

OTST_TS_OB_RFND_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.

The trip is based on a single outbound leg

Scenario

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

Is this one needed to be certified?

Example data

```
{
  "type": "OTST_TS_OB_RFND_1",
  "currency": "EUR",
  "service_class": "STANDARD",
  "travel_class": "SECOND",
  "origin": "urn:uic:stn:8400058",
  "destination": "urn:uic:stn:8727100",
  "start_datetime": "%TRIP_DATE%T07:00:00+02:00",
  "end_datetime": "%TRIP_DATE%T10:20:00+02:00",
  "product_category_ref": "",
  "product_category_name": "",
  "product_category_short_name": "",
  "vehicle_number": "OSDM_101",
  "operator_code": "urn:uic:rics:1"
}
```

Verifications:

1-4. Validated by scenario 1

5. Check that refundOffer are returned with status PROPOSED

- validUntil > now() + 10 min??

- appliedOverrideCode is omitted

- The posted fulfillmentId is returned

- refundFee and refundAmount are populated

6. If implemented, same validations as for step 5

7. The refundOffer is a part of the booking with same validations as for step 5.

8. - same verification as for step 5, but status = CONFIRMED

9. - fulfillments.status = REFUNDED

- bookedOffer.admissions.status = REFUNDED

- bookedOffer.admissions.refundAmount has a value

Suggested Validations per step

TBD

OTST_TS_OB_RFND_2

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 with a specific overrule code, retrieve it, patch it with status confirmed

Retrieve the booking to test the refund was processed correctly.

The trip is based on a single outbound leg

Scenario

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

Example data

```
{
  "type": "OTST_TS_OB_RFND_2",
  "currency": "EUR",
  "service_class": "STANDARD",
  "travel_class": "SECOND",
  "origin": "urn:uic:stn:8400058",
  "destination": "urn:uic:stn:8727100",
  "start_datetime": "%TRIP_DATE%T07:00:00+02:00",
  "end_datetime": "%TRIP_DATE%T10:20:00+02:00",
  "product_category_ref": "",
  "product_category_name": "",
  "product_category_short_name": "",
  "vehicle_number": "OSDM_101",
  "operator_code": "urn:uic:rics:1",
  "overruleCode": "PAYMENT_FAILURE"
}
```

Verifications:

- 1-4. Validated by scenario 1
5. Check that refundOffer are returned with status PROPOSED
 - validUntil > now() + 10 min??
 - appliedOverruleCode is omitted
 - The posted fulfillmentId is returned
 - refundFee and refundAmount are populated
 - appliedOverruleCode = "PAYMENT_FAILURE"
6. If implemented, same validations as for step 5
7. The refundOffer is a part of the booking with same validations as for step 5.
8. - same verification as for step 5, but status = CONFIRMED
9. - fulfillments.status = REFUNDED
 - bookedOffer.admissions.status = REFUNDED
 - bookedOffer.admissions.refundAmount has a value

OTST_TS_OB_RFND_3

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 with a overrule code that is not configured
The trip is based on a single outbound leg

Scenario

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
- No refundOfferId to make get with

Example data

```
{
  "type": "OTST_TS_OB_RFND_3",
  "currency": "EUR",
  "service_class": "STANDARD",
  "travel_class": "SECOND",
  "origin": "urn:uic:stn:8400058",
  "destination": "urn:uic:stn:8727100",
  "start_datetime": "%TRIP_DATE%T07:00:00+02:00",
  "end_datetime": "%TRIP_DATE%T10:20:00+02:00",
  "product_category_ref": "",
  "product_category_name": "",
  "product_category_short_name": "",
  "vehicle_number": "OSDM_101",
  "operator_code": "urn:uic:rics:1",
  "overruleCode": "CODE_DOES_NOT_EXIST"
}
```

Verifications:

- 1-4. Validated by scenario 1
- 5. http code 406 (Not Acceptable)
- 6. - fulfillments.status = FULFILLED
 - bookedOffer.admissions.status = FULFILLED
 - bookedOffer.admissions.refundAmount has no value

OTST_TS_OB_RFND_4

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 with a overrule code. Check the booking and delete the refund offer. Retrieve the booking again to make sure the refundOffer is deleted.

The trip is based on a single outbound leg

Scenario

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 delete refund offer request
9. Generate and send a get booking request
10. Generate and send a get refund offer request **No refundOfferId to fill parameter**

Example data

```
{
  "type": "OTST_TS_OB_RFND_4",
  "currency": "EUR",
  "service_class": "STANDARD",
  "travel_class": "SECOND",
  "origin": "urn:uic:stn:8400058",
  "destination": "urn:uic:stn:8727100",
  "start_datetime": "%TRIP_DATE%T07:00:00+02:00",
  "end_datetime": "%TRIP_DATE%T10:20:00+02:00",
  "product_category_ref": "",
  "product_category_name": "",
  "product_category_short_name": "",
  "vehicle_number": "OSDM_101",
  "operator_code": "urn:uic:rics:1",
  "overruleCode": "PAYMENT_FAILURE"
}
```

Verifications:

- 1-4. Validated by scenario 1
- 5. Check that refundOffer are returned with status PROPOSED
 - validUntil > now() + 10 min??
 - appliedOverruleCode = PAYMET_FAILURE
 - The posted fulfillmentId is returned
 - refundFee and refundAmount are populated
 - appliedOverruleCode = "PAYMENT_FAILURE"
- 6. If implemented, same validations as for step 5
- 7. The refundOffer is a part of the booking with same validations as for step 5.
- 8. http code 204
- 9. - The refundOffer is not presented