

Statsmatics Market Engine

Client Integration Guide

The client-facing contract for request input, asynchronous job polling, and normalized tennis pricing output.

<https://marketengine.statsmatics.com> | Contract 1.0.0 | Pricing service only

SECTION 1

1. Integration setup

Use HTTPS from a server-side application. Obtain a managed API key from your Statsmatics integration contact and store it as a secret.

Public endpoints

Method	Path	Purpose
POST	/v1/betfair/prediction-jobs	Validate input and create a new prediction job.
GET	/v1/prediction-jobs/{job_id}	Poll the job and retrieve completed output.
GET	/v1/capabilities	Read markets, periods, contract, and service mode.

Basic workflow

- Submit the documented JSON body with the managed API key.
- Store job_id, request_id, and the returned status_url.
- Poll until status is COMPLETED or FAILED.
- Consume prediction data only for a completed job with non-null data.

Keep the API key server-side. Never place it in query strings, browser code, shared source code, logs, screenshots, email, or chat. The API returns probabilities and decimal prices, not BET, NO_BET, stake sizes, or betting recommendations.

SECTION 2

2. Input contract

The submission body is a strict JSON object. Unknown properties are rejected. IDs that look numeric must still be sent as JSON strings.

Event and players

JSON path	Type	Required	Rule
event.event_id	string	Yes	Non-empty Betfair event identifier.
event.event_name	string	Yes	Non-empty event label.
players	array	Yes	Exactly two singles players.
players[].position	enum	Yes	One PLAYER_1 and one PLAYER_2.
players[].name	string	Yes	Name used for match mapping.

Market entries

JSON path	Type	Required	Rule
market_entries	array	Yes	Pre-game observations and first in-play marker.
[]market_id	string	Yes	Betfair market identifier.
[]market_type	string	Yes	Market type such as MATCH_ODDS.
[]market_name	string	Yes	Human-readable market label.
[]clk	string	Yes	Opaque Betfair cursor; never parse as time.
[]pt	int64	Yes	Betfair publish time in epoch milliseconds.
[]market_time	RFC 3339/null	No	Scheduled context, not actual match start.
[]timezone	string/null	No	Timezone context; UTC is the safe fallback.
[]in_play	boolean	Yes	Must be explicit.
[]status	enum	Yes	OPEN, SUSPENDED, or CLOSED.
[]selections	array	Yes	May be empty on first in-play marker.

SECTION 3

3. Selections and requested output

Selection fields

JSON path	Type	Required	Rule
selections[].selection_id	string/null	No	Betfair outcome identifier.
selections[].name	string	If present	Non-empty player or outcome label.
selections[].line	number/null	No	Handicap or totals line; null for moneyline.
selections[].ltp	number/null	No	Carried-forward Betfair rc[].ltp above 1.0.

Betfair normalization

Apply each raw mcm delta to cached market and runner state before submission. Preserve clk as an opaque string and pt as publish time. Map mc[].id to market_id, marketDefinition fields to the normalized entry, and joined runner definitions plus rc[].ltp to selections. A definition-only history contains no source odds.

SET_WINNER uses the set number in market_name; for example, Set 1 Winner maps to MONEYLINE / SET_1. The earliest in_play=true pt defines match start; scheduled market_time never replaces it.

Requested markets

requested_markets is optional or nullable. If omitted or null, the service requests every entitled default market family and period.

Field	Allowed values	Rule
requested_markets[].type	MONEYLINE, HANDICAP, TOTALS	Each type appears at most once.
requested_markets[].periods	MATCH, SET_1, SET_2, SET_3, SET_4, SET_5	Non-empty; availability depends on entitlement.

Optional metadata

The nullable metadata object may include event_type, tournament_name, round, surface, indoor, and gender. Metadata provides context but cannot replace player/date mapping inputs.

Closing-line rules

- The earliest entry with in_play=true establishes the match-start cutoff using pt.
- Only in_play=false prices strictly before the cutoff are eligible.
- For each market, selection, and line, the latest valid pre-cutoff price is used.
- A null LTP does not erase an earlier valid LTP; prices on the in-play marker are ignored.
- A missing in-play marker causes CLOSING_NOT_REACHED.

Do not send source_provider, schema_version, model_version, internal match IDs, or undocumented properties.

