OTST | UIC

SCENARIO SOLUTION DESCRIPTION

REVIEWED AND APPROVED FOR:	BY (FUNCTION, NAME):	DATE AND SIGNATURE:

ISSUE	AUTHOR	DATE	CHAPTER MODIFIED	MODIFICATIONS

TABLE DES MATIERES

1 OTST_TS_OB_RFND_1

1.1 Description

Scenario with an offer request based on a trip specification, booking request based on the response of the offer response. The booking will be fulfilled.

After fulfilling the booking request a refund, retrieve it, patch it with status confirmed. Retrieve the booking to test the refund was processed correctly.

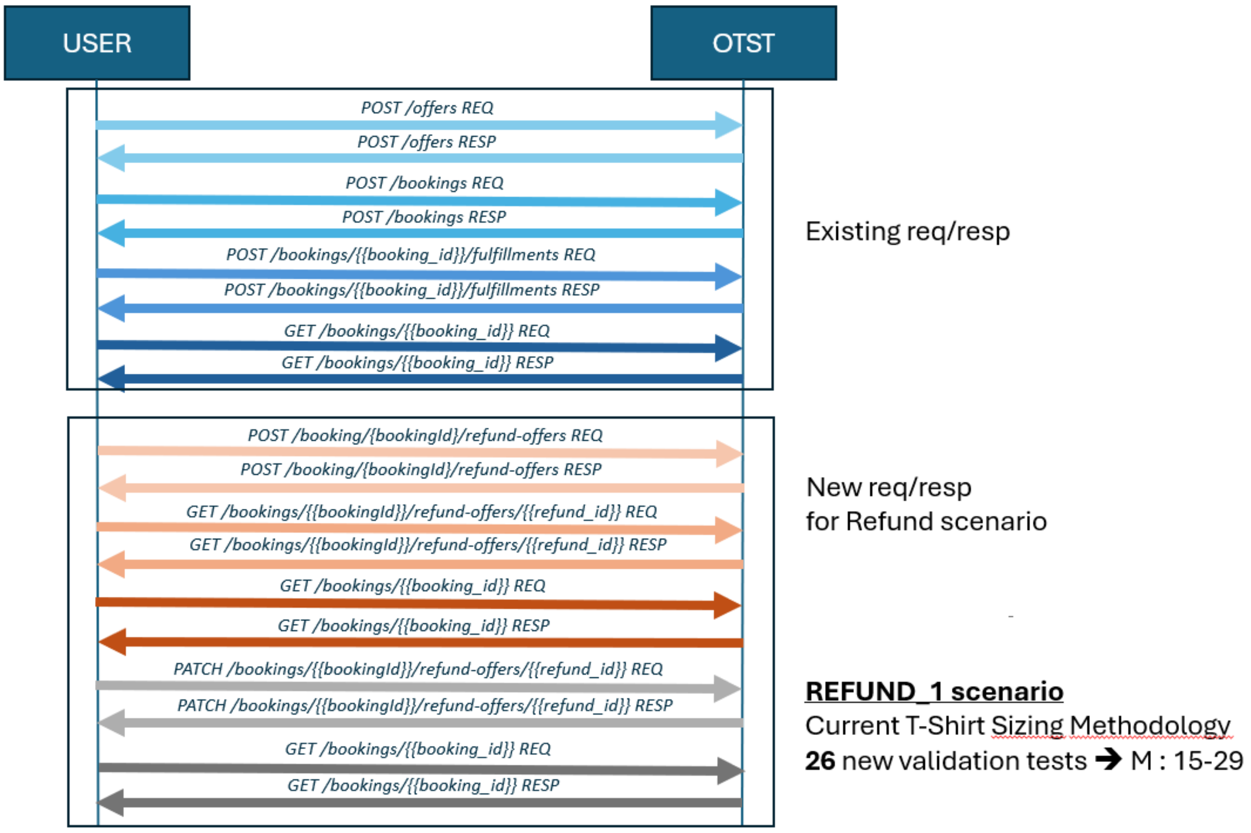
Please note the GET refund offer service could not be implemented and can result in a HTTP 501 (not implemented) response. The scenario should be able to handle this. Also note that in this scenario there are two possible outcomes. In the OSDM technical meeting it was decided that refundOffers do not need to be persisted. Therefore, after patching the refund offer you have two possibilities:

- Refunded BookedOfferParts without a refundOffer

- Refunded BookedOfferParts with a confirmed refundOffer

The trip is based on a single outbound leg.

