

Axis Aggregator Unified Operator Integration Guide

Unified API contract for operator integrations

Name	Value
Document status	Final operator integration guide
Audience	Operator engineering teams integrating with the Axis game aggregator
Codebase verification date	2026-05-22

Contents

1. Purpose And Scope	2
2. Base URL And Unified Routes	2
3. Authentication And Signing	2
4. Common Response And Error DTOs	3
5. Status Code Catalog	4
6. Operator-Implemented Unified Wallet API	5
7. Unified Aggregator API DTOs	6
8. Unified Free-Spins, Jackpots, And Promotions API	9
9. Credentials	18
10. Unified Integration Checklist	18
11. Minimal Examples	19
12. Contract Boundary	19

1. Purpose And Scope

This document is the operator-facing integration contract for Axis Aggregator. It describes only the unified APIs that an operator uses once across all supported game vendors. Included:

- Unified aggregator APIs under `/api/v1/game-operators`.
- Unified free-spins, jackpots, and promotions APIs under `/api/v1/game-operators/promotions`.
- Unified operator wallet callback API that the operator implements once for all game vendors.
- Credential handling needed to sign requests.
- Authentication, signing, DTOs, response envelopes, status codes, and integration checklist.

Any API not listed in this guide is outside the operator integration contract.

2. Base URL And Unified Routes

Use these integration bases:

Name	Value
API base URL	<code>https://<aggregator-host>/api/v1</code>
Unified operator prefix	<code>/game-operators</code>
Operator wallet callback base	Operator-provided <code>callbackUrl</code>

Unified operator route inventory:

Method	Route	Purpose	Auth
POST	<code>/game-operators/game/vendors</code>	List vendors available to the operator.	HMAC
POST	<code>/game-operators/game/list</code>	List games for a vendor.	HMAC
POST	<code>/game-operators/game/url</code>	Create game launch URL/session.	HMAC
POST	<code>/game-operators/game/terminate</code>	Terminate a player session.	HMAC
POST	<code>/game-operators/transaction/list</code>	List operator-scoped transactions.	HMAC
POST	<code>/game-operators/transaction/detail</code>	Fetch one transaction by betId.	HMAC
POST	<code>/game-operators/promotions/campaigns/create</code>	Create or register a promotion campaign.	HMAC
POST	<code>/game-operators/promotions/campaigns/list</code>	List promotion campaigns.	HMAC
POST	<code>/game-operators/promotions/campaigns/detail</code>	Get one promotion campaign.	HMAC
POST	<code>/game-operators/promotions/campaigns/cancel</code>	Cancel a promotion campaign.	HMAC
POST	<code>/game-operators/promotions/campaigns/players/add</code>	Assign players to a campaign.	HMAC
POST	<code>/game-operators/promotions/campaigns/players/remove</code>	Remove players from a campaign.	HMAC
POST	<code>/game-operators/promotions/campaigns/players/list</code>	List player campaign state.	HMAC
POST	<code>/game-operators/promotions/catalog/reward-values</code>	Query reward values for campaign setup.	HMAC
POST	<code>/game-operators/promotions/reports/winners</code>	List campaign winners.	HMAC
POST	<code>/game-operators/promotions/reports/leaderboard</code>	Get campaign leaderboard.	HMAC
POST	<code>/game-operators/promotions/reports/prizes</code>	Get campaign prizes.	HMAC
POST	<code>/game-operators/promotions/reports/detail</code>	Get campaign report detail.	HMAC
POST	<code>/game-operators/promotions/jackpots/active</code>	List active jackpot pools.	HMAC
POST	<code>/game-operators/promotions/jackpots/winners</code>	List jackpot winners.	HMAC
POST	<code>/game-operators/promotions/jackpots/winnings</code>	Get jackpot winnings.	HMAC
POST	<code>/game-operators/promotions/jackpots/contributions</code>	List jackpot contribution records.	HMAC

3. Authentication And Signing

3.1 Operator To Aggregator

Unified systematic routes use API-key HMAC headers:

Header	Required	Value
Content-Type	Yes for JSON bodies	application/json
Accept	Recommended	application/json
X-API-Key	Yes	Public API key, usually pk...
X-Signature	Yes	Lowercase hex HMAC-SHA256 of the canonical request payload described below.

HMAC payload rules for JSON requests:

- Build the JSON request body before signing.
- Use compact UTF-8 JSON: no spaces or newlines outside string values.
- Sign the same field names, values, and field order that are sent in the HTTP body.
- Do not sign a pretty-printed JSON body.
- Do not let the HTTP client mutate, reorder, or reserialize the body after signing.
- For a route with no body, sign an empty string.

The signature is the lowercase hexadecimal HMAC-SHA256 digest of that payload. Use the full provisioned `secretKey`, including the `sk_` prefix, as the HMAC key when signing operator-to-aggregator calls. The game launch route requires the same HMAC validation as the other unified routes. Missing or invalid launch authentication returns an operator envelope with `SC_INVALID_OPERATOR` or `SC_INVALID_SIGNATURE`.

3.2 Aggregator To Operator Wallet

The aggregator calls the operator's configured `callbackUrl` with:

Header	Required	Value
Content-Type	Yes	application/json
Accept	Yes	application/json
X-API-Key	Yes	Stored operator API key
X-Signature	Yes	Lowercase hex HMAC-SHA256 of the wallet request body.

