - Marketing Airline Flight Number
 - **eT** - Marketing Airline Equipment Type
- **duration** - Segment Journey duration
- **da** - Departure Airport Information (which has IATA Airport code,name,country,Terminal)
- **aa** - Arrival Airport Information (which has IATA Airport code,name,country,Terminal)
- **dt** - Departure Time of this segment (YYYY-MM-SS HH:MM Format)
- **at** - Arrival time of this segment (YYYY-MM-SS HH:MM Format)
- **bl** - Booking Related Information
 - **tl** (travellerInfo) - List of Traveler information
 - **ssrBaggageInfos** : It contains information of already selected Baggage , it will be empty in case no baggage is selected.



- **ssrMealInfos** : It contains information of an already selected Meal , it will be empty in case no baggage is selected.

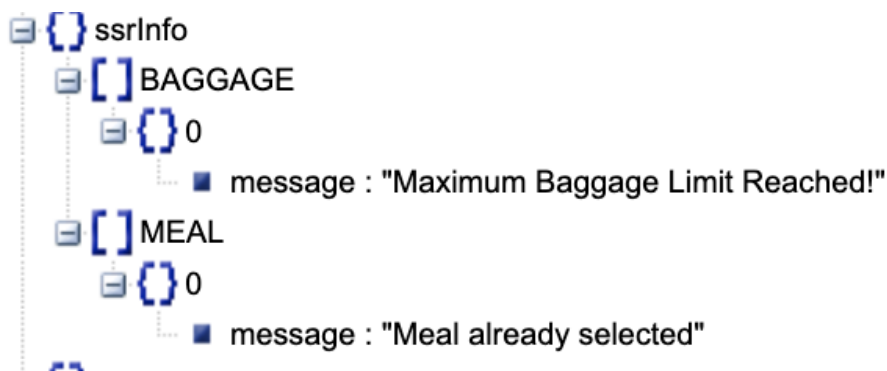


- **ssrInfo**: Each SSR Information has detailed information that needs to be shown and passed to the next request.
 - **code** : which is a unique code of that SSR Received from supplier and that's needed to be sent in booking request if customer selected any.
 - **amount** : Amount of Corresponding SSR.
 - **desc** - Description of Corresponding SSR that needs to be shown to Customer for clear information.

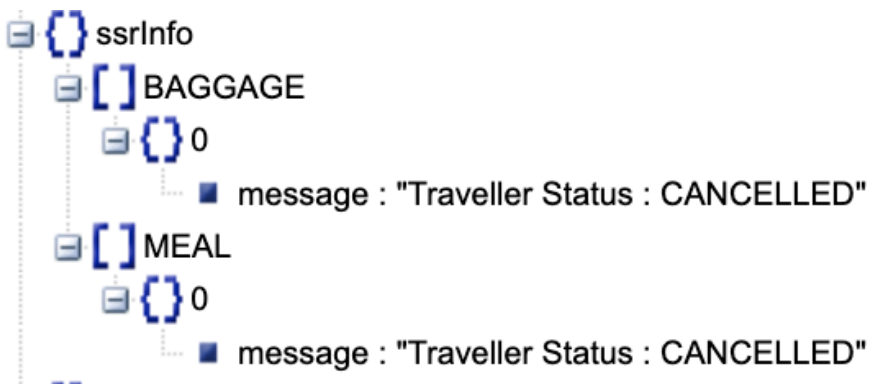
In the case of connecting Segments, the Baggage (SSRType), amount field will not be present- which means - whatever is selected on the first segment of the journey applies to that corresponding journey.

In Case of Meal (SSR type) - Each segment will have a unique Meal SSR which is applicable only to that segment.

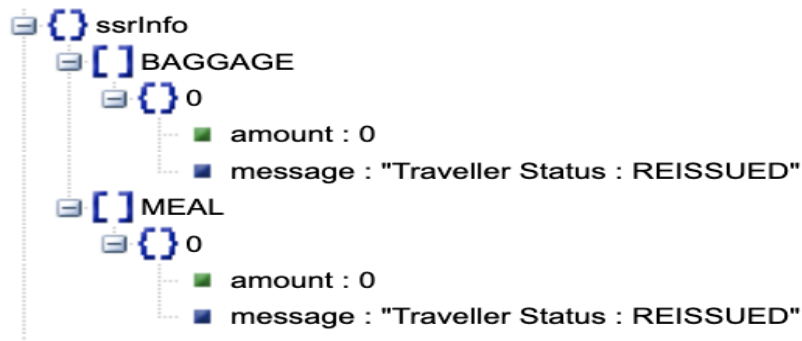
Case 1: when all Baggage has been selected and Meal is also selected.