1.2 Sequence diagram


USER
OTST
POST /offers REQ
POST /offers RESP
POST /bookings REQ
POST /bookings RESP
POST /bookings/{{booking_id}}/fulfillments REQ
POST /bookings/{{booking_id}}/fulfillments RESP
GET /bookings/{{booking_id}} REQ
GET /bookings/{{booking_id}} RESP
Existing req/resp
POST /booking/{bookingId}/refund-offers REQ
POST /booking/{bookingId}/refund-offers RESP
GET /bookings/{{bookingId}}/refund-offers/{{refund_id}} REQ
GET /bookings/{{bookingId}}/refund-offers/{{refund_id}} RESP
GET /bookings/{{booking_id}} REQ
GET /bookings/{{booking_id}} RESP
PATCH /bookings/{{bookingId}}/refund-offers/{{refund_id}} REQ
PATCH /bookings/{{bookingId}}/refund-offers/{{refund_id}} RESP
GET /bookings/{{booking_id}} REQ
GET /bookings/{{booking_id}} RESP
New req/resp for Refund scenario
REFUND_1 scenario
Current T-Shirt Sizing Methodology
26 new validation tests → M : 15-29

1.3 Steps

1. Generate and send a post offer request with a trip specification
2. Generate and send a post booking request
3. Generate and send a post fulfillment request
4. Generate and send a get booking request
5. Generate and send a post refund offer request
6. Generate and send a get refund offer request
7. Generate and send a get booking request
8. Generate and send a patch refund offer request
9. Generate and send a get booking request

1.3.1 Generate and send a post offer request with a trip specification / get offer

Endpoint : POST */offers*

Existing validation :

- ✓ Status code 200
- ✓ validateOfferConformsToOfferSearchCriteria
- ✓ validateOfferResponse

1.3.2 Generate and send a post booking request

Endpoint : POST */bookings*

Existing validation :

- ✓ Status code 200
- ✓ validateBookingResponse

1.3.3 Generate and send a post fulfillment request

Endpoint : POST */bookings/{{booking_id}}/fulfillments*

Existing validation :

- ✓ Status code 200

1.3.4 Generate and send a and send a get booking request

Endpoint : GET */bookings/{{booking_id}}*

Existing validation :

- ✓ Status code 200
- ✓ checkFulFilledBooking

1.3.5 Generate and send a post refund offer request

Endpoint : *POST /booking/{bookingId}/refund-offers*

Existing validation :

- ✓ Status code 200
- ✓ refundOffers element is present : *pm.expect(pm.response.json().refundOffers).to.not.be.empty;*

Response and new validation :

```
"refundOffers": [
  {
    "id": "string",
    "summary": "Refund Offer for Paris-Barcelona Andre Dupont 2022-07-23",
    "createdOn": "2024-07-04T16:01:02.115Z",
    "validFrom": "2024-07-04T16:01:02.115Z",
    "validUntil": "2024-07-04T16:01:02.115Z", #1 validUntil > now() + 10 min??
    "confirmedOn": "2024-07-04T16:01:02.115Z",
    "status": "PROPOSED", #2 <- check status = PROPOSED
    "reimbursementStatus": "IMMEDIATE",
    "reimbursementDateTime": "2024-07-04T16:01:02.115Z",
    "appliedOverruleCode": "string", #3 <- check is omitted

    "fulfillments": [
      {
        "id": "string", #4 <- returned and check equals the fulfillmentId from step 4
        "status": "AVAILABLE",
        "bookingRef": "string",
        "summary": "string",
        "createdOn": "2024-07-04T16:01:02.115Z",
        "controlNumber": "string",
        "bookingParts": […],
        "availableUsage": {…},
        "issuer": "string",
        "fulfillmentDocuments": […],
        "fulfillmentParts": […],
        "securityFeatureLinks": […],
        "_links": […]
      }
    ],
    "issuedFulfillments": […],
      "issuedVouchers": […],
      "refundFee": {
        "currency": "CHF",
        "amount": 0, #5  <- should be populated
        "scale": 2,
        "vats": […]
      },

      "refundableAmount": {
        "currency": "CHF",
        "amount": 25, #6  <- should be populated and equal the confirmedPrice from step 4
        "scale": 2,
        "vats": […]
      },
      "refundOfferBreakdown": {…},
        "refundableAmount": {
          "currency": "CHF",
          "amount": 0,
          "scale": 2,
          "vats": […]
        },
        "bookingParts": […]
      },
      "reimbursementMethod": {
        "paymentMethod": "string"
      },
      "_links": […],
  "_links": […]
}
```