Wallet callback verification rules:

- Axis sends wallet callbacks as compact UTF-8 JSON.
- Verify the signature against the raw HTTP request body received by the wallet endpoint.
- If the operator framework parses the request before application code can read it, enable raw-body capture for the wallet routes.
- Derive the wallet HMAC key by removing one leading `sk_` prefix from the provisioned `secretKey`. If the secret has no `sk_` prefix, use it unchanged.
- The expected signature is the lowercase hexadecimal HMAC-SHA256 digest.

The wallet callback key rule is intentionally different from operator-to-aggregator signing.

4. Common Response And Error DTOs

4.1 Operator Envelope

Most unified responses use:

Field	Type	Required	Description
traceId	UUID string	Yes when request has traceId	Echoes request trace.
status	string	Yes	SC_* status code.
message	string	Often	Human-readable message.
data	route-specific	On success	Success payload.

4.2 Validation Error

DTO validation failures return:

Field	Type	Description
statusCode	number	Usually 400.
message	string array	Validation messages.
error	string	Error label.

4.3 Unauthorized Error

Authentication failures return:

Field	Type	Description
statusCode	number	401.
message	string	Missing/invalid key or signature.
error	string	Unauthorized.

5. Status Code Catalog

Code	Meaning	Retry	Operator action
SC_OK	Success.	No	Continue.
SC_INSUFFICIENT_FUNDS	Wallet balance too low for debit.	No	Reject debit/bet.
SC_ACCOUNT_NOT_FOUND	Account or wallet missing.	No	Check provisioning.
SC_USER_NOT_EXISTS	Player username not found.	No	Create/map player before launch.
SC_WRONG_CURRENCY	Player lacks requested currency wallet.	No	Provision currency wallet or reject.
SC_LIMIT_REACHED	Operator-side limit blocks operation.	No	Surface limit rejection.
SC_INVALID_OPERATOR	Operator/API-key binding invalid.	No	Check key and operator setup.
SC_INVALID_SIGNATURE	HMAC signature mismatch.	No	Check exact payload and secret prefix rule.
SC_INVALID_GAME	Game/vendor/play mode unavailable.	No	Refresh catalog and check request.
SC_INVALID_REQUEST	Invalid business request.	No	Fix payload.
SC_SYSTEM_ERROR	Unexpected system error.	Yes, carefully	Retry only idempotent operations.
SC_DUPLICATE_TRANSACTION	Duplicate semantic transaction.	No	Use original transaction state.
SC_TRANSACTION_NOT_FOUND	Original transaction missing.	No	Reconcile transaction history.
SC_TRANSACTION_NOT_EXISTS	Compatibility alias.	No	Treat as SC_TRANSACTION_NOT_FOUND.
SC_PROMOTION_UNSUPPORTED	Promotion type or operation unavailable for the selected catalog scope.	No	Query reward values or change campaign setup.
SC_PROMOTION_ALREADY_EXISTS	campaignId already exists with different content.	No	Use the existing campaign or create a new ID.
SC_PROMOTION_NOT_FOUND	Campaign does not exist.	No	Check campaignId.
SC_PROMOTION_NOT_ACTIVE	Campaign is not assignable.	No	Check campaign status.
SC_PROMOTION_EXPIRED	Campaign or reward validity expired.	No	Create a new campaign.
SC_PLAYER_ALREADY_ASSIGNED	Player is already assigned to the campaign.	No	Query player state.
SC_PLAYER_NOT_ASSIGNED	Remove/query operation references a non-assigned player.	No	Reconcile player state.
SC_JACKPOT_UNSUPPORTED	Jackpot reporting is unavailable for the selected catalog scope.	No	Remove the filter or choose another supported scope.
SC_JACKPOT_NOT_FOUND	Jackpot pool does not exist or is not available to the operator.	No	Check jackpotId and catalog scope.
SC_INVALID_TIME_RANGE	Report time range is invalid or too large.	No	Submit a smaller valid range.

Wallet status normalization:

Input status	Normalized status
SUCCESS	SC_OK
INSUFFICIENT_BALANCE	SC_INSUFFICIENT_FUNDS
SC_TRANSACTION_NOT_EXISTS	SC_TRANSACTION_NOT_FOUND

6. Operator-Implemented Unified Wallet API

The operator implements this API once. The aggregator maps all supported game activity into these callbacks.

6.1 Wallet Routes

Method	Route	Purpose	Request DTO	Success data
POST	/wallet/balance	Fetch player balance.	OperatorBalanceRequest	OperatorWalletBalanceData
POST	/wallet/bet	Debit bet amount.	OperatorBetRequest	OperatorWalletTransactionData
POST	/wallet/bet_result	Settle win/loss result.	OperatorBetResultRequest	OperatorWalletTransactionData
POST	/wallet/rollback	Refund/rollback a previous transaction.	OperatorRollbackRequest	OperatorWalletTransactionData
POST	/wallet/adjustment	Apply reward/gift/cancel-gift adjustment.	OperatorAdjustmentRequest	OperatorWalletTransactionData

