BOA – Transact API

BOA Gaming

Operator Endpoint Requirements

Version 1.1
19/04/2018

Version Control

Version	Date	Author	Description
1.0	04/02/2018	Marlon Grech	Initial draft of document
1.1	19/04/2018	Marlon Grech	Updated API Methods

Table of Contents

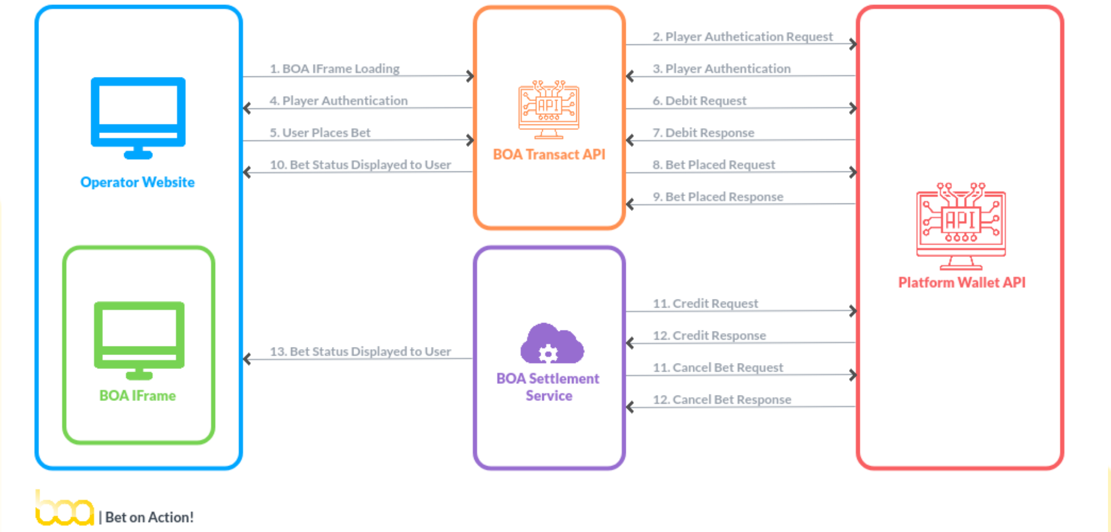
Overview

The BOA Transact API is the BOA API layer that facilitates the communication between the BOA and an Operator’s systems. All interaction between the two systems will pass through the Transact API.

To allow the BOA Transact API to interact with the Operator’s system, the Operator’s system must expose certain functionality to the BOA Transact API. This document will describe the minimum requirements, from an Operator’s API, to allow the BOA system to integrate effectively with the Operator’s system.

Technology

The BOA Transact API system is built using Microsoft .NET CORE using REST Web API. The Operator’s API can be built using any technology on any platform as long as it is accessible via the Internet (as a REST service, HTTP get, HTTP post, etc.). The request and response would be in JSON format.



API Methods

At a minimum, the Operator's API should expose the following methods:

- AuthenticatePlayer
- GetPlayerBalance
- DebitPlayer
- CreditPlayer
- BetPlaced
- CancelBet

AuthenticatePlayer

This API call will be called whenever a player needs to be authenticated by the BOA system. This will be typically when the player first accesses the games web site.

Request
string PlayerToken

Response
string PlayerId
string NickName
string CurrencyCode
string LanguageCode
decimal? PlayerLimit
string Email
BrandApiErrorCode Code *See **BrandApiErrorCode** definition
string Status

GetPlayerBalance

This API call will be called whenever a player to see his balance from the app.

Request
String CurrencyCode
string PlayerToken
string PlayerId

Response
decimal Amount
BrandApiErrorCode Code *See **BrandApiErrorCode** definition
string Status

DebitPlayer

This method will be called whenever the player should have some money debited from their wallet. This would typically be when the player submits their bet slip.

Request
String TransactionId
string PlayerToken
string PlayerId
decimal Amount
String CurrencyCode
Int? BetId
String Reason

Response
String TransactionId
BrandApiErrorCode Code *See **BrandApiErrorCode** definition
string Status

CreditPlayer

This method will be called whenever the player should have some money credited to their wallet. This would typically be when the player wins a bet or when a bet is refunded.

Request
String TransactionId
string PlayerToken
string PlayerId
decimal Amount
String CurrencyCode
Int? BetId
String Reason

Response
String TransactionId
BrandApiErrorCode Code *See **BrandApiErrorCode** definition
string Status

BetPlaced

This method will be called whenever the player would place a bet on the system.

Request
Int BetId
string PlayerToken
string PlayerId

Response
BrandApiErrorCode Code *See **BrandApiErrorCode** definition
string Status

CancelBet

This method will be called whenever the player would cancel a bet on the system.

Request
decimal Amount
String CurrencyCode
String TransactionId

Response
String TransactionId
BrandApiErrorCode Code *See **BrandApiErrorCode** definition
string Status

Common Models
BrandApiErrorCode
Ok = 0,
InvalidToken = 100,
PlayerAccountLockedOrInactive = 101,
InvalidPlayerId = 110,
InvalidCurrencyCodeForPlayer = 120,
InsufficientFunds = 121,
BetExceedsPlayerLimit = 122,
TransactionAlreadyProcessed = 123,
OriginalTransactionNotFound = 131