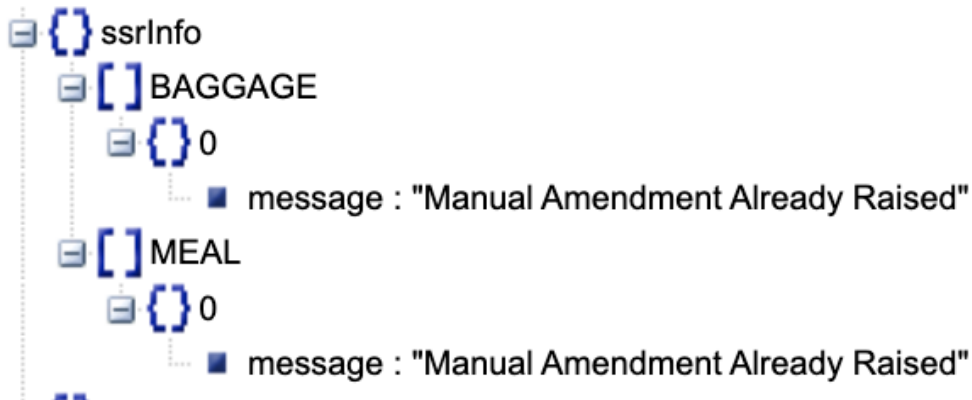
Case 2 : when pax status is canceled



Case 3: when pax has Reissued booking



Case 4 : when manul SSR Amendment has done for pax



- **ti** - Title of the Traveler
 - Adult: Mr,Mrs,Ms
 - Child: Ms, Master
 - Infant : Ms, Master
- **pt** - PaxType - ADULT,CHILD, INFANT (Values should be Caps)
- **fN** - First Name of the Traveler
- **IN** - Last Name of the Traveler
- **dob** - Traveler Date Of Birth (YYYY-MM-DD Format)

Conditions :

St : session time(secs)

Sct: session created time

Request payload

Key	Parent	Type	Description	Mandatory/ Optional
BookingId	Root	String		Mandatory

Response Payload

Key	Parent	Type	Description	Mandatory/ Optional
tripInfos	root	Array of tripInfos	Array of tripInfo object. Each Object contains details of each trip.	Mandatory
sl	tripInfo	Array of segmentInfo	Array of segments for the current trip.	Mandatory
bookingId	Root	String		Mandatory
conditions	Root	Object		Mandatory
st	conditions	Integer	Session time	
sct	conditions	String	Session created time	

segmentInfo:

Key	Parent	Type	Description	Mandatory/ Optional
id	segmentInfo	String	Id which generated during review for ssr	Optional
fD	segmentInfo	Object	Flight Designator (Marketing airline,Equipment and Flight number)	Mandatory
duration	segmentInfo	Integer	Travel duration of the current segment in minutes.	Mandatory
da	segmentInfo	Object	Departure Airport	Mandatory

aa	segmentInfo	Object	Arrival Airport	Mandatory
dt	segmentInfo	String	Departure date and time. Format: YYYY-MM-DDTHH:MM	Mandatory
at	segmentInfo	String	Arrival date and time. Format: YYYY-MM-DDTHH:MM	Mandatory
ssrInfo	segmentInfo	Object	Special service requests	Mandatory

fD: [Flight Designator - Marketing Airline,Equipment Type & Flight Number]

Key	Parent	Type	Description	Mandatory/ Optional
al	fD	Object	Airline Information	Mandatory
code	al	String	Airline code	Mandatory
name	al	String	Airline name	Mandatory
isLcc	al	Boolean	True if the airline is LCC else false	Mandatory
fN	fD	String	Flight Number	Mandatory
eT	fD	String	Equipment Type	Mandatory

ssrInfo:

Key	Parent	Type	Description	Mandatory/ Optional
MEAL	ssrInfo	Array of Object	Meal Information	Mandatory
BAGGAGE	ssrInfo	Array of Object	Baggage Information	Mandatory
code	Object	String	Unique code of each ssr	Mandatory
desc	Object	String	Description which needs to be shown in UI	Mandatory

1.0.2 FETCH SEAT MAP API

This API retrieves seat information after the booking process is completed and the booking status is marked as SUCCESS. Seat maps can be obtained by invoking this API and passing the "booking id" in a request

Please note that a seat can only be added once per passenger per segment.