1.3.6 Generate and send a get refund offer request

Endpoint : *GET /bookings/{{bookingId}}/refund-offers/{{refund_id}}*

Existing validation :

- ✓ Status code 200

Response and new validation :

```
{
    "refundOffer": {
        "id": "string",
        "summary": "Refund Offer for Paris-Barcelona Andre Dupont 2022-07-23",
        "createdOn": "2024-07-05T08:48:07.387Z",
        "validFrom": "2024-07-05T08:48:07.387Z",
        "validUntil": "2024-07-05T08:48:07.387Z", #7 <- check validUntil is >now
        "confirmedOn": "2024-07-05T08:48:07.387Z",
        "status": "PROPOSED", #8 <- check status = PROPOSED
        "reimbursementStatus": "IMMEDIATE",
        "reimbursementDateTime": "2024-07-05T08:48:07.387Z",
        "appliedOverruleCode": "string", #9 <- check is omitted
        "fulfillments": [
            {
                "id": "string", #10 <- check equals the fulfillmentID from step 4
                "status": "AVAILABLE",
                "bookingRef": "string",
                "summary": "string",
                "createdOn": "2024-07-05T08:48:07.387Z",
                "controlNumber": "string",
                "bookingParts": […],
                "availableUsage": {…},
                "issuer": "string",
                "fulfillmentDocuments": […],
                "fulfillmentParts": […],
                "_links": […]
            }
        ],
        "issuedFulfillments": […],
        "issuedVouchers": […],
        "refundFee": {
            "currency": "CHF",
            "amount": 0, #11  <- should be 0
            "scale": 2,
            "vats": […]
        },
        "refundableAmount": {
            "currency": "CHF",
            "amount": 25, #12  <- equal the refundableAmount from step 5
            "scale": 2,
            "vats": […]
```

```
          },
          "refundOfferBreakdown": {…},
          "_links": […]
      }
  }
```

1.3.7 Generate and send a get booking request

Endpoint : *GET /bookings/{{booking_id}}*

Existing validation :

- ✓ Status code 200

Response and new validation :

```
{
  "booking": {
    "id": "string",
    "bookingCode": "string",
    "externalRef": "string",
    "summary": "Booking 2345 for Clemens Gantert",
    "createdOn": "2024-07-05T08:59:06.634Z",
    "passengers": […],
    "purchaser": {…},
    "provisionalPrice": {…},
    "confirmedPrice": {…},
    "bookedOffers": […],
    "trips": […],
    "requestedInformation": "string",
    "confirmationTimeLimit": "2024-07-05T08:59:06.636Z",
    "fulfillmentType": "string",
    "fulfillments": […],
    "issuedVouchers": […],
    "documents": […],
    "paymentMethods": […],
    "trips": […],
    "refundOffer": {
            "id": "string",
            "summary": "Refund Offer for Paris-Barcelona Andre Dupont 2022-07-23",
            "createdOn": "2024-07-05T08:48:07.387Z",
            "validFrom": "2024-07-05T08:48:07.387Z",
            "validUntil": "2024-07-05T08:48:07.387Z", #13 <- check validUntil is >now
            "confirmedOn": "2024-07-05T08:48:07.387Z",
            "status": "PROPOSED", #14 <- check status = PROPOSED
            "reimbursementStatus": "IMMEDIATE",
            "reimbursementDateTime": "2024-07-05T08:48:07.387Z",
            "appliedOverruleCode": "string", #15 <- check is omitted
```

```
                "fulfillments": [
                    {
                        "id": "string", #16 <- check equals the fulfillmentID from step 4
                        "status": "AVAILABLE",
                        "bookingRef": "string",
                        "summary": "string",
                        "createdOn": "2024-07-05T08:48:07.387Z",
                        "controlNumber": "string",
                        "bookingParts": […],
                        "availableUsage": {…},
                        "issuer": "string",
                        "fulfillmentDocuments": […],
                        "fulfillmentParts": […],
                        "_links": […]
                    }
                ],
                "issuedFulfillments": […],
                "issuedVouchers": […],
                "refundFee": {
                    "currency": "CHF",
                    "amount": 0, #17  <- should be 0
                    "scale": 2,
                    "vats": […]
                },
                "refundableAmount": {
                    "currency": "CHF",
                    "amount": 25, #18  <- equal the refundableAmount from step 6
                    "scale": 2,
                    "vats": […]
                },
                "refundOfferBreakdown": {…},
                "_links": […]
            },
             "releaseOffers": […],
             "cancelFulfillmentsOffers": […],
             "exchangeOperations": […],
             "relatedBookingIds": […],
             "_links": […]
        }
```