6.2 Wallet Request DTOs

DTO	Field	Type	Required	Notes
OperatorBalanceRequest	traceId	string	Yes	Trace from aggregator.
OperatorBalanceRequest	username	string	Yes	Operator player username.
OperatorBalanceRequest	currency	string	Yes	Currency code.
OperatorBalanceRequest	token	string	Yes	Aggregator session token.
OperatorBetRequest	traceId	string	Yes	Trace from aggregator.
OperatorBetRequest	username	string	Yes	Player username.
OperatorBetRequest	currency	string	Yes	Currency code.
OperatorBetRequest	amount	number/string	Yes	Debit amount.
OperatorBetRequest	transactionId	string	Yes	Wallet idempotency key.
OperatorBetRequest	roundId	string	Yes	Round ID.
OperatorBetRequest	gameCode	string	Yes	Game code from catalog/launch.
OperatorBetRequest	token	string	Yes	Aggregator session token.
OperatorBetResultRequest	isWin	boolean	Yes	true credits positive amount; false records loss/no credit.
OperatorRollbackRequest	originalTransactionId	string	Yes	Original transaction to refund.
OperatorAdjustmentRequest	adjustmentType	enum	Yes	REWARD, GIFT, CANCEL_GIFT.
OperatorAdjustmentRequest	adjustmentTime	string	Yes	Timestamp string.
OperatorAdjustmentRequest	originalTransactionId	string	Conditional	Required for CANCEL_GIFT.
OperatorAdjustmentRequest	details	object	No	Optional metadata.

OperatorBetResultRequest and OperatorRollbackRequest include all common transaction fields from OperatorBetRequest. OperatorAdjustmentRequest includes traceId, username, currency, amount, transactionId, adjustmentType, adjustmentTime, token, optional originalTransactionId, and optional details.

6.3 Wallet Response DTOs

DTO	Field	Type	Required	Notes
Any wallet response	traceId	string	Yes	Echo request trace.
Any wallet response	status	SC_*	Yes	Business status.
Any wallet response	message	string	No	Recommended for failures.
OperatorBalanceResponse.data	username	string	Success	Player username.
OperatorBalanceResponse.data	currency	string	Success	Currency code.
OperatorBalanceResponse.data	balance	number	Success	Total balance.
OperatorBalanceResponse.data	cashBalance	number	Recommended	Cash balance.
OperatorBalanceResponse.data	bonusBalance	number	Recommended	Bonus balance.
OperatorBalanceResponse.data	usedPromo	number	Optional	Usually 0.
Transaction response data	transactionId	string	Success	Wallet transaction ID.

DTO	Field	Type	Required	Notes
Transaction response data	balance	number	Success	Balance after operation.
Transaction response data	cashBalance	number	Recommended	Cash after operation.
Transaction response data	bonusBalance	number	Recommended	Bonus after operation.
Transaction response data	usedPromo	number	Optional	Bonus amount consumed.

6.4 Wallet Idempotency

- Idempotency key is transactionId.
- Enforce uniqueness by (operatorId, transactionId).
- Duplicate final bet, bet_result, rollback, and adjustment calls should return cached SC_OK snapshots without moving balance again.
- Unknown rollback original should return SC_TRANSACTION_NOT_FOUND.
- Already-rolled-back originals may return no-op SC_OK if the operator can prove the previous rollback completed.
- The aggregator wallet client does not retry automatically, but upstream systems can replay requests, so operator wallet behavior must be idempotent.

7. Unified Aggregator API DTOs

7.1 POST /game-operators/game/vendors

Request ListVendorsV1Dto:

Field	Type	Required	Constraints
traceId	UUID string	Yes	Request trace.
providedByCode	string	No	1..64 chars.
displayLanguage	string	No	2..16 chars, language code.
currency	string	No	3..6 uppercase chars.

Response data []:

Field	Type	Description
name	string	Vendor display name.
slug	string	Vendor slug.
currencyCode	string	Currency.
code	string	Vendor code used in requests.
categoryCode	string	Category code.
providedBy, providedByName, providedBySlug	string/null	Parent vendor metadata.
isAvailable	boolean	Availability for operator use.
availableGameCount, totalGameCount	integer	Counts.
lastSyncedAt	string/null	Last sync timestamp.

7.2 POST /game-operators/game/list

Request ListGamesV1Dto:

Field	Type	Required	Constraints
traceId	UUID string	Yes	Request trace.
vendorCode	string	Yes	1..64 chars.
gameVendorCode	string	No	1..64 chars.
pageNo	integer	Yes	Positive; default 1.
pageSize	integer	No	Positive, max 500.
displayLanguage	string	No	2..16 chars.
currency	string	No	3..6 uppercase chars.

Field	Type	Required	Constraints
fetchFromApi	boolean	No	Refresh catalog where supported.

Response data:

Field	Type	Description
headers	object	Index map for games rows.
games	array[]	Row arrays decoded by headers.
currentPage, totalItems, totalPages	integer	Pagination.
source	enum	database or api.

Game row indexes:

Index	Field
0	gameCode
1	gameName
2	categoryCode
3	square_1x1
4	wide_2x1
5	landscape_3x2
6	portrait_3x4
7	standard_4x3
8	languageCode
9	platformCode
10	currencyCode
11	isOpen
12	isAvailable
13	isVip
14	rtp
15	volatility
16	vendorCode
17	vendorName
18	availabilityUpdatedAt

7.3 POST /game-operators/game/url

Request RequestGameUrlV1Dto:

Field	Type	Required	Constraints
traceId	UUID string	Yes	Request trace.
username	string	Yes	1..64 chars.
clientPlayerId	string	No	1..128 chars.
nickname	string	No	1..128 chars.
subOperator	string	No	1..128 chars, metadata.
gameCode	string	Yes	1..128 chars.
vendorCode	string	No	Vendor code returned by /game-operators/game/vendors.
language	string	Yes	2..16 chars.
platform	string	Yes	1..32 chars.
playMode	enum	No	REAL or DEMO.
currency	string	Yes	3..6 uppercase chars.
lobbyUrl	URL string	Yes	Return URL.
depositUrl	URL string	No	Cashier/deposit URL.
country	string	No	ISO alpha-2.
additionalParams	string	No	Continuity marker where supported.
ipAddress	string	Yes	3..64 chars.
playerType	enum	No	VIP or TEST.