Note : Infants are not allowed to add Seat

URL: <https://apitest.tripjack.com/fms/v1/ancillaries/fetch/seat>

Request Method : POST

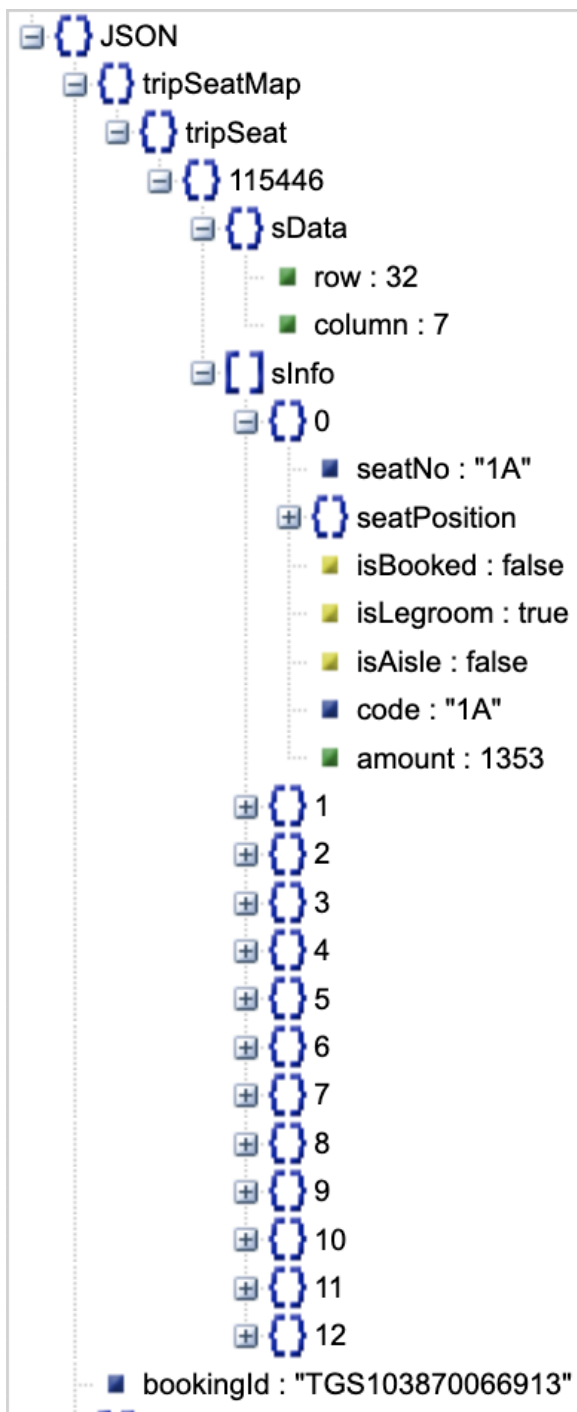
Sample Payloads : will attach soon

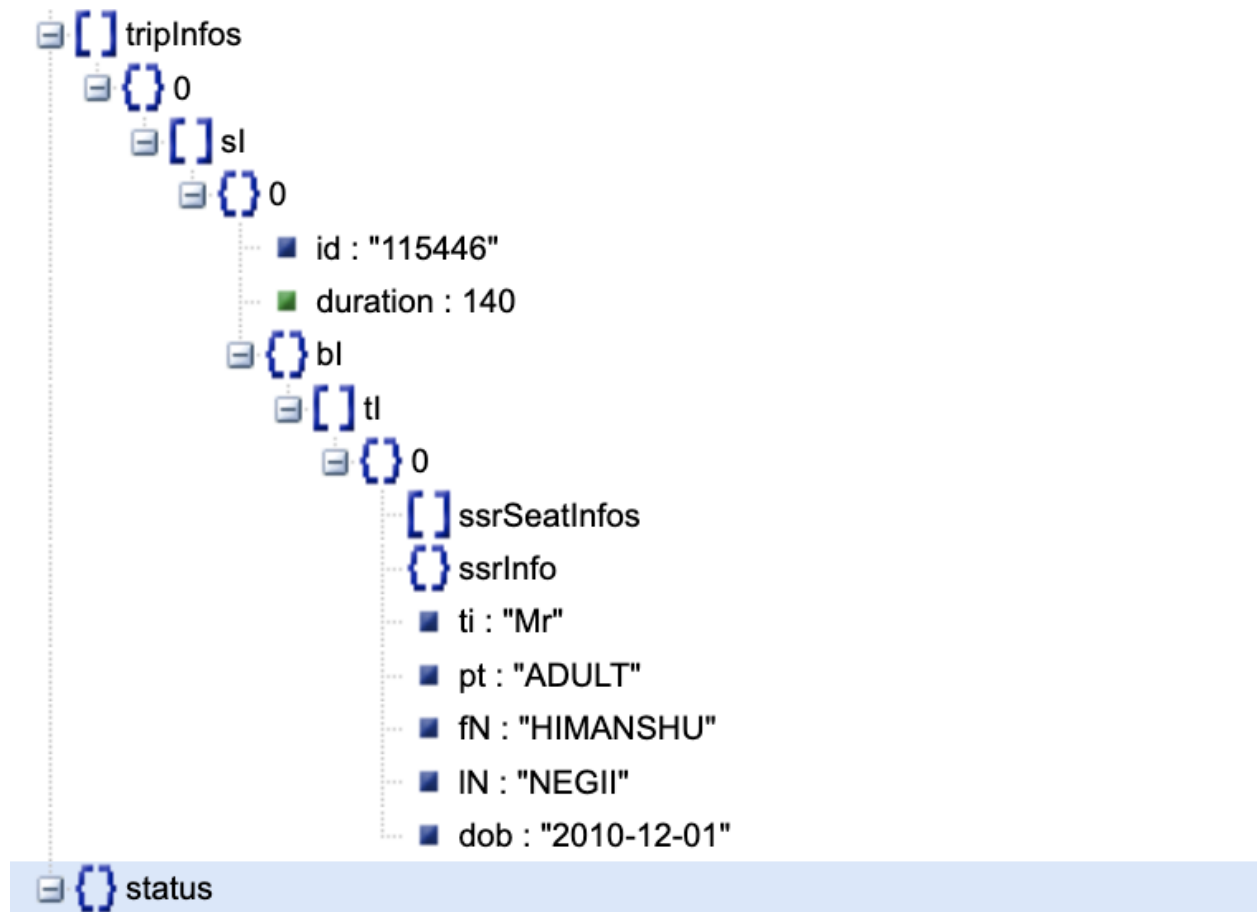
FETCH SEAT REQUEST STRUCTURE :

 JSON

■ bookingId : "TGS108370066489"

FETCH SEAT RESPONSE STRUCTURE :





TripSeat : This will contain seat map Information.