SECTION 4

4. Complete input example

Submit this JSON to POST /v1/betfair/prediction-jobs. The pre-match entry supplies valid source prices; the second entry is the first in-play marker.

```
{
  "event": {
    "event_id": "35321298",
    "event_name": "Guillen Meza v Mi Damas"
  },
  "players": [
    {"position": "PLAYER_1", "name": "Alvaro Guillen Meza"},
    {"position": "PLAYER_2", "name": "Miguel Damas"}
  ],
  "market_entries": [
    {
      "market_id": "1.254640336",
      "market_type": "MATCH_ODDS",
      "market_name": "Match Odds",
      "clk": "4491333114",
      "pt": 1772568507586,
      "market_time": "2026-03-03T20:00:00.000Z",
      "timezone": "UTC",
      "in_play": false,
      "status": "OPEN",
      "selections": [
        {"selection_id": "15745245", "name": "Alvaro Guillen Meza", "line": null, "ltp": 1.47},
        {"selection_id": "15313770", "name": "Miguel Damas", "line": null, "ltp": 3.05}
      ]
    },
    {
      "market_id": "1.254640336",
      "market_type": "MATCH_ODDS",
      "market_name": "Match Odds",
      "clk": "4491362894",
      "pt": 1772568624002,
      "market_time": "2026-03-03T20:00:00.000Z",
      "timezone": "UTC",
      "in_play": true,
      "status": "OPEN",
      "selections": []
    }
  ],
  "requested_markets": [
    {"type": "MONEYLINE", "periods": ["MATCH", "SET_1", "SET_2"]},
    {"type": "HANDICAP", "periods": ["MATCH", "SET_1", "SET_2"]},
    {"type": "TOTALS", "periods": ["MATCH", "SET_1", "SET_2"]}
  ],
  "metadata": {
    "event_type": "Challenger Men Singles",
    "tournament_name": "Brasilia 2",
    "round": "1/16-finals",
    "surface": "Clay",
    "indoor": false,
    "gender": "Male"
  }
}
```

SECTION 5

5. Submission, polling, and completed output

Accepted submission - HTTP 202

```
{
  "job_id": "job_d6faa1ae",
  "prediction_id": null,
  "status": "QUEUED",
  "stage": "VALIDATION_COMPLETE",
  "status_url": "/v1/prediction-jobs/job_d6faa1ae",
  "data": null,
  "error": null,
  "request_id": "req_93cce4c2",
  "accepted_at": "2026-07-31T09:30:00Z"
}
```

Processing poll - HTTP 200

```
{
  "job_id": "job_d6faa1ae",
  "prediction_id": null,
  "status": "PROCESSING",
  "stage": "MODEL_RUNNING",
  "data": null,
  "error": null,
  "request_id": "req_93cce4c2"
}
```

Polling requirements

- Send the same API key on every poll and use the returned relative status_url.
- Use bounded polling with a delay and stop only at COMPLETED or FAILED.
- A terminal failed job is a found resource and returns HTTP 200 with status=FAILED.

Completed output

```
{
  "job_id": "job_d6faa1ae",
  "prediction_id": "pred_218f1619",
  "status": "COMPLETED",
  "stage": "COMPLETED",
  "data": {
    "match": {"event_id": "35321298", "match_date": "2026-03-03"},
    "markets": [{
      "type": "MONEYLINE",
      "period": "MATCH",
      "line": null,
      "source_odds_available": true,
      "selections": [
        {
          "selection": "PLAYER_1",
          "name": "Alvaro Guillen Meza",
          "source_price_timestamp_utc": "2026-03-03T20:08:27.586Z",
          "model_probability": 0.684,
          "decimal_price": 1.462
        },
        {
          "selection": "PLAYER_2",
          "name": "Miguel Damas",
          "source_price_timestamp_utc": "2026-03-03T20:08:27.586Z",
          "model_probability": 0.316,
          "decimal_price": 3.165
        }
      ]
    }
  ]
},
  "error": null,
  "request_id": "req_93cce4c2"
}
```

SECTION 6

6. Output and error reference

Consume market data only when status=COMPLETED, prediction_id is a string, data is an object, and error is null.

Field	Meaning
prediction_id	Stable identifier for a completed prediction.
data.match.event_id	Original client event identifier.
data.match.match_date	Resolved UTC date in YYYY-MM-DD format.
markets[].type / period	Market family and supported match or set period.
markets[].line	Null for moneyline; numeric for handicap or totals.
markets[].source_odds_available	True when corresponding closing source odds were available.
selections[].selection	PLAYER_1, PLAYER_2, OVER, or UNDER.
selections[].model_probability	Finite probability from 0 through 1.
selections[].decimal_price	Finite model-derived decimal price greater than 1.0.

The public decimal_price is generated by NumLab. Source Betfair LTP is retained privately for audit and is not returned to the client.

State and nullability

Status	prediction_id	data	error
QUEUED / PROCESSING	null	null	null
COMPLETED	string	object	null
FAILED / REJECTED	null	null	object

Client handling

Condition	Action
401 / 403	Check the API key, client status, and market entitlement.
422 VALIDATION_ERROR	Correct every path reported in error.fields.
429 rate or quota limit	Honor Retry-After and monitor rate/quota headers.
503 transient service error	Retry only if creating another job is acceptable.
FAILED job	Never consume partial data; inspect error.code and error.retryable.

Integration checks

- Keep the API key server-side and absent from logs.
- Send only documented JSON fields and include the first in-play marker.
- Bound polling and stop at COMPLETED or FAILED.
- Record request_id and job_id for support correlation.

For support, provide request_id, job_id when available, HTTP status, public error code, and UTC timestamp. Never send the API key.