UOF - Recovery using API

This section includes instructions and details relevant to performing a recovery in Unified feed using the API. Please make sure to read the provided description and information regarding the different endpoints and if available, notes on performing recovery in edge cases.

Table of contents:

Max recovery periods for producers

- **Custom Bet**: Unlimited
- **Gaming producers**: 3 hours
- **Live Odds producers**: 10 hours
- **All other producers**: 72 hours

Recovery endpoints

HTTP	Endpoint	Description
POST	`(product)/recovery /initiate_request?after= (timestamp)[&request_id= (x)][&node_id=(y)]`	All odds changes for sport events handled by this product. The timestamp must be no longer than the max recovery period per producer in the past.
POST	`(product)/odds/events/(id) /initiate_request`	All current odds for all the markets for one single match/race. The event must be either not_started or live.
POST	`(product)/stateful_messages /events/(id) /initiate_request`	All stateful messages for the specified match /race. The event can be up to 30 days in the past.

All of these endpoints return either success or failure. If success, the actual messages come as one or more message per match over AMQP. The recovery sequence ends with a snapshot_complete message being sent:

```
<snapshot_complete request_id="1234" timestamp="1234578" product="3"/>
```

The (product) argument in the API (table above) can be one of the following: liveodds, pre, betpal, mts or virtuals

```
liveodds/recovery/initiate_request?after=(timestamp)[&request_id=(x)]
[&node_id=(y)]
```

All requests can have an additional parameter *request_id=X*, Where X is an integer of your own choice. The contract is that if this request ID if present, it will be included in all stateful messages resulting from this call as an extra attribute, and it will be included in the snapshot_complete message once all messages have been sent. You set a request_id if you want to track the results of individual requests.

All requests can also have an additional (optional) parameter *node_id=Y*, where Y is an integer of your choice. If you have multiple sessions, Betradar's suggestion is that you use different numbers for different sessions. During the recovery the node_id you passed in will be sent in as the 8th word in the routing key. This way you can add a binding key that ensures that you only receive recovery messages for your particular session, and not for your other sessions.

Note: The SDKs automatically takes care of recovery.

The after parameter is specified in **milliseconds since Epoch UTC.** This is for example: https://currentmillis.com/

The after parameter should normally be set to the timestamp of the last message received. This should be the timestamp in the message itself, not something the client system computed when it read the message. If the timestamp is not too far in the past, the client system is guaranteed to have current odds for any active matches, and also all bet_settlement, bet_cancel and rollbacks that have happened inbetween the timestamp and now.

Note: The client system receives current odds for active sport_events. This does not necessarily mean all odds changes. If there have been multiple updates to a market, the client system will only receive the most recent one. If the complete odds change history is desired, this has to be requested for an individual match separately.

This odds history is not available through the API, but can be downloaded through *Ctrl* or the *Live Booking Calendar* if needed.
Please be aware that multiple bet_settlements can be merged into one single bet_settlment message during recovery as well. For example: "*If a bet_settlment was sent out at half-time for half-time markets, and another bet_settlement was sent after the match, the recovery bet_settlement for the same match will than include both the half-time and post-match markets in one message*".

The client system may receive messages sent before the timestamp provided. The client system should be built robust enough to handle this.

The closer the after timestamp is to now, the better it is. If the last message you received was more than the max recovery period for the producer, this is too far in the past.

If the client system does not specify the after parameter, this means: "*I am new, just send me current active odds".* The response to this is that all current odds for live matches (where odds are active) are sent. The Live Odds producer handles all the live matches, but the prematch producers sends all matches where we have generated odds, and the match_status description="not_started".

If the after parameter is too far in the past or in the future, it will result in the request getting rejected and if the client system was not subscribed previously, the client system will remain unsubscribed.

Recovery and bet_settlement messages

Please be aware that multiple bet_settlements can be merged into one single bet_settlement message during recovery as well. For example: "*If a bet_settlement was sent out at half-time for half-time markets, and another bet_settlement was sent after the match, the recovery bet_settlement for the same match will than include **both** the half-time and post-match markets in one message*".