→ **tripSeat (key)**

- ◆ **key** - segment :id , belongs to that particular segment
- ◆ **sData** - Seat Deck Data (How many columns and rows to be created)
- ◆ **sInfo** - List of Seats in Selected Segment
 - **seatNo** : Seat Information Number with code
 - **seatPosition** : will have row and column number to place seat
 - Row
 - Column
 - **isBooked** - False , if seat is available
 - **isLegRoom** :True, if Leg room is available
 - **isAisle** : True , if seat is aisle

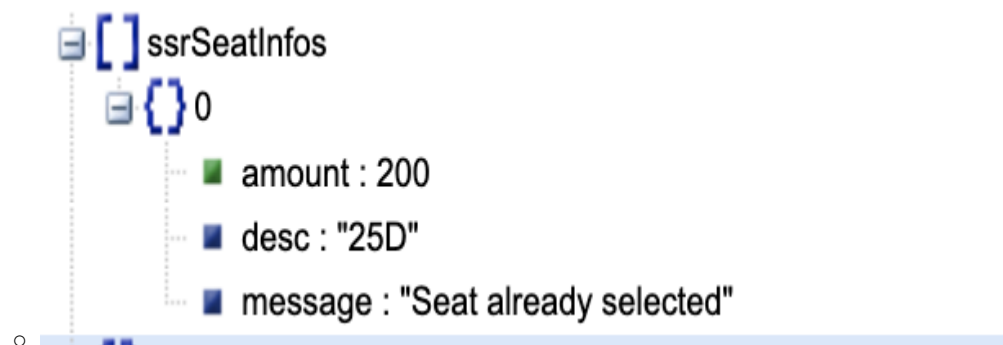
- **code** : Supplier Booking Code which needs to be passed for booking if a customer selects this particular seat.
- **amount** : Seat Chargeable amount.

TripInfos: List of trip info

sl (Segment Information) - List<Segments> : which has control of the fields.

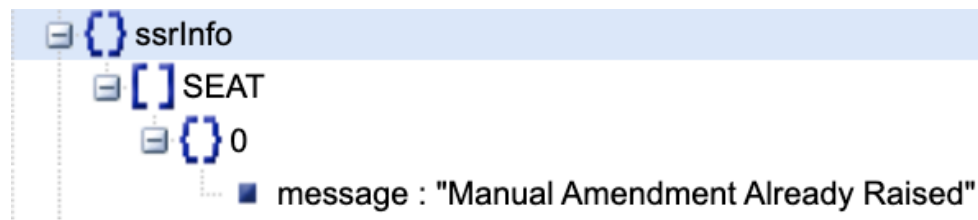
- **ID**: Segment Id,id of the segment.
- **duration** - Segment Journey duration (In Minutes)
- **bl** - Booking Related Information
 - **ssrSeatInfos**:It will contain information of the already selected seat, and will be empty in case no seat is select

→ Case :when the user has already selected a seat..

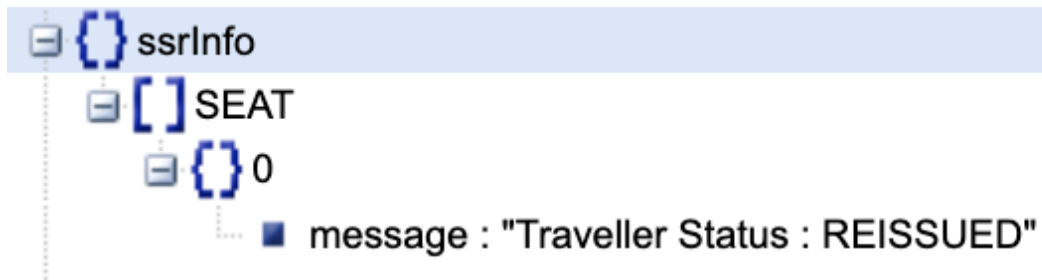


- **ssrInfo**: It will only be populated in following cases:

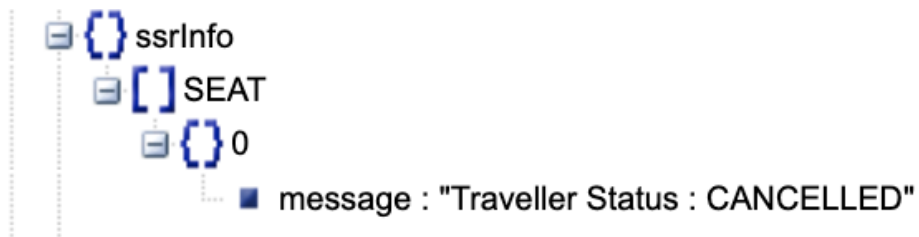
Case 1: When Manual amendment is done for seat



Case 2 :when pax has Reissued booking



Case 3: when pax status is canceled



○ **tl (travellerInfo) : List of Traveller**

- **tl** - Title of the Traveler
- **Adult:** Mr,Mrs,Ms
- **Child:** Ms, Master
- **Infant** : Ms, Master

- **pt** - PaxType - ADULT,CHILD, INFANT (Values should be Caps)
- **fN** - First Name of the Traveler
- **IN** - Last Name of the Traveler
- **dob** - Traveler Date Of Birth (YYYY-MM-DD Format)

