Refund Protect Platform

RESTfull API call

Introduction

Via available online service and through specified API, developers can connect to the Protect platform and submit individual sales transactions for Refund Protect. Service is available 24/7 and in order to have access it is mandatory to have an Agent account on the Protect platform. New Agents who do not have an account can start the registration process <here>.

There are two basic API calls available through the platform:

- Submit new sales offering with sales result.
- Cancelling already submitted (& sold) product.

Submitting transaction with REST API

The Refund Protect API is a RESTful web service. Character encoding is UTF-16. For adding new transaction, a PUT/POST http method is used. Other methods will return an error message.

Each transaction log entry has to have a predefined set of attributes and must conform to the specified rules. Requests to service must include following:

Request URL:

URL for **UAT** environment:

```
https://api-uat.protect-platform.com/api/v1/refundprotect/salesoffering
```

URL for **Production** environment:

```
https://api.protect-platform.com/api/v1/refundprotect/salesoffering
```

Request Method:

- Accepted METHODs: **POST**

Request HEADER:

- Content-Type: application/json
- X-RefundProtect-VendorId: *<unified vendor Id provided by the platform>*
- X-RefundProtect-AuthToken: *<please refer to authentication section>*

Important notice

All transaction offers must be submitted, including the ones rejected by the customers, to the Protect platform. This enables us and you to understand the performance of Refund Protect and support you with recommendations on improving sales conversion to maximize revenue.

Request CONTENT:

Data format is JSON, with the following structure:

```
{
  "vendorCode": "DemoVendorId ",
  "vendorSalesReferenceId": "UniqueID-9166-9877-1234-3211",
  "customerFirstName": "John",
  "customerLastName": "Doe",
  "products": [
    {
      "productCode": "HTL",
      "currencyCode": "USD",
      "productPrice": 100,
      "premiumRate": 3.5,
      "offeringMethod": "OPT-OUT",
      "sold": true,
      "insuranceEndDate": "2018-06-15T23:59:59.999"
    },
    {
      "productCode": "HTL",
      "currencyCode": "USD",
      "productPrice": 200,
      "premiumRate": 3.5,
      "offeringMethod": "OPT-IN",
      "sold": false,
      "insuranceEndDate": "2018-06-15T23:59:59.999"
    }
  ]
```

Request timeout:

In case no response is returned from the API, a request timeout should be set to 10 seconds.

Data structure

Data structure is separated into two entities:

- Transaction related data
- Product & sales related data (part of transaction related data)

Transaction data

Data field	Mandatory	Data type	Description
vendorCode	**YES**	Character	Your Vendor ID provided to you by Event Protect
vendorSalesReferenceId	**YES**	Character	Your unique sales reference ID that is visible to customer (needed in case of claim)
vendorSalesDate	**YES**	Character	Sales date, current UTC date if not sent
customerFirstName	**YES**	Character	Customer's first name
customerLastName	**YES**	Character	Customer's last name
products	**YES**	See below*	Sales data array nested as products field.

*described in following table

Products

Data field	Mandatory	Data type	Description
productCode	**YES**	Character	Requested product code. See list of available product codes below.
currencyCode	**YES**	Character	Currency code. This value must conform to ISO 4217 specification (three letter code). Reference: https://en.wikipedia.org/wiki/ISO_4217
productPrice	**YES**	Numeric	Product cost, i.e. the insured value. This value must have decimal point set to ".".
sold	**YES**	boolean	TRUE if sold, FALSE if not sold.
premiumRate	**YES**	Numeric	Premium offered/charged in terms of rate (i.e. %). E.g. 3.5 means 3.5 percent. This value must have decimal point set to ".".
offeringMethod	**NO**	Character	Offering method that was used when making the offer to the customer. Possible values: • OPT-IN, • OPT-OUT, • SELECT (Customer must explicitly select whether he/she does or does not want protection) If not supplied, the system will use default value that is set up within the platform, during account creation with Refund Protect team.
insuranceEndDate	**YES IF SOLD**	Character	The date when insurance expires. Empty if no insurance has been sold. This value must conform to ISO 8601 UTC format (YYYY-MM-DDTHH:mi:SS.ssssss). Reference: https://en.wikipedia.org/wiki/ISO_8601.

Product codes

Below are listed allowed product codes.