Response data:

Field	Type	Required	Description
gameUrl	URL string	Yes	Launch URL.
token	string	Yes	Aggregator session token.
providerSessionId	string	No	Upstream session ID.
clientSessionId	string	No	Client-facing session ID.
clientPlayerId	string	No	Operator player ID.
username	string	No	Username.

7.4 POST /game-operators/game/terminate

Request TerminateGameV1Dto:

Field	Type	Required	Constraints
traceId	UUID string	Yes	Request trace.
username	string	Yes	1..128 chars.

Response:

Field	Type	Description
traceId	UUID string	Echoed trace.
status	string	Usually SC_OK.
message	string	Success message.

7.5 POST /game-operators/transaction/list

Request ListTransactionsV1Dto:

Field	Type	Required	Constraints
traceId	UUID string	Yes	Request trace.
fromTime	integer	Yes	Non-negative timestamp.
toTime	integer	Yes	Non-negative timestamp.
pageNo	integer	Yes	Min 1.
pageSize	integer	No	1..5000, default 2000.

Transaction row indexes:

Index	Field
0	betId
1	roundId
2	externalTransactionId
3	username
4	currencyCode
5	gameCode
6	vendorCode
7	gameCategoryCode
8	betAmount
9	winAmount
10	winLoss
11	effectiveTurnover
12	jackpotAmount
13	refundAmount
14	status
15	vendorBetTime
16	vendorSettleTime
17	isFreeSpin
18	subOperator

7.6 POST /game-operators/transaction/detail

Request GetTransactionDetailV1Dto:

Field	Type	Required	Constraints
traceId	UUID string	Yes	Request trace.
betId	string	Yes	1..128 chars.
displayLanguage	string	No	2..16 chars.

Response data:

Field	Type	Description
detailUrl	URL/null	Detail/replay URL when available.
betDetail	object	Unified transaction detail.

betDetail fields: betId, externalTransactionId, roundId, username, subOperator, currencyCode, gameCode, vendorCode, gameCategoryCode, betAmount, winAmount, winLoss, effectiveTurnover, jackpotAmount, refundAmount, status, isFreeSpin, vendorBetTime, and vendorSettleTime.

8. Unified Free-Spins, Jackpots, And Promotions API

This section defines the only operator-facing contract for free spins, free chips, jackpots, tournaments, prize drops, bonus cards, and promotion reporting. Operators call the same Axis routes regardless of the upstream game vendor. Axis owns all vendor routing and payload translation behind this API. All routes in this section use the authentication headers and HMAC signing rules from Section 3.1, and all responses use the operator envelope from Section 4.1.

8.1 Design Model

Unified promotion and jackpot behavior is represented by these public concepts:

Concept	Purpose
Campaign	Operator-facing promotion container identified by <code>campaignId</code> .
Reward	Campaign benefit such as free spins, free chips, tournament prize, prize drop, or bonus card.
Player assignment	Grant or remove campaign eligibility for one or more players.
Jackpot	Jackpot pool, winner, winning amount, or contribution record exposed independently from a campaign.
Report	Read-only campaign state, winners, leaderboard, prizes, assignments, jackpot rows, or reconciliation data.

Promotion types:

Type	Meaning
FREE_SPINS	Campaign that grants a configured number of free spins/free rounds on one or more games.
FREE_CHIPS	Campaign that grants configured live-game chips or chip value.
TOURNAMENT	Leaderboard or tournament campaign.
PRIZE_DROP	Prize-drop campaign with winners and prize metadata.
BONUS_CARD	Bonus-card style campaign with active campaign and winner reporting.

Jackpots are not a `promotionType` unless Axis later introduces an explicit jackpot campaign product. Operators use the jackpot routes for jackpot pools, jackpot winners, jackpot winnings, and contribution reports. Campaign statuses:

Status	Meaning
DRAFT	Created locally but not active for assignment.
ACTIVE	Can receive assignments or be queried.
PAUSED	Temporarily blocked from new assignments.
CANCELLED	Cancelled by operator or Axis.
EXPIRED	End date has passed.
COMPLETED	Campaign has ended and final report data is available.

8.2 Route Inventory

All routes are relative to `https://<aggregator-host>/api/v1`.