Request payload

Key	Parent	Type	Description	Mandatory/ Optional
BookingId	Root	String		Mandatory

Response payload

Key	Parent	Type	Description	Mandatory/ Optional
tripSeatMap	root	Object	It contains a Map of AirCraft and its Seat configuration for the whole trip.	Mandatory
tripSeat	tripSeatMap			
sData		Object		
row	sData	Integer		
column	sData	Integer		
sInfo		Array of Object	Each Object contains single seat Information.	
seatNo	sInfo	String	Seat No	Mandatory
isBooked	sInfo	boolean	Seat status	Mandatory
isLegroom, isAisle	sInfo sInfo	Boolean boolean	Seat with Legroom Aisle seat	
code	sInfo	String	Seat Code	Mandatory
amount	sInfo	Double	Seat Price	Mandatory
seatPosition	sInfo	Object		Mandatory
row	seatPosition	Integer	Deck Data Row	
column	seatPosition	Integer	Deck Data column	
BookingId	root	String		Mandatory
tripInfos	root	Array of tripInfo	Array of tripInfo object	Mandatory
travellerInfos		Object	Array of travellerInfo	Mandatory

1.0.3 ADD SSR API :

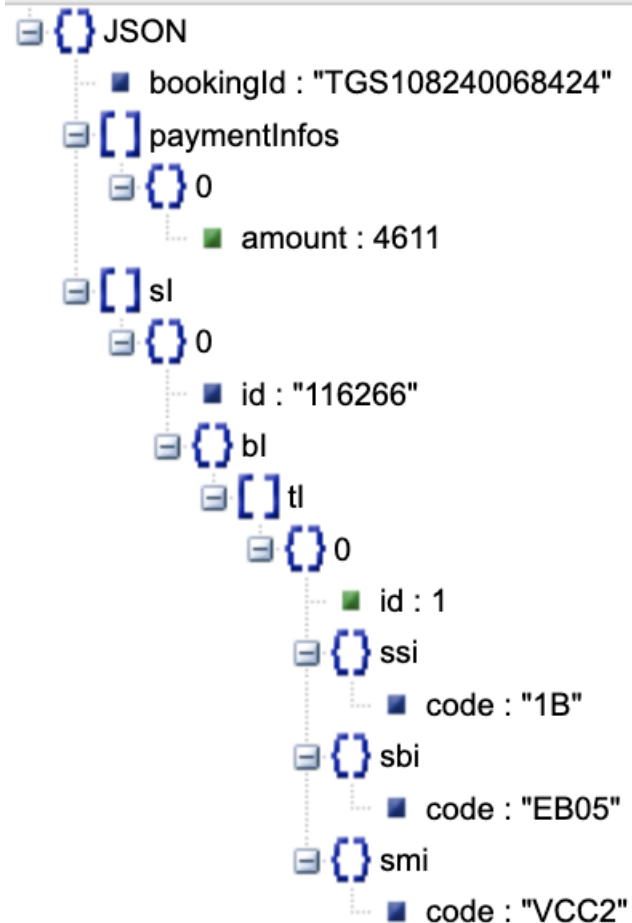
This API is utilized to include SSR for a selected Booking ID. It should be called after successfully invoking the FETCH SSR and FETCH SEAT APIs.

URL: <https://apitest.tripjack.com/oms/v1/air/amendment/add/ssr>

Request Method : POST

Sample Payloads : will attach soon

ADD SSR REQUEST STRUCTURE :



paymentInfos - Payment for the booking to Tripjack (List<PaymentInofs>)

- amount - total payable amount to Tripjack for the SSR

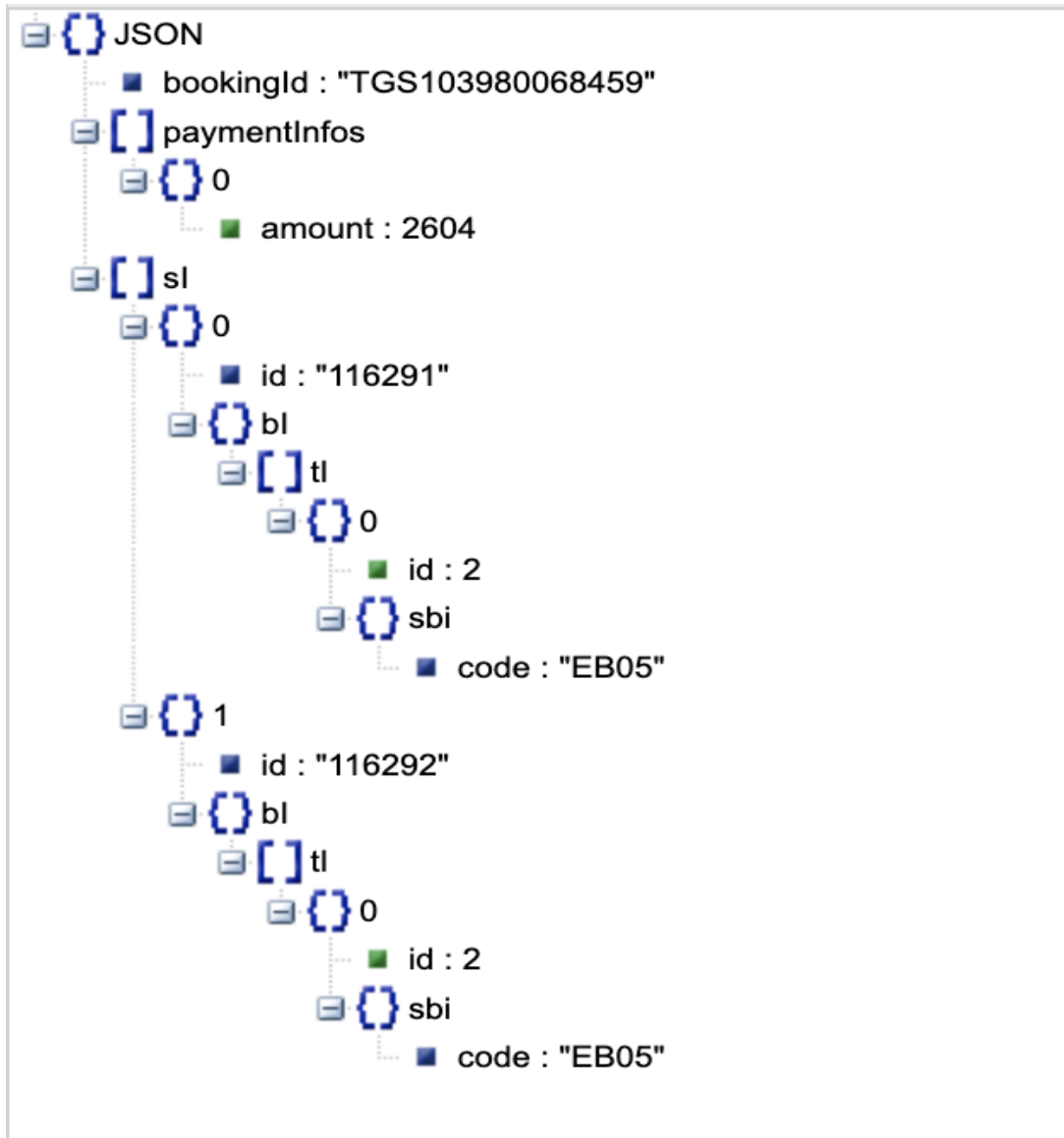
sl (Segment Information) - List<Segments> : It will contain a segment array for both trips(ONEWAY and RETURN).