Recovering all bet settlements for a specific event

Under some circumstances you may have missed/lost some bet_settlements in a match. Or you may want to recover the state of a match the results for a match that ended more than three days ago. In such cases, you can call a specific end-point to recover all bet_settlements, bet_cancels and rollbacks done for a particular event.

Recovery sequence

Proper recovery requires the following steps:

- Before starting the recovery: Call the daily schedule for the next 3 days and cache the fixtures for all matches received. This will reduce the number of individual API calls later on
- Ensure you have an established AMQP session
- Call the recovery endpoint
- Until you receive a snapshot_complete for the previous recovery call, process like this:
    - If there are odds changes with no recovery id, process them as normal. If they are odds_change messages with *recovery id* and you have no recent update on the specified sport event, process them as normal odds change updates. If they are odds_changes with recovery_id and you have already received a recent update, ignore this message (you have already received a more recent update)
    - If they are stateful messages (bet settlement, rollback bet settlement, bet cancel, rollback bet cancel) - store them in a queue for later processing
- When snapshot complete for the recovery request is received
    - Process all stateful messages in the queue received in the previous step
- After the snapshot complete: you can process all stateful messages directly without queueing them up
- As soon as you have received an odds_change for a sport_event even under the recovery, you open up the markets for that sport event. A conservative solution can wait until snapshot_complete has been received and open up all markets for all received sport-events then, but this is not strictly necessary

Note

Some market/line statuses could have changed during a possible disconnection, and these will not be involved in the regular odds_change after reconnecting. This means you should go through the recovery odds_changes as well in order to have the latest status on every market that you are offering for an event.

Recovery sequence when an <alive product="x" subscribed="0"/> is received

Whenever an alive message is received and the subscribed attribute in the message is set to 0, This means that a particular odds producer has been down (due to maintenance or some system error), and your system needs to initiate a partial recovery. This is done following the same steps as above, but in step two, only an API-call for the product that reported it was down is called. I.e. you receive:

```
<alive product="1" subscribed="0"/>
```

you call pre/recovery/initiate_request and process messages as outlined above until *snapshot_complete* is received.

Handling special states

Note

The system has an *at-least-once-guarantee*. If something goes down, there is some risk that the same message is sent multiple times; your system should be built to handle this (for odds changes this is typically not a problem).

Forced betstop

In case of the following scenario:

- If the scheduled time for a match has started and the prematch odds producer provided the last odds.
- You have not received any odds from another producer, and no betstop from the prematch producer has been received.

In this case, the client system should automatically betstop all markets for the match. This is an extra precaution if the prematch system is down and you have not yet noticed this happening.

Handed_over market state during recovery

During recovery a producer may send a message saying that some markets have been handed_over. This is not a market state, but an indicator that since the disconnect happened this producer has handed over odds production for this market to another odds producer. If you already have received odds for this market from another odds producer, you can ignore this message, otherwise you should now mark this market as *suspended*. You should shortly receive odds from the other producer after it is handed over. For more information about the life cycle of market, see this chapter.

Settled and cancelled market state during recovery

During recovery some markets can be sent with the market state *settled*. This means that bet_settlements already have been sent for this market. This can for example happen during the 2nd half in soccer, where the 1st half markets have already been *settled*. There is no more activity expected for this market (the only way the system can change this is to send a rollback_betsettlement.

Similarly, during recovery a market can be sent with the market state *cancelled*, which should be treated in the same manner as above - no more activity is expected, and if you haven't already received a bet_cancel when recovery ends, a bet_cancel message has been missed somewhere.

General recommendations for market state handling when an Odds Producer is unavailable

If a match is live and the client system loses connection to the odds producer providing odds for that market, the client system must set all markets to *suspended* and stop accepting tickets.

If a match is not yet *live*, you have a choice: Either *suspend* all markets directly and stop accepting tickets. Alternatively, sport-events that are more than e.g. 2 hours from scheduled start time can be kept open as it is unlikely that there will be significant odds changes, and you are likely to get connected again shortly. **However**, we still recommend that if you haven't been able to reconnect to the odds producer in ~10 minutes, stop all markets for all sport_events for this odds producer as soon as possible.

Back to top