1.3.8 Generate and send a patch refund offer request

Endpoint : *PATCH /bookings/{{bookingId}}/refund-offers/{{refund_id}}*

Existing validation :

- ✓ Status code 200

Response and new validation :

```
{
    "refundOffer": {
        "id": "string",
        "summary": "Refund Offer for Paris-Barcelona Andre Dupont 2022-07-23",
        "createdOn": "2024-07-05T09:14:58.822Z",
        "validFrom": "2024-07-05T09:14:58.822Z",
        "validUntil": "2024-07-05T09:14:58.822Z", #19 <- check validUntil is >now
        "confirmedOn": "2024-07-05T09:14:58.822Z",
        "status": "CONFIRMED", #20 <- check status = CONFIRMED
        "reimbursementStatus": "IMMEDIATE",
        "reimbursementDateTime": "2024-07-05T09:14:58.822Z",
        "appliedOverruleCode": "string", #21 <- check is omitted
        "fulfillments": [
            {
                "id": "string", #22 <- check equals the fulfillmentID from step 4
                "status": "AVAILABLE",
                "bookingRef": "string",
                "summary": "string",
                "createdOn": "2024-07-05T09:14:58.822Z",
                "controlNumber": "string",
                "bookingParts": […],
                 "availableUsage": {…},
                 "issuer": "string",
                 "fulfillmentDocuments": […],
                 "fulfillmentParts": […],
                 "securityFeatureLinks": […],
                 "_links": […]
            }
        ],
            "issuedFulfillments": […],
            "issuedVouchers": […],
        "refundFee": {
            "currency": "CHF",
            "amount": 0, #23  <- should be 0
            "scale": 2,
            "vats": [
                {
                    "countryCode": "DE",
                    "amount": 5,
                    "scale": 2,
                    "percentage": 0,
                    "taxId": "string",
                    "scope": "string"
                }
            ]
        },
        "refundableAmount": {
            "currency": "CHF",
            "amount": 25, #24  <- equal the refundableAmount from step 7
            "scale": 2,
            "vats": [
                {
                    "countryCode": "DE",
                    "amount": 5,
                    "scale": 2,
                    "percentage": 0,
                    "taxId": "string",
                    "scope": "string"
                }
            ]
        },
        "refundOfferBreakdown": {…},
        "_links": […]
    }
```

1.3.9 Generate and send a get booking request

Endpoint : *GET /bookings/{{booking_id}}*

Existing validation :

- ✓ Status code 200

Response and new validation :

```
{
   "booking": {
    "id": "string",
    "bookingCode": "string",
    "externalRef": "string",
    "summary": "Booking 2345 for Clemens Gantert",
    "createdOn": "2024-07-05T09:54:59.614Z",
    "passengers": […],
    "purchaser": {…},
    "provisionalPrice": {…},
    "confirmedPrice": {…},
    "bookedOffers": [
        {
            "offerId": "string",
            "summary": "string",
            "admissions": [
                {
                    "objectType": "string",
                    "id": "string",
                    "summary": "string",
                    "createdOn": "2024-07-05T09:54:59.614Z",
                    "confirmableUntil": "2024-07-05T09:54:59.614Z",
                    "validFrom": "2024-07-05T09:54:59.614Z",
                    "validUntil": "2024-07-05T09:54:59.614Z",
                    "confirmedOn": "2024-07-05T09:54:59.614Z",
                    "price": {…},
                    "refundAmount": {
                        "currency": "CHF",
                        "amount": 25, #25  <- has a value and equal the refundableAmount
from step 8
                        "scale": 2,
                        "vats": […]
                    ]
                      },
                    "tripCoverage": {…},
                    "summaryProductId": "string",
                    "products": […],
                    "status": "REFUNDED", #26 <- check status = REFUNDED
                    "offerMode": "COLLECTIVE",
                    "bookingPartCode": "string",
                    "passengerIds": […],
                    "availableFulfillmentOptions": […],
                    "refundable": "YES",
                    "exchangeable": "YES",
                    "afterSaleConditions": […],
                    "appliedCorporateCodes":[…],
                    "appliedPassengerTypes": […],
                    "appliedPromotionCodes":[…],
                    "appliedReductions": […],
                    "indicatedConsumption": {…},
                    "accountingRef": {…},
                    "isReservationRequired": false,
                    "feeRefs": […],
                    "reservationRefs": […],
                    "ancillaryRefs": […],
                    "regulatoryConditions": […]
                }
            ],
          "reservations": […],
          "ancillaries": […],
          "fees": […],
          "fares": […],
          "tripCoverage": {…},
          "appliedThroughTicketTags": […],
          "products": […]
                }
          ],
    "trips": […],
    "requestedInformation": "string",
    "confirmationTimeLimit": "2024-07-05T09:54:59.615Z",
    "fulfillmentType": "string",
    "fulfillments": [
          {
            "id": "string",
            "status": "REFUNDED", #27 <- check status = REFUNDED
            "bookingRef": "string",
            "summary": "string",
            "createdOn": "2024-07-05T09:54:59.615Z",
            "controlNumber": "string",
            "bookingParts": […],
            "availableUsage": {…},
            "issuer": "string",
            "fulfillmentDocuments": […],
            "fulfillmentParts": […],
            "_links": […]
        }
```