- **ID:** Segment Id,id of the segment.

- **bl** - Booking Related Information
- **tl : List of Travelers**
 - **Id** : id of a pax for whom the SSR is being added.
 - **sbi**: This contains information about the baggage that needs to be selected.

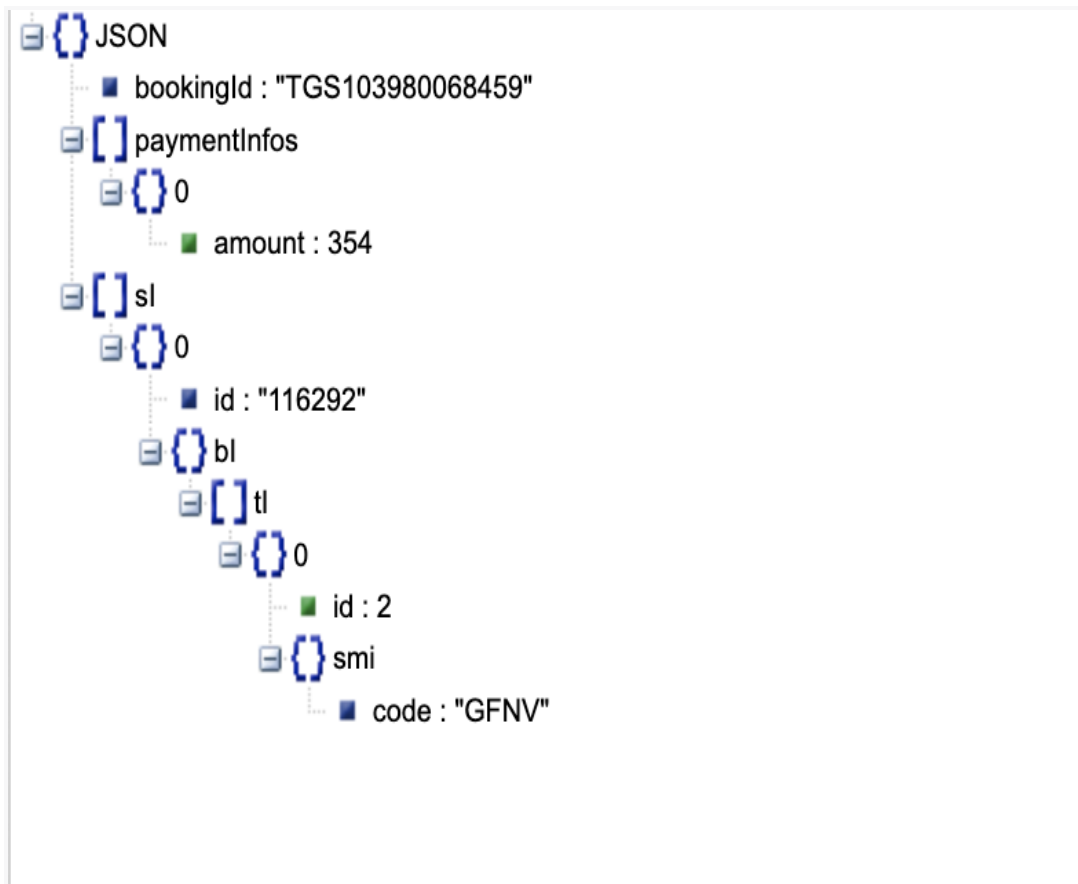
→ This information should be sent only when baggage needs to be added.

