UOF - Replay server (API)

Table of contents:

Please note

⚠

Due to the COVID-19 situation the whitelist on the Unified Feed Replay Environment has been removed, but only if you are using stgapi.betradar.com instead of api.betradar.com for all endpoints starting with /v1/replay

In the following table you will find all the available API endpoints for manipulating the replay server to suit your needs (https://iodocs.betradar.com/replay#Replay-Scenarios-GET-List-scenarios).

HTTP	Endpoint-path from https://api.betradar.com/v1	Description
GET	`/replay/`	Lists all sport-events in the replay list
PUT	`/replay/events/(id)`	Add an event to the replay list
DELETE	`/replay/events/(id)`	Remove an event from the replay list
POST	`/replay/play`	Start replay (with specified parameters)
POST	`/replay/stop`	Stop replay
POST	`/replay/reset`	Stop replay (if playing) and clear playlist
GET	`/replay/status`	Get status of replay (playing or stopped)
GET	`/replay/scenario`	List scenarios
POST	`/replay/scenario/play{scenarioid}`	Start replay scenario
GET	`/sports/{language}/sport_events/{urn_type}:{id}/summary.xml`	Summary information about a replay event
GET	`/sports/{language}/sport_events/{urn_type}:{id}/fixture.xml`	Fixture information about the event on the replay server, specific to type of event/replay
GET	`/sports/{language}/sport_events/{urn_type}:{id}/timeline.xml`	Timeline information about a replay event

Add an event to the replay list

Adds event {eventId} to the end of the replay queue. If there is no event in the replay queue, this event will appear on the first (and at the same time last) position of the replay queue. If there are already some events present, new event is added at the end of the queue. If {eventId} is already present in the queue before calling this endpoint and it is not at the end of queue, the event is moved to the last position in the queue and all events that were in the queue after this event are moved to 'their initial position - 1' position in the queue. If it is present and is already last, nothing will change.

If you want to start sending messages from say 5 minutes into the match, there is one extra parameter start_time that specifies how many minutes relative to match start to start sending messages from.

Remove an event from the replay list

If the specified event is present in the queue, it will be removed and all other affected events will change their position to initial position - 1 position. If the event is not present in the queue, nothing will change.

Start a replay

Start replaying an event from replay queue. Events are played in the order they were played in reality, e.g. if there are some events that were played simultaneously in reality, they will be played in parallel as well on replay server. Adjust the parameters and specify the speed of the replay, and what should be the maximum delay between messages. If not specified, default speed = 10 and max_delay = 10000 are used. This means that messages will be sent 10x faster than in reality, and that if there was some delay between messages that was longer than 10 seconds it will be reduced to exactly 10 seconds/10 000 ms (this is helpful especially in pre-match odds where delay, this can be even a few hours or more).

Note The start scenario API will return a response immediately. If you try to start a new replay before replay setup is complete, you will receive this response:

XML example

```
<response>
<action>Unable start the player. Player status is currently: SETTING_UP.<
/action>
<message>Player is currently fetching and preparing data for the replay.
Please wait.</message>
</response>
```

If you ask for a status **before** the replay setup is complete, you will get the following response:

XML example

```
<player_status status="SETTING_UP"
last_msg_from_event="sr:match:18000971"
description="Player is currently fetching and preparing data for the
replay. Please wait." />
```

The node_id parameter: Can be used if there are multiple developers for the same client using the replay server. If node_id is set, messages will have the node_id in the routing-key, which can be used to ensure developers are not affected by each other's replays.

The product_id parameter: Can be used to ensure that only messages from a specific producer is sent (for example if you only want to receive odds for the Live Odds producer).

The use_replay_timestamp: If you set this to true, the replay server will set the timestamp in the messages to the current time, if you set it to false the replay server will send the message with the original timestamp as it was sent.

Stop replay

Stop the player if it is currently playing. If player is already stopped, nothing will happen.

Rewrite timestamps

Replace the timestamp in messages with the current time.

Allowed settings are:

- **false**: Timestamps will NOT be in current time - this will use the fixture information from the actual match
- **true**: Timestamps will now be in current time - meaning match information will be relative to replay start

Note

This also affects fixture's starting times

Reset replay

Stops the player if it is playing and removes all sport events from the play list (if any matches are in the list).

Replay status

Return the status of player. Possible values are:

- Player is playing
- Player is stopped
- Player was never playing.

Replay summary, fixtures, and timelines

There are dedicated endpoints available on the replay server environment called */replay/sports/{language}/sport_events/{urn_type}:{id}/summary.xml (fixture.xml or timeline.xml)*.

These endpoints are copies of the endpoints for the real time API on the production environment (that returns the requested information of a sport event). However, these endpoints are used for replay matches and will provide you with replayed API responses.

Back to top