1.4 Variables

1.4.1 Reused variables

- leg_1_start_stop_place_ref
- leg_1_start_datetime
- leg_1_end_stop_place_ref
- leg_1_end_datetime
- leg_1_product_category_ref
- leg_1_product_category_name
- leg_1_product_category_short_name
- leg_1_vehicle_number
- leg_1_operator_code
- OFFER. SEARCH_CRITERIA_CURRENCY
- OfferPartType.RESERVATION
- ServiceClassType.STANDARD
- TravelClass.SECOND
- FulfillmentOptionType.ETICKET
- FulfillmentMediaType.PDF_A4
- OFFER.PASSENGER_SPECIFICATIONS
- OFFER.SEARCH_CRITERIA
- OFFER.FULFILLMENT_OPTIONS
- SCENARIO_TYPE
- offer_id
- offer
- offers
- booking_external_ref
- booking_id
- passenger_id
- PREBOOKED
- FULFILLED
- refund_id

1.4.2 New variables

- Status
- refundFeeAmount
- refundableAmount
- refundAmount
- fulfillmentsId
- validUntil
- admissionsStatus
- appliedOverruleCode
- refundOffer

1.5 Library update (function.js implementation)

Existing methods has been analysed to keep the format for new method

Below is the list of the new methods to implement with parameters

- osdmRefundSearchCriteria(legDefinitions, overruleCode) ➔ Matching happy and non-happy flow with mandatory parameter
- osdmRefundSpecification(legDefinitions, overruleCode) ➔ Do we need separate method for Refund specifications ?
- validateRefundResponse(OFFER.PASSENGER_SPECIFICATIONS, OFFER.OVERRULECODE_OPTIONS, refundId, SCENARIO_TYPE)
    - validationLogger
    - refundOffers element is present (Correct refund offer is returned)
    - passenger element is present and returned
    - refundOffer.validUntil ➔ Used to check: 1; 7; 13; 19
    - refundOffer.status ➔ Used to check: 2; 8; 14; 20
    - appliedOverruleCode ➔ Used to check: 3; 9; 15; 21
    - refundOffer.fulfillmentId ➔ Used to check: 4;10; 16; 22
    - refundFee ➔ Used to check: 5; 11; 17; 23
    - refundableAmount ➔ Used to check: 6;12; 18; 24
    - bookedOffer.admissions.refundAmount ➔ Used to check: 25
    - bookedOffer.admissions.status ➔ Used to check: 26
    - fulfillments.status ➔ Used to check: 27
- checkFulfilledRefund
    - validationLogger
    - Correct price is used on refund, compared to offer
    - passenger element is present and returned
- validateRefundResponse (PATCH ? Using same checks as *validateRefundResponse*)
- validateOfferConformsToOfferSearchCriteria(offer)
    - Refund is available
    - Refund with OverruleCode is available
- validateGetOfferResponseAfterRefund
    - validationLogger
    - Refund offers are returned
    - Correct Refund with overruleCode are returned

END OF DOCUMENT