Product Type	Product Type	Product Type
TKT	Ticket	This is any type of Ticket for an event, transportation or any other non-refundable ticket for the purpose of admission.
PKG	Package	This is any package that is non-refundable it may include accommodation, travel, tickets, etc.
HTL	Hotel	This is any type of accommodation booking which is non-refundable.

Response signals

Depending on input parameters and service status following response codes are possible:

Code	Text	Description
200	OK	OK signal
400	Bad Request	Non implemented URI call
401	Unauthorized	Missing authorization
405	Method Not Allowed	Non PUT/POST method
408	Request Timeout	Response taking too much time to complete
413	Request Entity Too Large	Submitted entity too large
422	Non-processable Entity	Missing data (data submitted not ok; e.g. missing price)
500	Internal Server Error	Internal error occurred. No data stored.
507	Insufficient Storage	In case no data

Request example

```
POST /api/v1/refundprotect/salesoffering HTTP/1.1
Host: api.protect-platform.com
Content-Type: application/json; charset=utf-8
X-RefundProtect-VendorId: DemoVendorId
X-RefundProtect-AuthToken:
FUTYDacR1XH+agt0fGNSrzk+MLwq4r5hUVCJMEQVWrY=
Cache-Control: no-cache

{
  "vendorCode": "DemoVendorId ",
  "vendorSalesReferenceId": "UniqueID-9166-9877-1234-3211",
  "customerFirstName": "John",
  "customerLastName": "Doe",
  "products": [
    {
      "productCode": "HTL",
      "currencyCode": "USD",
      "productPrice": 100,
      "premiumRate": 3.5,
      "offeringMethod": "OPT-OUT",
      "sold": true,
      "insuranceEndDate": "2018-06-15T23:59:59.999"
    },
    {
      "productCode": "HTL",
      "currencyCode": "USD",
      "productPrice": 200,
      "premiumRate": 4.0,
      "offeringMethod": "OPT-IN",
      "sold": false,
      "insuranceEndDate": "2018-06-15T23:59:59.999"
    }
  ]
```

Validation of requests

Before you get credentials, developers can start coding the connection to the Protect platform.

For that purpose, the TEST API service has been created. It is a different URL, but offers the same functionality as a regular API, just no data is saved. This service will return one of the response values, depending on submitted data.

To use test service, it is necessary to use test URL and demo credential data, so header call will have following parameters.

Request URL & Method:

- URL for **UAT** environment:
  - `https://api-uat.protect-platform.com/api/v1/refundprotect/salesoffering`
- URL for **Production** environment:
  - `https://api.protect-platform.com/api/v1/refundprotect/salesoffering`
- Accepted METHOD:
  - **POST** or **PUT**
- Test Vendor ID:
  - DemoVendorId
- Test Vendor API key:
  - DemoAPIKey

For test service, you can also use your own Vendor ID & API key provided to you by the platform.

Cancelling transaction with REST API

Request URL & Method:

URL for **UAT** environment:

```
https://api-uat.protect-
platform.com/api/v1/refundprotect/salesoffering/{booking_ref_number}
```

URL for **Production** environment:

```
https://api.protect-
platform.com/api/v1/refundprotect/salesoffering/{booking_ref_number}
```

Request Method:

- Accepted METHODs: **DELETE**

Request HEADER:

- Content-Type: application/json
- X-RefundProtect-VendorId: *<unified vendor Id provided by the platform>*
- X-RefundProtect-AuthToken: *<please refer to authentication section>*

Response Signals

Depending on input parameters and service status following response codes are possible:

Code	Text	
200	OK	OK signal
400	Bad Request	Non implemented URI call
401	Unauthorized	Missing authorization
404	Not Found	Transaction with specified booking number not found
405	Method Not Allowed	Non DELETE method
408	Request Timeout	Long response
409	Conflict	Transaction already cancelled
422	Non-processable Entity	Missing or invalid data
500	Internal Server Error	Internal error occurred. No data stored.
507	Insufficient Storage	In case no data

Authentication

In order to submit transactions each method call needs to be authenticated. Authentication method to sign & authenticate REST requests from 3rd party applications is HMAC-SHA256. For authentication process it is necessary to have authentication token which is sent in request HEADER in field "X-RefundProtect-AuthToken". Authentication token is created by combining two values

- **vendorID**
  Unified vendor Id provided by the platform

- **current date**
  UTC date on which data is sent in ISO 8601 DATE format (YYYY-MM-DD). For example, 15th of June 2018 would be 2018-06-15

Once these two values are combined, HMAC-SHA256 digest of this value is generated using your API Key.