Method	Route	Purpose	Idempotency
POST	<code>/game-operators/promotions/campaigns/create</code>	Create or register a promotion campaign.	<code>requestId</code> required
POST	<code>/game-operators/promotions/campaigns/list</code>	List campaigns.	Not required
POST	<code>/game-operators/promotions/campaigns/detail</code>	Get one campaign.	Not required
POST	<code>/game-operators/promotions/campaigns/cancel</code>	Cancel a campaign.	<code>requestId</code> required
POST	<code>/game-operators/promotions/campaigns/players/add</code>	Assign players to a campaign.	<code>requestId</code> required
POST	<code>/game-operators/promotions/campaigns/players/remove</code>	Remove players from a campaign.	<code>requestId</code> required
POST	<code>/game-operators/promotions/campaigns/players/list</code>	List player campaign state.	Not required
POST	<code>/game-operators/promotions/catalog/reward-values</code>	Query available reward values for games/currencies.	Not required
POST	<code>/game-operators/promotions/reports/winners</code>	List campaign winners.	Not required
POST	<code>/game-operators/promotions/reports/leaderboard</code>	Get campaign leaderboard.	Not required
POST	<code>/game-operators/promotions/reports/prizes</code>	Get campaign prizes.	Not required
POST	<code>/game-operators/promotions/reports/detail</code>	Get unified campaign detail.	Not required
POST	<code>/game-operators/promotions/jackpots/active</code>	List active jackpot pools.	Not required
POST	<code>/game-operators/promotions/jackpots/winners</code>	List jackpot winners for a time range.	Not required
POST	<code>/game-operators/promotions/jackpots/winnings</code>	Get winning amounts for one jackpot pool.	Not required
POST	<code>/game-operators/promotions/jackpots/contributions</code>	List jackpot contribution records.	Not required

8.3 Shared Promotion DTOs

Common request fields:

Field	Type	Required	Notes
<code>traceId</code>	UUID string	Yes	Operator-generated request trace.
<code>requestId</code>	string	Mutating routes	Idempotency key for create, cancel, add, and remove operations.
<code>campaignId</code>	string	Campaign-specific routes	Operator-facing campaign ID, unique per operator.
<code>promotionType</code>	enum	Where campaign type matters	<code>FREE_SPINS</code> , <code>FREE_CHIPS</code> , <code>TOURNAMENT</code> , <code>PRIZE_DROP</code> , or <code>BONUS_CARD</code> .
<code>vendorCode</code>	string	Optional	Vendor code from the unified game catalog when campaign support is vendor-scoped.
<code>currencyList</code>	string[]	Optional	ISO currency codes.
<code>gameList</code>	string[]	Optional	Game codes from the unified game catalog.
<code>cursor</code>	string	Optional	Opaque pagination cursor for list routes.
<code>limit</code>	integer	Optional	Defaults to 100; maximum 500 unless a route states otherwise.
<code>metadata</code>	object	Optional	Operator metadata. Axis stores and echoes this object but does not branch on it.

Timestamps are Unix epoch seconds unless the field explicitly says otherwise. `FreeSpinsReward`:

Field	Type	Required	Notes
<code>rounds</code>	integer	Yes	Number of free spins/free rounds.
<code>betValues</code>	object[]	Yes	At least one bet value by currency.
<code>betValues[].currency</code>	string	Yes	ISO currency code.
<code>betValues[].totalBet</code>	number	Conditional	Required when <code>betPerLine</code> is absent.
<code>betValues[].betPerLine</code>	number	Conditional	Required when <code>totalBet</code> is absent.
<code>periodOfTimeSeconds</code>	integer	No	Optional period-based validity rule.

`FreeChipsReward`:

Field	Type	Required	Notes
chipsNumber	number	Yes	Number of chips/credits granted.
chipValue	number	Conditional	Required when chipValuePerCurrency is absent.
chipValuePerCurrency	object	Conditional	Currency-to-value map. Required when chipValue is absent.
maxBetLimit	number	No	Maximum bet allowed by campaign.
maxWinLimit	number	No	Maximum win limit.
maxWinLimitPlayerCurrency	number	No	Maximum win limit in player currency.

TournamentReward, PrizeDropReward, and BonusCardReward:

Field	Type	Required	Notes
externalCampaignRef	string	Optional	External campaign reference if the campaign already exists outside Axis.
leaderboardSize	integer	No	Maximum leaderboard rows to return for tournaments.
listSize	integer	No	Maximum latest-wins rows for prize drops, max 300.
activityDate	string	No	YYYY-MM-DD or campaign activity date.

PromotionCampaign:

Field	Type	Required	Notes
campaignId	string	Yes	Operator-facing campaign ID.
promotionType	enum	Yes	Campaign type.
status	enum	Yes	DRAFT, ACTIVE, PAUSED, CANCELLED, EXPIRED, or COMPLETED.
vendorCode	string/null	No	Vendor selected by Axis or requested by operator.
gameList	string[]	No	Games covered by campaign.
currencyList	string[]	No	Currencies covered by campaign.
startAt	integer/null	No	Campaign start time.
expiresAt	integer/null	No	Campaign expiration time.
validUntil	integer/null	No	Last time a player can use the reward.
reward	object	Yes	One of the reward DTOs.
metadata	object/null	No	Operator metadata.
createdAt, updatedAt	datetime string	Yes	Axis timestamps.
supportedOperations	string[]	Yes	Operations currently available for this campaign.

PromotionPlayer:

Field	Type	Required	Notes
playerId	string	Yes	Operator player identifier.
username	string	No	Axis username if different from playerId.
currency	string	No	Player currency.
metadata	object	No	Operator metadata.

PromotionPlayerState:

Field	Type	Description
campaignId	string	Campaign ID.
playerId	string	Player ID.
status	enum	ASSIGNED, USED, EXPIRED, REMOVED, or FAILED.
rounds	integer/null	Free-spin rounds granted when applicable.
roundsPlayed	integer/null	Free-spin rounds already consumed.
chipsNumber	number/null	Free-chip count when applicable.
assignedAt, updatedAt	datetime string	Assignment timestamps.