Case : In the case of more than one segment, the SBI information should be passed in another segment as well, with amount 0.



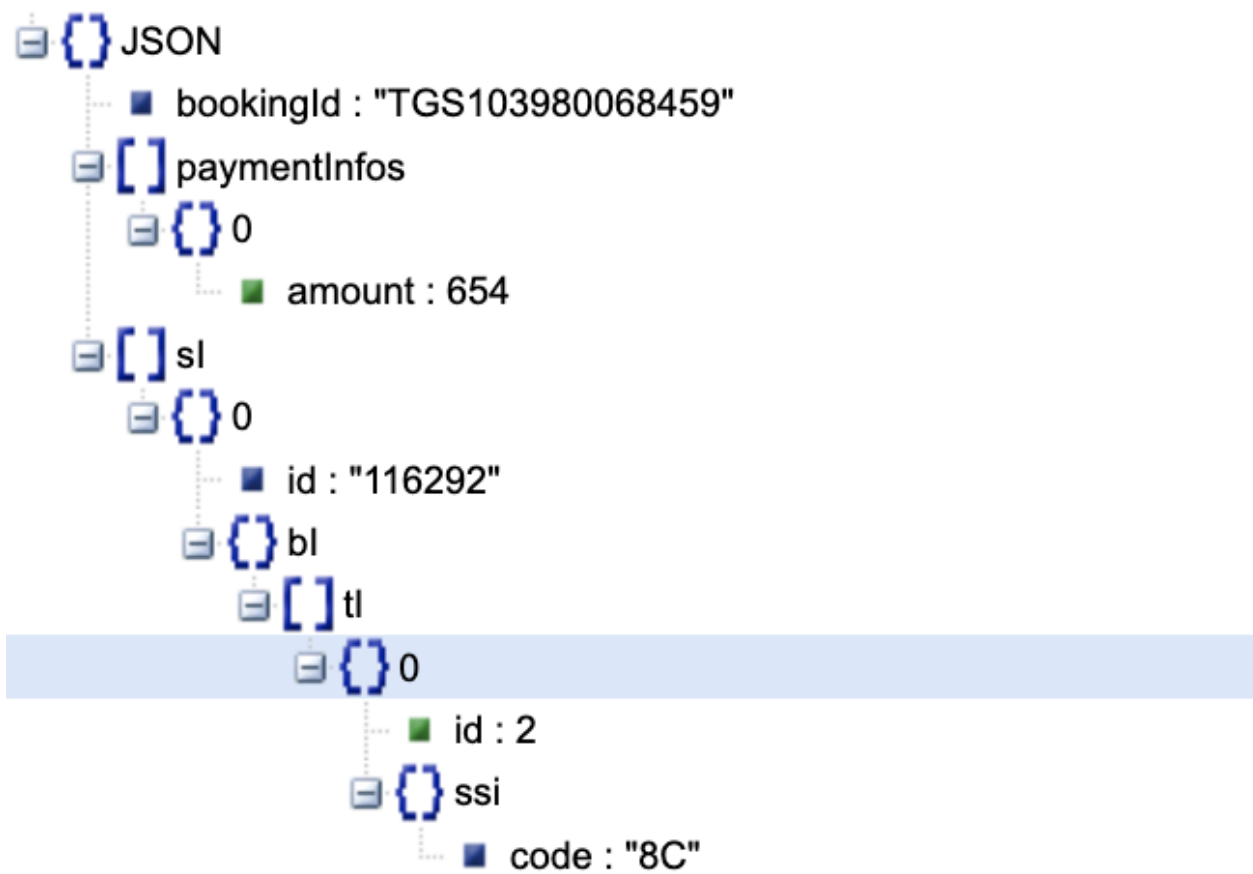
- **smi** : This contains information about the meal that needs to be selected.

→ This information should be sent only when a meal needs to be added.

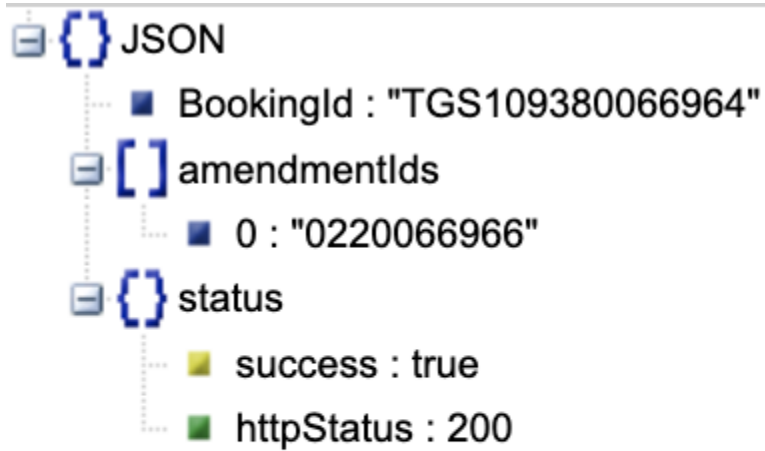


- o **ssi** : This contains information about the seat that needs to be selected.

→ This information should be sent only when a seat needs to be added.



ADD SSR RESPONSE STRUCTURE



amendmentids: This list contains amendments, with amendment IDs generated on a per-trip basis. A single ID is generated for all SSR amendments within a trip.