Finally, this value must be base64 encoded.
Both **vendorID** and the **API key** are provided by the platform and are available under members' access.

AuthToken Examples

Date	Vendor ID	Message	Key	AuthToken
2016-10-05	DemoVendorId	DemoVendorId2016-10-05	DemoAPIKey	TsfF7o5c1OxcEZkwH47R16+P7pQRpGYo7cEBebdFrRU=
2016-10-06	DemoVendorId	DemoVendorId2016-10-06	DemoAPIKey	FUTYDacR1XH+agt0fGNSrzk+MLwq4r5hUVCJMEQVWrY=

Code examples for AuthToken

C#

```
1.  string vendorID = "DemoVendorId";
2.  string APIKey = "DemoAPIKey";
3.
4.  var encoding = new ASCIIEncoding();
5.  byte[] keyByte = encoding.GetBytes(APIKey);
6.  byte[] messageBytes = encoding.GetBytes(vendorID + DateTime.UtcNow.ToString("yyyy-MM-dd"));
7.  using (var hmacsha256 = new HMACSHA256(keyByte))
8.  {
9.      byte[] hashmessage = hmacsha256.ComputeHash(messageBytes);
10.     return Convert.ToBase64String(hashmessage);
11. }
12.
```

PHP

```php
$vendorId = "DemoVendorId";
$key = "DemoAPIKey";

$date = gmdate("Y-m-d");
$message = $vendorId . $date;

$hash = hash_hmac('sha256', $message, $key);
$hash = pack('H*', $hash);

$authTokenB64 = base64_encode($hash);
```

Python

```python
import base64
import hmac
from hashlib import sha256
from datetime import datetime

VENDOR_ID = 'DemoVendorId'
API_KEY = 'DemoAPIKey'


def gen_base64_digest_str(message, key):
    hmac_gen = hmac.new(bytes(key, 'ascii'), digestmod=sha256)
    hmac_gen.update(bytes(message, 'ascii'))

    return base64.b64encode(hmac_gen.digest()).decode('ascii')


def create_token():
    utcDateISO = datetime.utcnow().strftime('%Y-%m-%d')
    message = VENDOR_ID + utcDateISO

    return gen_base64_digest_str(message, API_KEY)


print 'Valid token: ' + create_token()
```

Java

```java
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.text.SimpleDateFormat;
import org.apache.commons.codec.binary.Base64;


String vendor = "DemoVendorId";
String key = "DemoAPIKey";

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
sdf.setTimeZone(new SimpleTimeZone(SimpleTimeZone.UTC_TIME, "UTC"));

String tstamp = sdf.format(new Date());
String message = vendor + tstamp;

Mac sha256_HMAC = Mac.getInstance("HmacSHA256");
sha256_HMAC.init(new SecretKeySpec(key.getBytes(), "HmacSHA256"));

```

```
19. String authToken = new String (Base64.encodeBase64(sha256_HMAC.doFinal(message.getBytes())));
```

Suggested development process

1. Request your Vendor Id and API key from the platform manager.
2. Develop your code and test it using platform test/demo functionality. Use DEMO Vendor Id and Demo Key.
    - DEMO Vendor Id: DemoVendorId
    - DEMO Vendor Key: DemoAPIKey
    - Make calls to: https://api-uat.protect-platform.com/api/v1/refundprotect/salesoffering
    - Use one of code snippets from this document to make authentication token.
    - Use json data from this document as a content to start with.
    - Use your own data to create API content and validate it.
3. Once credentials have been provided by the platform (Vendor Id, Vendor API Key), test your code with new credentials.
4. Move to UAT:
    - Switch calls to: https://api-uat.protect-platform.com/api/v1/refundprotect/salesoffering
    - Check if your transactions are visible in the platform (leave up to 1hr for submitted transactions to propagate to the platform).
5. Once approved, move to production Environment.
    - Test it: https://api.protect-platform.com/api/v1/refundprotect/salesoffering
    - Move to production: https://api.protect-platform.com/api/v1/refundprotect/salesoffering