8.4 Jackpot DTOs

JackpotPool:

Field	Type	Required	Notes
jackpotId	string	Yes	Axis jackpot pool identifier.
name	string/null	No	Display name when available.
status	enum	Yes	ACTIVE, PAUSED, CLOSED, or UNKNOWN.
currency	string/null	No	ISO currency code.
amount	number/null	No	Current jackpot amount in currency.
gameList	string[]	No	Unified game codes linked to the pool.
updatedAt	datetime string/null	No	Last known update time.
metadata	object/null	No	Extra normalized jackpot metadata.

JackpotWinner:

Field	Type	Required	Notes
jackpotId	string	Yes	Jackpot pool identifier.
playerId	string	Yes	Operator player identifier.
username	string/null	No	Username when different from playerId.
gameCode	string/null	No	Unified game code.
currency	string/null	No	ISO currency code.
amount	number	Yes	Jackpot win amount.
transactionId	string/null	No	Axis/operator transaction reference when available.
roundId	string/null	No	Round reference when available.
wonAt	datetime string	Yes	Jackpot win time.

JackpotWinning:

Field	Type	Required	Notes
jackpotId	string	Yes	Jackpot pool identifier.
currency	string/null	No	ISO currency code.
amount	number	Yes	Winning amount.
gameCode	string/null	No	Unified game code when the winning is game-scoped.
updatedAt	datetime string/null	No	Last known update time.

JackpotContribution:

Field	Type	Required	Notes
jackpotId	string	Yes	Jackpot pool identifier.
gameCode	string/null	No	Unified game code.
currency	string/null	No	ISO currency code.
amount	number	Yes	Contribution amount.
transactionId	string/null	No	Related bet transaction when available.
roundId	string/null	No	Related game round when available.
contributedAt	datetime string	Yes	Contribution time.

8.5 Campaign Lifecycle Endpoints

POST/game-operators/promotions/campaigns/create Request CreatePromotionCampaignRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
requestId	string	Yes	Idempotency key.
campaignId	string	Yes	Operator-facing campaign ID.
promotionType	enum	Yes	Campaign type.
vendorCode	string	No	Optional vendor from catalog.
gameList	string[]	Conditional	Required for game-scoped rewards.
currencyList	string[]	No	Campaign currencies.

Field	Type	Required	Notes
startAt	integer	No	Defaults to now when supported.
expiresAt	integer	No	Campaign expiration.
validUntil	integer	No	Reward validity end.
reward	reward object	Yes	Shape depends on promotionType.
metadata	object	No	Operator metadata.

Response data:

Field	Type	Description
campaign	PromotionCampaign	Created or existing idempotent campaign.
created	boolean	true if created during this request.

Idempotency:

- Repeating the same requestId with the same body returns the same campaign result.
- Reusing the same requestId with different campaign content returns SC_DUPLICATE_REQUEST.
- Reusing an existing campaignId with a different requestId returns SC_PROMOTION_ALREADY_EXISTS.

POST/game-operators/promotions/campaigns/list Request ListPromotionCampaignsRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
promotionType	enum	No	Filter by type.
status	enum	No	Filter by status.
vendorCode	string	No	Filter by catalog vendor.
fromTime	integer	No	Created/updated lower bound.
toTime	integer	No	Created/updated upper bound.
cursor	string	No	Opaque pagination cursor.
limit	integer	No	Default 100, max 500.

Response data: items as PromotionCampaign[] and nextCursor as string/null. POST/game-operators/promotions/campaigns/detail Request GetPromotionCampaignRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
campaignId	string	Yes	Campaign ID.

Response data: campaign as PromotionCampaign. POST/game-operators/promotions/campaigns/cancel Request CancelPromotionCampaignRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
requestId	string	Yes	Idempotency key.
campaignId	string	Yes	Campaign ID.
reason	string	No	Operator cancellation reason.

Response data: campaign as PromotionCampaign, and cancelled as boolean.

8.6 Player Assignment Endpoints

POST/game-operators/promotions/campaigns/players/add Request AddPromotionPlayersRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
requestId	string	Yes	Idempotency key.
campaignId	string	Yes	Campaign ID.
players	PromotionPlayer[]	Yes	1..5000 players.
assignmentOverrides	object	No	Per-request override for dates, rounds, chips, or limits when supported.

Response data:

Field	Type	Description
campaignId	string	Campaign ID.
assigned	string[]	Player IDs assigned by this request.
alreadyAssigned	string[]	Player IDs already assigned on an idempotent repeat.
failed	object[]	Player-level failures with playerId, status, and message.

POST/game-operators/promotions/campaigns/players/remove Request RemovePromotionPlayersRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
requestId	string	Yes	Idempotency key.
campaignId	string	Yes	Campaign ID.
playerIds	string[]	Yes	1..5000 player IDs.
reason	string	No	Operator removal reason.

Response data:

Field	Type	Description
campaignId	string	Campaign ID.
removed	string[]	Player IDs removed by this request.
notAssigned	string[]	Player IDs that were not assigned.
failed	object[]	Player-level failures with playerId, status, and message.

POST/game-operators/promotions/campaigns/players/list Request ListPromotionPlayersRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
campaignId	string	No	Filter by campaign.
playerId	string	No	Filter by player.
status	enum	No	ASSIGNED, USED, EXPIRED, REMOVED, or FAILED.
cursor	string	No	Opaque pagination cursor.
limit	integer	No	Default 100, max 500.

Response data: items as PromotionPlayerState[] and nextCursor as string/null.

