EOS Broker Platform

RESTfull API call

Version 1.4 03.08.22

Introduction

The EOS Broker Platform RESTfull API Call is available via an online service and through a specified API. Developers can connect to the EOS platform and submit individual sales transactions. Our service is available 24/7 and in order to have access it is mandatory that you have an Agent account on the EOS platform.

Submitting transaction with REST API

The EOS API is a RESTful web service. Character encoding is UTF-16.

To add a new transaction, a 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 the following:

Request URL & Method:

URL for **UAT** environment:

https://app-eos-presentation-api.azurewebsites.net/Transaction

- **Open API specification**
    - https://app-eos-presentation-api.azurewebsites.net/swagger/v1/swagger.json

URL for **Production** environment:

https://app-eos-prod-api.azurewebsite.net/Transaction

- **Open API specification**
    - https://app-eos-prod-api.azurewebsites.net/swagger/v1/swagger.json

Request Method:

- Accepted METHODs: **POST**

Request HEADER:

- Content-Type: application/json
- x-api-key: *<please refer to authentication section>*

Request CONTENT:

Data format is JSON, with the following structure:

```
{
  "personalFirstName": "John",
  "personalLastName": "Smith",
  "vendorSalesReferenceId": "Haktou01234567889",
  "dateOfPurchase": "2022-08-02T08:41:12.350Z",
  "eventDate": "2022-08-02T08:41:12.350Z",
  "eventName": " North Island - 7 day tour ",
  "currencyCode": "NZD",
  "individualTicketPrice": 1000,
  "quantity": 1,
  "transactionTotalValue": 1100,
  "eventCountry": "NZ"
}
```

Request timeout:

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

Data structure

Data Field	Mandatory	Data Type	Description
vendorTransactionReferenceId	**YES**	Character	Unique transaction ID coming from the Vendor's system. The format allows any type of id (numeric/character). **The combination of vendor id & vendor transaction reference id MUST be UNIQUE**.
personalFirstName	**YES**	Character	Contains the name of the buyer in any format sent from the vendor. The platform does not check this data, but it will be used as additional reference for individual transaction manipulation (i.e. in case of claims or in case of transaction cancellation).

personalLastName	**YES**	Character	Contains the name of the buyer in any format sent from the vendor. The platform does not check this data, but it will be used as additional reference for individual transaction manipulation (i.e. in case of claims or in case of transaction cancellation).
dateOfPurchase	**YES**	Character	The date & time of the transaction. Sent to the platform in ISO 8601 UTC format (YYYY-MM-DDTHH:mi:SS.ssssss+01:00) with time zone. Reference for format: https://en.wikipedia.org/wiki/ISO_8601
eventDate	**YES**	Character	The date of the event. Sent to the platform in ISO 8601 DATE format (YYYY-MM-DD). Reference for format: https://en.wikipedia.org/wiki/ISO_8601
eventName	**YES**	Character	The name of the event. Value in any format is accepted.
currencyCode	**YES**	Character	The code of the currency. Standard ISO 4217 Currency code is accepted (three letter code). Reference for format: https://en.wikipedia.org/wiki/ISO_4217
individualTicketPrice	**YES**	Number	The price of ticket(s) including fees and charges. It is accepted as numeric value with following format (#0.00): ▪ Decimal separator: ".". ▪ Number of digits: 2 ▪ Thousands separator: NO ▪ Leading zero: YES
quantity	**YES**	Number	Number of items in the transaction. It is accepted as integer value with following format (#0)
transactionalTotalValue	**YES**	Number	The total value of the purchace including fees and charges or additions. This is the total value that is going to be insured. It is accepted as numeric value with following format (#0.00): ▪ Decimal separator: ".". ▪ Number of digits: 2 ▪ Thousands separator: NO ▪ Leading zero: YES
eventCountry	**YES**	Character	The code of the country where event is taking place. Standard ISO 3166-1 ALPHA 2 Country code is accepted. Reference for format: https://en.wikipedia.org/wiki/ISO_3166-1

Authentication

In order to submit a transaction each method call needs to be authenticated. For the authentication process it is necessary to have an authentication token which is sent in request HEADER in field “x-api-key”

Response signals

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

Code	Text	
200	OK	OK signal
400	Bad Request	Non implemented URI call
401	Unauthorized	Missing authorization
405	Method Not Allowed	Non POST method
408	Request Timeout	Long response
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

Suggested development process

1. Request your Vendor Id and API key from the EOS Platform Manager - wrodda@eosclaims.com.
2. Develop your code and test it on our UAT environment.
    - Make calls to: https://app-eos-presentation-api.azurewebsites.net/Transaction
    - Use json data from this document as a content to start with.
    - Use your own data to create API content and validate it.
    - Check if your transactions are visible on the platform.
3. Once approved, move to production Environment.
    - Move to production: https://app-eos-prod-api.azurewebsite.net/Transaction