1.0.4 Amendment Details:

This request is used to retrieve the status of an amendment, indicating whether it is "SUCCESS" or "REJECTED". To make this request, you need to provide the amendmentId, which can be fetched from the ADD SSR response.

URL: <https://apitest.tripjack.com/oms/v1/air/amendment/amendment-details>

Method: POST

Sample Payloads : [will upload soon](#)

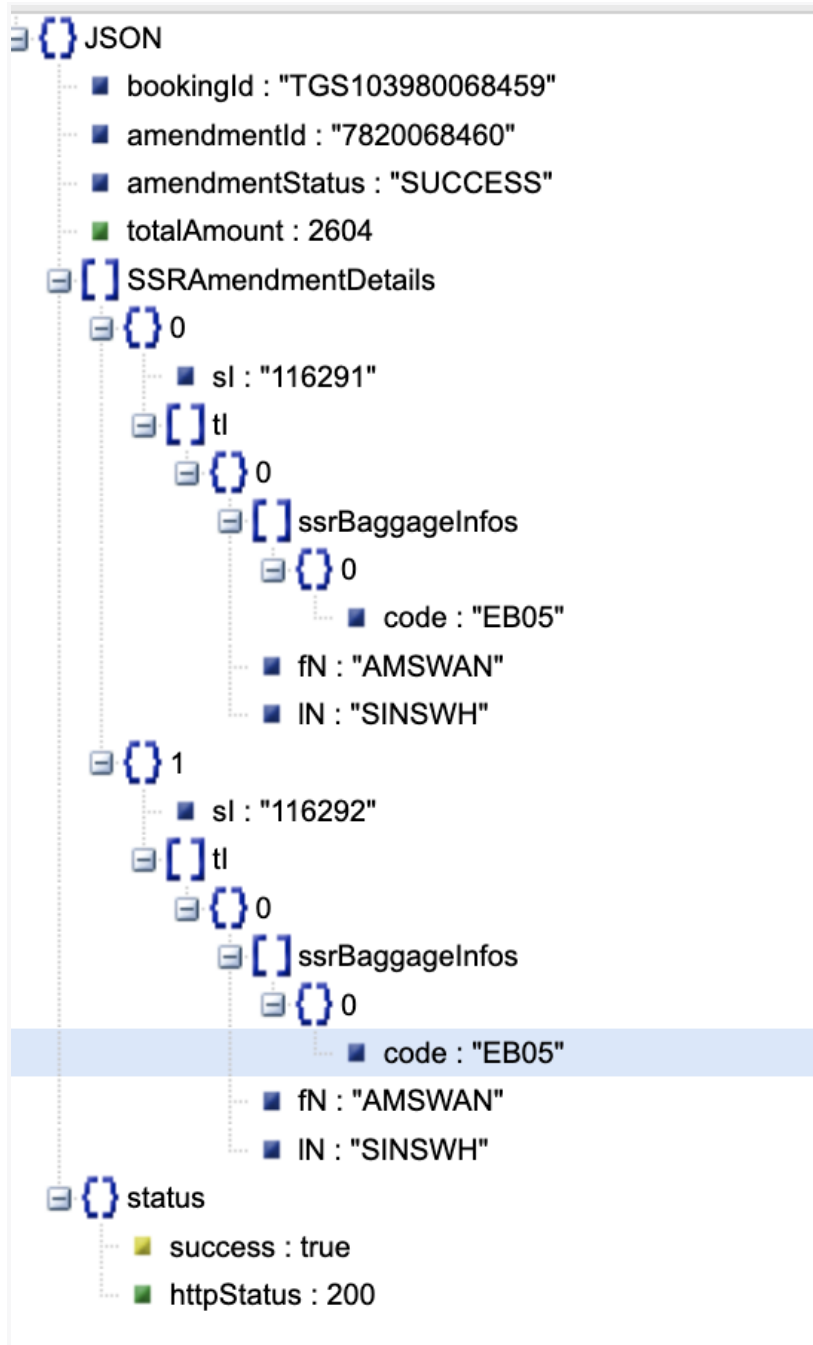
Request Sample:

 **JSON**
amendmentId : "7820068460"

Request Structure

- **amendmentId**: AmendmentId (from ADD SSR Response)

Response Structure :



Response Structure : It consists of the amendment details

bookingId-BookingId

amendmentId-AmendmentId

amendmentStatus-Status of amendment

- ◆ REJECTED - Amendment is rejected.
- ◆ SUCCESS - Amendment modification has been successful with all changes.
- ◆ PENDING - Amendment has been in progress

totalAmount-Total amount paid for SSR

SSRAmendmentDetails - : It will contain array of Segment and segment will contain following information

- ◆ **ID**: Segment Id, id of the segment.
- ◆ **tI** : List of Travelers
- ◆ **Id** : id of a pax for whom the SSR is being added.
- ◆ **ssrSeatInfos**: It will contain information of the already selected seat, and will be empty in case no seat is selected
- ◆ **ssrBaggageInfos** : It contains information of already selected Baggage , it will be empty in case no baggage is selected.
- ◆ **ssrMealInfos** : It contains information of an already selected Meal , it will be empty in case no baggage is selected.
- ◆ **fN** - First Name of the Traveler
- ◆ **IN** - Last Name of the Traveler