8.7 Catalog And Reporting Endpoints

POST/game-operators/promotions/catalog/reward-values Request GetPromotionRewardValuesRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
promotionType	enum	Yes	FREE_SPINS or FREE_CHIPS.
vendorCode	string	No	Optional catalog vendor.
gameList	string[]	No	Game codes to inspect.
currencyList	string[]	Yes	Currencies to inspect.

Response data:

Field	Type	Description
rewardValues	object[]	Available reward values by game/currency.
currencyMultipliers	object[]	Currency multiplier data when available.
unsupported	object[]	Game/currency pairs not supported for the requested reward type.

POST/game-operators/promotions/reports/winners Request ListPromotionWinnersRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
promotionType	enum	Yes	Campaign type.
campaignId	string	No	Campaign ID when known.
fromTime	integer	No	Lower time bound.

Field	Type	Required	Notes
toTime	integer	No	Upper time bound.
cursor	string	No	Opaque pagination cursor.
limit	integer	No	Default 100, max 500.

Response data: items as PromotionWinner [] and nextCursor as string/null. POST/game-operators/promotions/reports/leaderboard Request GetPromotionLeaderboardRequest: traceId as UUID string, campaignId as string, and optional limit integer. Response data contains campaignId and entries as LeaderboardEntry []. POST/game-operators/promotions/reports/prizes Request GetPromotionPrizesRequest: traceId as UUID string, campaignId as string, and optional currencyList as string []. Response data contains campaignId and prizes as object array. POST/game-operators/promotions/reports/detail Request GetPromotionReportDetailRequest: traceId as UUID string and campaignId as string. Response data contains campaign as PromotionCampaign and summary as object. Reporting row DTOs:

DTO	Field	Type	Description
PromotionWinner	campaignId	string	Campaign ID.
PromotionWinner	playerId	string	Player ID.
PromotionWinner	currency	string/null	Currency.
PromotionWinner	amount	number/null	Win or prize amount.
PromotionWinner	prizeId	string/null	Prize identifier when available.
PromotionWinner	wonAt	datetime/string/null	Win timestamp.
LeaderboardEntry	rank	integer	Ranking position.
LeaderboardEntry	playerId	string	Player ID.
LeaderboardEntry	score	number/string	Leaderboard score.
LeaderboardEntry	currency	string/null	Currency.
LeaderboardEntry	prize	object/null	Prize attached to this rank.

8.8 Jackpot Endpoints

Jackpot endpoints are read-only. They do not require requestId, and retries are safe. POST/game-operators/promotions/jackpots/active Request ListActiveJackpotsRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
vendorCode	string	No	Optional catalog vendor filter.
currency	string	No	ISO currency filter.
gameCode	string	No	Unified game code filter.
cursor	string	No	Opaque pagination cursor.
limit	integer	No	Default 100, max 500.

Response data: items as JackpotPool [] and nextCursor as string/null. POST/game-operators/promotions/jackpots/winners Request ListJackpotWinnersRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
vendorCode	string	No	Optional catalog vendor filter.
jackpotId	string	No	Optional jackpot pool filter.
currency	string	No	ISO currency filter.
startTimepoint	integer	Yes	Inclusive Unix epoch seconds.
endTimepoint	integer	Yes	Exclusive Unix epoch seconds.
cursor	string	No	Opaque pagination cursor.
limit	integer	No	Default 100, max 500.

Response data: items as JackpotWinner [] and nextCursor as string/null. Validation:

- endTimepoint must be greater than startTimepoint.
- Axis may cap the maximum time window. If capped, return SC_INVALID_TIME_RANGE.

POST/game-operators/promotions/jackpots/winnings Request GetJackpotWinningsRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
jackpotId	string	Yes	Jackpot pool identifier.
currency	string	No	ISO currency filter.
cursor	string	No	Opaque pagination cursor.
limit	integer	No	Default 100, max 500.

Response data:

Field	Type	Description
jackpotId	string	Jackpot pool identifier.
currency	string/null	Requested currency filter.
items	JackpotWinning[]	Winning rows for the jackpot pool.
nextCursor	string/null	Cursor for the next page.

POST/game-operators/promotions/jackpots/contributions Request ListJackpotContributionsRequest:

Field	Type	Required	Notes
traceId	UUID string	Yes	Request trace.
vendorCode	string	No	Optional catalog vendor filter.
jackpotId	string	No	Optional jackpot pool filter.
currency	string	No	ISO currency filter.
gameCode	string	No	Unified game code filter.
fromTime	integer	No	Inclusive Unix epoch seconds.
toTime	integer	No	Exclusive Unix epoch seconds.
cursor	string	No	Opaque pagination cursor.
limit	integer	No	Default 100, max 500.

Response data: items as JackpotContribution[] and nextCursor as string/null.

8.9 Promotion And Jackpot Idempotency

- Mutating promotion requests must include requestId.
- Axis dedupes by (operatorId, route, requestId).
- Exact replay returns the original response.
- Same requestId with a different payload returns SC_DUPLICATE_REQUEST.
- Player assignment is idempotent per (operatorId, campaignId, playerId, requestId).
- A bulk assignment may partially succeed. The response includes success, already-processed, and failed arrays so the operator can retry only failed players.
- Jackpot list/report routes are read-only and do not require requestId.

8.10 Promotion And Jackpot Examples

Create a free-spins campaign:

```
{
  "traceId": "00000000-0000-4000-8000-000000000101",
  "requestId": "promo-create-001",
  "campaignId": "fs-2026-05-operator-a",
  "promotionType": "FREE_SPINS",
  "vendorCode": "vendor-demo",
  "gameList": ["game-001"],
  "currencyList": ["USD"],
  "startAt": 1779436800,
  "expiresAt": 1779840000,
  "validUntil": 1779840000,
  "reward": {
    "rounds": 10,
  }
}
```

```

    "betValues": [
      {
        "currency": "USD",
        "totalBet": 1
      }
    ]
  }
}

```

Assign players to a campaign:

```

{
  "traceId": "00000000-0000-4000-8000-000000000102",
  "requestId": "promo-assign-001",
  "campaignId": "fs-2026-05-operator-a",
  "players": [
    {
      "playerId": "john.doe",
      "currency": "USD"
    }
  ]
}

```

Query reward values:

```

{
  "traceId": "00000000-0000-4000-8000-000000000103",
  "promotionType": "FREE_CHIPS",
  "vendorCode": "vendor-demo",
  "gameList": ["game-001"],
  "currencyList": ["USD", "EUR"]
}

```

Query active jackpots:

```

{
  "traceId": "00000000-0000-4000-8000-000000000104",
  "vendorCode": "vendor-demo",
  "currency": "USD",
  "limit": 100
}

```

Query jackpot winners:

```

{
  "traceId": "00000000-0000-4000-8000-000000000105",
  "jackpotId": "jp-main-usd",
  "currency": "USD",
  "startTimepoint": 1779436800,
  "endTimepoint": 1779523200,
  "limit": 100
}

```

8.11 Promotion Contract Boundary

The operator should not need to know which upstream adapter handles a campaign or jackpot report. Axis owns that routing behind this unified API.

Behavior family	Unified API concept
Free-spin/free-round create, cancel, and list	Campaign lifecycle with <code>promotionType:FREE_SPINS</code> .
Free-spin/free-round player grant and state	Player assignment lifecycle.
Free-chip create, cancel, and value lookup	Campaign lifecycle plus reward value lookup with <code>promotionType:FREE_C HIPS</code> .
Free-chip player grant, removal, and state	Player assignment lifecycle.
Tournament, prize-drop, and bonus-card reporting	Reporting endpoints under <code>/game-operators/promotions/reports/*</code> .
Active jackpot pools, jackpot winners, jackpot winnings, and contributions	Jackpot endpoints under <code>/game-operators/promotions/jackpots/*</code> .
Jackpot win amount on game transactions	Exposed through jackpot reports and transaction reconciliation fields, not as a provider callback.

9. Credentials

Axis provides one active credential pair per integration environment:

Field	Type	Notes
<code>publicKey</code>	string	Use as X-API-Key.
<code>secretKey</code>	string	Use for HMAC signing. Store securely.

Credential rules:

- Keep separate credentials per environment.
- Treat `secretKey` as write-only after provisioning.
- Rotate credentials before disabling an old key.
- Sign every unified aggregator request with the active credential pair.
- Verify every aggregator-to-wallet callback using the configured wallet secret rule in Section 3.2.

10. Unified Integration Checklist

1. Configure operator record and callback URL.
2. Create an API key and store `publicKey` plus one-time `secretKey`.
3. Implement wallet endpoints exactly once for all game vendors.
4. Verify HMAC in both directions, including the `sk_` prefix difference.
5. Enforce idempotency by (`operatorId`, `transactionId`).
6. Provision players and currency wallets before launch.
7. Pull vendors and games from `/game/vendors` and `/game/list`.
8. Launch games using `/game/url`.
9. Process balance, bet, result, rollback, and adjustment callbacks.
10. Reconcile transactions using `/transaction/list` and `/transaction/detail`.
11. Integrate free-spins, free-chips, promotion, and jackpot routes only through `/game-operators/promotions/*`.
12. Pass negative tests for invalid signature, missing player, wrong currency, insufficient funds, duplicate transaction, rollback missing original, duplicate promotion request, invalid campaign, and invalid jackpot time range.

11. Minimal Examples

11.1 Signed Game List Request

Body:

```
{
  "traceId": "00000000-0000-4000-8000-000000000001",
  "vendorCode": "vendor-demo",
  "pageNo": 1,
  "pageSize": 100,
  "currency": "USD"
}
```

Headers:

```
Content-Type: application/json
Accept: application/json
X-API-Key: pk_example
X-Signature: lowercase_hex_hmac_sha256(secretKey, compact_json_body)
```

11.2 Wallet Bet Callback

Request from aggregator to operator:

```
{
  "traceId": "00000000-0000-4000-8000-000000000001",
  "username": "john.doe",
  "currency": "USD",
  "amount": "10.00",
  "transactionId": "bet-123",
  "roundId": "round-123",
  "gameCode": "game-001",
  "token": "session-token"
}
```

Response from operator:

```
{
  "traceId": "00000000-0000-4000-8000-000000000001",
  "status": "SC_OK",
  "data": {
    "transactionId": "bet-123",
    "balance": 90,
    "cashBalance": 80,
    "bonusBalance": 10,
    "usedPromo": 0
  }
}
```

12. Contract Boundary

Only the APIs listed in this guide are part of the operator integration contract. New functionality should appear through one of these unified surfaces before an operator relies on it:

- Catalog and launch: `/game-operators/game/vendors`, `/game-operators/game/list`, `/game-operators/game/url`.
- Money movement: the operator wallet callback API.
- Reconciliation: `/game-operators/transaction/list` and `/game-operators/transaction/detail`.
- Free-spins, free chips, promotions, and jackpots: `/game-operators/promotions/*`.